Advanced Warehouse Energy Storage System Control Using Deep Supervised- and Reinforcement Learning

Sven Myrdahl Opalic

UiA University of Agder

Doctoral Dissertations at the University of Agder 435

Advanced Warehouse Energy Storage System Control Using Deep Supervised- and Reinforcement Learning

Sven Myrdahl Opalic

Advanced Warehouse Energy Storage System Control Using Deep Supervised- and Reinforcement Learning

Doctoral Dissertation for the Degree *Philosophiae Doctor (PhD)* at
the Faculty of Engineering and Science, Specialisation in Renewable Energy

University of Agder
Faculty of Engineering and Science
2023

Doctoral Dissertations at the University of Agder 435

ISSN: 1504-9272
ISBN: 978-82-8427-152-1

© Sven Myrdahl Opalic, 2023

Printed by Make!Graphics, Kristiansand

Preface

In 2015 I was hired as an engineering consultant to design an energy concept for a new food distribution warehouse. Many years later, the foundation of this research was established after a meeting with Professor Morten Goodwin regarding possibilities for intelligent control of a smart warehouse using artificial intelligence. The meeting was initiated in response to our effort to design a rule-based algorithm for the warehouse. This algorithm was intended to control and optimize the operational costs associated with the battery storage system and thermal storage system. These systems were part of a technologically advanced food distribution warehouse that we have been developing since its inception in 2015. The task turned out to be overly complicated and would require extensive human expert maintenance during operation. Our attempts to find a robust, adaptive, and sophisticated commercially available system also proved to be fruitless. We therefore decided to undertake this industrial Ph.D. study to explore the emerging possibilities within artificial intelligence.

This dissertation is the result of a collaboration between the property development company Relog AS and The University of Agder and their Centre for Artificial Intelligence Research (CAIR). The research is also partially funded by the Norwegian Research Council. My main supervisor has been Professor Mohan Lal Kolhe, with co-supervisors Professor Morten Goodwin, Associate Professor Lei Jiao, and Professor Henrik Kofoed Nielsen.

Production note: LaTeX has been adopted as the tool for writing this dissertation, as well as the papers produced during my Ph.D. study. The mathematical calculations and simulation results are obtained by using PYTHON and supporting AI libraries.

Acknowledgments

First of all, I wish to thank the Norwegian Research Council for their financial support and the opportunity they provide with their Industrial Ph.D. program through our project "Deep Neural Networks for Energy Optimisation" (#283957).

I would like to express my gratitude to my main supervisor, Professor Mohan Lal Kolhe for helping me navigate the world of academic publishing. I also wish to express my sincerest gratitude to co-supervisors Professor Morten Goodwin and Associate Professor Lei Jiao for their steadfast encouragement, valuable suggestions and dependable follow-up. Morten and Lei consistently made time out of their busy schedule to be available for discussion and provide me with valuable feedback. I also thank Professor Henrik Kofoed Nielsen for his contributions to the thermal energy component of our research. I wish to thank the Ph.D. coordinators at the Faculty of Engineering and Science, UiA, Kristine Evensen Reinfjord and Emma Elisabeth Horneman for their administrative support.

Last, but not least, I also wish to express my heartfelt and most sincere appreciation to Bjørnar Høyland and Relog AS for this amazing opportunity. All the work behind this thesis would not have been possible without Bjørnar and Relogs consistent support and trust. My gratitude also goes out to the people at Rema 1000 who had a part in making this a reality. Last, but definitely not least, I wish to thank my family and my wife in particular for encouraging me to undertake this transformative journey.

Sven Myrdahl Opalic
Oslo
31.01.2023

Dedicated to my girls

Alba Noelle, Alida Isabelle, and Angelica

Abstract

The world is undergoing a shift from fossil fuels to renewable energy sources due to the threat of global warming, which has led to a substantial increase in complex building-integrated energy systems. These systems increasingly feature local renewable energy production and energy storage systems that require intelligent control algorithms.

Traditional approaches, such as rule-based algorithms, are dependent upon time-consuming human expert design and maintenance to control the energy systems efficiently. Although machine learning has gained increasing amounts of research attention in recent years, its application to energy cost optimization in warehouses still remains in a relatively early stage. Suggested newer approaches are often too complex to implement efficiently, very computationally expensive, or lacking in performance.

This Ph.D. thesis explores, designs, and verifies the use of deep learning and reinforcement learning approaches to solve the bottleneck of human expert resource dependency with respect to efficient control of complex building-integrated energy systems. A technologically advanced smart warehouse for food storage and distribution is utilized as a case study. The warehouse has a commercially available Intelligent Energy Management System (IEMS).

This thesis has two main parts. The first part is a data-driven modelling approach of a smart warehouse to build a simulated training environment for reinforcement learning agents. We use Artificial Neural Networks (ANN) to model the compressors of an industrial cooling system in the warehouse. We create an ensemble of compressor models to model the whole cooling system and generate data related to the system's performance. The developed model has MAPE in the range of 5 % to 12 % in the operational case-study cooling system. The presented results also show that the accuracy can be drastically improved with increased quality of data collection frequency in the operational measurements, supported by a MAPE of around 1.8 % compared to measurements from a laboratory cooling system. We also use various machine learning techniques, such as linear and polynomial regression, to generate data-driven models of sub-systems and dynamics of the warehouse energy system.

The second part of the thesis describes a robust, data-efficient reinforcement learning algorithm based on Augmented Random Search (ARS). We introduce ANNs to replace the linear policy of the ARS to allow the agent to learn more abstract behavior to achieve higher performance in a complex environment. We show that the ARS-ANN algorithm achieves impressive performance by reducing energy cost through Battery Energy Storage System (BESS) control in the simulated warehouse environment. We extend the ARS algorithm with COST-WINNERS, allowing the algorithm to control multiple energy storage systems.

More specifically, we reduce energy cost through simultaneous BESS and Thermal Energy Storage (TES) control with the COST-WINNERS reinforcement learning algorithm in the simulated warehouse environment. The TES is supplied by reclaimed heating energy from the warehouse cooling system mentioned in the previous paragraph. The ANN model of the cooling system has been used to generate data related to available heat and cooling system efficiency in the simulated environment developed to train the COST-WINNERS reinforcement learning agent. We show that the COST-WINNERS algorithm achieves comparable or better performance than an optimization solver given perfect information in 9 out of 10 seeded trials. However, an important note is that the optimization algorithm only controls the BESS.

In conclusion, we show impressive performance in ANN modelling of cooling system efficiency through comparisons to theoretical and laboratory experiment data, as well as metered energy consumption from the warehouse cooling system. We show that the COST-WINNERS algorithm establishes a new state-of-the-art of simultaneous control of multiple energy storage systems in the simulated technologically advanced warehouse. The ANN cooling system model has been implemented in the warehouse and generates live data on cooling system performance. We plan to implement and test the developed algorithms in the warehouse in the future, subject to satisfactory quality assurance measurements.

Publications

The author of this dissertation is the first author and the principal contributor of all the included papers listed below. Papers A-E in the first set of the following list are selected to represent the main research achievements and are reproduced as Part II of this dissertation.

Papers Included in the Dissertation

Paper A **Opalic, S. M.**, Goodwin, M., Jiao, L., Nielsen, H. K., and Kolhe, M. L., "Modelling of Compressors in an Industrial CO_2-Based Operational Cooling System Using ANN for Energy Management Purposes.", In *International Conference on Engineering Applications of Neural Networks*, pp. 43-54. Springer, Cham, 2019.

Paper B **Opalic, S. M.**, Goodwin, M., Jiao, L., Nielsen, H. K., Pardiñas, Á. Á., Hafner, A., and Kolhe, M. L., "ANN modelling of CO2 refrigerant cooling system COP in a smart warehouse.", In *Journal of Cleaner Production*, 260, 2020.

Paper C **Opalic, S. M.**, Goodwin, M., Jiao, L., Nielsen, H. K., and Kolhe, M. L. "A Deep Reinforcement Learning scheme for Battery Energy Management.", In *2020 5th International Conference on Smart and Sustainable Technologies (SpliTech)*, pp. 1-6, IEEE, 2020.

Paper D **Opalic, S. M.**, Goodwin, M., Jiao, L., Nielsen, H. K., and Kolhe, M. L., "Augmented Random Search with Artificial Neural Networks for energy cost optimization with battery control.", In *Journal of Cleaner Production*, 2022.

Paper E **Opalic, S. M.**, Palumbo, F., Goodwin, M., Jiao, L., Nielsen, H. K., and Kolhe, M. L., "COST-WINNERS: COST reduction WIth Neural NEtworks-based augmented Random Search for simultaneous thermal and electrical energy storage control.", In *Journal of Energy Storage, 2023.*

Contents

List of Figures

List of Tables

List of Abbreviations

AI Artificial Intelligence

ANN Artificial Neural Network

ARS Augmented Random Search

BEMS Building Energy Management System

BESS Battery Energy Storage System

BMS Building Management System

CES Cold Energy Storage

CNLP Constrained Non-Linear Programming

COP Coefficient of Performance

CS Cooling System

DC Direct Current

DDPG Deep Deterministic Policy Gradient

DL Deep Learning

DPC Data Predictive Control

DPG Deep Policy Gradient

DQN Deep Q-Network

DSM Demand Side Management

ESS Energy Storage System

GDI Generalized Data Distribution Iteration

GLPK GNU Linear Programming Kit

HES Heat Energy Storage

HVAC Heating Ventilation Air-Conditioning

IEMS Intelligent Energy Management System

MDP Markov Decision Process

MILP Mixed Integer Linear Programming

MPC Model Predictive Control

PV Photovoltaic

ReLU Rectified Linear Unit

RL Reinforcement Learning

SAC Soft Actor-Critic

SOC State Of Charge

Tanh Hyperbolic Tangent

TD3 Twin Delayed Deep Deterministic Policy Gradient

TES Thermal Energy Storage

ToU Time-of-Use

TRPO Trust-Region Policy Optimization

Part I

Main Chapters

Chapter 1

Introduction

The world is undergoing a shift from fossil fuels to renewable energy sources due to the threat of global warming. A significant part of this transition is the necessary increase in building-integrated intermittent renewable energy production, including local energy storage and microgrid solutions. This leads to new challenges in achieving energy efficiency during the operation of these more complex building-integrated energy systems.

Electrical power production companies and grid owners are influencing consumers through variable pricing strategies and peak effect tariffs that also need to be taken into account. The energy systems are affected by variables such as weather, ambient temperature and building user interaction. Considering how all these variables simultaneously affect the energy system and how this could be optimised through prediction, planning, and real-time management of energy production, load shifting, and electrical and thermal energy storage is a very complex problem. Solving this problem is a necessary step to realize the full potential of positive environmental impact from building-integrated renewable energy production.

In this Ph.D. thesis, a technologically advanced smart warehouse for food storage and distribution is utilized as a case study of an energy system that features all the aforementioned characteristics. The smart warehouse has been completed and has a commercially available Intelligent Energy Management System (IEMS) that utilizes various machine learning techniques for predictions of essential parameters and an optimization algorithm to generate an hourly 48-hour schedule for the electrical and thermal energy storage systems. The IEMS predicts electrical and thermal energy demand to come up with an optimal scheduling strategy that also considers future energy price, weather forecasts and predicted thermal energy production efficiency. The limitations of the current system include simplified modeling of the energy system to make it solvable for the optimization algorithm, inability to react to discrepancies between predictions and real-time measurements, reliance on the continual human expert tuning of various algorithm dependencies and the relatively short planning horizon in relation to the energy storage capacity of the Thermal Energy Storage (TES).

Although rule-based and optimizer-based systems are both relevant approaches to IEMS design, they are quite human expert resource dependant in both design and maintenance. As building-integrated energy system complexity increases through the addition of local renewable energy production and storage systems, such as in our featured technologically advanced warehouse, human expert maintenance can reach unmanageable levels in terms of cost and time constraints. A possible solution to this problem is the application of Artificial Intelligence (AI) to enable systems that require less human attention in both implementation and operation due to its ability to learn from historical data and adapt to new data. Within AI, Reinforcement Learning (RL) is a very promising research area that allows an agent to learn from interactions with its environment through reward signals, whereas deep RL is the application of Deep Learning (DL) techniques to RL algorithms. More specifically, making the various neural network architectures able to deal with higher degrees of complexity and abstraction by increasing the amount of neurons is essential for learning the complex energy system dynamics necessary for IEMS application.

Although deep RL has gained increasing amounts of research attention in recent years, its application to energy cost optimization in advanced warehouses still remains in an early stage. This thesis will explore a deep RL approach to energy cost optimization with energy storage systems. This requires a sophisticated simulated environment that enables the algorithm to train in an offline manner until the decisions made are intelligent enough to be implemented and tested in the case-study warehouses. The simulation environment initially consisted of traditional mathematical models of the various building components and systems. As data from the building operation increases, we have replaced component models by more accurate models based on neural networks or other data-driven algorithms. This process can be automated and thus hopefully makes the complete algorithm robust and scalable.

The overall goal with the research in this Ph.D. thesis is to explore the use of deep RL to solve the bottle-neck of human expert resource dependency in relation to design and maintenance of intelligent control systems of complex building-integrated energy systems.

In this chapter, a description of the motivation of this Ph.D. dissertation and an overview of the research questions are presented. Furthermore, the research approaches and the dissertation structure are also outlined.

1.1 Motivation and research questions

Initial attempts to design a deterministic program tasked to control the electrical and thermal Energy Storage Systems (ESSs) in the case-study smart warehouse made it apparent that such an approach would be impractical for multiple reasons. Firstly, we discovered that there would need to be a magnitude of adjustable parameters to ensure the expected energy cost reduction related to the use of the ESSs. Secondly, adjusting the parameters over time to adapt to the changes in end-user energy consumption patterns and market pricing signals would be very time consuming. Therefore, we decided that a more robust and dynamic approach would be necessary. This led to the development and acquisition of a commercially available IEMS that employs stale machine learning techniques and optimization algorithms to generate a 48 hour schedule of ESSs charging and discharging based on predicted energy demand and local power production. However, this system still adheres to multiple adjustable parameters that have to be manually altered by the end-user and it also suffers an inability to react to unforeseeable events within each scheduled hour.

There is a research gap in the application of the state-of-the-art AI to efficient control of complex building-integrated energy systems featuring multiple energy storage solutions.

AI is an extensive field of research with a plethora of possible applications. In the domain of energy consumption it has been successfully applied to prediction (Gassar, Cha, 2020), design (Abdalla et al., 2021) and control problems (Ammari et al., 2022).

The main objective of this thesis is to examine a combination of DL and RL to model and control ESSs in a simulated environment of a smart warehouse. By doing so, we aim to improve the practical applicability and robustness of the state-of-the-art IEMSs through the application of AI.

Research Question 1: Can human and computational resource expenditure be reduced while accuracy is improved for coefficient of performance (COP) estimation of a smart warehouse's cooling system for IEMS application, considering the challenges of energy metering related to CO_2 refrigerant cooling plants?

Motivation: In order to enable intelligent energy and power management for warehouses that feature large-scale cooling installations with TES capabilities, such as food distribution warehouses, accurate estimations of the time-varying performance of individual components of the energy system are necessary. Considering the difficulties of energy metering associated with CO_2 refrigerant cooling plants, it is crucial to find a viable approach for accurate estimation of compressor efficiency and cooling system COP as input parameters for IEMS decision-making related to cost-efficient TES control.

Approach: An ensemble of fully connected Artificial Neural Networks (ANNs) with nonlinear activation functions is developed to predict compressor power consumption, internal working medium mass flow, and cooling system COP. Separate models for subcritical and transcritical operational modes are created for each transcritical compressor in the cooling system. Model input consists of cooling medium evaporation temperature, condensing temperature, suction gas temperature, and compressor frequency. Results are verified using metered laboratory data and comparisons to total cooling system power consumption. Please refer to Paper A and Paper B for more details.

Research Question 2: Can deep RL algorithms be designed, developed and applied for efficient and adaptable direct control of BESS in a smart warehouse, addressing the constraints of commercially available IEMSs and minimizing the requirement for extensive hyperparameter optimization?

Motivation: The commercially available IEMS installed in our smart warehouse has clear limitations, such as dependence on manual tuning by human experts and operation on an hourly basis, which may not adequately react to unforeseen events. To address these issues, we explore RL algorithms that can adapt to changing patterns through automated off-line training processes, interact with the warehouse energy system in an online manner, and minimize the time and effort required for hyperparameter tuning.

Approach: Deep RL agents are trained in a simulated environment featuring a battery control problem with the objective of energy cost reduction. The agents are tasked with controlling the battery in a continuous manner with full control over the charging and discharging setpoints. A new data-driven simulated training environment is built using operational data from the smart warehouse, and multiple RL agent algorithms are trained and compared with a benchmark established using an optimization algorithm given perfect information. Please refer to Paper C and Paper D for more details.

Research Question 3: To what extent can a deep RL-based approach be employed for the concurrent management and control of the BESS and TES within the smart warehouse environment, allowing for expeditious, scalable and robust implementation in an operational setting?

Motivation: Our ultimate research goal is to take important steps towards designing the backbone AI of an IEMS system that can adapt to a changing environment, handle complex dynamics and effectively control multiple components with different characteristics and overall impact on energy cost. The BESS and the TES in our case-study warehouse together inhabit the necessary characteristics for this endeavour. Simultaneously controlling these ESSs is the crucial last step in this Ph.D. thesis.

Approach: We examine the applicability of the COST-WINNERS Augmented Random Search (ARS)-ANN RL algorithm to a complex energy cost reduction problem through direct control of BESS and TES charging and discharging setpoints. The agent is trained in a simulated environment of the smart warehouse, which we mainly designed through the use of data-driven techniques. We have emphasized the use of data-driven techniques as a way to reduce the need for human expertise to design the simulated environment in order to increase the practical utility of our approach. We refer to Paper E for more details.

Limitations: In this thesis, the main aim is to explore the state-of-the-art AI techniques to reduce energy costs in a case-study smart warehouse. We have attempted to design a simulated environment using mainly data-driven machine learning, and we have trained deep RL algorithms to control BESS and TES with excellent performance. Our simulated environment is not fully data-driven, and the COST-WINNERS (introduced in paper E) algorithm has not yet been implemented and tested in our smart warehouse. The COST-WINNERS algorithm needs to be tested at a higher time step frequency to ensure that it has the ability to react in a close to on-line manner. Our approach requires high operational data quality both for training purposes and in operation. Practical implementation should also include an appropriate mechanism for live verification of data integrity to ensure appropriate and expected decision-making.

Overall, our approach to IEMS has shown very promising results in terms of high performance using deep RL in a data-driven simulated environment. This enables practical implementations of an IEMS that can be automatically retrained in an off-line simulated environment that will be continuously evolving along with its physical counterpart. Hence, we propose and employ a novel deep RL solution to energy cost reduction that should be robust, scalable, and self-improving. The heart of our solution is several versions of the COST-WINNERS algorithm combined with data-driven simulated training environments. In this thesis, for practical purposes the COST-WINNERS algorithm is interchangeably also called ARS-ANN, all though COST-WINNERS refers specifically to the ARS-ANN algorithm adapted to multiple simultaneous ESS control.

1.2 Publications

We solve each task of the thesis mainly using machine learning and deep RL. ANNs are employed to model cooling system performance, regression is employed to model dynamics in the thermal energy domain, and advanced RL algorithms are tested to control BESS and TES in our case-study smart warehouse. We list the contributions of this thesis below, each of which are described in detail in Chapter III, and the associated papers published are presented in their entirety in Part II of the thesis. Here, we present a summary of our papers:

Paper A Lacking important metrics such as thermal energy measurements and power consumption at the individual compression stages of our case-study warehouse cooling system, we attempt to develop theoretical models of the compressors by generating a data set through the use of compressor manufacturer software and available mathematical equations. An ANN is trained to model the compressors using operational data. The models are trained with cooling medium evaporation and condensation temperature, suction gas temperature, gas cooler outlet temperature and pressure, and compressor operating frequency. The output is the aggregated electrical power load and mass flow for the freezing stage compressors. The resulting average MSE of 0.08% conclusively shows that using an ANN to model the compressors in a cooling system is a valid approach that allows quick and accurate theoretical calculations of cooling load and compressor power. This paper mainly addresses Research Question 1.

Paper B In this paper, we expand our approach from paper A to the full ensemble of compressors featured in all the compression stages of the warehouse cooling system. Individual ANN compressor models are combined so that power input and thermal mass flow of the compression stages can be calculated. Furthermore, cooling system COP can be calculated as a whole or for each compression stage. To examine the practical applicability of the approach, we use laboratory data from the CO_2 cooling system at the Norwegian University of Science and Technology, as well as total power input measurements for the cooling system at our smart warehouse. The results show that the presented approach is relatively precise with a Mean Average Percentage Error (MAPE) as low as 5%, when constrained by low resolution and asynchronous data from the case-study cooling system. When tested in a laboratory setting, we achieve a MAPE as low as 1.8%. This paper mainly addresses Research Question 1.

Paper C In this paper, we examine deep Reinforcement Learning (RL) algorithms developed for game play applied to a battery control task with an energy cost optimization objective. We explore how agent behavior and hyperparameters can be analyzed in a simplified environment with the goal of modifying algorithm exploration of the action space for increased stability. Our modified Deep Deterministic Policy Gradient (DDPG) agent is able to perform consistently close to the optimum over multiple training sessions with a maximum cost reduction of 25% and an average cost reduction of 99% of the maximum in a simplified BESS environment. When environment complexity is increased by increasing the time frame of each episode, a modified Twin Delayed DDPG (TD3) agent is utilized to achieve an average of 99.9% of the optimal result compared to a GLPK optimization solver given perfect information. However, the amount of time required for algorithm tuning and enhancement was unsatisfactory in relation to our overall research goals. This paper mainly addresses Research Question 2.

Paper D In this work, the focus is on the application of the deep RL algorithm to the specific energy optimization problem of controlling a BESS in our smart warehouse. This paper adopts data from the real and operational BESS installed in the smart warehouse. We develop a simulated environment of the smart warehouse using data from the integrated photovoltaic power plant, local energy demand, historical ToU energy prices and more. We present the combination of the ARS reinforcement algorithm with ANNs as a potential backbone to the design of an IEMS tasked to control the energy flow of the BESS with the goal of energy cost minimization. An ANN replaces the simple input-output matrix used to parameterize the agent policy in the original ARS algorithm, allowing for more complex and abstract policies. The suggested algorithm shows very promising results, achieving an average of 99.2% accuracy across 10 seeded trials when compared with a GLPK optimization solver given perfect information. This paper mainly addresses Research Question 2.

Paper E In this paper we examine the applicability of the ARS-ANN RL algorithm to a complex energy cost reduction problem through direct control of BESS and TES charging and discharging setpoints in a simulated case-study smart warehouse. Our main research goal is to examine if the ARS-ANN algorithm can efficiently control multiple ESSs with different dynamics and substantially varying degrees of impact on energy cost. We hereby refer to this adaptation of the algorithm as COST-WINNERS. The agent is trained in a simulated environment of the smart warehouse, which we mainly designed through the use of data-driven techniques. We have emphasized the use of data-driven techniques as a way to reduce the need for human expertise to design the simulated environment to increase the practical utility of our approach. We continue development of the simulated smart warehouse by including a mathematical model of the TES. The various dynamical interfaces between the TES and the hydronic heating distribution system are modelled by simple machine learning techniques such as linear and polynomial regression. The COST-WINNERS algorithm is tasked to simultaneously control multiple ESSs, namely the BESS and TES. This paper mainly addresses Research Question 3.

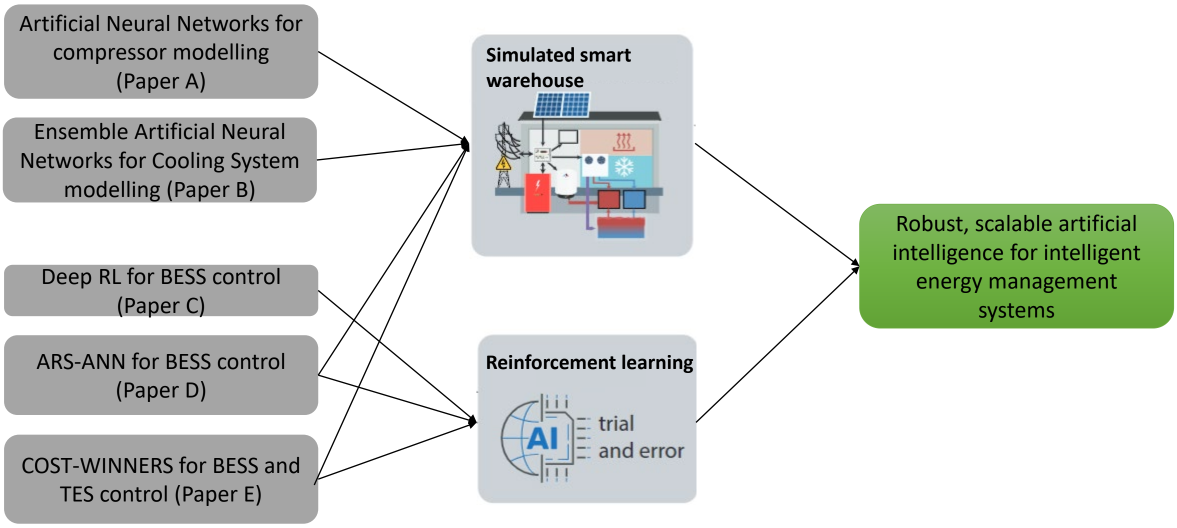

Artificial Neural Networks for compressor modelling (Paper A)
Ensemble Artificial Neural Networks for Cooling System modelling (Paper B)
Deep RL for BESS control (Paper C)
ARS-ANN for BESS control (Paper D)
COST-WINNERS for BESS and TES control (Paper E)
Simulated smart warehouse
Reinforcement learning
AI
trial and error
Robust, scalable artificial intelligence for intelligent energy management systems

Figure 1.1: Organization of Contributions in Research Area.

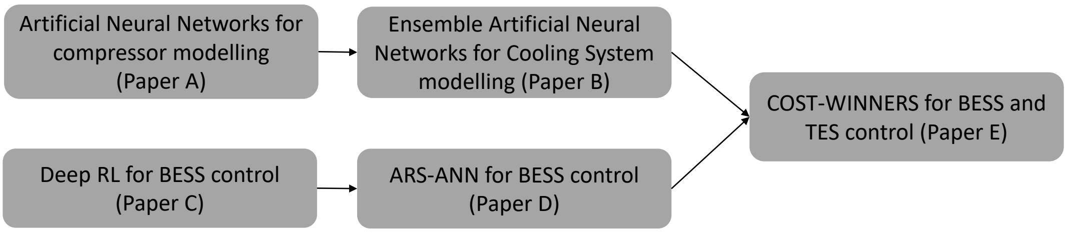

Artificial Neural Networks for compressor modelling (Paper A)
Ensemble Artificial Neural Networks for Cooling System modelling (Paper B)
Deep RL for BESS control (Paper C)
ARS-ANN for BESS control (Paper D)
COST-WINNERS for BESS and TES control (Paper E)

Figure 1.2: Organization of Contributions in Hierarchy.

Each of the articles listed above represents individual components of a practical methodology to the design of a robust and scalable IEMS. The main characteristics of our IEMS design methodology is the use of deep RL trained in an offline environment consisting of an ensemble of mainly data-driven machine learning models of individual energy system components and dynamics. Mathematical models of individual components can be included for practical purposes, but can also easily be replaced once available data increases to an amount that makes machine learning techniques more viable. Our design philosophy is modular in the sense that both individual component models and the deep RL algorithms can be replaced as the research field advances further. An illustration of the research articles by contribution to main research area is presented in Fig. 1.1, whereas Fig. 1.2 visualizes the hierarchy of the articles.

1.3 Thesis outline

The dissertation is organized into two parts. Part I contains an overview of the work carried out throughout this Ph.D. study and Part II includes a collection of five published, accepted or submitted papers, which are mentioned in the list of publications. In addition to the introduction chapter presented above, the following chapters are included.

- Chapter II presents the theoretical background within the various research fields related to this thesis, such as machine learning, reinforcement learning, and energy system modeling and control.

- Chapter III describes the components of the operational smart warehouse energy system we use as the basis for data-driven modeling and energy control.

- Chapter IV details the contributions of this thesis. The two main sections are dedicated to data-driven energy system modeling and reinforcement learning for intelligent energy management systems. The methodologies, experiments and results for each contribution are explained extensively.

- Chapter V concludes Part I of the thesis and discusses the implications of the outcomes of the thesis. It also contains potential future research directions that can further improve the work presented in the thesis.

- In Part II of the thesis, all publications associated with the thesis are presented in their entirety. There are five publications labeled as Paper A to E. The papers are listed in chronological order according to their time of publication.

Chapter 2

Background

In this Ph.D. study we propose the state-of-the-art machine learning techniques to solve a complex energy cost optimization problem to form the backbone of a robust and scalable IEMS.

In this chapter, we briefly describe the background and preliminary information needed to understand the thesis. First, we introduce the various AI research areas associated with this thesis, starting with ANNs and other machine learning techniques that have been used to model components in a simulated warehouse environment we developed for RL agent training. The second section introduces RL and the algorithms we have adopted in experiments throughout this Ph.D, where deep ANNs play a crucial role.

2.1 Control systems for energy cost reduction

Many approaches to energy cost reduction in an operational setting can be found in the scientific literature. For optimizing energy cost and power flow in a Direct Current (DC) microgrid Sechilariu et al. (2014) proposed Mixed Integer Linear Programming (MILP). The approach is similar to the Intelligent Energy Management System (IEMS) already implemented in our previously described case-study smart warehouse. It includes load and Photovoltaic (PV) prediction, a human-machine interface, and energy management. Huang et al. (2015) proposed a hybrid (MPC) for energy cost optimization in a case-study airport terminal building. The authors introduce Neural Networks as a way to handle non-linearity. Another MPC approach was suggested in Lešić et al. (2017) using hierarchies of multiple MPCs for energy cost optimization and thermal comfort control. A data-driven MPC, i.e., Data Predictive Control (DPC), was proposed in Smarra et al. (2018). The authors suggested using random forests for predictions and argued that intelligent control systems that require physical models of buildings are not practical due to high complexity and variance in building design. Wang et al. (2020) propose MPC for control of a dual BESS connected to a wind power farm. Based on simulations, the authors claim improved wind farm dispatchability, and extended battery life as their results. Barbato, Capone (2014) conducted a survey to describe various optimization techniques designed to solve Demand Side Management (DSM) problems for end-users in smart grid scenarios. They conclude that although researchers had undergone extensive work in this field of research, many research questions remained unanswered. Mariano-Hernández et al. (2021) conducted a review of various strategies for Building Energy Management Systems (BEMS), including MPC, DSM, and optimization. The authors found MPC to be the most used management strategy in non-residential buildings and conclude that the building model will be critical to ensure intelligent control in future research. Rätz et al. (2019) describe a methodology for automated data-driven modeling of energy systems in buildings that could be applicable to MPC and RL.

Battery Energy Storage Systems (BESS) built with lithium-ion technology are increasingly deployed in both macro and micro scale projects (Stroe et al., 2017). For optimal utilization of the BESS for multiple purposes such as energy cost reduction, reducing peak power demand and frequency regulation, intelligent control systems that balance the need for longer-term planning with immediate response are required. For such systems, many approaches have been suggested, including constrained non-linear programming (CNLP) optimization for aggregated two-stage control in a micro-grid in Long et al. (2018) achieving a 30% energy cost reduction when combined with peer-to-peer energy sharing, a rule-based approach for many distributed batteries in a data center with a focus on accurate battery health modeling in Aksanli et al. (2013) and a rule-based scheme for PV and wind application in Teleke et al. (2010). When considering the dynamic and ever-changing nature of building-integrated energy systems, it seems unlikely that a rule-based approach can be implemented without extensive follow-up and revision. In related research, Siqueira de, Peng (2021) conducted a review of control strategies for smoothing wind power output, finding Model Predictive Control MPC to be the most common for multi-objective optimization. Lipu et al. (2021) discussed various approaches to intelligently control battery management in electric vehicles.

2.1.1 Rule-based

We observe a decrease in rule-based approaches with an increase in energy system complexity. Nevertheless, a rule-based approach is a simple and understandable solution to many control tasks in a buildings energy system. A rule-based system is characterized by control algorithms that follow a set of rules, often decided by using human expert knowledge. Examples of rule-based control algorithms for energy cost reduction are ventilation temperature set points proportionally adjusted to ambient temperature or Heating, Ventilation, Air-Conditioning (HVAC) operational calendar scheduling. In simple, unchanging systems, this is often the most transparent and robust solution. However, for dynamic and evolving systems the challenge is to design rules that are relevant for all operational scenarios. Maintaining the ruleset and updating values manually is another dimension that requires human expert knowledge.

2.1.2 Mixed integer linear programming

MILP relates to optimization problems with both continuous and integer variables. The simplest, but least effective way to solve such problems is to use exhaustive search. The most common, featured in the open-source GLPK solver, is the branch and bound method.

Essentially, the branch and bound method works by calculating candidate branch solutions within an upper and lower bound. When applied to energy cost optimization problems in the temporal domain, a MILP approach could be a component of MPC. An important prerequisite for MILP in this instance is accurate predictions and/or models of input variables, which can be challenging. In cases where a large component of the energy costs come from peak power tarriffs, making decisions based on inaccurate predictions or models can lead to overall increased energy costs. Since prediction and optimization algorithms can be computer resource intensive, combining these systems with rule-based algorithms to avoid peak power costs can be prudent. The overall design of a robust IEMS solution based on MILP requires expert knowledge of the energy system and can be quite time-consuming.

2.1.3 Model predictive control

MPC is an established theoretical approach to complex HVAC control, although not many examples of practical application exist. According to Afram, Janabi-Sharifi (2014), advantages of MPC control include the use of a model to enable proactive rather than reactive control, the ability to handle temporally variable dynamics, and the use of a variety of optimization algorithms to achieve multiple objectives through a well-defined cost function. The model can essentially be designed in three different ways:

1. A transparent physical model.

2. A so-called “Grey-box” with a physical model where unknown parameters are tuned with operational data.

3. Black-box with a purely data-driven approach.

In an operational setting a hybrid between these designs where data availability and component complexity would be used to determine the most practical modeling approach is perhaps the most realistic scenario.

As an example, Goldsworthy et al. (2022) has successfully implemented a cloud-based Model Predictive Control (MPC) battery control algorithm for energy cost reduction at an office building. The system has been operational for a year and achieved an energy cost reduction of 5.5%.

2.2 Supervised machine learning

In this section, we describe the areas of supervised machine learning that are fundamental to the topics investigated and adopted in this study. Supervised machine learning is a broad field of study of computer algorithms that learn or improve from data (Jordan, Mitchell, 2015). We focus on some of the main techniques applied to our research, namely ANNs, whereas RL, another large research field within machine learning, is described in the next section.

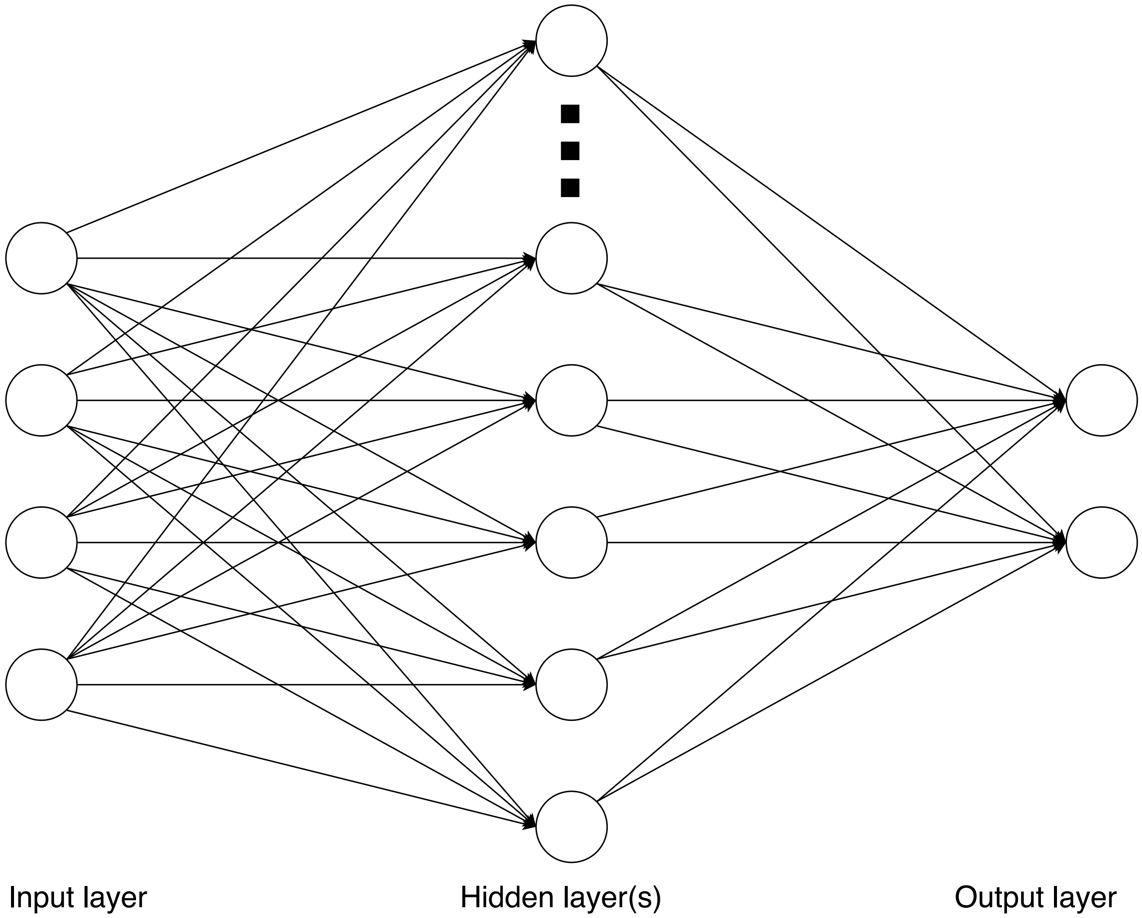


Figure 2.1: Fully connected artificial neural network with a single hidden layer.

2.2.1 Artificial neural networks

ANNs have the ability to approximate both simple and complex unknown functions that fit the underlying data. ANNs come in many forms, but the most common kinds feature an input layer, one or more hidden layers, and an output layer (see Fig. 2.1). Each layer consists of so-called neurons, named after the neurons in the human brain. The neurons in the input layer represent the chosen input parameters, passing these directly to the neurons in the first hidden layer. Usually, the hidden layers are fully connected, meaning each neuron in the hidden layers is connected to each neuron in the previous layer. The values are passed along the connections and summed before an activation function is applied to determine the output value of each neuron. The ANN is trained or updated by propagating the measured error of the output backwards through the same layers. At each node in the ANN, its numerical activation value consists of the activation function applied to the sum of the weighted input values. Backpropagation facilitates learning by updating the input weights according to the gradient of the error. Two of the most common activation functions are the hyperbolic Tangent (Tanh, Eq. 2.1) and the Rectified Linear Unit (ReLU, Eq. 2.3). Finally, the output layer neurons process the values according to the desired application by applying the output layer activation function to each output neuron. The number of neurons in the output layer corresponds to the desired number of outputs.

Activation functions: There are many available activation functions, chosen depend-

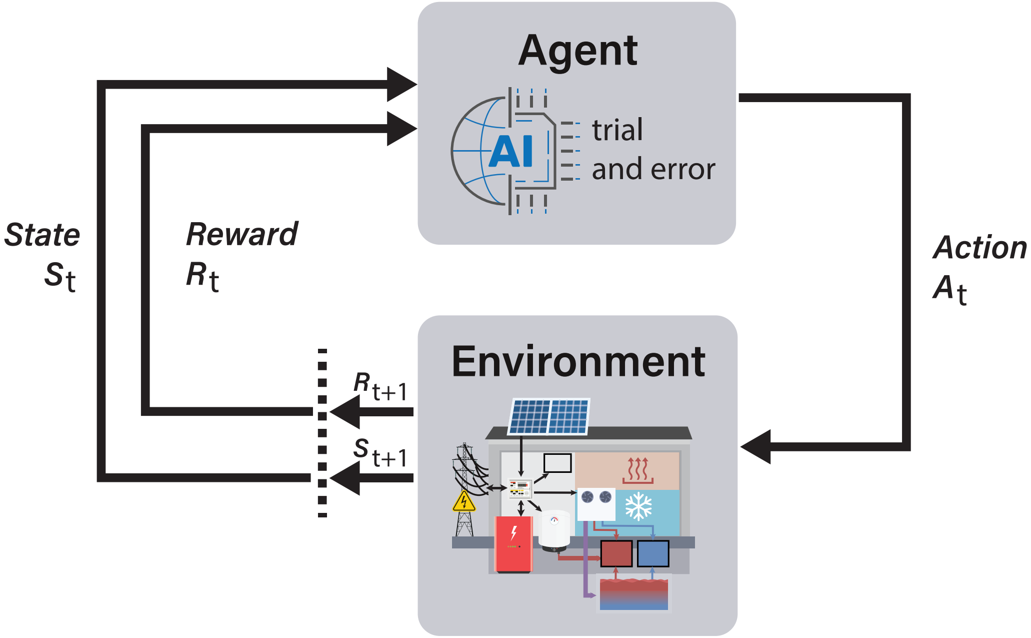


Figure 2.2: Markov decision process interaction between agent and environment.

ing on the nature of the neural net and the desired application. The most common activation functions include Sigmoid (Eq. 2.4), Tanh (Eq. 2.1), ReLU (Eq. 2.3) and softmax.

$$Tanh(x) = \frac{\exp^x - \exp^{-x}}{\exp^x + \exp^{-x}}. \qquad [\mathrm{W\,m^{-2}}] \tag{2.1}$$

$$Tansig(x) = \frac{2}{(1 + e^{-2x}) - 1}. \tag{2.2}$$

$$ReLU(x) = \max(0, x). \tag{2.3}$$

$$Sigmoid(x) = \frac{e^x}{1 + e^x}. \tag{2.4}$$

2.2.2 Deep learning

Deep learning is a field within AI and machine learning that focuses on extracting patterns from data through a hierarchy of increasingly complex abstractions (Goodfellow et al., 2016). The most common implementation of deep learning is characterized by passing data through the multiple hidden layers of a deep ANN.

2.3 Reinforcement Learning

According to Sutton, Barto (2018), RL is learning by discovering what actions to take to maximize a reward. Experiments with simulated environments are often designed for agents to learn, by trial and error, how to maximize a numerical reward signal, often binary in nature. It is common for researchers to design a reward function to reward desired behavior and, in some cases, to penalize unwanted behavior. The reward function may be updated if the desired behavior changes over time in an operational scenario, even if the overall goal is unchanged.

RL researchers commonly model the problem as a finite Markov Decision Process (MDP). The iterative process is illustrated in Figure 2.2. An agent interacting with an environment through actions receives numerical feedback from the environment in the form of a reward or penalty. The agents' actions may affect the environments' internal state as a direct or partial consequence. The environment determines which actions are available to the agent, and the action space is usually either a constant set of discrete numbers, a continuous range of floats, or decided for each new state as would be the case in a game of chess. The agent determines what actions to take by following its internal policy π. The policy usually includes a mechanism that allows the agent to explore alternative actions outside the most strict interpretation of its policy to be able to discover new states and actions that have the potential to generate higher rewards. Upon such discoveries, various methods exist to update the policy according to the newfound knowledge. Finally, a crucial element in RL is the $value function$ that defines the value of a state through probabilities related to actions, rewards, and future states. Sutton, Barto (2018) referred to the Bellman equation as the definition of the value of a state while following the policy π:

$$V_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s'r|s,a) \left[r + \gamma V_\pi(s')\right], \forall s \in S. \tag{2.5}$$

The Bellman equation describes the value of the state s while following the policy π as the sum of the probability of taking action a in the state s, multiplied by the sum of the probability of arriving in each state s' and receiving reward r, multiplied by the sum of r and the discounted (γ) expected value of the future state s'. The Bellman equation is a central part of RL theory and research.

Wang, Hong (2020) conducted a survey of RL application to control of technical systems in buildings. The authors argued that established techniques such as MPC require extensive expert human knowledge to properly design and implement, making it less applicable in the building control domain compared to mass production domains such as the automobile industry. Furthermore, Wang, Hong (2020) stated that RL combined with transfer learning should be further explored for building control.

Importantly, there are also examples of more advanced state-of-the-art algorithms in the literature. Mocanu et al. (2019) use Deep Policy Gradient (DPG), similar to DQN, for binary scheduling of flexible residential consumer loads. Wan et al. (2018) propose a variant of Deep Deterministic Policy Gradient (DDPG), from Lillicrap et al. (2015), for residential BESS control. An improved DQN is suggested in Cao et al. (2020) for BESS arbitrage. This algorithm includes a lithium-ion battery degradation model, with discretized action space for full or 50% capacity charging and discharging in addition to stand-by. Shang et al. (2020) propose a DQN with bootstrapping combined with Monte Carlo tree search to control a BESS in a microgrid. The authors in Xu et al. (2021) propose a combination of RL (Q-learning) with differential evolution to reduce energy cost for industrial users with combined solar power and thermal energy production, as well as BESS and TES, while satisfying local energy demand and trading energy in an energy trading platform.

However, in all the aforementioned cases except Wan et al. (2018), the algorithms work in discrete domains and therefore have limited action space. In addition, in many cases, the reward functions are quite sophisticated and tailored to a specific experiment. The above-mentioned approaches are not ideal for enabling large-scale adoption and quick implementation of IEMS using RL due to complicated algorithms and reward functions, or simplified action spaces that reduce the cost savings potential of the systems.

2.3.1 Q-learning

At the foundation of many RL algorithms is Q-learning (Watkins, 1989). The original Q-learning algorithm is a table-based mapping of states to the Q-values of all possible actions. The Q-value is a mathematical estimate of the expected discounted future value of the action. The state space and the action space have to be discrete and finite. The agents' policy is encoded in the Q-table, where each state has a corresponding Q-value for each possible action, and the deterministic version of the policy consists of choosing the action with the highest Q-value. The mechanism for exploring actions outside the policy in Q-learning consists of adding a random component to a fraction of the actions taken. As stated in Perera, Kamalaruban (2021), most of the RL employed in the energy domain uses Q-learning, even if simpler algorithms are still deployed. Q-learning is essentially a table-based approach, mapping an environment state to Q-values for each possible action. The Q-table is updated according to

$$\begin{aligned} Q(S_t, A_t) \leftarrow & Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - \\ & Q(S_t, A_t)], \end{aligned} \tag{2.6}$$

where S is the state and A is the action selected. The learning rate α is applied to the sum of the reward R at time $t+1$ and the discounted (γ) estimated max future Q-value at state S_{t+1}, minus the existing Q-value.

The following researchers have applied variations of Q-learning to many problems and challenges within renewable energy, energy storage, and complex energy systems. Kuznetsova et al. (2013) simulated a microgrid consisting of a wind turbine and a BESS connected to a power grid. The author's approach uses Q-learning taking as inputs energy price, battery State Of Charge (SOC), wind energy predictions, and energy demand. The agent can choose between three discrete actions: Battery charging, battery discharging, or stand-by. Kuznetsova et al. (2013) claim that their approach is a framework for understanding and exploring stochastic energy systems. Wen et al. (2015b) also propose using Q-learning and end-user device utilization for controlling the temporal shift of flexible loads in small offices and residential buildings. Mbuwir et al. (2017) suggest Fitted Q-iteration as the basis for transfer learning of battery control to and from BESS with similar characteristics. Henze, Schoenmann (2003) examine Q-learning for control of a Thermal Energy Storage (TES) in a simulated environment.

In recent years, several significant breakthroughs have been made in applying a combination of deep learning and RL to various games. Simulated environments that allow numerous swift learning iterations with clearly defined numerical reward signals have proven to be fertile ground for the exploration of these algorithms. The starting point for much of this development was when Mnih et al. (2013) introduced a deep ANN adaptation of basic table based Q-Learning called deep Q-learning (also known as Deep Q-Networks, DQN), demonstrating state-of-the-art results in six out of seven Atari 2600 games. DQNs replace the Q-table with an ANN such that the output of the neural network is the Q-values of all possible discrete actions in a given state. The objective in Mnih et al. (2013) and further in Mnih et al. (2015) is to explore the effects of advances in computing power and deep learning on the common RL bench-marking task of Atari 2600 gameplay performance. The Q-table is encoded into the weight parameters of a deep ANN, specifically a deep convolutional neural network, and the weights, θ, are updated according to

$$\begin{aligned}\theta_{t+1} =& \theta_t + \alpha[R_{t+1} + \gamma \max_a \hat{Q}(S_{t+1}, a, \theta_t) - \\ & \hat{Q}(S_t, A_t, \theta_t)] \nabla \hat{Q}_\theta(S_t, A_t, \theta_t),\end{aligned} \tag{2.7}$$

where θ_{t+1} are the weights at time $t+1$, θ_t are the weights at time t and $\nabla \hat{Q}_\theta(S_t, A_t, \theta_t)$ are the partial derivatives of the state-action pair value approximations with respect to the weight vector θ_t. Instead of updating the weights after every action according to the current sequence of actions, the algorithm draws random mini-batch samples from an experience replay memory database to update the DQN weights through stochastic gradient descent. In paper C, we demonstrate the applicability of DQN to control a BESS following a time-of-use energy pricing profile as a benchmark.

2.3.2 Deep reinforcement learning

Many breakthroughs in RL in recent years come from algorithms developed for gameplay in various benchmarking tasks. Adapting and applying the most promising algorithms to energy optimization tasks has been an avenue of research we have explored in this thesis. Below, we present some of the important developments in deep RL in recent years. AlphaGo mastered the extremely complex, but highly intuitive turn-based game of Go through a combination of supervised learning (pre-training with human-generated example data) and deep RL Silver et al. (2016), resulting in a 4-1 defeat of 18-time World Champion Lee Sedol. The achievements of AlphaGo have since been surpassed by AlphaGo Zero through *tabula rasa* deep RL without any human knowledge in Silver et al. (2017), where AlphaGo Zero defeated AlphaGo 100-0. Central to these algorithms is the concept of self-play to generate an experience replay database from which random samples are utilized for training. This was further explored in Silver et al. (2018b) for the games of Shogi and Chess, leading to similarly impressive results. The AlphaZero algorithm uses neural networks to estimate action probabilities and a Monte-Carlo tree search algorithm for future move-sequence analysis. A more recent development is the MuZero algorithm introduced in Schrittwieser et al. (2019). Where AlphaZero is informed of the environment dynamics, i.e., the rules of the game, MuZero differs by having to learn a model of the environment starting from scratch. This constitutes a significant step toward the real-world application of deep RL with stochastic and partially unknown environment dynamics. More recently, Badia et al. (2020) achieved state-of-the-art performance in the popular Atari games benchmark. The Agent57 algorithm is the first to outperform the standard human benchmark in all 57 games. The algorithm includes training an artificial neural network encoding multiple policies with varying degrees of exploratory behavior. Addressing the challenge of increasing data and training sample efficiency, Schrittwieser et al. (2021) introduced MuZero Unplugged as a more sample efficient version of the MuZero algorithm adapted to off-line RL, and Fan, Xiao (2022) introduced Generalized Data Distribution Iteration (GDI) which according to the authors reduced data consumption by 500 times compared to Agent57.

Efforts to explore the state-of-the-art deep RL for energy optimization have also been made. Deep RL for online dynamic binary consumer load scheduling in households is described in Mocanu et al. (2019). Availability of locally produced solar electricity, energy price, and peak shaving are all considered. Data is extracted from the PecanStreet database and used to model households on individual and aggregated levels. The proposed algorithm, Deep Policy Gradient (DPG), replaces the output Q-values in a DQN with an estimated probability of taking action a in state s_t, thus allowing for multiple simultaneous discrete actions to be selected. DPG is found to outperform a DQN modified for simultaneous action selection through action grouping. Wei et al. (2015) proposed dual iterative Q-learning neural networks to reduce energy cost with optimal battery control. The dual iteration relates to an internal iteration j to reduce energy cost for each episode of 24 hours, and an external iteration $i \to \infty$ to update a defined performance index function towards its optimum. The overall claim is that the dual iteration is necessary due to the time dependent nature of the optimal Q-function, $Q^*(S_t, A_t, t)$. The neural networks are used in an actor-critic setup, denoted action and critic networks by the authors. Numerical results show improved performance over particle swarm optimization and time-based DQN. Residential battery control with deep RL is explored in Wan et al. (2018). The algorithm can be characterized as Deep Deterministic Policy Gradient (DDPG), first proposed in Lillicrap et al. (2015), and consists of actor-critic deep neural networks, specifically recurrent neural networks using gated recurrent units (Cho et al., 2014). The actor network utilizes policy gradient for parameter updates while the critic network utilizes a squared Q-value loss function. Results are compared with the theoretically lowest energy cost calculated by an optimization algorithm and a do-nothing scenario with a clearly favorable, but not optimal outcome. Zhang et al. (2021) proposed Soft Actor-Critic SAC to optimize BESS control with multiple energy production facilities. However, the authors have not clarified if the experiment is based on more than a single 24 hour episode and results are only compared with other simpler RL algorithms. We expand on the details of the DDPG and SAC algorithms later in Subsection 2.3.3. DQN and DDPG are both explored in paper C, whereas SAC is featured in papers D and E.

2.3.3 Actor-critic policy gradient algorithms

Actor-critic RL algorithms are characterized by the agent having separate policy and value functions. For instance, deep RL actor-critic algorithms separately train ANNs for value estimation (critic) and action selection (actor). One such algorithm is the DDPG algorithm, first proposed in Lillicrap et al. (2015) and adopted by Wan et al. (2018). DDPG is an actor-critic RL algorithm with four ANNs – the actor policy network μ, the critic network Q and their respective *target networks*. The target networks weights, θ', trail the main networks weight parameter updates, θ, through

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta', \tag{2.8}$$

where the target networks function as a mechanism for improving stability by using them to estimate the value of the following state while the main networks are used for current state value estimation. Training the main networks is carried out through the use of an experience replay database R that holds transitions (s_i, a_i, r_i, s_{i+1}) for each step in every training episode. The algorithm samples a random mini-batch N of non-sequential transitions from R and uses the target actor $\mu'(s|\theta^{\mu'})$ to predict actions a'_{i+1} for every new state s_{i+1} in the mini-batch. A temporary state-action value is then calculated using the target critic network as

$$y_i = r_i + \gamma Q'(s_{i+1}, a'_{i+1}), \tag{2.9}$$

and the main critic network updated by minimizing the mean squared error between y_i and $Q(s_i, a_i)$ for every transition in the mini-batch. Finally, the main actor network can be updated from the same mini-batch by first calculating new actions a_i from current states s_i with the main actor μ. The gradients for the main Q network weights θ^Q with respect to a_i, and the gradients for the main policy network μ with respect to its parameters θ^μ are then used to approximate the gradient of the policy network cost function J with respect to θ^μ, by sampling as shown in Silver et al. (2014):

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_{a_i} Q(s_i, a_i|\theta^Q) \nabla_{\theta^\mu} \mu(s_i|\theta^\mu). \tag{2.10}$$

The approximated gradients are then applied to θ^μ using an optimizer, such as Adam Kingma, Ba (2014), with an adjustable learning rate α that determines the step size of each update.

Recent improvements suggested in Fujimoto et al. (2018) with Twin Delayed DDPG (TD3) include the adoption of clipped dual Q-networks to avoid Q-value overestimation by only considering the most conservative output, delayed updates of the actor networks compared to the critic networks and adding noise to the target network predictions during training. Another development of DDPG is the SAC proposed in Haarnoja et al. (2018), introducing entropy regularization for exploration combined with clipped dual Q-networks. Actor network output layers are configured with a hyperbolic tangent (tanh) activation function, whereas critic network outputs are linear. DDPG and TD3 are both explored in paper C, whereas SAC and TD3 are featured in papers D and E.

2.3.4 Augmented random search

ARS is a more efficient version of what the authors (Mania et al., 2018) term basic random search due to the various mechanisms in the algorithm that targets the search towards higher rewards. The authors designed ARS to work with a simple linear policy, unlike the direction that many other RL researchers are taking, and it also operates in continuous action space. Additionally and in contrast to most other RL algorithms, exploration with the ARS is done directly in the parameters of the policy function by randomly making minute changes to the parameter weights. In other words, the algorithm directly manipulates the parameters of the linear policy function to search for a policy that generates higher rewards. In contrast, well-known algorithms for continuous action space such as DDPG (Lillicrap et al., 2015), SAC (Haarnoja et al., 2018), Trust-Region Policy Optimization (TRPO) (Schulman et al., 2015) and TD3 (Fujimoto et al., 2018) all add a random component to the agent output action to encourage exploration. For ARS, the parameter space is explored by generating a table of random noise and adding the noise to the policy parameters in both positive and negative directions. The new parameters are tested by running an episode and collecting the reward. N such tests, termed rollouts, are performed and sorted by reward in descending order (Mania et al., 2018). The top b directions are then chosen and used to update the policy according to

$$\theta_{j+1} = \theta_j + \frac{\alpha}{b\sigma_R} \sum_{k=1}^{b} \left[r\left(\pi_{j,(k),+}\right) - r\left(\pi_{j,(k),-}\right) \right] \delta_{(k)}, \tag{2.11}$$

where θ represents the parameters of the policy, α is the learning rate, σ_R is the standard deviation of the rewards, $r(\pi_{j,(k),+})$ and $r(\pi_{j,(k),-})$ are the sorted rewards from positive and negative rollouts and $\delta_{(k)}$ is the randomly generated noise of the same size as θ. The mean and standard deviation of input variables are continuously updated and used to normalize input values. The authors demonstrate impressive performance across a wide range of known RL benchmark problems while also vastly decreasing computational resources required for training.

2.4 Summary

In this chapter, we have introduced and explained the most important background concepts and research areas underlining this thesis. Our research touches upon a broad area of subjects, such as control of complex renewable energy systems, machine learning, reinforcement learning, and energy system modeling. Specifically, in a substantial part of our research, we use ANNs in various settings to enable complex and abstract pattern recognition. This thesis combines techniques from the aforementioned areas to design a RL backbone to an IEMS capable of reducing the energy cost of complex energy systems with multiple ESS' in a robust and self-improving manner. We have employed ANNs in all parts of the thesis (Papers A, B, C, D, and E) and reinforcement learning for ESS control in a significant portion of the research (Papers C, D, and E).

Chapter 3

Smart warehouse energy system

This chapter details the smart warehouse energy system that is part of this Ph.D. research. The research in this study is based on existing infrastructure and data from a 27,000 m^2 technologically advanced warehouse for food storage and distribution, located in Sandnes, Norway. We use data and component specifications from this warehouse energy system in all of the research in this thesis. The warehouse was completed in 2017 and features a commercially developed IEMS based on hourly scheduling that uses various machine learning techniques to generate an optimized schedule for the utilization of a locally installed Battery Energy Storage System (BESS) and an insulated firewater tank is also used as a Thermal Energy Storage (TES) for storage of heated or chilled water. The IEMS has to predict future electrical and thermal energy demand and uses external energy pricing information to create an optimal scheduling strategy. The main components of the warehouse energy system are a 1 MWp solar PV power plant, a 460 kWh storage capacity electrochemical li-ion BESS with two 100 kW inverters, a 300 m^3 TES, a CO_2-based large scale cooling system consisting of three two-stage cooling plants and a back-up cooling machine for ventilation air and IT-server cooling, an electric boiler, and accompanying technical infrastructure (HVAC, Lighting, etc.). The heat from the Cooling System (CS) is reclaimed and used to heat the building. An overview of the warehouse temperature zones with their respective operating temperatures is listed in Table 3.1, whereas the main components of the energy system and their interdependencies are visualized in Fig. 3.1 and listed in Table 3.2.

Table 3.1: Warehouse cooling floor area and operating temperatures.

Area	Size	Operating temperature
Warehouse	27 000 m^2	-20°C to +20°C
Frozen storage area	3 000 m^2	-20°C
Cold storage area	3 600 m^2	+2°C
Cooled shipping area	3 600 m^2	+2°C

In the following sections, the components of the warehouse energy system and their intersections are explained in more detail. We also highlight some of the technical challenges that need to be solved when designing an IEMS to control the systems.

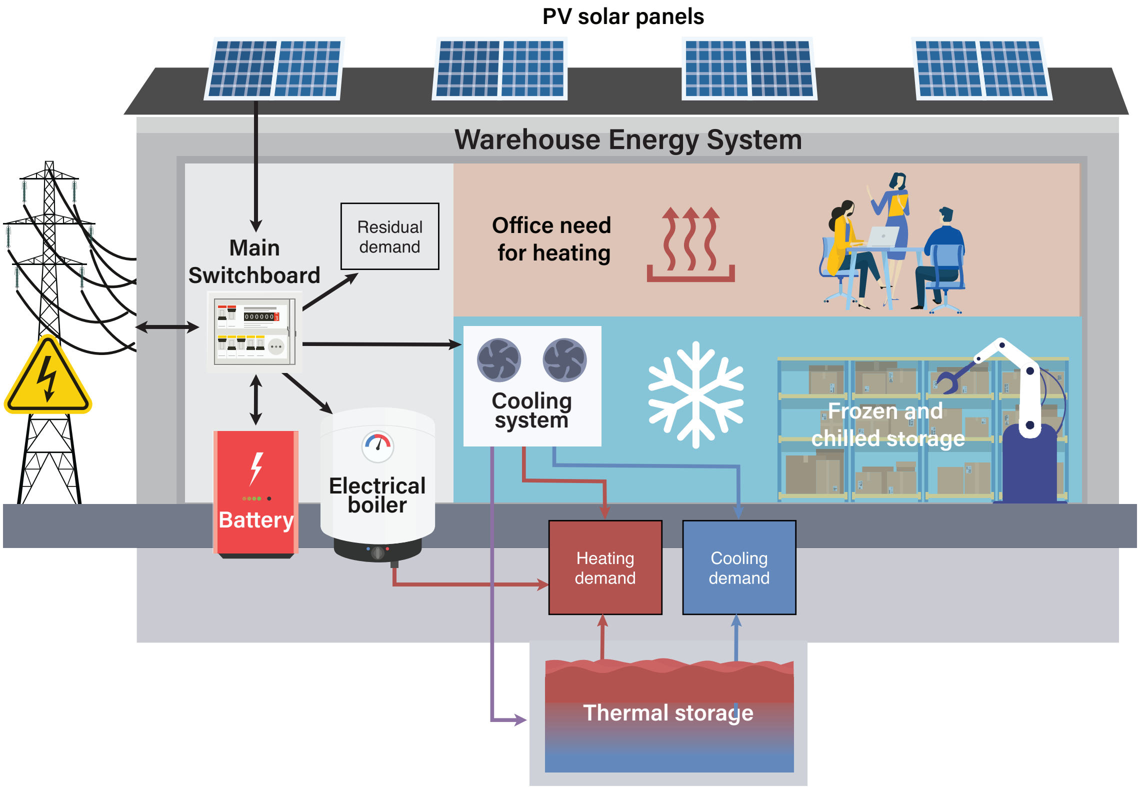


Figure 3.1: The smart warehouse energy system with BESS, TES, cooling system and PV power plant (Opalic et al., 2022). Arrows indicate the direction of energy flow.

Table 3.2: Main components of the smart warehouse energy system.

Component	Capacity	Unit of measurement
Photovoltaic solar panels	1,018	[kW$_p$]
Lithium-ion battery energy storage system	460/200	[kWh/kW]
Cooling system	1,140	[kW$_{thermal}$]
Thermal energy storage system	300/300	[m^3/kW$_{thermal}$]

3.1 Photovoltaic solar panels and battery energy storage system

The warehouse energy system features the combination of a PV solar plant and a BESS. The PV power plant has an installed power production capacity of 1 MWp and produces around 830 MWh of energy annually. It features 1000 V modules in a southwest by northeast configuration at an incline of 10^o.

Being able to plan for future PV production and react to changes in current PV production, implicitly or explicitly, are important requirements of an IEMS to extract maximum value out of the locally produced electrical energy. For instance, maximizing local consumption of PV production through the use of ESSs will reduce total energy costs by reducing grid tariff costs related to energy purchase, given that the total energy consumption remains the same.

BESSs built with lithium-ion technology are increasingly deployed in both macro and

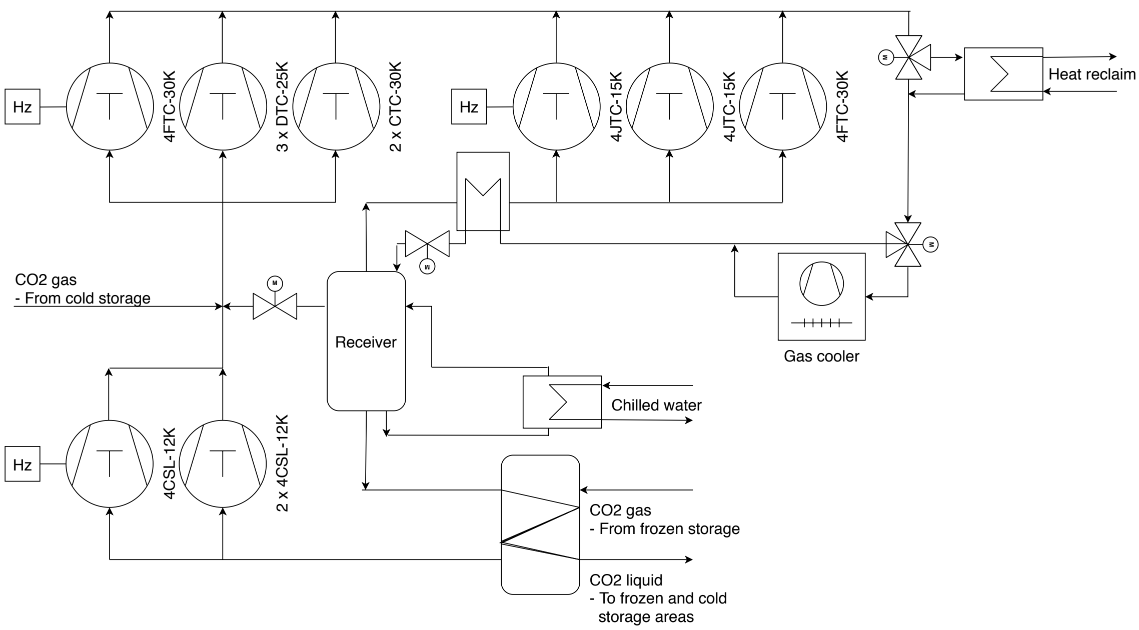


Figure 3.2: On-site cooling plant architecture showing both compression stages and actual compressor types (Opalic et al., 2020).

micro scale projects (Stroe et al., 2017). The case-study warehouse energy system uses a BESS to both enable peak power demand shaving and increase the self-consumption of locally produced energy from the PV power plant. AC current is converted to DC current through two 100 kW inverters for energy storage, and the opposite occurs during discharge. The BESS is currently configured to be remotely controlled by the selection of battery operating mode and a setpoint value for charging or discharging magnitude. During the design phase, the BESS's total energy storage capacity of 460 kWh was calculated so that as much of the PV energy as possible could be used locally.

A BESS configuration with 200 kW total inverter capacity and 460 kWh electrical energy storage was used in all research papers in the thesis involving BESS control (Papers C, D, and E).

3.2 Cooling system

The warehouse's main cooling system is an industrial CO_2-refrigerant cooling system consisting of three separate cooling plants. The cooling process operates by circulating liquid CO_2 to evaporators in the frozen and chilled food storages where it evaporates after valve injection. The cooling system also produces chilled water for cooling the remaining building areas, including food storage, office space, and support areas. The architecture of the main cooling system is shown in Fig. 3.2. An additional backup and peak-load cooling machine provides chilled water for ventilation and server cooling at peak demand. Surplus heat is recovered and utilized to heat tap water, to keep the ground beneath the frozen storage frost-free, and to supply the non-cooled areas of the building with heating energy when needed. If there is insufficient excess heat available, the operating pressure can be increased to satisfy the heating demand, up to a predefined maximum pressure level.

Recovered heat can also be stored in the TES for future use, mainly to reduce the need for the electrical boiler at peak heating demand.

The compressors are semi-hermetic reciprocating compressors manufactured by Bitzer GmbH, with one frequency-controlled compressor at each stage. Fig. 3.2 shows the placement of all the compressors in a simplified cooling system architecture. There are two pressure stages of compression as well as parallel compressors to handle flash gas in the receiver and chilled water production. The compressors for the frozen storage areas are displayed in the bottom left, with the cold storage compressors in the top left and the parallel compressors in the top right. Fig. 3.2 also displays mass flow direction and the most crucial CS components. It can be noted that the CO_2 based cooling system is a highly complex part of the energy system in the considered technologically advanced warehouse.

Data from the cooling system is used in papers A, B and E. In papers A and B we model the compressor efficiency and cooling system performance using ANNs, whereas in paper E we use data generated with the ANN models to build a data-driven RL training environment.

3.3 Thermal energy storage

Table 3.3: Thermal energy storage system characteristics.

Attribute	Values	Unit of measurement
Measurements in LxWxH	12x10x2,5	[m]
Volume	300	[m^3]
Average U-value	0.20	$[\frac{W}{m^2K}]$
Storage medium	Water	N/A
Heat exchanger max flow	25	$[\frac{m^3}{h}]$
Heat exchanger temperature loss	2	$[^oK]$

The TES is employed to hydronically store both heating and cooling energy by switching between two separate seasonal modes of operation, hereby denoted Heat Energy Storage (HES) and Cold Energy Storage (CES). Switching between heating and cooling storage incurs significant cost due to the difference in the operational temperature levels of the heating and cooling distribution systems at 50°C / 25°C and 9°C / 15°C supply/return temperature respectively. Therefore, the TES is used only as HES in the winter and as CES during the summer half of the year. It currently operates in CES mode from around March to November, and HES for the remainder of the year. Natural reduction of the cooling demand occurs as outside temperature decreases towards the winter season. As a result, surplus heat available for recovery is no longer able to sustain the warehouse's overall demand for heating. However, by storing heating energy reclaimed from the cooling system in advance, the load on the electric boiler can be severely reduced, which in turn reduces the consumption of energy and the cost of peak power.

In CES mode, the TES can be adopted to maximize self-consumption of solar power and reduce energy cost through two main objectives:

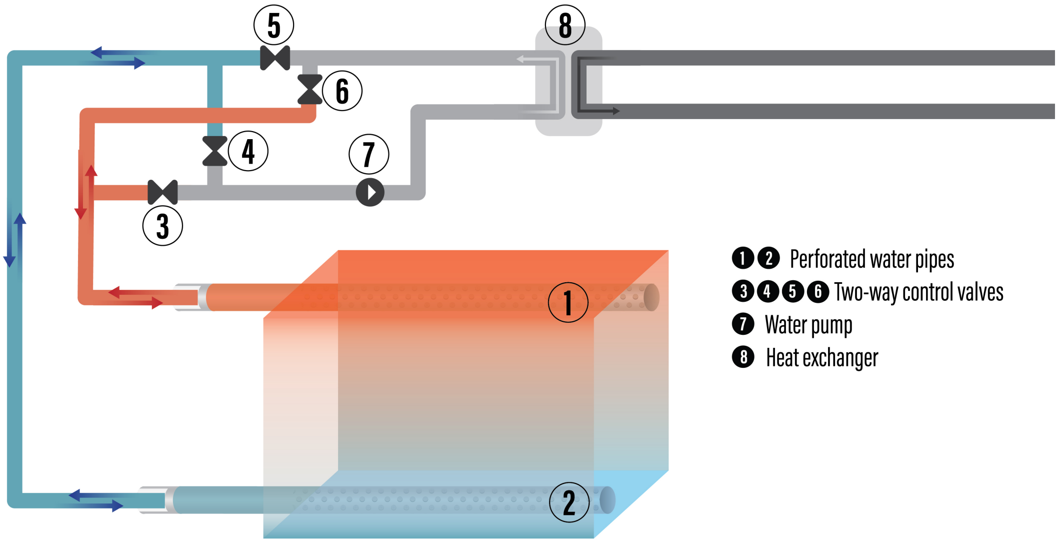


Figure 3.3: Thermal energy storage with valves for reversing direction of water flow.

1. Storing surplus electricity generated by the PV installation in the CES through energy conversion.
2. Producing and storing chilled water at high COP conditions and low energy prices.

To achieve the first objective, the cooling system can be used to convert surplus electricity that would otherwise be exported to the grid into chilled water for storage in the CES. In the evening, when the natural reduction of power output from the PV-plant occurs the CES can be discharged, thereby reducing energy consumption for the cooling system. The second objective involves decoupling the production of cooling energy from the cooling energy demand through the use of CES. Decisions regarding when to charge, discharge or stand-by can be made by an IEMS on the basis of current and future energy prices as well as the cooling system COP to maximize cost reduction related to chilling water.

In HES mode, excess heat recovered from the cooling system after the warehouse heating demand is fulfilled can be stored in the HES. Available excess heat depends on the cooling demand in the refrigerated areas of the warehouse and will thus vary proportionally to the cooling work done by the cooling system. If available heat is not sufficient to cover the heating demand, the remaining demand can either be covered by discharging stored energy from the HES or by producing heat with an electrical boiler. The boiler can produce heat at an efficiency of around 0.9, whereas using excess heat from the cooling plant only incurs a small cost based on various operating conditions such as internal operating pressure, operational temperature, external cooling demand and ambient temperature. When the HES is charged, the cooling system pressure can be automatically increased to make more heat transferable across the gas cooling heat exchanger, although this leads to a reduction in the cooling system COP.

A schematic of the TES is included in Fig. 3.3. The schematic shows perforated water

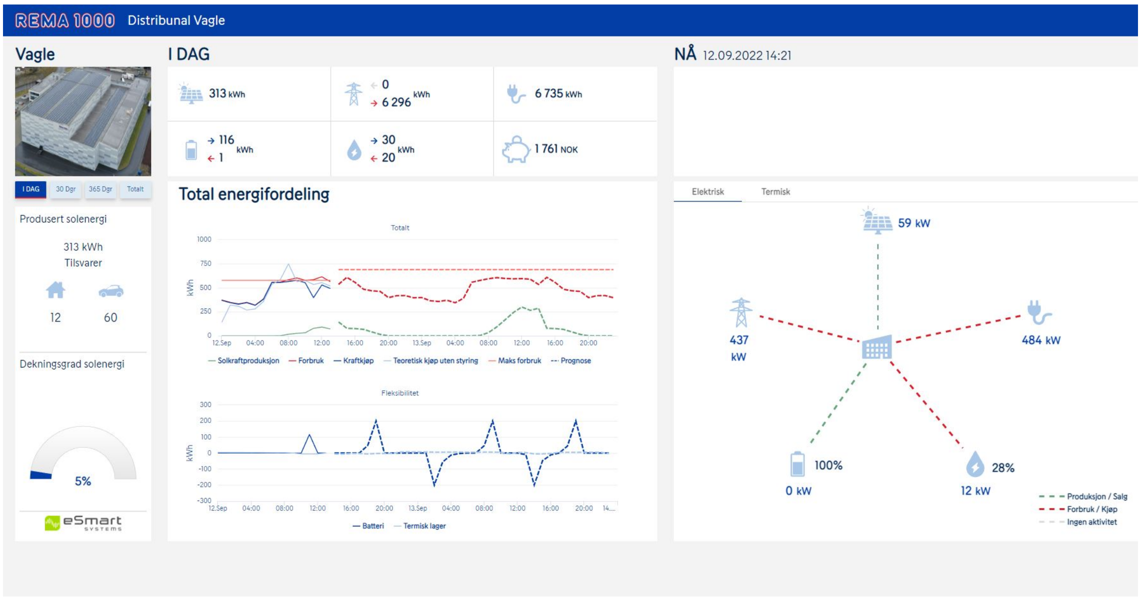


Figure 3.4: IEMS dashboard view for the smart warehouse.

pipes in the TES (1, 2), placed diagonally along opposite walls within. This allows for an even distribution of water flowing into and out of the thermal storage, consistent with a strategy of maintaining water temperature layering inside the tank. The direction of water flowing through the tank can be reversed using an arrangement of four two-way valves (3-6). The TES is physically separated from the main hydronic energy distribution systems by a heat exchanger (8). The flow volume on the TES side of the heat exchanger is automatically balanced with the main hydronic energy distribution system using flow measurements and a frequency-controlled pump (7).

We introduce the TES in paper E as the second ESS to be simultaneously controlled with the BESS by an RL agent to reduce energy cost. In this research, the TES is simulated as a hybrid of data-driven and physical models.

3.4 Existing smart warehouse intelligent energy management system

The warehouse has a commercially installed IEMS that currently controls the BESS and TES. It features an online user interface with analytic tools and a dashboard view, as shown in Fig. 3.4. The IEMS can facilitate energy management and reduction of the operational demands in an intelligent way to reduce energy cost and environmental impact. The system is based on machine learning predictions of electrical and thermal energy demand, and PV production, and an optimization algorithm that generates a 48 -hour schedule for BESS and TES charging and discharging Marton, others (2019). The schedule is automatically implemented through the local Building Management System (BMS). The existing system does not react to live operational data and instead follows the schedule precisely for the following hour. The whole process is repeated on an hourly basis, i.e., the system generates a 48-hour schedule and implements the first hour suggested actions. As such, the system relies very heavily on accurate predictions to be able to harness the ESSs for maximum energy and cost reduction.

To optimize the interaction between thermal energy production and the TES, the time-varying performance of the cooling system is required. The IEMS estimates this through the utilization of cooling demand predictions, weather predictions, and a table of COP values. The table of COP values is provided by the cooling system manufacturer and constitutes a simplified approach to performance evaluation at any given ambient temperature. Future COP values can then be estimated using weather predictions.

Chapter 4

Contributions

In this chapter, we outline and describe the primary contributions of this thesis. The thesis is multidisciplinary and touches many established research fields, such as renewable energy, control systems and artificial intelligence. More specifically, we utilize ANNs to model cooling system performance, the ARS-ANN RL algorithm to control ESSs, and machine learning techniques to build data-driven models of a technologically advanced food distribution warehouse energy system. This thesis's contributions align with our goal of advancing the research on state-of-the-art AI applied to IEMS. Our contributions are focused on two main research fields:

- Data-driven building energy system modeling.

- Reinforcement learning for intelligent energy management systems.

The main contribution of this thesis is to investigate the potential of the state-of-the-art advances in RL, trained in simulated training environments based on operational data and machine learning models, to optimize the use of electrical and thermal energy storage in advanced building integrated energy systems featuring local power production and ESSs. For RL agent training, a simulated environment of a smart warehouse energy system is introduced, designed as an ensemble of individual models of components and system dynamics. First, we demonstrate the theoretical and practical applicability of ANN modeling of industrial-scale cooling plants for performance estimation. Next, we examine the state-of-the-art RL algorithms' applicability to a simplified BESS control problem. We then introduce the ARS-ANN algorithm for BESS control in the simulated smart warehouse. Finally, we apply the ARS-ANN algorithm to simultaneous BESS and TES control in the simulated smart warehouse and demonstrate highly promising results.

Most importantly, by combining the use of RL with data-driven energy system modeling, our approach will enable an IEMS system to automatically adapt to changes in the energy system by first retraining data-driven energy system component models with new operational data before the RL agent is retrained without risk to the operational system in an off-line manner. This allows the system to adjust to physical changes in the components, as well as changes in energy usage patterns, in a way that other suggested approaches cannot.

4.1 Data-driven building energy system modeling

This section describes the various essential component models that together compose the simulated smart warehouse RL training environment. The main contribution is the use of ANNs to model the smart warehouse cooling system, which is described in subsection 4.1.2 after an overview of the relevant literature is presented in the first subsection. We describe the methodology and results in detail. The last subsection, 4.1.3, is dedicated to brief descriptions of data-driven components and dynamics models developed for the simulated warehouse.

The contributions outlined in this section address research questions 1 and 3 and include:

- The investigation into data-driven modeling of CO_2 refrigerant cooling system COP using ANNs. This groundbreaking work has facilitated precise estimations of time-varying performance under dynamic operational conditions, signifying a substantial advancement in the field.

- A study of the data-driven modeling of thermal energy system components using machine learning. This inquiry has made possible a more realistic TES response to control signals within simulated warehouse environments, contributing to the evolution of more accurate and efficient control systems.

4.1.1 Literature review

This subsection briefly reviews the related articles to our main contribution of cooling system performance modeling when ANN is applied.

In food distribution warehouses, comprehensive cooling systems (CSs) account for a significant portion of the building's energy consumption and are affected by changes in the operational environment, such as weather conditions, logistical operations, and workforce behavior (Chua et al., 2010; Sarkar et al., 2004). The impact of these factors is greater when using environmentally friendly refrigerants like CO_2 (Schmidt et al., 2019; Neksa, 2002; Neksa et al., 1998; Sarkar et al., 2004). Energy-efficient operation is a cost-effective way to reduce environmental impact in existing CSs, which can be enhanced through optimized interaction with a Thermal Energy Storage (TES) (Široký et al., 2011; Arteconi et al., 2013; Pardo et al., 2010), surplus heat recovery (Chua et al., 2010), and optimized time-of-use with access to local renewable energy resources (Wu, Wang, 2018; Kow et al., 2018).

An IEMS enables building operators to automate the process of selecting cost-reducing or energy-saving actions, leveraging the shift from Human-to-Machine to Machine-to-Machine communication, and potentially incorporating the latest AI developments for prediction and control purposes (T et al., 2018; Venayagamoorthy et al., 2016; Wen et al., 2015a; Zhao et al., 2013; Chen et al., 2011; Hakimi, Hasankhani, 2020; Wu, Wang, 2018; Manic et al., 2016). An IEMS can handle various tasks, such as optimized utilization of a TES to reduce overall CS energy consumption (Široký et al., 2011). However, accurate energy measurements and individual system performance data, including cooling load and working fluid flow measurement, are required for informed decision-making. CO_2 flow measurements are difficult to obtain, and theoretical calculations of the Coefficient of Performance (COP) are often necessary to determine system performance.

Industrially sized CSs are often unique and built with IP-protected components, limiting owners' and operators' options for continuous performance evaluation. Suppliers may calculate system performance using proprietary models, which are not shared. Thus, openly available alternatives are needed to model the system for performance evaluation, providing reliable input to the IEMS.

Installing flow measuring equipment in existing CO_2 refrigerant, direct expansion CSs is a costly and complicated operation. The complexity and risk increase when the CS operates on multiple temperature levels with separate distribution systems. The most logical option for performance evaluation then becomes a theoretical calculation based on available operational data. In Zou, Xie (2017), a simplified model for COP modeling of a water source heat pump is suggested. Sun et al. (2017) proposes a general simulation model based on graph theory that utilizes accurate mathematical models of individual components, such as Li (2013) suggested approach to variable speed compressors to model refrigerant flow. Kim et al. (2018) conducted a case study of variable refrigerant flow simulation tailored for building energy modeling, where the focus was the calibration of a CS model to the U.S. DOE's EnergyPlus software. Zhu et al. (2013) proposed a generic model for variable refrigerant flow in air conditioning systems with multiple evaporators intended for simulation of performance and control analysis. None of the aforementioned studies proposed models for multi-stage compression CS. Adaptation and implementation of the proposed methods would also require quite extensive knowledge of refrigeration technology and specific system design. Future IEMS systems might be dependent upon a realistic simulated environment to enable the training of advanced RL agents (Schrittwieser et al., 2019; Silver et al., 2018a) that can adapt to and learn from operational data. A robust method for cost-effective, real-world implementation in complex, industrial scale, CO_2 direct expansion CS is needed.

Other related work includes Dong et al. (2021) examining global greenhouse gas emissions related to servicing in the cooling industry, and Zhang et al. (2022) exploring CO_2-based combined cooling and power with an established prototype that shows promising results.

4.1.2 Modeling of compressors in an industrial cooling system using ANNs

In this subsection, we address Research Question 1 and describe our approach to using ANNs to model the compressors of an industrial and operational two-stage CO_2-based CS, as shown in Fig. 3.2. For more details, please refer to Paper A and Paper B. ANNs have already shown promising results in performance prediction modeling of heat pump technology (Esena et al., 2008), but in Esena et al. (2008) the training data set consists of a very limited amount of measurements in an experimental setting, where only thermal energy and electrical energy input could be measured.

To examine the usefulness and real-world application of this approach, we compare electrical power measurements of a case study CS to the summed calculations of an ensemble of ANNs that each models a compressor type featured in the CS. We also verify our method by comparing our calculations to measurements from a comparable laboratory CS. We train the ANNs using available data collected from the compressor manufacturer's web-based software. The ANN training algorithm adjusts the weighting of the input parameters, as well as the weighted connections between neurons, to expertly fit the labeled training data. After we define the appropriate input and output parameters, our approach only requires limited knowledge of refrigeration technology and system design to be implemented in an operational setting.

In CSs with access to a limited amount of desired performance measures, our approach can be used to supplement and enhance the value of the existing data. In such installations, the overlap between measurements and calculations can also be used to discover inconsistencies between theoretical and actual performance. To the best of our knowledge, our approach to linking theory and practice in multi-stage, CO_2 refrigeration technology using ANNs has not been attempted before. The proposed method is both practically feasible and useful in evaluating the energy performance of CO_2-based cooling installations. Owners and operators can use our ANN model ensemble approach for quality assurance of CO_2-based CSs.

We have designed our approach to:

- independently model the parts of the CS that interact with the TES at any given time, such that we can use the efficiency of this isolated part of the CS as input to an algorithm that optimizes the use of the TES;
- have a more accurate performance measure than what is currently available;
- create a data set that enables the development of CS future performance prediction models by applying our method to historical CS data;
- be able to calculate historical values of available excess heat, whereas what is currently known is only the amount of heat that was reclaimed and used;
- investigate to what extent ANNs can model complex scenarios consisting of several cooling compressors in a multi-stage CS – specifically including transcritical pressure conditions for CO_2.

4.1.2.1 ANN compressor modeling - System structure and configuration

This section summarizes the system structure and configuration of our work in compressor modeling using ANNs. For more details please refer to papers A and B.

We suggest an ANN approach to calculate compressor mass flow and electricity consumption. The compressors are semi-hermetic reciprocating compressors manufactured by Bitzer GmbH, with one frequency-controlled compressor at each stage. Fig. 3.2 shows the placement of all the compressors in a simplified cooling system architecture.

The website of the manufacturer was used to collect data (Bitzer-GmbH, 2019). Theoretical values for cooling capacity (Q), electrical power (P), electrical current (I) or mass flow ($\dot{m}$), which can all be substituted for the parameter y in Eqs. 4.1 and 4.2, can then be separately calculated by using the appropriate constants $c_i, \forall i \in 1, 2, ..10$ in the following polynomials, for subcritical pressure conditions

$$\begin{aligned} y_{sc} = \quad & c_1 + c_2 t_o + c_3 t_c + c_4 t_o^2 + c_5 t_o t_c + c_6 t_c^2 + c_7 t_o^3 + \\ & c_8 t_c t_o^2 + c_9 t_o t_c^2 + c_{10} t_c^3, \end{aligned} \tag{4.1}$$

and, for transcritical pressure

$$\begin{aligned} y_{tc} = \quad & c_1 + c_2 t_o + c_3 p_{HP} + c_4 t_o^2 + c_5 t_o p_{HP} + c_6 p_{HP}^2 + \\ & c_7 t_o^3 + c_8 p_{HP} t_o^2 + c_9 t_o p_{HP}^2 + c_{10} p_{HP}^3. \end{aligned} \tag{4.2}$$

In Eqs. (4.1) and (4.2), t_o (°C) represents the temperature of evaporation and t_c (°C) is the condensation temperature, whereas $p_{HP}[bar]$ is the discharge pressure of the compressors at transcritical operating conditions defined by $p_{HP} > 73.77[bar]$. The constants c_1 through c_{10} depend on suction gas temperature (SGT, °C) and compressor operating frequency (CF, Hz) for subcritical operating conditions, while gas cooler outlet temperature (GOT, °C) must also be given for transcritical operation. Separate and independent sets of constants can be used to calculate Q (kW$_{thermal}$), P (kW), I (A) or $\dot{m}$ (kg/h) when used with Eqs. (4.1) and (4.2).

Finally, we can determine cooling production, available excess heat, and the COP of any part of the system through calculations. For example, $\dot{m}$ can be used to calculate cooling load with the enthalpy difference equation

$$Q_c = \frac{\dot{m} \Delta h_c}{3600}, \qquad \text{[kW]} \tag{4.3}$$

where Δh_c (kJ/kg) is the specific enthalpy difference of the refrigerant between the outlet and inlet of a specific evaporation stage. We can then calculate the COP_c of a single, or multiple, compressor(s).

4.1.2.2 ANN compressor modeling - Methodology

Clearly, in Eqs. (4.1) and (4.2), we can observe the characteristics of a polynomial function. Even though the relationship between the input variables and the constants c_i, $\forall i \in 1, 2, ..10$ are unknown, Eqs. (4.1) and (4.2) provide important information that we consider an indication of the hidden function we are attempting to approximate with ANNs. To approach a function that has polynomial features as the overall trend, we believe that a simple ANN with sigmoidal activation functions in the hidden layer is most probably sufficient (Cybenko, 1989). Therefore, instead of applying modern deep learning techniques, we start with a neural network structure with one hidden layer and gradually increase the number of layers and neurons to observe the learning behavior and efficiency.

Fully connected ANNs are configured to calculate P and $\dot{m}$ by feed-forwarding input data through the neurons in the hidden layer.

Further, we assembled the individually trained models in accordance with the design of the smart warehouse CS shown in Fig. 3.2. Operational data from the cooling system was gathered in order to compare the aggregated output of the ANN models with the metered power input.

In addition to t_o, t_c, P_{HP}, SGT, CF, and GOT, compressor operating status for each compressor was collected. For every timestep, our algorithm utilizes the operational data to determine which compressors are operational, the CF of the frequency-controlled compressors, and whether the CS pressure level exceeds the transcritical threshold. The data for the active compressors, in the appropriate operational mode, is then selected and sorted into the appropriate format, and fed into the input layers of the selected models. The resulting model output is finally summed for each separate stage of compression and compared with the metered power input to the CS.

4.1.2.3 ANN compressor modeling - Results and discussion

A single hidden layer model with 45 neurons in the hidden layer (Tanh-MSE-45) outperformed all multiple hidden layer models in all tested configurations for the freezing stage compressors in subcritical conditions. This result is logical based on the expected polynomial shape of the hidden ground-truth function. The system complexity is limited and therefore does not require too many neurons in the hidden layer.

For the one hidden layer models, there was little difference between training and validation errors. In contrast, multiple hidden layer models tended towards a higher validation error as well as bigger differences between training and validation, which is a sign of overfitting the training data. The multiple layer models also tended towards larger variations in loss between every update of the trainable parameters, which is expected since there are more parameters being updated after every training batch.

Table 4.1: Training and validation MSE for all models. Separate models for frequency-controlled (FC) compressors and transcritical (TC) operation.

Compressor model	Training MSE	Validation MSE
Bitzer 4CSL12K	2,97E-05	2,48E-05
Bitzer 4CSL12K FC	2,37E-05	1,60E-05
Bitzer 4CTC30K	3,90E-05	3,17E-05
Bitzer 4CTC30K TC	7,79E-06	4,57E-06
Bitzer 4DTC25K	1,84E-05	2,01E-05
Bitzer 4DTC25K TC	6,20E-06	2,89E-06
Bitzer 4FTC30K	6,76E-05	6,50E-05
Bitzer 4FTC30K FC	2,68E-05	1,74E-05
Bitzer 4FTC30K FC TC	1,28E-05	7,85E-06
Bitzer 4FTC30K TC	1,54E-05	1,09E-05
Bitzer 4JTC15K	1,87E-05	1,34E-05
Bitzer 4JTC15K FC	2,34E-05	1,82E-05
Bitzer 4JTC15K FC TC	2,19E-05	1,54E-05
Bitzer 4JTC15K TC	7,26E-06	6,91E-06

We also trained ANNs for all the compressors in both subcritical and transcritical pressure conditions. The difference between training and validation error, as shown in Table 4.1, is minimal in all cases. Therefore, we could likely have used a more significant part of the data sets for training without the risk of overfitting. The results show that the models are highly accurate when compared to training and validation data sets generated with Eq. (4.1) and (4.2) and can therefore be expected to give very similar results to the hidden ground-truth theoretical models.

We also use data, collected through sensor networks, from an ongoing NTNU CS experiment to validate our approach in a laboratory setting. Measurements of power and flow in the ongoing experiment are compared to the outputs of an aggregated ANN model specifically designed to match the laboratory CS. The NTNU experiment was conducted in transcritical operating conditions, with pressure ranging from 74.9 bar to 98.3 bar. We obtain a MAPE of 3.13% when comparing the output from the ANNs with measurements from the power meters, whereas using measurements from the inverter for the frequency-controlled compressor reduces MAPE to 1.87%. Measurements from the power meters include the power consumption of the inverter as well as power conversion losses. The increased accuracy, when using measurements in the inverter, suggests that the aforementioned losses are not included in the Bitzer software (Bitzer-GmbH, 2019) calculations. The result for the ANN flow output compared to NTNU CS measurements is 1.76% MAPE. These results show that the presented method is accurate when given synchronized data with a low sampling time period.

4.1.3 Modeling of warehouse energy system component dynamics

To be able to create a simulated environment of the smart warehouse for RL agent training, we need models of all the important components and how they interact. In this subsection, we describe the ways we have used various machine learning techniques to make data-driven models of energy system components and dynamics. This is a necessary precursor to answering Research Question 3, and creating a simulated training environment of the smart warehouse for RL agents.

In addition to the previously described ANN modeling of the cooling system, we have built other parts of the simulated environment on operational data using linear and polynomial regression in order to make the simulated environment accessible for result analysis. As this simplified approach potentially decreases the accuracy of the system model, one could consider building a more accurate model of the environment using deep learning neural networks in an operational scenario. The methodology described in (Rätz et al., 2019) or similar approaches would then be considered. The simulated environment features an ensemble of models of energy system components and dynamics such that individual component models could be easily replaced to enhance the simulated environment's level of precision.

The following historical data sources were examined and used as input for the simulated warehouse model:

- Total power consumption and local power production.
- Cooling plant power consumption.
- Cooling plant mass flow (Opalic et al., 2020).

- Heating demand.
- TES charging and discharging.
- Energy price for electrical energy bought from and sold to the grid.

We refer to Paper E for more details.

4.1.3.1 Modeling of energy system - Thermal energy storage

In the smart warehouse, intelligently managing the interaction between the CS and the TES is an important way to reduce energy consumption and peak power load. Charging and discharging the TES at appropriate times can reduce electrical energy consumption by taking advantage of operational conditions that increase the CS COP value during heat reclaim, reduce energy cost by taking advantage of temporarily lower energy price, and reduce the energy demand for the electrical boiler through the use of stored heating energy.

Important components and dynamics of the models for the TES, production, and distribution are the following:

- Operational data of TES charging and discharging compared to the setpoint.
- TES storage loss and internal temperature levels.
- Cooling plant electrical power consumption and recoverable excess heat.

The dynamics of the hydronic heating system are complicated. We have therefore examined TES operational data in response to charging and discharging set points. The examination shows a high degree of variation between the actual delivered and the requested charge, as well as a non-linear relationship between charging and discharging dynamics. Therefore, we chose to model charging and discharging dynamics with two different functions, using more recent operational data. The R^2 score for the charging and discharging functions are 0.83 and 0.53, respectively. A qualitative analysis of the results highlights a larger spread in the data point for the discharge function. Importantly, although the R^2 for the discharge function is rather low, the goal of this function is to have a simple and explainable model of the TES while discharging. The variation in TES discharging, related to the setpoint, is known to depend on a multitude of other variables when considering a priori and empirical knowledge of the hydronic heating system and is beyond the scope of this work. A more practical way to model the TES dynamic, with a higher degree of accuracy, is likely through the use of ANN and multiple input variables. However, this would reduce model explainability, and it is not desirable at the current stage.

4.1.3.2 Modeling of cooling system dynamics - Cooling system and various component dynamics

Various less complicated models have been used to model the cooling system and other component dynamics. These are briefly described in this subsection.

We have implemented the cooling system model described in (Opalic et al., 2020) and subsection 4.1.2, and configured it to continuously calculate the refrigerant mass flow in the cooling plants. We have fitted a linear regression model, using the pressure and mass flow of the refrigerant as inputs and recoverable heat as output. Consequently, this model can be used to find the recoverable heat upper bound at the maximum pressure of 80 bar and at any given refrigerant mass flow.

Moreover, we also model the electric consumption of the cooling plant as a second-order polynomial, using refrigerant mass flow and heat recovered as inputs, and the electric consumption as output. The R^2 score of the electric consumption function is 0.87, while the RMSE is 11.17.

The cooling work, expressed as the refrigerant mass flow, represents the limiting factor for the maximum heat that can be recovered. We model this dynamic with a simple linear function, using as input the refrigerant mass flow and returning as an output the maximum recoverable heat.

Finally, there is a lower limit to the amount of electrical energy required by the cooling plant to fulfill its primary function of keeping the storage areas refrigerated. Also in this case we chose a linear model using the refrigerant mass flow as input and outputting the expected least required energy consumption. The R^2 score of the minimum electric consumption function is 0.61.

4.2 Deep reinforcement learning for intelligent energy management systems

In this section, we describe our contributions toward the application of RL for IEMS with ESS control. Our main contribution is the pioneering development and successful implementation of ARS-ANN for BESS and TES control, showcasing significant progress in the field. First, we describe our exploration to apply state-of-the-art RL algorithms to a simplified BESS control problem. Secondly, we introduce ARS-ANN for BESS control, and we finally describe simultaneous BESS and TES control with the COST-WINNERS ARS-ANN algorithm.

The contributions outlined in this section address research questions 2 and 3 and include:

- The exploration and research into deep RL control algorithms with a focus on simplicity and generalizability. This was applied to optimize the energy of a BESS in a simulated warehouse environment, with the efficacy of the system verified through multiple seeded trials requiring minimal hyperparameter tuning.

- A pioneering study into the simultaneous intelligent control of multiple ESSs, specifically the BESS and TES, within a simulated smart warehouse. We introduced the COST-WINNERS algorithm to manage and optimize energy storage systems.

- The exploration and development of an approach to IEMS that promotes standardization in design, implementation, and maintenance. Our research has demonstrated a potential for a significant reduction in dependence on human expertise in energy systems control, marking a shift towards more autonomous and efficient energy management systems.

4.2.1 Literature review

The developments in RL in recent years, with the introduction of deep learning techniques (Lillicrap et al., 2015; Silver et al., 2017, 2018b), show the potential for RL to play a major role in real-world energy optimization. BESSs built with lithium-ion technology are increasingly deployed in both macro and micro scale projects (Stroe et al., 2017). For optimal utilization of the BESS for multiple purposes such as energy cost reduction, reducing peak power demand and frequency regulation, intelligent control systems that balance the need for longer-term planning with immediate response are required. For such systems, many approaches have been suggested including constrained non-linear programming (CNLP) optimization for aggregated two-stage control in a micro-grid in Long et al. (2018), achieving a 30% energy cost reduction when combined with peer-to-peer energy sharing, a rule-based approach for many distributed batteries in a data center with a focus on accurate battery health modeling in Aksanli et al. (2013) and a rule-based scheme for PV and wind application in Teleke et al. (2010). When considering the dynamic and ever-changing nature of building-integrated energy systems, it seems unlikely that a rule-based approach can be implemented without extensive follow-up and revision. In related research, Siqueira de, Peng (2021) conducted a review of control strategies for smoothing wind power output, finding Model Predictive Control MPC to be the most common for multi-objective optimization. Lipu et al. (2021) discussed various approaches to intelligent control for battery management in electric vehicles.

As shown in Perera, Kamalaruban (2021), many researchers turn to RL as a potentially self-improving and robust approach to intelligent control of building energy systems. RL algorithms can reduce costs by reducing necessary human resource expenditure, and risks associated with their behavior can be managed through offline, data-driven training. However, most of the studies regarding RL application to energy systems do not attempt to implement state-of-the-art RL algorithms, instead, they rely on basic Q-learning. This could limit the application to well-defined and uncomplicated systems and solutions, or lead to sub-optimization through compartmentalization of complex problems into simpler tasks that disregard the intricacies of the energy system. Also, many of the more complicated state-of-the-art algorithms are primarily developed to teach agents to solve benchmark gameplay tasks from the OpenAI Gym (Brockman et al., 2016), or prediction of load forecasting Johannesen et al. (2018).

Energy Storage Systems (ESS) can consist of various technologies and be applied in a multitude of ways (Palizban, Kauhaniemi, 2016). From the perspective of the main electrical distribution grid, an important distinction exists between centralized and decentralized ESS. As opposed to decentralized ESS, centralized systems can be directly controlled by the grid operator. However, decentralized ESSs are seen as an important component of a more environmentally friendly energy system, but they come with a new set of challenges (Bögel et al., 2021). The decentralized systems should monitor the energy market, integrate it with market dynamics, and use it to reduce the peak load of the system while also minimizing costs. In the case of multiple ESSs with different dynamics, such as a combination of a BESS and TES, the complexity of the optimization problem further increases. Research on control systems for multiple ESSs with different dynamics is lacking.

Other related work includes Chung (2021) conducting a review of smart technology applied in the logistics and transport sector, and Nguyen et al. (2022) examining AI for smart warehouses in Vietnam.

4.2.2 Deep reinforcement learning for energy optimization with battery control

In this subsection, we first compare the performance of modified versions of well-known deep RL algorithms applied to a simplified battery control cost optimization task, mainly operating in continuous action space. The aim is to analyze algorithm learning and behavior from a practical standpoint as grounds for further modification of the most promising algorithms for real-world application.

Progressing further, we introduce a novel adaptation of an RL algorithm applied to energy cost optimization through direct BESS control in the smart warehouse illustrated in Fig. 3.1. The proposed ARS-ANN algorithm is a less complicated RL algorithm, compared with the state-of-the-art, which both has superior performance and significantly reduces the need for hyperparameter tuning and computational resource expenditure during training.

For more details, see Papers C and D.

4.2.2.1 Deep RL BESS control - Motivation and research goals

We aim to design the backbone of an IEMS that can be easily implemented and is able to adapt automatically to its environment. We therefore focus on data-driven energy system component modeling, which can extract patterns from operational data over time, and RL, which can be automatically trained and retrained in a simulated offline environment.

Our approach features the Augmented Random Search (ARS) (Mania et al., 2018), adapted for policy parameterization with Artificial Neural Networks (ANN) instead of the suggested linear function employed in Mania et al. (2018). The method is benchmarked to the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm (Fujimoto et al., 2018), found to be the most promising and able to find an optimal solution to a simpler BESS control problem Opalic et al. (2020). We also compare results with the vanilla ARS. A new simulated environment is introduced and developed for training the agents on historical operational data from an examined smart warehouse.

In summary, the contributions outlined in this section include:

- Introducing a robust RL algorithm that can handle complex energy optimization problems.

- Combining ARS with ANN for energy optimization of BESS.

- Creating a data-driven simulation environment of a smart warehouse for RL training.

4.2.2.2 Deep RL BESS control - Methodology

To examine how the state-of-the-art RL algorithms perform in a battery control cost optimization task we first create a simulated environment consisting of a simple ideal battery, without any losses related to power conversion or storage. The baseline energy demand for the energy system is set to 300 kWh per hour for every hour.

The battery storage capacity and inverter power output is set to 460 kWh and 200 kW respectively, in accordance with the specifications of the battery from the smart warehouse in Opalic et al. (2019, 2020), illustrated in Fig. 3.1. Consequently, our agent is allowed to charge or discharge the battery by $B^{kW} \in [-200.0, 200.0]$. We initialize the environment by inputting vectors for hourly energy price P and demand D. A baseline energy cost is then calculated as

$$C^{base} = \sum_{t=0}^{T} P_t D_t, \tag{4.4}$$

where T is the terminal time step of each episode. For every non-terminal time step the agent is awarded a numerical reward of 0 by the environment. A reward system where the agent receives the energy cost as a reward signal after each action was also examined. The cost incurred at each time step, where λ is the adjustable time step length in minutes, is calculated by

$$C_t^{agent} = P_t \left(D_t + \frac{B_t^{kW} \lambda}{60} \right), \tag{4.5}$$

and accumulated in

$$C^{agent} = \sum_{t=0}^{T} C_t. \tag{4.6}$$

Finally, the normalized reward for the agent at timestep T is given as

$$r_T = \frac{C^{base} - C^{agent}}{C^{base}}. \tag{4.7}$$

For continuous action space, we utilize modified versions of the DDPG and TD3 algorithms. For further details on these algorithms see Subsection 2.3.3.

Exploration with the TD3 algorithm was modified to also include completely random actions. As a benchmark, we also train a Deep Q-Network using a discretized action space of $\{0, 1, 2\}$ to either standby, charge or discharge at full capacity. A grid search was conducted to find the optimal network architecture.

In the second part of our deep RL BESS control research, our goal is to train an agent to learn intelligent control of a BESS in a simulated environment of the smart warehouse. We have designed a simulated environment consisting of the previously mentioned simple battery model and historical operational data from the smart warehouse. We emphasize that our suggested modeling approach is data-driven, which allows for lower demand on human resources in initial design when compared to purely physical models, as well as potentially automated adaptation to changes in building occupant behavior and other operational parameters. In combination with RL algorithms able to continually adapt to a changing environment, we argue that these characteristics are crucial for the successful adoption of IEMS in smart warehouses. Historical energy consumption and PV power production for the smart warehouse are included in the simulated environment. We define our goal as a problem of energy cost optimization.

Our suggested approach features a simple modification of the work on the ARS algorithm introduced in Mania et al. (2018). We adopt ANNs instead of the suggested linear function to parameterize the policy, see Algorithm 1 in subsection 4.2.3. The original ARS algorithm suggested the use of a simple linear policy, namely a matrix directly mapping input to output. The strength and simplicity of this algorithm are self-evident when examining the results presented in Mania et al. (2018). However, a linear policy is not always sufficient when dealing with highly complex environments such as building-integrated energy systems featuring local energy production and energy storage.

The state S_t, shown in Fig. 2.2, is composed of historical operational data as well as parameters calculated by the simulated training environment. Operational data given as current temporal values include time, energy demand, PV production, energy buy price, and energy sell price. Battery SOC and peak power limit are calculated by the simulated training environment for each timestep and included in the state.

We decided that a practical way to reduce the need for human maintenance of the agent in operation is to engineer a simple reward system that is closely coupled with the actual financial benefit. We suggest that this also potentially enables the use of the same reward function regardless of the system dynamics or degree of complexity, which in turn simplifies implementation and thereby increases scalability. The function for the reward system calculates baseline energy cost C^b for each episode where no actions are taken and compares this to the actual calculated cost C^a after the agent has selected an action,

$$R = C^b - C^a. \tag{4.8}$$

The energy cost calculated by the training environment is structured according to the energy pricing scheme utilized by the grid operator, consisting of the following parts:

- spot price per kWh,
- a fixed annual fee,
- a fixed rate per kWh consumed (summer/winter),
- monthly peak power,

4.2.2.3 Deep RL BESS control - Results and discussion

The first experiment environment consists of 50 timesteps with an unchanging energy demand of 300 kWh every hour. For this experiment, utilizing energy cost as a reward signal on every timestep yielded reduced agent performance and resulted in slowed learning due to an observed natural preference towards battery discharging.

The experiment mainly features the DQN, DDPG and TD3 algorithms. It is noted that the algorithms' performances are volatile, and they are not able to converge toward the optimal behavior in every training session without hyper-parameter tuning. The maximum achievable cost reduction in this environment was found to be 14000, and our average TD3 result was 13999. Achieving stable results across multiple seeded training sessions required an extensive hyperparameter grid search and tuning of the algorithm. It also required us to continuously save the top-performing version of the agent as the performance frequently and rapidly declined after achieving the highest score of each training session.

Secondly, we explore RL algorithms applied to cost optimization of energy storage in BESS, based on operational data from the smart warehouse. Thus, we conducted an experiment of 48-hour episodes that features comparisons between ARS-ANN, original ARS, and the GLPK solver. As stated in Mania et al. (2018), too few experiments in RL verify results across multiple seeds, thus shedding doubt on whether the reported performance is a result of algorithm ingenuity and generalizability or extensive hyperparameter tuning to a single instance of the given RL problem. To verify our results across multiple trials, we conducted an experiment with 10 randomly seeded 48-hour episodes pulled from our data set. Results from this experiment can be observed in Table 4.2.We observe that both the ARS and ARS-ANN algorithms are achieving results that are very close to the GLPK solver. When comparing numerical results in Table 4.2, we observe that the ARS-ANN has a slight increase in performance when compared with the original ARS. In addition to peak shaving, the ARS-ANN can extract some values from energy price differentiation even though the reward increase from this behavior is almost inconsequential due to an exceptionally low energy cost at around 0.3 NOK/kWh. We note that the performances of the ARS algorithms are very high, with the original ARS achieving an average of 98.5% and the ARS-ANN achieving 99.2% of the GLPK solver solution. The same ARS-ANN architecture and hyperparameters are used for all the seeded trials, indicating that the performance is the result of a well-designed algorithm.

Table 4.2: Results for experiment two - 48 h trials with 50% SOC reward incentive. ARS-ANN architecture “24 tanh”.

Episode	GLPK	ARS-ANN			ARS-Original		
	Reduction	Reward	Reduction	Penalty	Reward	Reduction	Penalty
1	2609	2547	2556	8.9	2555	2556	1.2
2	206	185	187	2.7	193	193	0.6
3	4247	4211	4223	10.9	4204	4216	11.8
4	5497	5455	5481	26.6	5464	5474	10.7
5	3581	3576	3576	0.1	3573	3573	0.5
6	2611	2568	2605	36.2	2566	2601	35.4
7	6777	6728	6757	29.6	6727	6744	17.1
8	2613	2600	2605	5.0	2602	2606	4.5
9	1970	1933	1953	19.8	1955	1965	9.3
10	7247	7242	7241	0.1	6965	7020	56.0
Average	**3736**	**3705**	**3718**	**14.0**	**3680**	**3695**	**14.7**

We also ran an experiment where we expanded a single episode to nearly include the entire dataset and compared our ARS-ANN agent performance with the original ARS and a GLPK solver solution. To find the best configuration of the network, we conducted hyperparameter searches. After observing a destabilizing effect when increasing the amount of ANN weight parameters during hyperparameter tuning, we also conducted a grid search with different learning rates α and noise standard deviations v in an ANN architecture featuring 4 hidden layers with 64 neurons each. Results show that learning and validation performance for deeper neural network architectures can be stabilized by decreasing the learning rate. Reducing the learning rate from 0.01 to 0.001 significantly increases algorithm stability and performance.

4.2.3 Simultaneous BESS and TES control with COST-WINNERS

In this subsection, we introduce the COST-WINNERS RL algorithm. The algorithm is an iteration of the previously introduced ARS-ANN adapted to control multiple energy storage systems simultaneously and builds upon all the research previously introduced in this chapter. The simulated training environment is based on operational data from the warehouse energy system shown in Fig. 3.1. For more details, see Papers D, E, and subsection 4.1.3.

4.2.3.1 COST-WINNERS - Outline of research

Newer RL algorithms often include training ANNs to output desired actions or action values, showing improved performance (Lillicrap et al., 2015; Cao et al., 2020; Shang et al., 2020). In contrast, Mania et al. (2018) showed that the Augmented Random Search (ARS) algorithm could achieve high performance with very little computational resource expenditure by training a simple linear function for action selection with their proposed policy search algorithm.

In the previous sections of this chapter, we have described our approach to data-driven modeling of a smart warehouse to enable offline training of RL algorithms in a simulated environment. We have emphasized the use of data-driven techniques as a way to reduce the need for human expertise to design the simulated environment and increase the practical utility of our approach. We have also shown how RL can be used to control a BESS for energy cost optimization. In this subsection, we present a novel approach to control, for the first time, both the BESS and TES of a smart warehouse. Specifically:

- We implement the Augmented Random Search (ARS) (Mania et al., 2018) RL algorithm, modified with ANNs to encode the agent policy, to simultaneously control TES and BESS energy storage systems.

- We train the RL agents in a data-driven simulated training environment, also modeling the dynamics of the TES.

- Overall, we introduce a novel approach to control both the BESS and TES of a smart warehouse simultaneously to reduce total energy cost. This is an important detail because combining different ESSs can lead to improved performance and cost savings but also introduces new challenges due to each system's different dynamics and control requirements. We argue that simultaneous control disincentivizes suboptimization.

4.2.3.2 COST-WINNERS - Methodology

Algorithm 1 Augmented Random Search with ANN.

```
Set hyperparameters:
    • $\alpha$ - learning rate
    • $n$ - number of directions sampled per iteration
    • $v$ - exploration noise standard deviation
    • $b$ - number of top-performing directions to use
Run algorithm 2 to initialize policy parameters $\theta_j$, i.e. ANN weights
Initialize:
    • Mean - $\mu_0 = 0 \in \mathbb{R}^{inputs}$
    • Covariance - $\Sigma_0 = \mathbf{I}_n \in \mathbb{R}^{inputsxinputs}$
while ending condition not satisfied do
    Sample $\delta_1, \delta_2, ..., \delta_N$ of the same size as $\theta_j$, with i.i.d. standard normal entries.
    Normalize input values $x$ with $x_{normalized} = diag(\Sigma_j)^{-\frac{1}{2}}(x - \mu_j)$. Collect $2N$ rollouts of horizon $H$ and their corresponding rewards using noise modified ANN policies $\pi_{j,k,+}$ and $\pi_{j,k,-}$, where the $v\delta_k$ exploration noise is added to the weight parameters $\theta_j$ of the ANN for $\pi_{j,k,+}$ and subtracted from $\theta_j$ for $\pi_{j,k,-}$ with $k \in \{1, 2, ..., N\}$.
    Sort the directions $\delta_k$ by $\max\{r(\pi_{j,k,+}), r(\pi_{j,k,-})\}$, denote by $\delta_{(k)}$ the $k$-th largest direction, and by $\pi_{j,(k),+}$ and $\pi_{j,(k),-}$ the corresponding policies.
    Make the update step for the ANN weights:
    $\theta_{j+1} = \theta_j + \frac{\alpha}{b\sigma_R} \sum_{k=1}^{b} [r(\pi_{j,k,+}) - r(\pi_{j,k,-})]\delta_k$, where the standard deviation of the $2b$ rewards for the policy update is $\sigma_R$.
    Set the mean and covariance, $\mu_{j+1}, \Sigma_{j+1}$, of the $2NH(j+1)$ training states encountered.
    $j \leftarrow j + 1$.
end while
```

We examine the applicability of the ARS-ANN RL algorithm to a complex energy cost reduction problem through direct control of BESS and TES charging and discharging setpoints. Our main research goal is to examine if the ARS-ANN algorithm can efficiently control multiple ESSs with different dynamics and substantially varying degrees of impact on energy cost. The agent is trained in a simulated environment of a technologically advanced warehouse, which we mainly designed through the use of data-driven techniques. We refer to subsection 4.1.3 for further description of the data-driven component models.

We use a physical model for the thermal energy storage featuring:

- Temperature loss through the heat conduction to surroundings.
- 4 internal vertical temperature levels.

Algorithm 2 ANN for ARS in RLLIB.

```
1: Set hyperparameters:
      • $\theta^{hl}$ - ANN hidden layers.
      • $\theta^{nu}$ - number of neurons in each hidden layer.
      • $\theta^{af}$ - list of activation function for each layer.
2: Initialize: $j = 0$, policy parameters $\theta_j$ of shape defined by $\theta^{hl}$ and $\theta^{nu}$ and random
   values $X$ from $N(\mu_\theta, \sigma_\theta^2)$ normal distribution of mean $\mu_\theta = 0$ and variance $\sigma_\theta^2 = 1$,
   multiplied by standard deviation $\sigma = 1.0$ for the hidden layers and $\sigma = 0.1$ for the
   output, divided by the square root of the random value $X^{hl,nu}$, $\theta_j^{hl,nu} = X^{hl,nu}\frac{\sigma}{\sqrt{X}}$.
```

Our model of the TES includes the ability to reverse the direction of the flow of water such that hotter water is always added to or extracted from the top of the tank and vice versa for colder water. We have not included a model of the heat exchanger due to the physical system automatically balancing volume flow on both sides of the heat exchanger and the observed temperature loss in the heat exchanger is minimal. Modeling the heat exchanger could possibly be considered for future work.

As described in subsection 4.2.2, we implement a modified version of the ARS algorithm (Mania et al., 2018). We deploy an ANN for policy parameterization in place of the linear function proposed by Mania et al. (2018), see Algorithm 1. We take advantage of the functionality for neural networks already implemented in the RLLIB programming library.

We use Pyomo (Hart et al., 2011; Bynum et al., 2021) with a GNU Linear Programming Kit (GLPK) solver to calculate near-optimal solutions for performance comparisons and benchmarking. However, due to the complex nature of our energy system, we do not attempt to implement the TES in the GLPK solver solution.

We examined the operational data and found that the electrical boiler had contributed very little to satisfy the heating demand in the selected time period due to the fact that available excess heat from the cooling system seemed to be sufficient. Reducing energy consumption on the boiler is the main way that the TES can contribute to lower electrical energy consumption during winter operations. We argue that the impact of the TES on the energy cost in the time period we pulled our operational data from is very limited. Adopting the performance of the GLPK solvers BESS control as a benchmark is therefore still valid and useful.

4.2.3.3 COST-WINNERS - Results and discussions

We investigate the application of the ARS-ANN algorithm in a simulated warehouse energy system, featuring both electrical (BESS) and thermal (TES) energy storage systems. Therefore, we have the opportunity of analyzing algorithm performance on a complex temporal energy optimization problem. The objective of the algorithm is to reduce energy costs by controlling charging and discharging setpoints of both energy storage systems.

First, we apply the ARS-ANN agent to control both BESS and TES for a random 48-hour episode. Our results clearly indicate that the agent is able to find a near-optimal

Table 4.3: Results for 10 seeded trials for ARS-ANN vs GLPK - battery only.

Trial	GLPK - Battery only	ARS-ANN Result	Percent of GLPK
1	4910	5046	103%
2	7115	7106	100%
3	7540	7498	99%
4	298	361	121%
5	643	639	100%
6	7117	7109	100%
7	5861	5864	100%
8	3771	3780	100%
9	640	641	100%
10	6652	3233	49%

Table 4.4: Results for 10 seeded trials with state-of-the-art RL algorithms.

Trial	SAC		TD3	
	Reward	Percentage ARS-ANN	Reward	Percentage ARS-ANN
1	13	0.3 %	346	7 %
2	7083	99.7 %	290	4 %
3	7147	95.3 %	43	1 %
4	141	39.1 %	-62	-1 7%
5	86	13.4 %	76	12 %
6	1246	17.5 %	305	4 %
7	133	2.3 %	55	1 %
8	3772	99.8 %	21	1 %
9	728	113.5 %	232	36 %
10	691	21.4 %	-338	-10 %

value for BESS charging such that the peak energy cost is reduced to a minimum. The agent took advantage of the TES, when heating was required, to reduce the electrical energy required by the cooling plant. It is relevant to mention that the heating demand was very low during the random episode used for experiment one. However, the ARS-ANN agent was still able to find and store excess heat when there was no cost induced, and then in turn used this to slightly reduce electrical consumption by discharging when necessary. By doing this, the agent was able to minimize cooling system energy demand during periods of higher heating demand.

Secondly, to better quantify the performance of the ARS-ANN agent, we compare it with a GLPK optimization solver in multiple seeded trials, as well as benchmark it with other state-of-the-art RL algorithms. The GLPK will be solely controlling the BESS, with perfect information, and the comparison will be done for 10 seeded trials. Opposed to the GLPK, the ARS-ANN agent will have control of both BESS and TES. We have decided that comparing performance to an optimization algorithm with simultaneous BESS and

TES control is out of the scope of this research due to modeling complexity. Additionally, the operational data used to pull randomly seeded trials is from early winter, where the potential cost reduction of optimal TES control is minor compared with BESS control. There are two main reasons behind this choice. Firstly, this was the time period with the most available data requiring minimal amounts of data cleaning. Secondly, we decided that observing how the algorithm performs in controlling multiple systems with vastly different impacts on the result would be of interest.

The results of the multiple seeded trials are displayed in Table 4.3. We observe that for the majority of the trials, the energy cost reduction of the ARS-ANN with both BESS and TES control either equals or exceeds the cost reduction of the GLPK with BESS control only. For trial 10, the algorithm seems to get stuck in a local optimum where it charges the battery too aggressively on the first timestep. Additional research is required to explore why this happens and how it can be avoided in the future. In the 4th seeded trial, we observe that the ARS-ANN outperforms GLPK by 21%. In this trial, the potential for cost reduction using the BESS is quite low due to a relatively low baseline peak power cost. Finally, we compare results for the SAC and TD3 RL algorithms with the ARS-ANN algorithm solution, shown in Table 4.4. There, we can observe that TD3 results seem to stagnate around origo while SAC actually performs reasonably well and even exceeds ARS-ANN in a single trial achieving an average performance of 50% compared with ARS-ANN. However, it was only after training SAC for more than three weeks that these results could be achieved. On a more reasonable time frame of running the algorithm for about a week of training time on 6 GPUs and 96 CPUs, both SAC and TD3 achieved similarly poor results. Also, the SAC algorithm results were not the actual end results because the performance did not stabilize. In fact, SAC performance drops off entirely in most cases. The results in Table 4.4 include the maximum award achieved during each training session.

4.3 Summary of contributions

In this chapter, we have presented our main contributions to energy cost reduction by building a simulated warehouse energy system environment with machine learning data-driven techniques to train RL agents. We have described our main contributions to RL research, which includes replacing a linear policy matrix with ANNs for ARS to control multiple energy storage systems simultaneously. The first part of this chapter, section 4.1, relates to data-driven modeling of the warehouse energy system and its industrial CO_2 refrigerant cooling system. We showed that the accuracy of this modeling approach is practically on par with results from the compressor manufacturers' software tool. We also verified results in a laboratory setting and compared an ensemble ANN cooling system model with metered energy consumption, achieving very promising results despite unsatisfactory data quality. Based on this approach, an ensemble model of the cooling system in the smart warehouse has been developed and implemented in the warehouse. The values calculated by our cooling system model are stored in the BMS and can be used in various ways by the BMS, IEMS, and for research purposes.

Our contribution to data-driven modeling of cooling systems is an important part of our RL-based IEMS concept, enabling accurate COP estimation in a simulated smart warehouse environment. Combined with other data-driven techniques, the simulated environment can use the values generated by the cooling system model to estimate how TES charging or discharging affects the electrical power consumption of the cooling system. Cost reduction due to higher thermal power production efficiency or shifting thermal production to periods with lower energy prices can then be estimated.

Further, ANNs to model cooling systems and heat pump COP could be designed and applied in a multitude of ways. For instance, the data source could be as elementary as measured heating or cooling output and electrical power input, as well as any known time-varying operational parameters that commonly affect these values. We also expect to be able to model any important energy-consuming component of a building-integrated energy system with ANNs or even simpler data-driven machine learning techniques. Further, we expect that the process of coupling relevant available data to each component for modeling purposes, choosing the appropriate modeling technique, and training each model for integration in a simulated environment, can be automated and solved with applied machine learning. As more operational data is made available over time, it will also be possible to update individual models with more accurate and sophisticated methods automatically.

In the second section of this chapter, section 4.2, we have detailed our exploration of various reinforcement learning algorithms for energy storage control. We start with policy gradient algorithms, including DDPG and TD3, obtaining good results with extensive algorithm hyperparameter tuning in a simple BESS control problem. We increase the control problem complexity by building a simulated environment using operational data from the smart warehouse and modifying the robust ARS algorithm using ANNs to parameterize the policy. We show that ARS-ANN achieves near-optimal performance on a simulated battery control problem based on operational data from the smart warehouse. We then expand our simulated control problem to include the smart warehouse TES. We model the TES in the simulated environment using a combination of various machine learning and physical modeling techniques. We then show that the COST-WINNERS version of the ARS-ANN agent vastly outperforms other state-of-the-art RL algorithms in multiple seeded trials of simultaneous BESS and TES control for energy cost reduction.

Although we are satisfied with the COST-WINNERS algorithm in its current state, we expect that further development of the algorithm will be necessary for its applicability to be extended from ESSs to flexible loads, from the smart warehouse to other warehouse buildings, and hopefully to any building category. However, with our data-driven modeling approach and modular simulated environment, it will also be possible to effortlessly implement new and superior RL algorithms in the future. We propose that adhering to principles of flexibility and modularity is essential for IEMS scalability. Therefore, we maintain that a data-driven modular approach to a simulated RL training environment for IEMSs, where any single component model can be easily replaced and any RL algorithm can be applied, is a very promising avenue of research that should be further explored.

The contributions described in this chapter are proposed solutions to the research questions described earlier in the thesis, and in line with our overall goal of developing a scalable and robust approach to IEMS. The summarized total contribution is an approach to building-integrated energy system optimization that enables an IEMS to evolve with changes in the building energy system and energy usage patterns. This can be achieved by relying on a data-driven simulated environment designed for training RL agents through trial-and-error interactions between the agent and the environment. The RL agents are allowed to control a predefined number of subsystems in the environment, in this instance a TES and BESS, and can adjust their behavior according to a reward signal. One of the main advantages of this approach is that the simulated environment can be automatically updated using more recent operational data, potentially without the need for any human intervention. This could allow the simulated environment to continually represent an accurate model of the current state of the operational environment. Combined with the presented COST-WINNERS algorithm, we have proposed an approach to IEMS that outperforms the state-of-the-art RL algorithms in simultaneous TES and BESS control in smart warehouses, while also potentially drastically reducing human expert resource dependence both in the design and operation of the system.

Chapter 5

Conclusions and Future Work

In this thesis, we propose deep RL with a data-driven environment modeling approach for IEMS application. We make contributions in both research areas, and we structure this thesis accordingly. In this chapter, we first detail to what degree our contributions answer our research questions. Secondly, we make concluding remarks on our contributions in two parts - data-driven modeling, and reinforcement learning, respectively.

5.1 Conclusions to the Research Questions

In this section, we conclude the findings of our proposed methods in accordance to the research questions in Chapter 1.

Research question 1: We introduce feed-forward Artificial Neural Networks (ANNs) to model all the compressors in all the compression stages of the smart warehouse CO_2 refrigerant cooling system. Using compressor manufacturer data, we train models for both subcritical and transcritical operational pressure for the high-pressure compression stages, nearly perfectly replicating the proprietary compressor manufacturer software calculation results. Single hidden layer ANNs are found to be sufficient for this task, and deeper architectures do not achieve higher performance.

These ANNs are combined into an ensemble for modeling the entire smart warehouse cooling system refrigerant flow and power consumption. The COP is then calculated using known thermodynamic equations. We find that power consumption accuracy, when compared with meter data from the smart warehouse, exceeds other available methods. Furthermore, the method is highly accurate for refrigerant flow estimation when compared with flow meter data in a laboratory setting. We also observe that deeper architectures for single compressor models appear to be redundant, as they require additional computational resources without increasing performance or accuracy.

Our research demonstrates the possibility of using a data-driven approach to COP estimation in CO_2 refrigerant cooling systems, presenting a solution to the challenges associated with accurate and cost-prohibitive CO_2 flow measurement. This enables accurate performance modeling of such cooling systems in a simulated training environment, which in turn can be used to train RL agents for energy cost optimization. We argue that this is an important prerequisite to scalable smart warehouse IEMS implementation.

Research Question 2: We first explore the use of game-play oriented policy gradient RL algorithms for a simplified BESS control task. To encourage exploration, we modify the policy gradient RL algorithms to increase the randomness of their actions, achieving stellar performance with the TD3 algorithm in particular. However, the amount of hyperparameter tuning and modification necessary could be a potential barrier to the introduction of these algorithms in an operational setting.

Addressing the challenges with policy gradient algorithms, we examine the use of the ARS algorithm using reward-guided random searches in policy space, substituting the original linear policy with an ANN for more complex and abstract policies. The ARS-ANN algorithm scores very high on our performance metric, maintaining a steady upward trajectory of reward achieved over elapsed time in training, as well as stabilizing performance if learning continues for an extended amount of time after peak performance has been achieved. Time spent on hyperparameter tuning was significantly reduced compared with our previous work, and no alterations were made to the exploration mechanics of ARS, showcasing the algorithm's apparent potential for practical application in an operational setting.

Research Question 3: We propose a novel approach to intelligent ESS control of a technologically advanced warehouse. We examine the applicability of the ARS-ANN RL algorithm to simultaneously control the TES and BESS ESSs. We develop a data-driven simulated training environment, also modeling the dynamics of the TES and building upon our previous work on cooling system modeling. Specifically, the ARS-ANN RL algorithm is tasked to solve a complex energy cost reduction problem through direct control of BESS and TES charging and discharging setpoints. The algorithm compares favorably to benchmark state-of-the-art algorithms on performance, training time, and computational resource expenditure.

We show that our RL-based approach to IEMS is capable of energy cost optimization at a high level of precision, even when controlling multiple ESSs with diverging dynamics and levels of impact on total energy cost. The RL approach to IEMS potentially reduces the need for human expertise in design, implementation, and maintenance, thus making the solution more scalable and robust for practical application.

5.2 Data-driven building energy system modeling

5.2.1 Modeling of compressors in an industrial cooling system using ANNs

We show that using an ANN to model the compressors in a cooling system is a valid approach that allows quick and quite accurate calculations of cooling load and compressor power consumption. The compressor COP at the given operating conditions can then be calculated. The best result was achieved using a single hidden layer ANN with a hyperbolic tangent activation function. The model was trained with an MSE loss function using the Adam optimizer. For this approach to be valuable to an IEMS, the transcritical compressors that interact directly with TES must be modeled so that the full system performance can be calculated. For the best use of the TES as a HES during winter, the maximal available and reclaimable heat must also be determined.

We further present a performance estimation model of an operational CO_2-based industrial cooling sub-system of a complex warehouse energy system using an ensemble of ANN compressor models. The operating temperature and pressure measurements, as well as the operating frequency of frequency-controlled compressors, are used in developing the operational model. The output of the model ensemble is electrical consumption and refrigerant mass flow for the compression process. The presented technique is relatively superior to a general theoretical model both in terms of accuracy, flexibility, cost-effectiveness, and implementability in real-world applications.

The developed model has MAPE in the range of 5% to 12% in the operational case-study cooling system. However, the presented results also indicate that the accuracy can be drastically improved with increased quality of data collection frequency in the operational measurement, supported by a MAPE of 1.87% and 1.76% in a comparable laboratory CS, for power and flow respectively. The accuracy of the presented ANN flow calculations is promising from a practical standpoint and can be implemented through Machine-to-Machine communication using IoT-related devices. The developed model of the cooling system has been implemented in the case study energy system (Fig. 3.1).

5.3 Reinforcement learning for intelligent energy management systems

5.3.1 Deep reinforcement learning for energy optimization with battery control

We designed modifications to the DDPG and TD3 algorithms for enhancing exploration to enable cost-efficient control of the charging and discharging of an ideal BESS. Our experiments reveal that the algorithms are quite sensitive to changes in hyperparameter and exploration settings and need to be configured appropriately to deliver consistent performance. This could pose a serious challenge in complex environments where ideal agent behavior is less transparent. However, our appropriately configured DDPG agent was able to reduce the energy cost by 25% while maintaining an average of 99% of maximum over multiple training sessions in the basic environment. In the advanced experiment setting, our TD3 agent was able to achieve optimal results when future energy price for 5-time steps was included as state input variables. Stabilizing results over multiple seeded training sessions required hyperparameter tuning and an additional hidden layer, leading to virtually optimal average performance.

Important conclusions we drew from these initial experiments are that algorithm performance stability in multiple training sessions should be further explored to reduce the need for hyperparameter tuning. The environment complexity needs to be enhanced to approach real-world operational settings, introducing constraints for battery health preservation as well as more realistic battery charge cycles. The training environment will be expanded to include a realistic dynamic load and locally produced solar power. Real energy pricing schemes need to be introduced, including peak power tariffs and price differentiation between import and export energy.

Further, we present the application of reinforcement learning-based techniques to the specific energy optimization problem of controlling the BESS in a smart warehouse to minimize the energy cost. We have adopted data from a real operational smart warehouse for food distribution on the west coast of Norway, integrated with a PV power plant and BESS. Multiple experiments have been conducted within a simulated training environment built with operational data the smart warehouse featuring a 460kWh lithium-ion BESS. RL agents, and specifically the proposed ARS-ANN agent, are trained to control the BESS charging and discharging to minimize energy costs. Obtained results show that both the ARS and ARS-ANN algorithms perform very well on 48-hour episodes, achieving an average of 98.5 and 99.2% accuracy, respectively, across 10 seeded trials. Also, ARS-ANN shows promising results on a longer time horizon, outperforming original ARS by 21%. As seen in the initial experiment on ARS-ANN with reduced learning rates, learning for deeper neural network architectures can be stabilized by lowering the learning rate α. The developed algorithm finds very promising solutions in the considered case study of a smart house for energy cost minimization through BESS control. The presented methodology can likely be implemented in a wider range of smart energy-efficient buildings (e.g., smart warehouses) with less engineering detail for a reduction in energy costs.

5.3.2 Simultaneous BESS and TES control with COST-WINNERS

We examine the applicability of the COST-WINNERS RL algorithm for a complex energy cost reduction problem by direct control of BESS and TES charging and discharging setpoints in a simulated environment of an operational smart warehouse. We successfully demonstrate that we are able to model the dynamics of the TES and to use it in combination with BESS and controlled by the COST-WINNERS agent to minimize energy consumption. It is important to mention that, due to time constraints and a lack of additional data, we only tested this approach in a scenario in which the heating demand was limited.

Overall, by combining BESS and TES direct control with the presented COST-WINNERS agent, we demonstrate that the agent was able to stabilize maximum energy consumption, thereby reducing the network peak energy demand. Additionally, the agent exploited the TES, when the heat was in demand to reduce the required electrical energy demand by the cooling plant.

We show that for 9 out of 10 of our seeded trials, the algorithm meets or exceeds the performance of a GLPK optimization solver controlling the BESS only while given perfect information. For the single trial where it only performs at around 50% of the GLPK, the algorithm seems to become stuck in a local optimum. Why this happens and how it can be avoided should be explored in future work. To conclude, we also compare our solution to the state-of-the-art RL algorithms, showing an average of 100% performance increase compared with the SAC algorithm. However, the SAC algorithm was able to match or slightly exceed the performance of COST-WINNERS in a few seeded trials when SAC training time was increased by nearly a factor of 3.

5.4 Future Work

The work developed through this thesis represents a new approach to IEMS based on RL, including building a data-driven simulated environment of a technologically advanced warehouse with various machine learning techniques and proposing the COST-WINNERS RL agent. However, exploring the generalizability of our approach to other building categories and building-integrated energy systems is a natural next step.

Data-driven modeling of cooling systems could potentially be fine-tuned using operational data. This could include additional training of the developed compressor and cooling system models, based on increasing amounts of operational data, for increased operational accuracy. With regard to the other energy system components and dynamics modeled in this thesis, we are sure that these could be enhanced in a multitude of ways to both reduce time expenditure and improve accuracy. Developing such models using a data-driven approach is necessary to enable scalability, and constitutes an important step towards establishing best-practice solutions to modeling the various component categories necessary to enable RL-based IEMS to be applicable to an increasingly growing number of smart buildings.

Specifically related to the smart warehouse described in this thesis, it would be of interest to explore a broader landscape of scenarios with higher heating demand, and during the summer season to evaluate the general efficacy of the method. If simulated performance is adequately promising, after further development and hyperparameter tuning, and if the costs are not prohibitive, our chosen algorithm will be tested in an operational setting in a technologically advanced warehouse.

With regard to the COST-WINNERS algorithm, there is an occurrence of a single trial where it only performs at around 50% of the GLPK solver solution. The algorithm seems to converge to a local optimum. Why this happens and how it can be avoided should be explored in future work.

Finally, although generally outperformed by COST-WINNERS, the SAC algorithm showed promising results in certain instances. It would be interesting to investigate possible solutions combining COST-WINNERS/ARS-ANN and SAC to fully explore the action space in an efficient manner. Our suggested approach allows for continual adaptation to new state-of-the-art deep RL algorithms, either that are designed for or that have not been previously tested for ESS control. This should be further explored in the future. Overall, combining existing RL algorithms or developing new algorithms for simultaneous ESS and flexible load control are interesting avenues of research that build upon the groundwork laid out in this thesis.

Bibliography

Abdalla Ahmed N., Nazir Muhammad Shahzad, Tao Hai, Cao Suqun, Ji Rendong, Jiang Mingxin, Yao Liu. Integration of energy storage system and renewable energy sources based on artificial intelligence: An overview // Journal of Energy Storage. 2021. 40. 102811.

Afram Abdul, Janabi-Sharifi Farrokh. Theory and applications of HVAC control systems – A review of model predictive control (MPC) // Building and Environment. 2014. 72. 343–355.

Aksanli Baris, Rosing Tajana, Pettis Eddie. Distributed battery control for peak power shaving in datacenters // 2013 International Green Computing Conference Proceedings. 2013. 1–8.

Ammari Chouaib, Belatrache Djamel, Touhami Batoul, Makhloufi Salim. Sizing, optimization, control and energy management of hybrid renewable energy system—A review // Energy and Built Environment. 2022. 3, 4. 399–411.

Arteconi A., Hewitt N.J., Polonara F. Domestic demand-side management (DSM): Role of heat pumps and thermal energy storage (TES) systems // Applied Thermal Engineering. 2013. 51, 1. 155 – 165.

Badia Adrià Puigdomènech, Piot Bilal, Kapturowski Steven, Sprechmann Pablo, Vitvitskyi Alex, Guo Zhaohan Daniel, Blundell Charles. Agent57: Outperforming the Atari Human Benchmark // Proceedings of the 37th International Conference on Machine Learning. 119. 13–18 Jul 2020. 507–517. (Proceedings of Machine Learning Research).

Barbato Antimo, Capone Antonio. Optimization models and methods for demand-side management of residential users: A survey // Energies. 2014. 7, 9. 5787–5824.

Bitzer-GmbH . Bitzer Software v6.10.2 rev2250. 2019.

Bögel Paula Maria, Upham Paul, Shahrokni Hossein, Kordas Olga. What is needed for citizen-centered urban energy transitions: Insights on attitudes towards decentralized energy storage // Energy Policy. 2021. 149. 112032.

Brockman Greg, Cheung Vicki, Pettersson Ludwig, Schneider Jonas, Schulman John, Tang Jie, Zaremba Wojciech. OpenAI Gym. 2016.

Bynum Michael L., Hackebeil Gabriel A., Hart William E., Laird Carl D., Nicholson Bethany L., Siirola John D., Watson Jean-Paul, Woodruff David L. Pyomo–optimization modeling in python. 67. 2021. Third.

Cao Jun, Harrold Dan, Fan Zhong, Morstyn Thomas, Healey David, Li Kang. Deep Reinforcement Learning-Based Energy Storage Arbitrage With Accurate Lithium-Ion Battery Degradation Model // IEEE Transactions on Smart Grid. 2020. 11, 5. 4513–4521.

Chen C., Duan S., Cai T., Liu B., Hu G. Smart energy management system for optimal microgrid economic operation // IET Renewable Power Generation. 2011. 5, 3. 258–267.

Cho Kyunghyun, Merrienboer Bart van, Gulcehre Caglar, Bahdanau Dzmitry, Bougares Fethi, Schwenk Holger, Bengio Yoshua. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. 2014.

Chua K.J., Chou S.K., Yang W.M. Advances in heat pump systems: A review // Applied Energy. 2010. 87, 12. 3611 – 3624.

Chung Sai-Ho. Applications of smart technologies in logistics and transport: A review // Transportation Research Part E: Logistics and Transportation Review. 2021. 153. 102455.

Cybenko George. Approximation by superpositions of a sigmoidal function // Mathematics of control, signals and systems. 1989. 2, 4. 303–314.

Dong Yabin, Coleman Marney, Miller Shelie A. Greenhouse Gas Emissions from Air Conditioning and Refrigeration Service Expansion in Developing Countries // Annual Review of Environment and Resources. 2021. 46, 1. 59–83.

Esena Hikmet, Inallib Mustafa, Sengurc Abdulkadir, Esena Mehmet. Performance prediction of a ground-coupled heat pump system using artificial neural networks // Expert Systems with Applications. 2008. 35, 4. 1940–1948.

Fan Jiajun, Xiao Changnan. Generalized Data Distribution Iteration. 2022.

Fujimoto Scott, Hoof Herke van, Meger David. Addressing Function Approximation Error in Actor-Critic Methods // CoRR. 2018. abs/1802.09477.

Gassar Abdo Abdullah Ahmed, Cha Seung Hyun. Energy prediction techniques for large-scale buildings towards a sustainable built environment: A review // Energy and Buildings. 2020. 224. 110238.

Goldsworthy M., Moore T., Peristy M., Grimeland M. Cloud-based model-predictive-control of a battery storage system at a commercial site // Applied Energy. 2022. 327. 120038.

Goodfellow Ian, Bengio Yoshua, Courville Aaron. Deep Learning. 2016. http://www.deeplearningbook.org.

Haarnoja Tuomas, Zhou Aurick, Abbeel Pieter, Levine Sergey. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor // CoRR. 2018. abs/1801.01290.

Hakimi Seyed Mehdi, Hasankhani Arezoo. Intelligent energy management in off-grid smart buildings with energy interaction // Journal of Cleaner Production. 2020. 244. 118906.

Hart William E, Watson Jean-Paul, Woodruff David L. Pyomo: modeling and solving mathematical programs in Python // Mathematical Programming Computation. 2011. 3, 3. 219–260.

Henze Gregor P., Schoenmann Jobst. Evaluation of Reinforcement Learning Control for Thermal Energy Storage Systems // HVAC&R Research. 2003. 9, 3. 259–275.

Huang Hao, Chen Lei, Hu Eric. A new model predictive control scheme for energy and cost savings in commercial buildings: An airport terminal building case study // Building and Environment. 2015. 89. 203–216.

Johannesen Nils Jakob, Kolhe Mohan, Goodwin Morten. Comparison of regression tools for regional electric load forecasting // 2018 3rd International Conference on Smart and Sustainable Technologies (SpliTech). 2018. 1–6.

Jordan M. I., Mitchell T. M. Machine learning: Trends, perspectives, and prospects // Science. 2015. 349, 6245. 255–260.

Kim Dongsu, Cox Sam J., Cho Heejin, Im Piljae. Model calibration of a variable refrigerant flow system with a dedicated outdoor air system: A case study // Energy and Buildings. 2018. 158. 884 – 896.

Kingma Diederik P., Ba Jimmy. Adam: A Method for Stochastic Optimization // CoRR. 2014. abs/1412.6980.

Kow Ken Weng, Wong Yee Wan, Rajkumar Rajprasad, Isa Dino. An intelligent real-time power management system with active learning prediction engine for PV grid-tied systems // Journal of Cleaner Production. 2018. 205. 252 – 265.

Kuznetsova Elizaveta, Li Yan-Fu, Ruiz Carlos, Zio Enrico, Ault Graham, Bell Keith. Reinforcement learning for microgrid energy management // Energy. 2013. 59. 133 – 146.

Lešić Vinko, Martinčević Anita, Vašak Mario. Modular energy cost optimization for buildings with integrated microgrid // Applied Energy. 2017. 197. 14–28.

Li Wenhua. Simplified steady-state modeling for variable speed compressor // Applied Thermal Engineering. 2013. 50, 1. 318 – 326.

Lillicrap Timothy P., Hunt Jonathan J., Pritzel Alexander, Heess Nicolas, Erez Tom, Tassa Yuval, Silver David, Wierstra Daan. Continuous control with deep reinforcement learning. 2015.

Lipu MS Hossain, Hannan MA, Karim Tahia F, Hussain Aini, Saad Mohamad Hanif Md, Ayob Afida, Miah Md Sazal, Mahlia TM Indra. Intelligent algorithms and control strategies for battery management system in electric vehicles: Progress, challenges and future outlook // Journal of Cleaner Production. 2021. 292. 126044.

Long Chao, Wu Jianzhong, Zhou Yue, Jenkins Nick. Peer-to-peer energy sharing through a two-stage aggregated battery control in a community Microgrid // Applied Energy. 2018. 226. 261–276.

Mania Horia, Guy Aurelia, Recht Benjamin. Simple random search provides a competitive approach to reinforcement learning. 2018.

Manic Milos, Amarasinghe Kasun, Rodriguez-Andina Juan J., Rieger Craig. Intelligent Buildings of the Future: Cyberaware, Deep Learning Powered, and Human Interacting // IEEE Industrial Electronics Magazine. 2016. 10, 4. 32–49.

Mariano-Hernández D., Hernández-Callejo L., Zorita-Lamadrid A., Duque-Pérez O., Santos García F. A review of strategies for building energy management system: Model predictive control, demand side management, optimization, and fault detect & diagnosis // Journal of Building Engineering. 2021. 33. 101692.

MIP in demand side response. // . 2019.

Mbuwir Brida V, Ruelens Frederik, Spiessens Fred, Deconinck Geert. Battery energy management in a microgrid using batch reinforcement learning // Energies. 2017. 10, 11. 1846.

Mnih Volodymyr, Kavukcuoglu Koray, Silver David, Graves Alex, Antonoglou Ioannis, Wierstra Daan, Riedmiller Martin A. Playing Atari with Deep Reinforcement Learning // CoRR. 2013. abs/1312.5602.

Mnih Volodymyr, Kavukcuoglu Koray, Silver David, Rusu Andrei A., Veness Joel, Bellemare Marc G., Graves Alex, Riedmiller Martin, Fidjeland Andreas K., Ostrovski Georg, Petersen Stig, Beattie Charles, Sadik Amir, Antonoglou Ioannis, King Helen, Kumaran Dharshan, Wierstra Daan, Legg Shane, Hassabis Demis. Human-level control through deep reinforcement learning // Nature. II 2015. 518, 7540. 529–533.

Mocanu E., Mocanu D. C., Nguyen P. H., Liotta A., Webber M. E., Gibescu M., Slootweg J. G. On-Line Building Energy Optimization Using Deep Reinforcement Learning // IEEE Transactions on Smart Grid. July 2019. 10, 4. 3698–3708.

Neksa Petter. CO2 heat pump systems // International Journal of Refrigeration. 2002. 25, 4. 421 – 427.

Neksa Petter, Rekstad Havard, Zakeri G.Reza, Schiefloe Per Arne. CO2-heat pump water heater: characteristics, system design and experimental results // International Journal of Refrigeration. 1998. 21, 3. 172 – 179.

Nguyen Vu-Anh-Tram, Le Ngoc-Bich, Kieu Manh-Kha, Nguyen Xuan-Hung, Nguyen Duc-Canh, Le Ngoc-Huan, Ninh Tran-Thuy-Duong, Debnath Narayan C. Artificial Intelligence Based Solutions to Smart Warehouse Development: A Conceptual Framework // The 8th International Conference on Advanced Machine Learning and Technologies and Applications (AMLTA2022). Cham: Springer International Publishing, 2022. 115–124.

Opalic S. M., Goodwin M., Jiao L., Nielsen H. K., Lal Kolhe M. A Deep Reinforcement Learning scheme for Battery Energy Management // 2020 5th International Conference on Smart and Sustainable Technologies (SpliTech). 2020. 1–6.

Opalic Sven Myrdahl, Goodwin Morten, Jiao Lei, Nielsen Henrik Kofoed, Kolhe Mohan Lal. Modelling of Compressors in an Industrial CO2-Based Operational Cooling System Using ANN for Energy Management Purposes // Engineering Applications of Neural Networks. Cham: Springer International Publishing, 2019. 43–54.

Opalic Sven Myrdahl, Goodwin Morten, Jiao Lei, Nielsen Henrik Kofoed, Kolhe Mohan Lal. Augmented Random Search with Artificial Neural Networks for energy cost optimization with battery control // Journal of Cleaner Production. 2022. 134676.

Opalic Sven Myrdahl, Goodwin Morten, Jiao Lei, Nielsen Henrik Kofoed, Pardiñas Ángel Álvarez, Hafner Armin, Kolhe Mohan Lal. ANN modelling of CO2 refrigerant cooling system COP in a smart warehouse // Journal of Cleaner Production. 2020. 260. 120887.

Palizban Omid, Kauhaniemi Kimmo. Energy storage systems in modern grids—Matrix of technologies and applications // Journal of Energy Storage. 2016. 6. 248–259.

Pardo N., Montero Á., Martos J., Urchueguía J.F. Optimization of hybrid-ground coupled and air source-heat pump systems in combination with thermal storage // Applied Thermal Engineering. 2010. 30, 8. 1073 – 1077.

Perera A.T.D., Kamalaruban Parameswaran. Applications of reinforcement learning in energy systems // Renewable and Sustainable Energy Reviews. 2021. 137. 110618.

Rätz Martin, Javadi Amir Pasha, Baranski Marc, Finkbeiner Konstantin, Müller Dirk. Automated data-driven modeling of building energy systems via machine learning algorithms // Energy and Buildings. 2019. 202. 109384.

Sarkar J., Bhattacharyya Souvik, Gopal M.Ram. Optimization of a transcritical CO2 heat pump cycle for simultaneous cooling and heating applications // International Journal of Refrigeration. 2004. 27, 8. 830 – 838.

Schmidt Drew, Singleton Jake, Bradshaw Craig R. Development of a light-commercial compressor load stand to measure compressor performance using low-GWP refrigerants // International Journal of Refrigeration. 2019. 100. 443 – 453.

Schrittwieser Julian, Antonoglou Ioannis, Hubert Thomas, Simonyan Karen, Sifre Laurent, Schmitt Simon, Guez Arthur, Lockhart Edward, Hassabis Demis, Graepel Thore, Lillicrap Timothy, Silver David. Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model. 2019.

Schrittwieser Julian, Hubert Thomas, Mandhane Amol, Barekatain Mohammadamin, Antonoglou Ioannis, Silver David. Online and Offline Reinforcement Learning by Planning with a Learned Model. 2021.

Schulman John, Levine Sergey, Moritz Philipp, Jordan Michael I., Abbeel Pieter. Trust Region Policy Optimization // CoRR. 2015. abs/1502.05477.

Sechilariu Manuela, Wang Bao Chao, Locment Fabrice. Supervision control for optimal energy cost management in DC microgrid: Design and simulation // International Journal of Electrical Power & Energy Systems. 2014. 58. 140–149.

Shang Yuwei, Wu Wenchuan, Guo Jianbo, Ma Zhao, Sheng Wanxing, Lv Zhe, Fu Chenran. Stochastic dispatch of energy storage in microgrids: An augmented reinforcement learning approach // Applied Energy. 2020. 261. 114423.

Silver David, Huang Aja, Maddison Chris J., Guez Arthur, Sifre Laurent, Driessche George van den, Schrittwieser Julian, Antonoglou Ioannis, Panneershelvam Veda, Lanctot Marc, Dieleman Sander, Grewe Dominik, Nham John, Kalchbrenner Nal, Sutskever Ilya, Lillicrap Timothy, Leach Madeleine, Kavukcuoglu Koray, Graepel Thore, Hassabis Demis. Mastering the Game of Go with Deep Neural Networks and Tree Search // Nature. I 2016. 529, 7587. 484–489.

Silver David, Hubert Thomas, Schrittwieser Julian, Antonoglou Ioannis, Lai Matthew, Guez Arthur, Lanctot Marc, Sifre Laurent, Kumaran Dharshan, Graepel Thore, Lillicrap Timothy, Simonyan Karen, Hassabis Demis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play // Science. 2018a. 362, 6419. 1140–1144.

Silver David, Hubert Thomas, Schrittwieser Julian, Antonoglou Ioannis, Lai Matthew, Guez Arthur, Lanctot Marc, Sifre Laurent, Kumaran Dharshan, Graepel Thore, others . A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play // Science. 2018b. 362, 6419. 1140–1144.

Silver David, Lever Guy, Heess Nicolas, Degris Thomas, Wierstra Daan, Riedmiller Martin. Deterministic Policy Gradient Algorithms // International Conference on Machine Learning. 2014. 387–395.

Silver David, Schrittwieser Julian, Simonyan Karen, Antonoglou Ioannis, Huang Aja, Guez Arthur, Hubert Thomas, Baker Lucas, Lai Matthew, Bolton Adrian, Chen Yutian, Lillicrap Timothy, Hui Fan, Sifre Laurent, Driessche George van den, Graepel Thore, Hassabis Demis. Mastering the game of Go without human knowledge // Nature. X 2017. 550. 354–359.

Siqueira Luanna Maria Silva de, Peng Wei. Control strategy to smooth wind power output using battery energy storage system: A review // Journal of Energy Storage. 2021. 35. 102252.

Široký Jan, Oldewurtel Frauke, Cigler Jiří, Prívara Samuel. Experimental analysis of model predictive control for an energy efficient building heating system // Applied Energy. 2011. 88, 9. 3079 – 3087.

Smarra Francesco, Jain Achin, de Rubeis Tullio, Ambrosini Dario, D'Innocenzo Alessandro, Mangharam Rahul. Data-driven model predictive control using random forests for building energy optimization and climate control // Applied Energy. 2018. 226. 1252–1272.

Stroe D., Knap V., Swierczynski M., Stroe A., Teodorescu R. Operation of a Grid-Connected Lithium-Ion Battery Energy Storage System for Primary Frequency Regulation: A Battery Lifetime Perspective // IEEE Transactions on Industry Applications. 2017. 53, 1. 430–438.

Sun Haoran, Ding Guoliang, Hu Haitao, Ren Tao, Xia Guanghui, Wu Guoming. A general simulation model for variable refrigerant flow multi-split air conditioning system based on graph theory // International Journal of Refrigeration. 2017. 82. 22 – 35.

Sutton Richard S., Barto Andrew G. Reinforcement Learning: An Introduction. 2018. Second.

T R., Jasmin E. A., Ahamed T. P. I. Residential Load Scheduling With Renewable Generation in the Smart Grid: A Reinforcement Learning Approach // IEEE Systems Journal. 2018. 1–12.

Teleke Sercan, Baran Mesut E., Bhattacharya Subhashish, Huang Alex Q. Rule-Based Control of Battery Energy Storage for Dispatching Intermittent Renewable Sources // IEEE Transactions on Sustainable Energy. 2010. 1, 3. 117–124.

Venayagamoorthy G. K., Sharma R. K., Gautam P. K., Ahmadi A. Dynamic Energy Management System for a Smart Microgrid // IEEE Transactions on Neural Networks and Learning Systems. 2016. 27, 8. 1643–1656.

Wan Z., Li H., He H. Residential Energy Management with Deep Reinforcement Learning // 2018 International Joint Conference on Neural Networks (IJCNN). July 2018. 1–7.

Wang Bo, Cai Guowei, Yang Deyou. Dispatching of a Wind Farm Incorporated With Dual-Battery Energy Storage System Using Model Predictive Control // IEEE Access. 2020. 8. 144442–144452.

Wang Zhe, Hong Tianzhen. Reinforcement learning for building controls: The opportunities and challenges // Applied Energy. 2020. 269. 115036.

Learning from delayed rewards. // . 1989.

Wei Q., Liu D., Shi G. A novel dual iterative Q-learning method for optimal battery management in smart residential environments // IEEE Transactions on Industrial Electronics. April 2015. 62, 4. 2509–2518.

Wen Z., O'Neill D., Maei H. Optimal Demand Response Using Device-Based Reinforcement Learning // IEEE Transactions on Smart Grid. 2015a. 6, 5. 2312–2324.

Wen Zheng, O'Neill Daniel, Maei Hamid. Optimal Demand Response Using Device-Based Reinforcement Learning // IEEE Transactions on Smart Grid. 2015b. 6, 5. 2312–2324.

Wu Nan, Wang Honglei. Deep learning adaptive dynamic programming for real time energy management and control strategy of micro-grid // Journal of Cleaner Production. 2018. 204. 1169 – 1177.

Xu Zhengwei, Han Guangjie, Liu Li, Martínez-García Miguel, Wang Zhijian. Multi-energy scheduling of an industrial integrated energy system by reinforcement learning-based differential evolution // IEEE Transactions on Green Communications and Networking. 2021. 5, 3. 1077–1090.

Zhang Bin, Hu Weihao, Cao Di, Li Tao, Zhang Zhenyuan, Chen Zhe, Blaabjerg Frede. Soft actor-critic–based multi-objective optimized energy conversion and management strategy for integrated energy systems with renewable energy // Energy Conversion and Management. 2021. 243. 114381.

Zhang Yonghao, Shi Lingfeng, Tian Hua, Li Ligeng, Wang Xuan, Huang Guangdai, Shu Gequn. Experimental investigation on CO2-based combined cooling and power cycle // Energy Conversion and Management. 2022. 256. 115342.

Zhao Z., Lee W. C., Shin Y., Song K.-B. An Optimal Power Scheduling Method for Demand Response in Home Energy Management System // IEEE Transactions on Smart Grid. 2013. 4, 3. 1391–1400.

Zhu Yonghua, Jin Xinqiao, Du Zhimin, Fan Bo, Fu Sijie. Generic simulation model of multi-evaporator variable refrigerant flow air conditioning system for control analysis // International Journal of Refrigeration. 2013. 36, 6. 1602 – 1615.

Zou Shenghua, Xie Xiaokai. Simplified model for coefficient of performance calculation of surface water source heat pump // Applied Thermal Engineering. 2017. 112. 201 – 207.

Part II

Appended Papers

Appendix A

Paper A

Title:	Modelling of compressors in an industrial CO_2-based operational cooling system using ANN for energy management purposes
Authors:	Sven Myrdahl Opalic, Morten Goodwin, Lei Jiao, Henrik Kofoed Nielsen, and Mohan Lal Kolhe
Affiliation:	University of Agder, Faculty of Engineering and Science, 4879, Grimstad, Norway
Conference:	*20th International Conference on Engineering Applications of Neural Networks (EANN)*, Xersonisos, Crete, Greece, May 2019.
DOI:	10.1007/978-3-030-20257-64.

Paper A - Modelling of compressors in an industrial CO_2-based operational cooling system using ANN for energy management purposes

Sven Myrdahl Opalic, Morten Goodwin, Lei Jiao, Henrik Kofoed Nielsen, and Mohan Lal Kolhe
Department of Engineering Sciences
Faculty of Engineering and Science, University of Agder
4879, Grimstad, Norway
E-mails: {sven.opalic, morten.goodwin, lei.jiao, henrik.kofoed.nielsen, mohan.l.kolhe}@uia.no

Abstract — **Large scale cooling installations usually have high energy consumption and fluctuating power demands. There are several ways to control energy consumption and power requirements through intelligent energy and power management, such as utilizing excess heat, thermal energy storage and local renewable energy sources. Intelligent energy and power management in an operational setting is only possible if the time-varying performance of the individual components of the energy system is known. This paper presents an approach to model an industrial, operational two-stage cooling system, with CO_2 as the working fluid, located in an advanced food distribution warehouse in Norway. An artificial neural network is adopted to model the compressors using the operational data. The models are trained with cooling medium evaporation and condensation temperature, suction gas temperature, gas cooler outlet temperature and pressure, and compressor operating frequency. The output is the aggregated electrical power load and mass flow for each stage. The model ensemble will be part of a system implemented in a real-world setting to determine the coefficient of performance of the cooling system. An intelligent energy management system will utilize the model for energy and power optimization of the cooling system by storing energy in a thermal energy storage, using predictions of energy demand and cooling system performance.**

A.1 Introduction

It is complex to design and operate an efficient building energy system that incorporates multiple elements of new and emerging technologies Manic et al. (2016). The increase in building-integrated intermittent renewable energy production, local energy storage, and micro-grid solutions provides the building operator with a multitude of options in choosing the optimal operational mode of all the components at any given time. Implementation of an Intelligent Energy Management System (IEMS) is one way to automate this decision-making process in order to reduce the total energy cost for a building Chen et al. (2011); T et al. (2018); Venayagamoorthy et al. (2016); Wen et al. (2015); Zhao et al. (2013). An IEMS can be tasked to predict short- and long-term energy demand and local energy production in order to continuously design an optimal schedule for all energy storage options, while also considering energy price fluctuations and thermal energy production efficiency.

For the heating and cooling demands, heat pumps and Cooling Systems (CS) are widely accepted as an efficient way to produce thermal energy, with continual improvements being made to maximize efficiency Chua et al. (2010). In technologically advanced food distribution warehouses, large scale CS represent a large part of the buildings total energy demand. Changing operating conditions, such as weather (including ambient temperature), flow of goods and building occupant behaviour, will continuously impact CS performance Chua et al. (2010); Sarkar et al. (2004). This is especially true for more environmentally friendly working fluids, such as CO_2, that have made their relatively recent re-entry into the field of refrigeration technology Neksa (2002); Neksa et al. (1998); Sarkar et al. (2004). CO_2-based large scale multi-stage cooling systems are becoming common as natural refrigerants with low global warming potential and ozone depletion potential replace synthetic refrigerants. One way to increase the efficiency of these systems during operation is to produce thermal energy (heating or cooling) at the most ideal operating conditions and store the energy in a thermal energy storage (TES) for later use Arteconi et al. (2013); Pardo et al. (2010). This requires an IEMS that is given accurate energy measurements and individual system performance data. For the CS, this includes cooling load which requires working fluid flow measurement. However, because accurate CO_2 flow measurements are difficult, energy measurement of cooling demand supplied with CO_2 as the working fluid of energy distribution, typically in warehouses with large cooling and freezing storage areas, is usually unavailable. Theoretical calculation of system efficiency, or Coefficient of Performance (COP), is therefore necessary to determine system performance. However, industrially sized CS's are usually unique and built by intellectual property (IP) protected components that limit the system owner and operators options for continual performance evaluation. In many cases, system performance at given operating conditions can only be calculated by the supplier using a proprietary model, but the details of the model itself are not shared. Therefore, openly available alternatives are necessary in order to model the system for performance evaluation purposes to provide reliable input to an IEMS.

In this work, we use Artificial Neural Networks (ANNs) to model the compressors of an industrial and operational two-stage CO_2-based CS. ANNs have already shown promising results in performance prediction modelling of heat pump technology Esena et al. (2008), but in Esena et al. (2008) the training data set consisted of a very limited amount of measurements in an experimental setting, where thermal energy and electrical energy input could be measured.

ANNs have the ability to approximate both simple and complex unknown functions that fit the underlying data. Therefore, we apply ANNs on data generated by an openly available online compressor calculation model. The aim is for the ANN to learn the underlying patterns within the data. With ANNs trained on compressor calculation data, the need for extensive knowledge and understanding of refrigeration technology is less critical in development of the model. This is key when it comes to ease of implementation and scalability in real world applications where every system has its own unique aspects that must be taken into account. Since the ANN model is built on freely available information, there is no need to access IP protected empirical data or algorithms, allowing the model to be designed independently of the compressor manufacturer. Lastly, since machine learning models, such as ANNs, can be further trained with new training examples, the parts of the system that are measurable and observable can be used to modify the model to adjust for observable deviations between theory and practice. This fact can also be used to reveal large discrepancies between expected and real performance that should be further investigated.

The remaining sections of this article are organized as follows. Section A.2 presents the case-study energy system and cooling system where the ANN CS performance model will be applied. The research setup, various ANN designs and configurations are presented in Section A.3. Section A.4 contains the results and discussion from the various models that were trained. Finally, the article conclusions and suggested future work are presented in Section A.5.

A.2 System Structure and Configuration

The research in this study is based on existing infrastructure and data from a 27,000 m^2 technologically advanced warehouse for food storage and distribution, located in Sandnes, Norway. The warehouse was completed in 2017 and is currently implementing a commercially available IEMS based on hourly scheduling that uses various machine learning techniques to optimize energy storage, both electrochemical in batteries and thermal utilizing an isolated fire sprinkler basin, for storage of hot or cold water. The IEMS has to predict future electrical and thermal energy demand in order to come up with an optimal scheduling strategy. The main components of the warehouse energy system is a 1 MWp solar Photovoltaic power plant (PV), a 460 kWh storage capacity electrochemical li-ion battery storage system with two 100 kW inverters, a 300 m^3 water tank for thermal storage, a CO$_2$-based large scale CS consisting of two identical two-stage cooling plants and a back-up cooling machine for ventilation air and IT-server cooling, an electrical boiler and accompanying technical infrastructure (HVAC, Lighting etc.). Excess heat from the CS is reclaimed and used to supply the building with heating energy. An overview of the warehouse temperature zones with their respective operating temperatures are listed in Table A.1, whereas the main components of the energy system and their inter-dependencies

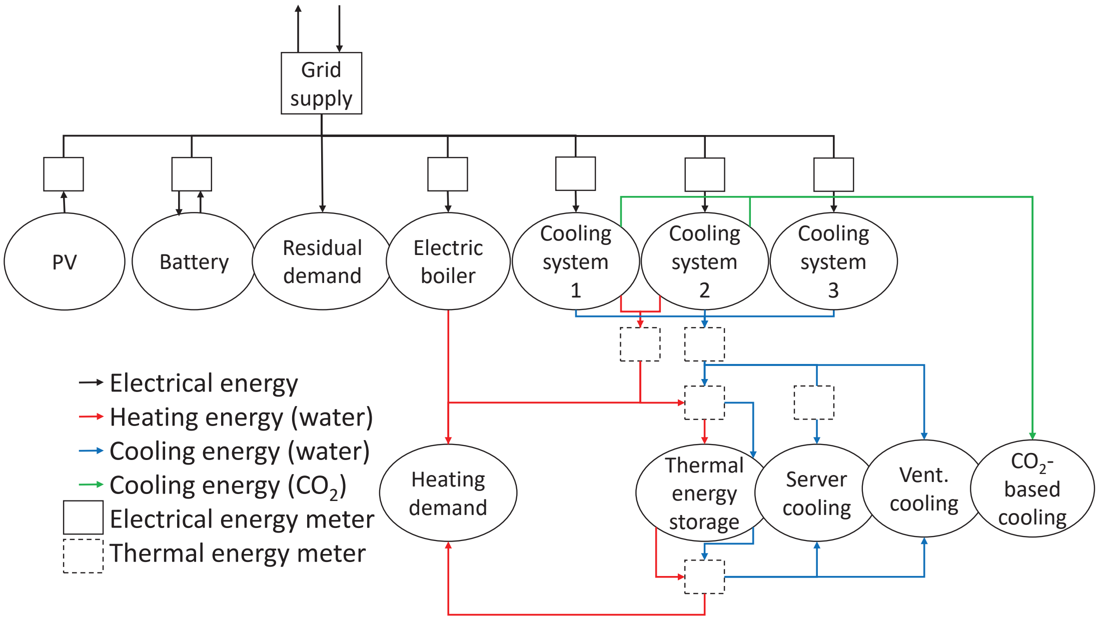


Figure A.1: The case-study warehouse energy system.

are visualized in Fig. A.1 and listed in Table A.2.

Table A.1: Warehouse cooling floor area and operating temperatures

Area	Size	Operating temperature
Warehouse	27,000 m^2	-20°C to +20°C
Frozen storage area	3,000 m^2	-20°C
Cold storage area	3,600 m^2	+2°C
Cooled shipping area	3,600 m^2	+2°C

Table A.2: Energy system components, capacity in [kW_p], [kW], [kWh], [m^3] and [$kW_{thermal}$]

Component	Capacity	Unit of measurement
PV - solar power plant	1,000	[kW_p]
EB - battery bank	460/200	[kWh/kW]
TES - water tank	300	[m^3]
CS - cooling system	1,140	[kW_{th}]
Electrical boiler	495	[kW]

The TES is operated as a Cooling Energy Storage (CES) during spring, summer and fall, and as a Heating Energy Storage (HES) during winter. When ambient air temperature decreases, cooling demand and thereby available excess heat is reduced to the point where excess heat is insufficient to meet the heating energy demand. Therefore, the HES can reduce required heat supply and power load on the electrical boiler by storing available excess heat from the CS.

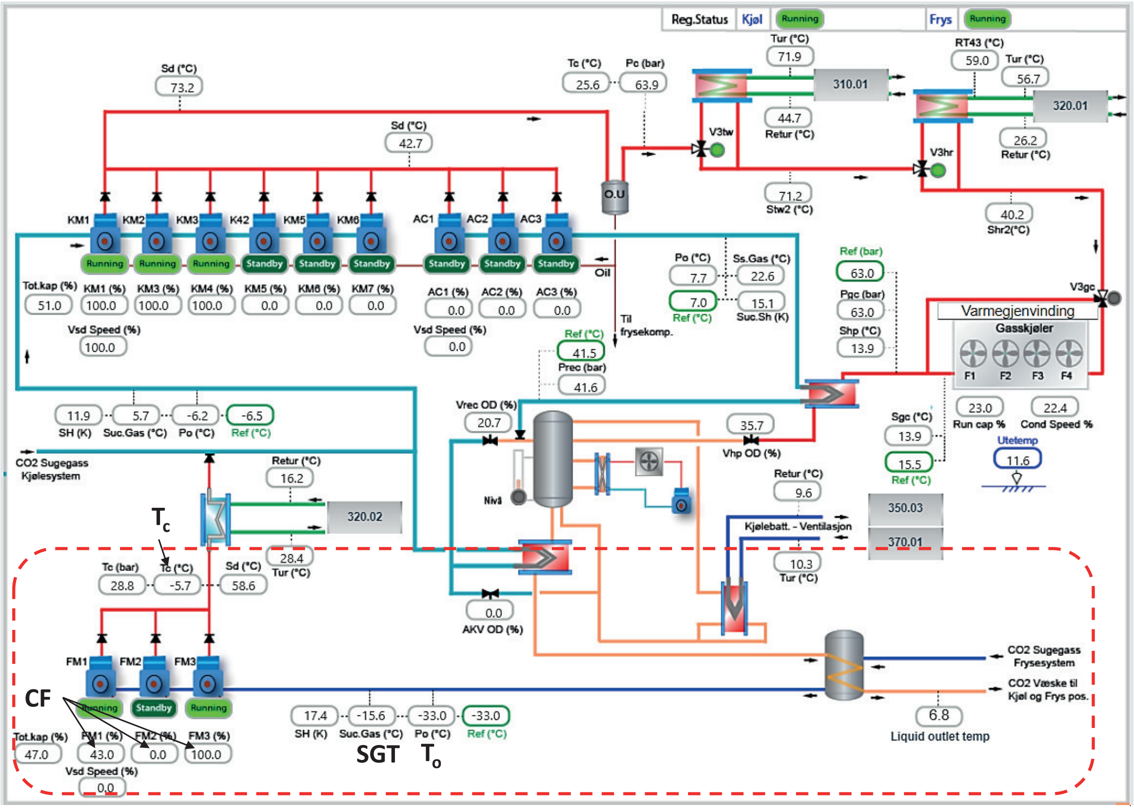


Figure A.2: The case-study cooling system as visualized in the building management system, courtesy of IWMAC. Freezing compressors and CO_2 distribution to evaporators roughly outlined in the red dashed box, ANN model input values marked in bold (CF in %).

For the remainder of the year, the CES is used in one of two ways – to store excess energy from the PV-plant when production exceeds demand, or to store cooling energy produced during optimal operating conditions for the CS. In order to store excess energy from the PV-plant, the electrical energy is converted to thermal energy by the CS and stored in the CES as chilled water in a temperature range between 7°C and 15°C. In the evening, when production from the PV-plant is naturally reduced, the CES is discharged by directly supplying cooling energy for ventilation air and IT-servers. Alternatively, the CES can be used to optimize cooling energy production by charging and discharging based on varying operating conditions, such as current and predicted cooling load, and current and predicted ambient air temperature.

For the IEMS to make the optimal choice of operating mode for the TES, the performance of the CS must be evaluated both at current and future operating conditions. In this work, ANN models for theoretical calculation of cooling load and compressor power consumption based on available compressor data is explored. The models have been developed for the three identical semi-hermetic reciprocating sub-critical compressors, denoted FM1, FM2 and FM3 within the red dashed box in Fig. A.2, but the whole CS will be modelled in future work.

Compressor performance calculation data for the 4CSL-12K compressors was collected from the website of the manufacturer, Bitzer™. Theoretical values were calculated using the following given equation (according to EN12900):

$$y = c_1 + c_2 t_o + c_3 t_c + c_4 t_o^2 + c_5 t_o t_c + c_6 t_c^2 + c_7 t_o^3 + c_8 t_c t_o^2 + c_9 t_o t_c^2 + c_{10} t_c^3 \qquad \text{(A.1)}$$

In Eq. (A.1), t_o is the evaporating temperature and t_c is the condensing temperature. c_1 to c_{10} are constants that are given based on the selected suction gas temperature (SGT), compressor frequency (CF) and subcooling temperature. Four different sets of constants are given that are used to calculate either cooling capacity Q, power input P, current I or mass flow, represented as y in Eq. (A.1), for the compressor within its defined and given operating range. Constants for Q and P only were collected in 5 degree steps, from -30°C to -5°C for suction gas temperature, and for 5 Hz steps from 70 Hz to 30 Hz for compressor frequency. Since the subcooling temperature was unknown at the time of data extraction, it was set to 2°K as a reasonable average value provided by the cooling system supplier. A total number of 96 rows of 10 constants were collected and labeled with suction gas temperature and compressor frequency, as well as the compressor evaporation and condensation range at the given operating conditions. Finally, P [kW] and Q [kW$_{th}$] were calculated using integers for the compressor evaporation and condensation range, resulting in a data set of approximately 30 000 example values. The COP of the compressor can then calculated by Eq. (A.2).

$$COP = \frac{Q}{P} \qquad \text{(A.2)}$$

A.3 ANN Approach Design and Configurations

Clearly, from Eq. (A.1), we understand that the function of the output is in a polynomial form. Although the functions between the inputs and the parameters c_i, $\forall i \in 1, 2, ..10$ are unknown, Eq. (A.1) provides important information which can be considered as an indicator of the overall hidden function that our ANN model is approximating. To approach a function that has polynomial features as the overall trend, we believe that a simple ANN with sigmoidal activition functions in the hidden layer is most probably sufficient Cybenko (1989). Therefore, instead of applying modern deep learning techniques, we start with a neural network structure with one hidden layer (HL) and gradually increase the number of layers and neurons to observe the learning behavior and efficiency. Fully connected ANNs, as shown in Fig. A.3 with single and multiple HLs with nonlinear activation functions were trained to predict P and Q by forward propagating data from the input neurons To, Tc, SGT and CF through the HLs as shown in Fig. A.3. Hyperbolic tangent Eq. (A.3) (Tanh), tangent sigmoid Eq. (A.4) (Tansig) from Esena et al. (2008), rectified linear unit Eq. (A.5) (ReLU) and sigmoid Eq. (A.6) activation functions were tried. Since the Tansig function used in Esena et al. (2008) is exponential, the ReLU function was also attempted. An Adam optimizer Kingma, Ba (2014) was used to train the networks until convergence using the Keras early-stop function. During training, the training data set was divided into batches of 100 examples so that the trainable model parameters could be updated after each batch. Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) for a single model, were used as loss functions and model accuracy metrics.

$$Tanh = \frac{e^z - e^{-z}}{e^z + e^{-z}} \qquad \text{(A.3)}$$

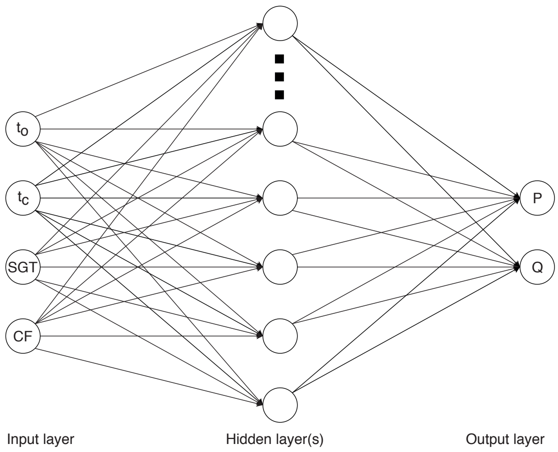


Figure A.3: Fully connected ANN with four neurons in the input layer and two neurons in the output layer.

$$Tansig = \frac{2}{(1+e^{-2z})-1} \tag{A.4}$$

$$ReLU = max(0,z) \tag{A.5}$$

$$Sigmoid = \frac{e^{z}}{1+e^{z}} \tag{A.6}$$

The models were programmed in Python 3.6 and the Keras library. Several network configurations were tested by changing the amount of HLs, the amount of neurons and the corresponding activation functions in each HL. Input values were normalized by subtracting the mean and normalizing the variance using Eqs. (A.7)-(A.10). The calculated values of μ and σ^2 for the training data set $\{X_i\}$ were also applied to the validation data set using Eq. (A.8) and Eq. (A.10).

$$\mu = \frac{1}{m}\sum_{i=1}^{m} X_i \tag{A.7}$$

$$X_i^{(\mu)} = X_i - \mu \tag{A.8}$$

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m}\left(X_i^{(\mu)}\right)^2 \tag{A.9}$$

$$X_i^{(\sigma^2)} = \frac{X_i^{(\mu)}}{\sigma^2} \tag{A.10}$$

A.4 Results and discussion

The data set consists of 29,408 values, divided randomly into a training and validation data set using a 90 / 10 split. The results for single HL models with a hyperbolic tangent (Eq. (A.3)) activation function are listed in Table A.3.

A single HL model with 45 neurons in the HL (Tanh-MSE-45) outperformed all multiple hidden layer models in all tested configurations. This includes models with multiple hidden layers, with both different and identical activation functions across the hidden layers. This result is logical based on the expected polynomial shape of the hidden

Table A.3: Results - One hidden layer ANN.

Activation-loss-neurons	Training epochs	Training loss	Validation loss
Tanh-MSE-7	6,184	0.008790	0.008639
Tanh-MSE-10	5,202	0.002355	0.002236
Tanh-MSE-15	4,029	0.001065	0.001047
Tanh-MSE-20	3,242	0.000644	0.000708
Tanh-MSE-25	2,398	0.000558	0.000554
Tanh-MSE-30	2,271	0.000530	0.000518
Tanh-MSE-35	2,620	0.000411	0.000379
Tanh-MSE-40	3,404	0.000367	0.000361
Tanh-MSE-45	2,868	**0.000291**	**0.000273**
Tanh-MSE-50	2,987	0.000319	0.000456
Tanh-MSE-55	2,742	0.000297	0.000294
Tanh-MSE-60	3,616	0.000326	0.000309

Table A.4: Results - Two hidden layers ANN.

Activation-loss-neurons-HL	Training epochs	Training loss	Validation loss
Tanh-MSE-5-2	2,010	0.008039	0.008952
Tanh-MSE-7-2	1,602	0.002343	0.002125
Tanh-MSE-10-2	1,641	0.001450	0.001293
Tanh-MSE-15-2	1,076	0.000880	0.000721
Tanh-MSE-20-2	1,127	0.000762	0.000945
Tanh-MSE-25-2	1,561	0.000483	0.000730
Tanh-MSE-30-2	712	0.000809	0.000797
Tanh-MSE-35-2	683	0.000675	0.000480
Tanh-MSE-40-2	828	0.000632	**0.000415**
Tanh-MSE-45-2	1,074	**0.000541**	0.0030
Tanh-MSE-50-2	879	0.000639	0.0013

ground-truth function. The system complexity is limited, and therefore does not require too many neurons in the hidden layer, as shown in Fig. A.4.

Table A.5: Results - One seven neuron hidden layer ANN with different activation and loss functions, and the best performing SVR model.

Model	Training epochs	Training loss (as MSE)	Validation loss
Tanh-MSE-7	6,184	0.0088	0.0086
Tansig-RMSE-7	1,763	0.0110	0.0106
Sig-MSE-7	21,043	0.0083	0.0079
ReLU-MSE-7	606	1.1444	1.1697
SVR-C1e10-RBF	-	0.0095	0.0094

For the single layer models, there was little difference between training and validation

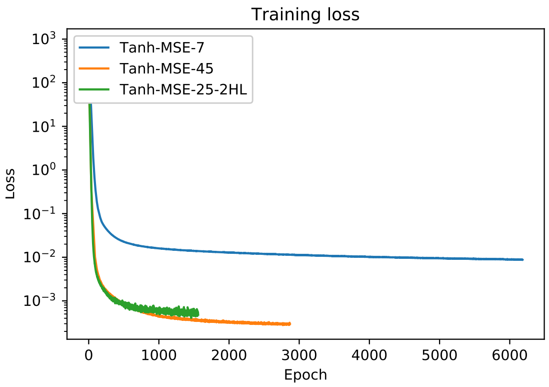


Figure A.4: Training loss comparison of the training error between Tanh-MSE-7, Tanh-MSE-45 and Tanh-MSE-45-2HL.

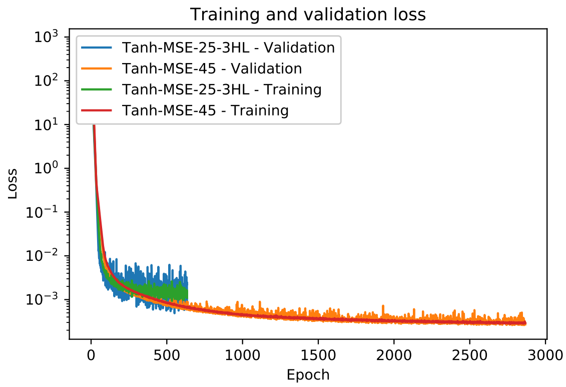


Figure A.5: Training loss comparison of the training and validation error between Tanh-MSE-45 and Tanh-MSE-25-3HL.

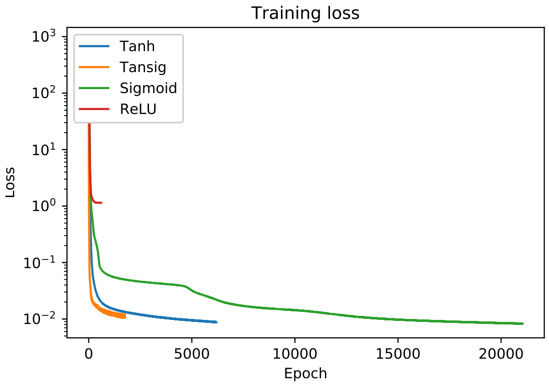


Figure A.6: Training loss comparison of the training error between different activation functions for single HL, seven neuron models.

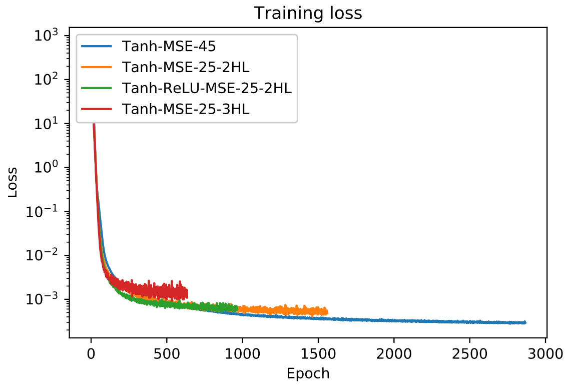


Figure A.7: Training loss comparison of the training error between the best performing models with one (Tanh-MSE-45), two (Tanh-MSE-25-2HL and Tanh-ReLU-MSE-25-2HL using tanh for the first HL and ReLU for the second) or three HL (Tanh-MSE-25-3HL).

error. In contrast, multiple layer models tended towards a higher validation error as well as bigger differences between training and validation, as exemplified in Fig. A.5, which is a sign of overfitting the training data. The multiple layer models also tended towards larger variations in loss between every update of the trainable parameters, which is to be expected since there are more parameters being updated after every training batch. This can be clearly observed in Fig. A.4 and in Fig. A.5 where the fluctuations increase with the amount of trainable parameters. This effect could perhaps be mitigated by tuning training parameters, but because the expected complexity of the hidden function is limited we do not believe that accuracy can be improved by adding more hidden layers. This assumption is supported by the results.

An ANN close to the best performing ANN in Esena et al. (2008)(Tansig-RMSE-7 in

Table A.5) was also trained in order to compare result accuracy with the most similar study found. The architecture of this network consists of a single HL with seven neurons and input values normalized to values between zero and one, using a tangent sigmoid activation function (Eq. (A.4)) and an RMSE loss function. One HL, seven hidden neuron models were also trained using sigmoid (Eq.(A.6)) and ReLU (Eq. (A.5)) activation functions. The tangent sigmoid model converges most quickly among the tested activations, but fluctuates significantly between every batch update. The sigmoid activation is the slowest to converge, but achieves the lowest MSE by a very small margin. This could be explained by the random initiation of parameters and we were not able to duplicate this result multiple times. Several Support Vector Regression (SVR) models were also trained with Sci Kit, using linear, polynomial and gaussian kernels. The best performing SVR, using a very large value for C and the gaussian kernel function is also included in Table A.5. Finally, a single HL 7 neuron model using the hyperbolic tangent activation function (Eq. (A.3)) was trained and yielded the best results, within a reasonable amount of training time, as shown in Table A.5 and Fig. A.6 which show the comparison of different activation functions. The best performing SVR model result is included in Table A.5. The result for the Tanh activation was achievable on multiple training sessions, indicating that the Tanh activation function is a reasonable fit for the training and validation data.

For models with two HL using the hyperbolic tangent (Eq. (A.3)) activation function, results are displayed in Table A.4. None of these models outperformed the Tanh-MSE-45 ANN, but the comparison of training error in Fig. A.7 show that the more complex two HL models converge and improve more quickly than the single layer models. This is expected since the amount of trainable parameters increases drastically once extra layers are added, but the fast learning rate quickly abates and results in fluctuating, indicating that the models are too complex for the underlying data.

Table A.6: Results - Tanh-MSE-45 model output compared with calculations done with software from Bitzer™.

t_o / t_c	SGT	CF	P Bitzer™	P Tanh-MSE-45	Q Bitzer™	Q Tanh-MSE-45	% SE
-35 / -5	-12.5	67	13.24	13.25	52.7	52.6	0.03
-35 / -5	-12.5	63	12.30	12.30	49.5	49.4	0.04
-35 / -5	-17.5	47	8.95	8.96	36.7	36.6	0.03
-35 / -5	-17.5	43	8.21	8.22	33.4	33.3	0.08
-35 / -5	-22.5	37	7.18	7.18	28.6	28.6	0.01
-35 / -5	-22.5	33	6.54	6.54	25.2	25.2	0.01

Finally, to examine how the Tanh-MSE-45 model performs on completely new data, some example calculations with Bitzer™ software were done using input data that fall in between the values that were used to generate the training and validation data sets. Specifically, instead of 5°C steps for SGT $\in -30, -25, .. -5$, the values -12.5, -17.5 and -22.5 were used. Similarly, values for CF were set to 67, 63, 47, 43, 37 and 33. Table A.6 show these results as well as % Squared Error (SE).

A.5 Conclusions

With a resulting MSE of 0.08%, we conclusively show that using an ANN to model the compressors in a cooling system is a valid approach that allows quick and quite accurate calculations of cooling load and compressor power. The compressor COP at the given operating conditions can then be calculated. The best result was achieved using a single HL ANN with a hyperbolic tangent activation function. The model was trained with a MSE loss function using the Adam optimizer. For this approach to be valuable to an IEMS, the transcritical compressors that interact directly with TES must also be modelled so that the full system performance can be calculated. For best use of the TES as a HES during winter, the maximal available and reclaimable heat must also be determined. The trained model will be part of a full CS model adopted in a real-world setting and used to determine the cooling load, compressor power load and COP as input to an IEMS for optimization purposes.

Bibliography

Arteconi A., Hewitt N.J., Polonara F. Domestic demand-side management (DSM): Role of heat pumps and thermal energy storage (TES) systems // Applied Thermal Engineering. 2013. 51, 1. 155 – 165.

Chen C., Duan S., Cai T., Liu B., Hu G. Smart energy management system for optimal microgrid economic operation // IET Renewable Power Generation. 2011. 5, 3. 258–267.

Chua K.J., Chou S.K., Yang W.M. Advances in heat pump systems: A review // Applied Energy. 2010. 87, 12. 3611 – 3624.

Cybenko George. Approximation by superpositions of a sigmoidal function // Mathematics of control, signals and systems. 1989. 2, 4. 303–314.

Esena Hikmet, Inallib Mustafa, Sengurc Abdulkadir, Esena Mehmet. Performance prediction of a ground-coupled heat pump system using artificial neural networks // Expert Systems with Applications. 2008. 35, 4. 1940–1948.

Kingma Diederik P., Ba Jimmy. Adam: A Method for Stochastic Optimization // CoRR. 2014. abs/1412.6980.

Manic M., Amarasinghe K., Rodriguez-Andina J. J., Rieger C. Intelligent Buildings of the Future: Cyberaware, Deep Learning Powered, and Human Interacting // IEEE Industrial Electronics Magazine. 2016. 10, 4. 32–49.

Neksa Petter. CO2 heat pump systems // International Journal of Refrigeration. 2002. 25, 4. 421 – 427.

Neksa Petter, Rekstad Håvard, Zakeri G.Reza, Schiefloe Per Arne. CO2-heat pump water heater: characteristics, system design and experimental results // International Journal of Refrigeration. 1998. 21, 3. 172 – 179.

Pardo N., Montero Á., Martos J., Urchueguía J.F. Optimization of hybrid – ground coupled and air source – heat pump systems in combination with thermal storage // Applied Thermal Engineering. 2010. 30, 8. 1073 – 1077.

Sarkar J., Bhattacharyya Souvik, Gopal M.Ram. Optimization of a transcritical CO2 heat pump cycle for simultaneous cooling and heating applications // International Journal of Refrigeration. 2004. 27, 8. 830 – 838.

T R., Jasmin E. A., Ahamed T. P. I. Residential Load Scheduling With Renewable Generation in the Smart Grid: A Reinforcement Learning Approach // IEEE Systems Journal. 2018. 1–12.

Venayagamoorthy G. K., Sharma R. K., Gautam P. K., Ahmadi A. Dynamic Energy Management System for a Smart Microgrid // IEEE Transactions on Neural Networks and Learning Systems. 2016. 27, 8. 1643–1656.

Wen Z., O'Neill D., Maei H. Optimal Demand Response Using Device-Based Reinforcement Learning // IEEE Transactions on Smart Grid. 2015. 6, 5. 2312–2324.

Zhao Z., Lee W. C., Shin Y., Song K.-B. An Optimal Power Scheduling Method for Demand Response in Home Energy Management System // IEEE Transactions on Smart Grid. 2013. 4, 3. 1391–1400.

Appendix B

Paper B

Title:	ANN modelling of CO2 refrigerant cooling system COP in a smart warehouse
Authors:	Sven Myrdahl Opalic, Morten Goodwin, Lei Jiao, Henrik Kofoed Nielsen, Ángel Álvarez Pardiñas, Armin Hafner, and Mohan Lal Kolhe
Affiliation:	University of Agder, Faculty of Engineering and Science, 4879, Grimstad, Norway and Norwegian University of Science and Technology, Kolbjorn Hejes vei 1B, 7034, Trondheim, Norway
Journal:	*Journal of Cleaner Production, Elsevier*, Volume 260, July, 2020
DOI:	10.1016/j.jclepro.2020.120887.

ANN modelling of CO2 refrigerant cooling system COP in a smart warehouse

Sven Myrdahl Opalic, Morten Goodwin, Lei Jiao, Henrik Kofoed Nielsen,
Ángel Álvarez Pardiñas, Armin Hafner, and Mohan Lal Kolhe
Department of Engineering Sciences
Faculty of Engineering and Science, University of Agder
4879, Grimstad, Norway
E-mails: {sven.opalic, morten.goodwin, lei.jiao, henrik.kofoed.nielsen,
mohan.l.kolhe}@uia.no
Department of Energy and Process Engineering
Norwegian University of Science and Technology
7034, Trondheim, Norway
E-mails: {armin.hafner, angel.a.pardinas}@ntnu.no

Abstract — **Industrial cooling systems consume large quantities of energy with highly variable power demand. To reduce environmental impact and overall energy consumption, and to stabilize the power requirements, it is recommended to recover surplus heat, store energy, and integrate renewable energy production. To control these operations continuously in a complex energy system, an intelligent energy management system can be employed using operational data and machine learning. In this work, we have developed an artificial neural network based technique for modelling operational CO_2 refrigerant based industrial cooling systems for embedding in an overall energy management system. The operating temperature and pressure measurements, as well as the operating frequency of compressors, are used in developing operational model of the cooling system, which outputs electrical consumption and refrigerant mass flow without the need for additional physical measurements. The presented model is superior to a generalized theoretical model, as it learns from data that includes individual compressor type characteristics. The results show that the presented approach is relatively precise with a Mean Average Percentage Error (MAPE) as low as 5 %, using low resolution and asynchronous data from a case study system. The developed model is also tested in a laboratory setting, where MAPE is shown to be as low as 1.8 %.**

B.1 Introduction

The building and construction sector, including energy intensive food distribution warehouses, is responsible for almost 40 % of total emissions related to energy and process (IEA, 2019a). Within the built environment, cooling demand is continually increasing as the weather grows warmer and a larger part of the worlds population and industrial enterprises gain access to air conditioning equipment and cooled building space (IEA, 2019b). The environmental impact of this trend can mainly be alleviated through a two-fold focus on energy efficient operation (Li et al., 2020; Zhu et al., 2019) and use of increasingly viable environmentally friendly refrigerants, such as carbon dioxide (CO_2), in the Cooling Systems (CS) (Mohammadi, McGowan, 2019; Sarkar et al., 2004; Neksa, 2002; Neksa et al., 1998).

Typically, in warehouses and distribution centers, comprehensive CSs are responsible for a big portion of the building's energy use. CS performance will also be affected by changes in the operational environment, including weather conditions, logistical operations, and workforce behavior (Chua et al., 2010; Sarkar et al., 2004). These effects are enhanced when dealing with environmentally friendly refrigerants, such as CO_2, that recently have seen an increase in utility due to environmental concerns (Schmidt et al., 2019). A cost efficient way to reduce environmental impact in the existing CSs is through energy efficient operation. This can be achieved in several ways, depending on the existing energy system design, such as optimized interaction with a Thermal Energy Storage (TES) (Širokỳ et al., 2011), surplus heat recovery (Chua et al., 2010) and optimized time-of-use with simultaneous access to local renewable energy resources (Wu, Wang, 2018; Kow et al., 2018). Implementing an Intelligent Energy Management System (IEMS) allows the building operator to automate the process of continuously choosing actions with the highest cost-reduction or energy-savings potential (T et al., 2018; Venayagamoorthy et al., 2016; Wen et al., 2015; Zhao et al., 2013; Chen et al., 2011). The IEMS takes advantage of the shift from Human-to-Machine to Machine-to-Machine communication, with access to large quantities of data through Internet/Intelligence of Things (IoT) components, and can incorporate the latest developments within Artificial Intelligence (AI) for prediction and control purposes (Hakimi, Hasankhani, 2020; Wu, Wang, 2018; Manic et al., 2016). The IEMS can handle various tasks, such as optimized utilization of energy storage options to reduce overall CS energy consumption (Širokỳ et al., 2011). TES systems can be used to enhance the CS performance by exploiting available heating and cooling capacity for optimum operation of energy storage during high-performance operating conditions. In a CS, the most important energy efficiency measure is the Coefficient of Performance (COP). The COP is a ratio of the useful thermal energy provided compared to the electrical work required. To determine the thermal component of this ratio in direct expansion systems that use the refrigerant for cooling energy distribution, we need an accurate measure of refrigerant flow.

Installing flow measuring equipment in existing CO_2 refrigerant, direct expansion CSs is a costly and complicated operation. The complexity and risk increases when the CS operates on multiple temperature levels with separate distribution systems. The most logical option for performance evaluation then becomes a theoretical calculation based on available operational data. In Zou, Xie (2017), a simplified model for COP modelling of a water source heat pump is suggested. Sun et al. (2017) proposes a general simulation model based on graph theory that utilizes accurate mathematical models of individual components, such as the Li (2013) suggested approach to variable speed compressors, to model refrigerant flow. Kim et al. (2018) conducted a case study of variable refrigerant flow simulation, tailored for building energy modelling, where the focus was calibration of a CS model to the U.S. DOE's EnergyPlus software. Zhu et al. (2013) proposes a generic model for variable refrigerant flow in air conditioning systems with multiple evaporators intended for simulation of performance and control analysis. None of the aforementioned studies propose models for multi-stage compression CS. Adaptation and implementation of the proposed methods would also require quite extensive knowledge of refrigeration technology and specific system design. Future IEMS systems might be dependant upon a realistic simulated environment to enable training of sophisticated Reinforcement Learning agents (Schrittwieser et al., 2019; Silver et al., 2018) that can adapt to and learn from operational data. A robust method that allows for cost effective, real-world implementation in complex, industrial scale, CO_2 direct expansion CS is needed. Since industrial scale CSs have to be specifically designed and built for each use case, a general calculation will be quite inaccurate. Intellectual Property (IP) rights tied to the individual components in the CS can also restrict options for full access to precise performance data. Some industrial CS suppliers provide access to web-based software designed for product selection and simple, static performance calculation, but the details necessary to build a more robust theoretical calculation model are not shared. An open, accurate, scalable, and reliable method for theoretical COP calculation is therefore needed.

Within the field of AI, an Artificial Neural Network (ANN) is a particularly powerful tool for hidden function approximation. ANNs trained on limited experimental data were successfully used for COP calculation in Esena et al. (2008). In Opalic et al. (2019) we showed that ANNs trained to model the electrical power utilized by Bitzer, a widely utilized compressor manufacturer, 4CSL12K compressors give highly accurate results, with an MSE of 0.08%, when compared to results attained from Bitzer software.

In this paper, we expand our scope by using ANNs to model all Bitzer compressors in a large and fully operational CO_2-based CS. To further examine the usefulness and real-world application of this approach, we compare electrical power measurements of a case study CS to the summed calculations of an ensemble of ANNs that each model a compressor type featured in the CS. We also verify our method by comparing our calculations to measurements from a comparable laboratory CS. We train the ANNs using available data collected from the compressor manufacturer's web-based software. The ANN training algorithm adjusts the weighting of the input parameters, as well as the weighted connections between neurons, to expertly fit the labeled training data. After we define the appropriate input and output parameters, our approach only requires limited knowledge of refrigeration technology and system design to be implemented in an operational setting. In CSs with access to a limited amount of desired performance measures, our approach can be used to supplement and enhance the value of the existing data. In such installations, the overlap between measurements and calculations can also be used to discover inconsistencies between theoretical and actual performance. To the best of our knowledge, our approach to linking theory and practice in multi-stage, CO_2 refrigeration technology using ANNs has not been attempted before. The proposed method is both practically feasible and useful in evaluating the energy performance of CO_2-based cooling installations. Owners and operators can use our ANN model ensemble approach for quality assurance of CO_2-based CSs.

We have designed our approach to:

- independently model the parts of the CS that interact with the TES at any given time, such that we can use the efficiency of this isolated part of the CS as input to an algorithm that optimizes the use of the TES;
- have a more accurate performance measure than what is currently available;
- create a data set that enables the development of CS future performance prediction models by applying our method to historical CS data;
- be able to calculate historical values of available excess heat, whereas what is currently known is only the amount of heat that was reclaimed and used;
- investigate to what extent ANNs can model complex scenarios consisting of several cooling compressors in a multi-stage CS – especially including transcritical conditions for CO_2.

We organize this article in the following manner. Section B.2 describes the components of a real-world advanced warehouse and logistical center that includes a case study cooling system, as well as the data collection process for model development. We present our CS model ANN architecture in Section B.3. Section B.4 is our discussion of results and implementation. Lastly, we present our conclusions and suggest future research efforts in Section B.5.

B.2 System structure and configuration

We based our work on information and data collected from a warehouse and food distribution center near Stavanger in Norway, completed in the fall of 2017. The main component of the warehouse energy system is an industrial CO_2 refrigerant CS consisting of three separate cooling plants circulating liquid CO_2 to evaporators in the frozen and chilled food storages. The CS also produces chilled water for cooling of the remaining building areas, including food storage, office space, and support areas. The architecture of the case-study cooling plant examined in this study is shown in Fig. B.2. An additional back-up and peak-load cooling machine also provides chilled water for ventilation and server cooling. CS surplus heat is recovered and utilized to heat tap water, to keep the ground beneath the frozen storage frost-free and to supply the non-cooled areas of the building with heating energy when needed. If there is insufficient excess heat available, the operating pressure of the CS is increased to satisfy the heating demand, up to a predefined maximum pressure level. Recovered heat can also be stored in a TES for future use, mainly to reduce the need for the electrical boiler at peak heating demand.

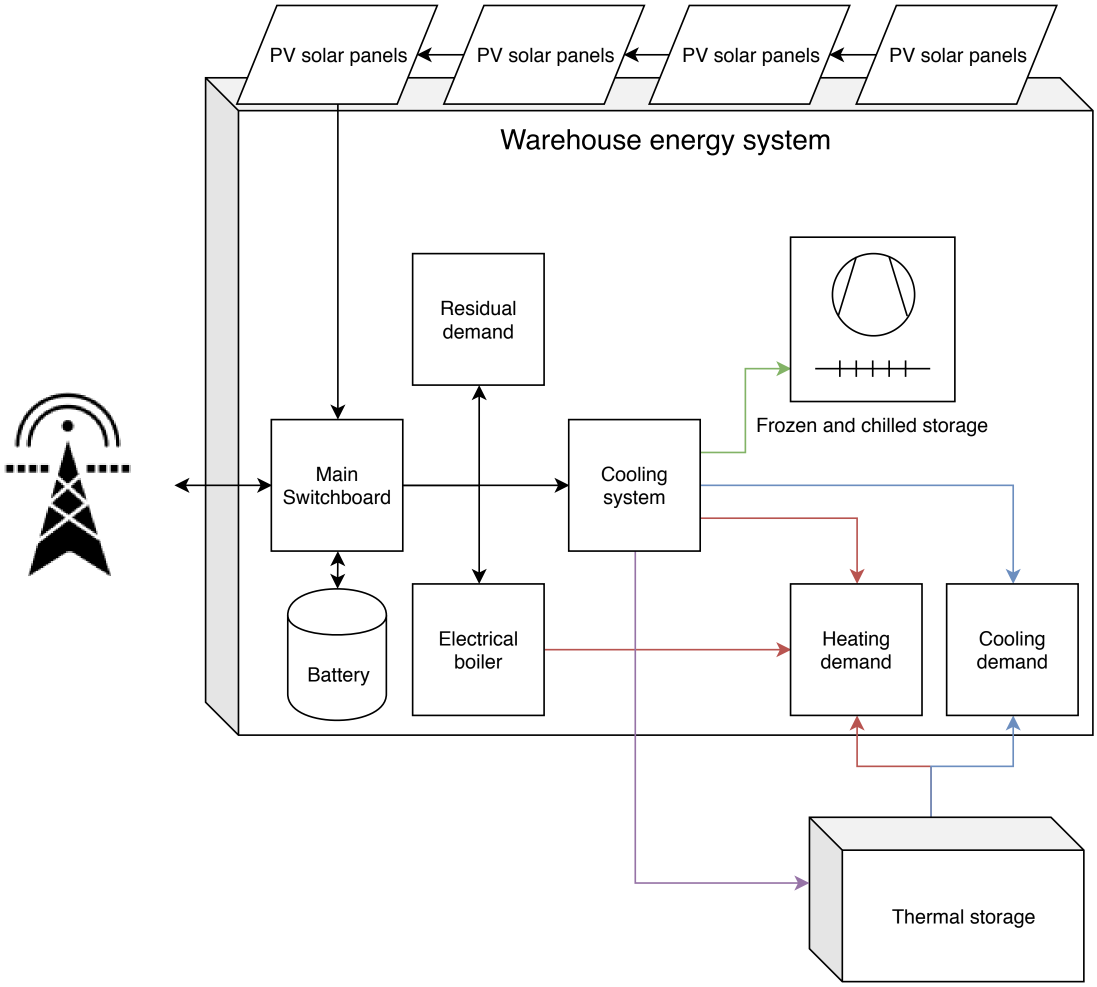


Figure B.1: The warehouse energy system.

The warehouse also exhibits a considerable photovoltaic (PV) power generation plant, a lithium-ion battery system (LBS), and a buried and insulated 300 m^3 firewater tank connected to a heat exchanger that is utilized as a TES. An electrical boiler is employed for back-up and peak demand heating. Table B.1 contains a list of the operational temperature range in the various warehouse areas, whereas Fig. B.1 and Table B.2 visualizes and lists the main components of the warehouse energy system. The PV plant supplies A/C power directly to the main switchboard. If demand is sufficient, all the PV energy is utilized in the building. Otherwise, energy is stored in the LBS, converted to thermal energy and stored in the TES or exported to the main grid. In addition to storing surplus solar energy, the LBS is used for power peak reduction. Thermal energy in the form of chilled or heated water can be stored in the TES, represented by the purple arrow in Fig. B.1. The IEMS tasked to control the energy storage systems applies proven machine learning algorithms to predict PV power generation, as well as the future demand for thermal and electrical energy. An optimization algorithm then employs the predictions to calculate the most cost-effective hourly schedule for charging and discharging.

The IEMS controls the TES in two separate seasonal modes of operation, Heat Energy Storage (HES) and Cold Energy Storage (CES). It employs CES mode from around March to November, and HES for the remainder of the year. Natural reduction of the

Table B.1: Warehouse dimensions and temperatures.

Area	Size	Operating temperature
Dry storage, office space, etc.	19,000 m^2	18-22°C
Frozen	3,000 m^2	-20°C
Chilled	3,500 m^2	0-4 °C
Chilled distribution	3,500 m^2	0-4 °C

Table B.2: Components' specifications.

Component	Capacity	Unit of measurement
PV - photovoltaic power generation	1,000	[kW$_p$]
LBS - lithium-ion battery system	460/200	[kWh/kW]
TES - thermal energy storage	300/300	[m^3/kW$_{thermal}$]
CS - cooling system	1,140	[kW$_{thermal}$]

cooling demand occurs as outside temperature decreases towards the winter season. As a result, surplus heat available for recovery is no longer able to sustain the warehouse's overall demand for heating. However, by storing heating energy reclaimed from the CS in advance, the load on the electric boiler can be severely reduced, which in turn reduces the consumption of energy and the cost of peak power.

In CES mode, the IEMS attempts to balance two main strategies:

1. Storing surplus electricity generated by the PV installation in the CES through energy conversion.

2. Producing and storing chilled water at high COP conditions.

When the IEMS applies strategy number one, the CS converts surplus electricity to chilled water for storage in the CES at a temperature range between 7°C and 15°C. In the evening, when the natural reduction of power output from the PV-plant occurs, the IEMS may choose to discharge the CES and thereby reducing power requirements for the CS. The second strategy involves optimizing the production of cooling energy by decoupling it from the consumption through the CES. The IEMS optimization algorithm accomplishes this through the utilization of cooling demand predictions, weather predictions, and table base COP values.

The IEMS currently uses a simplified approach with a provided table of COP values to evaluate performance at given ambient temperature and operating conditions. Future COP values can then be estimated using weather predictions. The COP table is a rough metric that does not supply the optimization algorithm with quantitative input, such as expected cooling production at the separate CS stages and total available excess heat. Also, the Building Management System (BMS) provides a general CO$_2$ CS model that calculates all the necessary parameters, but with unsatisfactory accuracy.

We, therefore, suggest an ANN approach to calculate compressor mass flow and electricity consumption. Calculating cooling capacity instead of mass flow would be preferable. However, due to unavailability of cooling capacity data for all the compressors,

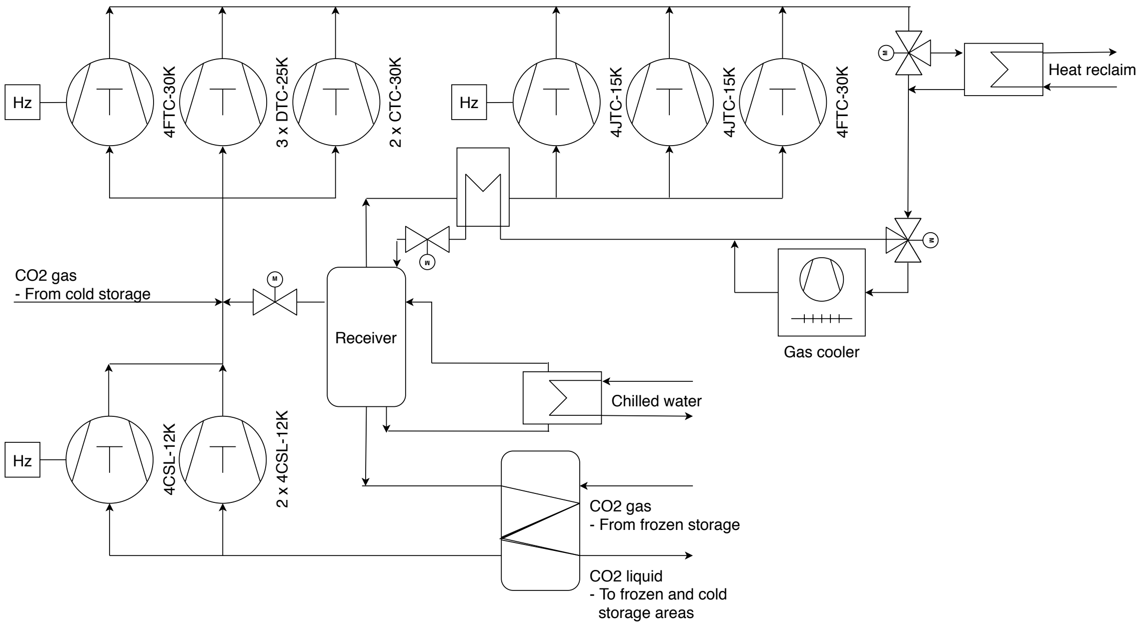


Figure B.2: On-site cooling plant architecture.

we use mass flow as an alternative approach. We have developed models for all the compressors in the cooling system. Two models have been developed for each transcritical compressor so that we use separate models of the same compressor for calculations in the subcritical and transcritical operational modes. The compressors are semi-hermetic reciprocating compressors manufactured by Bitzer GmbH, with one frequency-controlled compressor at each stage. Fig. B.2 shows the placement of all the compressors in a simplified cooling system architecture. There are two pressure stages of compression as well as parallel compressors to handle flash gas in the receiver and chilled water production. The compressors for the frozen storage areas are displayed in the bottom left, with the cold storage compressors in the top left and the parallel compressors in the top right. Fig. B.2 also displays mass flow direction and the most crucial CS components. It can be noted that the CO_2 based cooling system is a highly complex part of the energy system in the considered technologically advanced warehouse. Fig. B.2 is an element of Fig. B.1.

The website of the manufacturer was used to collect data (Bitzer-GmbH, 2019). Theoretical values for cooling capacity (Q), electrical power (P), electrical current (I) or mass flow ($\dot{m}$), which can all be substituted for the parameter y in Eqs. B.1 and B.2, can then be separately calculated by using the appropriate constants $c_i, \forall i \in 1, 2, ..10$ in the following polynomials (according to EN (2013)), for subcritical pressure conditions

$$\begin{aligned} y_{sc} = \quad & c_1 + c_2 t_o + c_3 t_c + c_4 t_o^2 + c_5 t_o t_c + c_6 t_c^2 + c_7 t_o^3 + \\ & c_8 t_c t_o^2 + c_9 t_o t_c^2 + c_{10} t_c^3, \end{aligned} \tag{B.1}$$

and, for transcritical pressure

$$\begin{aligned} y_{tc} = \quad & c_1 + c_2 t_o + c_3 p_{HP} + c_4 t_o^2 + c_5 t_o p_{HP} + c_6 p_{HP}^2 + \\ & c_7 t_o^3 + c_8 p_{HP} t_o^2 + c_9 t_o p_{HP}^2 + c_{10} p_{HP}^3. \end{aligned} \tag{B.2}$$

In Eqs. (B.1) and (B.2), t_o (°C) is representing temperature of evaporation and t_c (°C) is the condensation temperature, whereas $p_{HP}[bar]$ is the discharge pressure of the compressors at transcritical operating conditions where $p_{HP} > 73.77[bar]$. The constants c_1 through c_{10} depend on suction gas temperature (SGT, °C) and compressor operating frequency (CF, Hz) for subcritical operating conditions, while gas cooler outlet temperature (GOT, °C) must also be selected for transcritical operation. Separate and independent sets of constants are used to calculate Q (kW$_{thermal}$), P (kW), I (A) or $\dot{m}$ (kg/h) when used with Eqs. (B.1) and (B.2). Constants for P and $\dot{m}$ were collected in 5 degree steps for SGT and GOT within each compressors defined operational range, and 5 Hz steps for CF between 70 and 30 Hz. P and $\dot{m}$ example values were then calculated and labelled appropriately using integers for t_o, t_c and p_{HP}, resulting in data sets ranging from approximately 10 000 to 100 000 training examples for each compressor model.

Finally, we can determine cooling production, available excess heat, and the COP of any part of the system through calculations. For example, $\dot{m}$ can be used to calculate cooling load with the enthalpy difference equation

$$Q_c = \frac{\dot{m}\Delta h_c}{3600}, \tag{B.3}$$

where Δh_c (kJ/kg) is the specific enthalpy difference of the refrigerant between the outlet and inlet of a specific evaporation stage. Pressure and temperature of the subcooled liquid refrigerant before the expansion device (evaporator inlet conditions), along with the pressure and temperature of the superheated gas (evaporator outlet conditions), are measured. Specific enthalpy at the inlet and outlet of the evaporation stage is therefore known and can be used to calculate the specific enthalpy difference. We can then calculate the COP_c of a single, or multiple, compressor(s) using Eq. (B.4)

$$COP_c = \frac{Q_c}{P}. \tag{B.4}$$

B.3 ANN approach design and configuration

We chose the appropriate ANN design for compressor modelling by analyzing the Bitzer software and the available data. Clearly, in Eqs. (B.1) and (B.2), we can observe the characteristics of a polynomial function. Even though the relationship between the input variables and the constants c_i, $\forall i \in 1, 2, ..10$ are unknown, Eqs. (B.1) and (B.2) provide important information which we consider an indication of the hidden function we are attempting to approximate with ANNs.

In the considered ANN approach design and configuration, the patterns are discovered by such a function via training the ANN employing a hyperbolic tangent (tanh) activation function (Opalic et al., 2019; Cybenko, 1989). We, therefore, use the most suited neural network architecture found in (Opalic et al., 2019), namely using one hidden layer (HL) containing 45 neurons. Fully connected ANNs are configured to calculate P and $\dot{m}$ by feed-forwarding input data through the neurons in the HL as shown in Fig. B.3. We have trained compressor models for subcritical operating conditions with data sets generated with Eq. (B.1), while Eq. (B.2) was utilized to generate the data sets for the transcritical operation model training. The Adam optimizer (Kingma, Ba, 2014) has been applied to update the weights of the neural networks during training. The training continued until

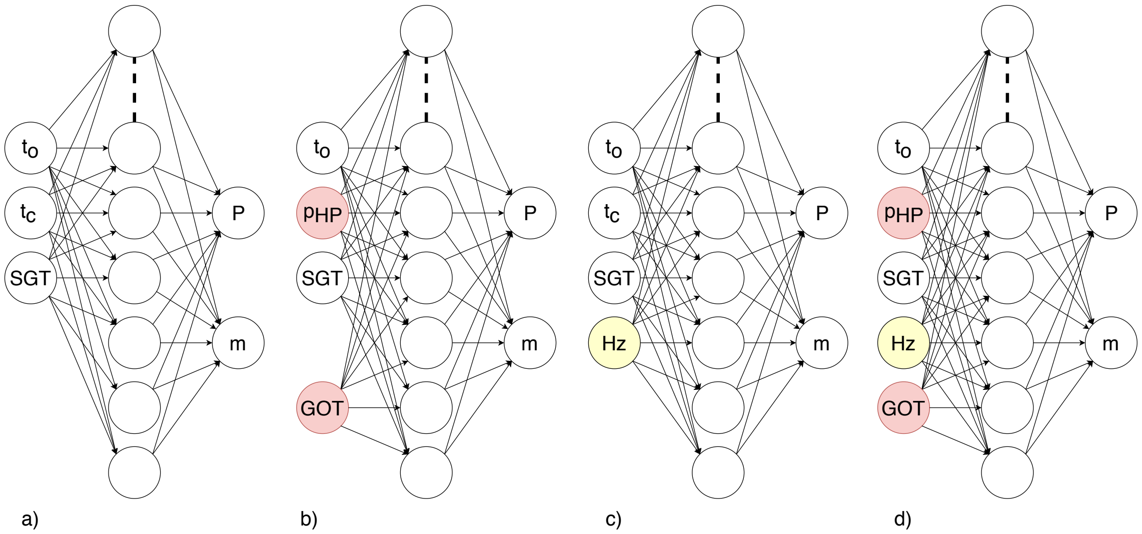


Figure B.3: ANN model architectures: a) Subcritical operation, b) Transcritical operation, c) Subcritical and frequency controlled, d) Transcritical and frequency controlled.

model learning converged by using the early-stop method in the Keras (Chollet, others, 2015) programming library, with the ”patience” parameter set to 150 epochs.

We set the training optimizer to update the trainable parameters after each training batch, consisting of 100 training examples. We have used Mean Squared Error (MSE) as the loss function while MSE and Mean Average Percentage Error (MAPE) were used as model accuracy metrics.

The models are programmed using Python 3.6 and Keras (Chollet, others, 2015). We divided the data sets into training and validation data through randomization and a factor of 0.9 to 0.1, respectively. We normalized the input values by mean (μ) subtraction and adjusting for variance (σ^2). The resulting values of μ and σ^2 calculated on the training data set $\{X_i\}$ were then employed to also adjust the validation data set.

We finally assembled the individually trained models in accordance with the design of the case-study CS shown in Fig. B.2. Operational data from the cooling system was gathered in order to compare the aggregated output of the ANN models for running compressors to the metered power input. In addition to t_o, t_c, P_{HP}, SGT, CF and GOT, compressor operating status for each compressor was collected. For every timestep, our algorithm utilizes the operational data to determine which compressors are operational, the CF of the frequency controlled compressors, and whether the CS pressure level exceeds the transcritical threshold. The data for the active compressors, in the appropriate operational mode, is then selected and sorted into the format shown in Fig. B.3, and fed into the input layers of the selected models. The resulting model output is finally summed for each separate stage of compression and compared to the metered power input to the CS.

However, none of the data is temporally synchronized. Accordingly, the raw data had to be processed and aligned in order for comparisons to be made. The data processing introduces an error source that has to be taken into account when observing the results. Also, a third-party BMS, utilizing serial bus communication for data gathering, is responsible for collecting the power measurements and operational data from the cooling system.

The BMS only timestamps the data when it is received. There is no timestamp for when the data was requested or when the cooling system controller received the request (the actual time of measurement). This lack of clarity adds another layer of uncertainty to the temporal accuracy and integrity of the raw data. By request, the BMS operator increased the frequency of data collection in June 2019 in order to increase input data quality.

An analysis of the raw data also shows that even when measured power input drops to zero, the BMS will still show active compressors, and accordingly, the models will predict the individual compressor power usage. Therefore, we have removed all data points with a power measurement of zero in the data cleaning process.

An alternative research approach would have been to structure the training data so that a single model could be used to predict the aggregated output. We only briefly considered this alternative as such an approach would have included removing known information and system boundaries from the training process only to have the information, hopefully, relearned by the single model. Also, we would have removed the advantage in our chosen approach of being able to model separate stages in the cooling system, while transfer learning by reusing already trained compressor models in other cooling systems would have been more difficult.

There is no flow measuring equipment in the case-study CS that can be used to verify the accuracy of the aggregated model. Therefore, we also tested our method with data from an ongoing experiment at the Norwegian University of Science and Technology (NTNU) laboratory CS. The NTNU CS has a very similar design to the case-study CS, while also measuring the flow of CO_2 through each compressor stage and the individual electrical power input of each compressor. The compressors in the NTNU CS parallel stage, consisting of a Bitzer 2KTE-7K-40S (Inverter driven), Bitzer 2KTE-7K-40S (set to fixed speed) and Bitzer 4JTC-15K-40S (fixed speed), were modeled using our previously described ANN configuration approach. Part of the pressure and temperature sensors in the NTNU CS are connected to Danfoss controllers which sample and log the data in 5-second intervals. Mass flow meters, temperature sensors, and active power consumption meters for the compressors are connected to National Instruments Hardware, and the data is logged by LabVIEW software with a sampling time of 1 second. LabVIEW software also handles information coming from the inverters (frequency, power, etc.), connected by Modbus, with a 5 second sampling time. NTNU researchers finally synchronize all the data in MATLAB with in-house software.

B.4 Results and implementation plan

B.4.1 Results analysis

In this paper, we attempt to model the compressors in an operational, industrial CS using ANNs. We trained the ANNs with data generated by calculating power input and mass flow of Bitzer CO_2 CS compressors using polynomials, subject to openly available constants, for subcritical and transcritical conditions. The difference between training and validation error, as shown in Table B.3, is minimal in all cases. Therefore, we could likely have used a more significant part of the data sets for training without risk of overfitting. Table B.3

Table B.3: Training and validation MSE for all models. Separate models for frequency controlled (FC) compressors and transcritical (TC) operation.

Compressor model	Training MSE	Validation MSE
Bitzer 4CSL12K	2,97E-05	2,48E-05
Bitzer 4CSL12K FC	2,37E-05	1,60E-05
Bitzer 4CTC30K	3,90E-05	3,17E-05
Bitzer 4CTC30K TC	7,79E-06	4,57E-06
Bitzer 4DTC25K	1,84E-05	2,01E-05
Bitzer 4DTC25K TC	6,20E-06	2,89E-06
Bitzer 4FTC30K	6,76E-05	6,50E-05
Bitzer 4FTC30K FC	2,68E-05	1,74E-05
Bitzer 4FTC30K FC TC	1,28E-05	7,85E-06
Bitzer 4FTC30K TC	1,54E-05	1,09E-05
Bitzer 4JTC15K	1,87E-05	1,34E-05
Bitzer 4JTC15K FC	2,34E-05	1,82E-05
Bitzer 4JTC15K FC TC	2,19E-05	1,54E-05
Bitzer 4JTC15K TC	7,26E-06	6,91E-06

Table B.4: Monthly MSE and MAPE comparison from January 2019 to July 2019. Separate columns for subcritical (SC) and transcritical (TC) operating conditions.

Month	MSE	MSE TC	MSE SC	MAPE	MAPE TC	MAPE SC
January	112.3	120.7	104.1	15.8 %	14.9 %	16.7 %
February	102.8	106.9	101.7	15.4 %	12.4 %	16.2 %
March	90.7	112.0	85.4	14.7 %	13.8 %	14.9 %
April	145.7	187.9	136.3	18.3 %	18.7 %	18.2%
May	88.0	130.9	79.7	16.3 %	18.3 %	15.9 %
June	44.2	34.6	45.5	12.0 %	6.1 %	12.8 %
July	38.8	31.1	42.6	10.1 %	5.8 %	12.3 %

lists the training and validation MSE results for each compressor model. Table B.3 shows that the models are highly accurate when compared to training and validation data sets generated with Eq. (B.1) and (B.2) and can therefore be expected to give very similar results to the hidden ground-truth theoretical models.

Table B.4 shows results for aggregated model output compared to metered power input to the case study CS every month from January 2019 to July 2019. We observe an increase in aggregated model predictive accuracy compared to power measurements in June and July, which is likely due to the increased data collection frequency implemented in the BMS. Fig. B.4 and B.5 show monthly plots for the worst (April) and best (July) months. Making any visual distinction between these months is difficult, but an apparent trend in both months is that the largest discrepancies between predicted and actual power input exists in the lower spectrum of power usage. Sudden drops in measured power input, not reflected in the CS BMS data, is a probable cause of this trend. It is therefore likely that

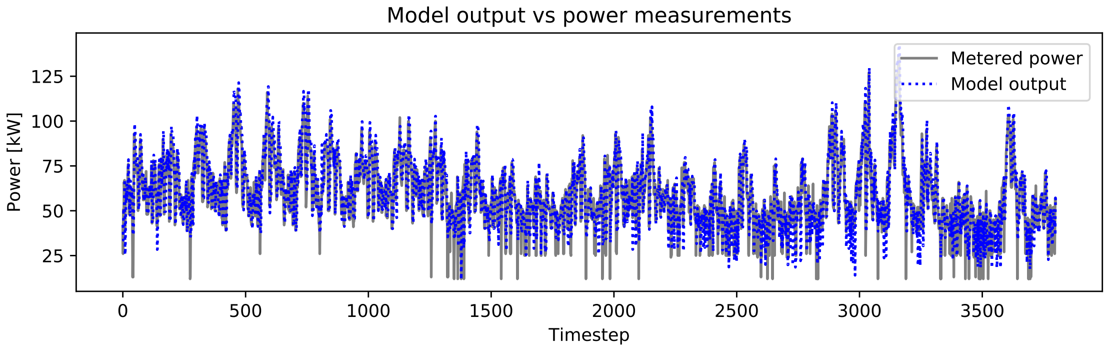


Figure B.4: April 2019 - Aggregated model output compared with metered power input to CS.

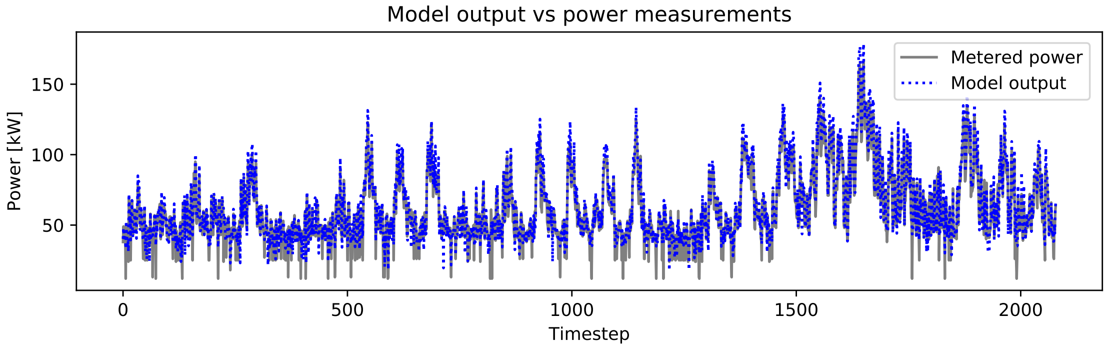


Figure B.5: July 2019 - Aggregated model output compared with metered power input to CS.

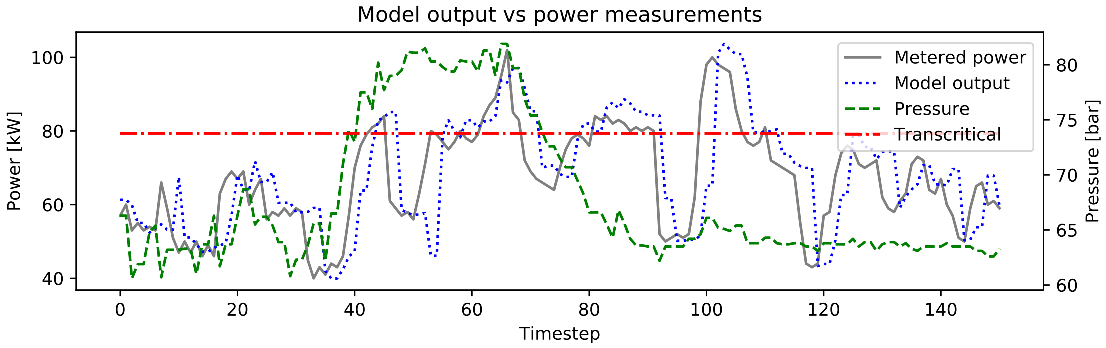


Figure B.6: 2019-04-10, 24 hours - Aggregated model output compared with metered power input to CS.

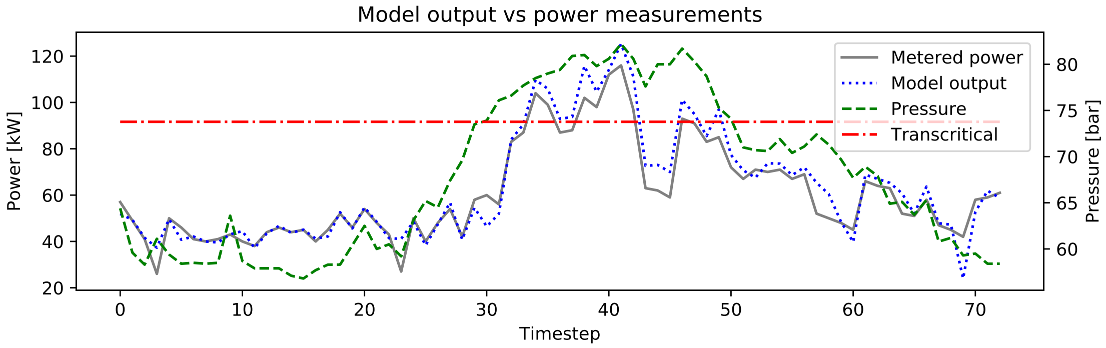


Figure B.7: 2019-07-16, 24 hours - Aggregated model output compared with metered power input to CS.

there is an error in the raw CS data connected to sudden drops in power input, perhaps due to sudden switches between compressors or a rapid decrease in cooling demand when local evaporator set-point temperature conditions are met. We find further evidence of this when examining the differences between MSE and MAPE in TC or SC operation during warmer or colder months. Table B.4 shows that the MSE and MAPE in transcritical operating conditions are higher than in subcritical operation for January through May, while the opposite is true in June and July. Since heat is reclaimed from the CS and used for heating purposes, pressure is increased in the winter months when the heating distribution system requests more energy concurrently with or caused by drops in cooling demand. Inversely, during the summer months, pressure increases are usually caused by an increase in ambient temperature and cooling demand. Therefore, the conditions likely to cause the most significant discrepancies occur most often in TC operation in the colder parts of the year and SC operation during the summer, possibly leading to the observable differences in TC and SC MSE and MAPE in Table B.4.

Since the monthly plots are quite hard to read due to a large number of data points, we include plots of a single random day in April and July in Fig. B.6 and B.7. These plots show the importance of the increased quality of the aggregated model input data. Fig.B.6 indicates a temporal displacement between the aggregated model output and the power measurements when compared to Fig. B.7.

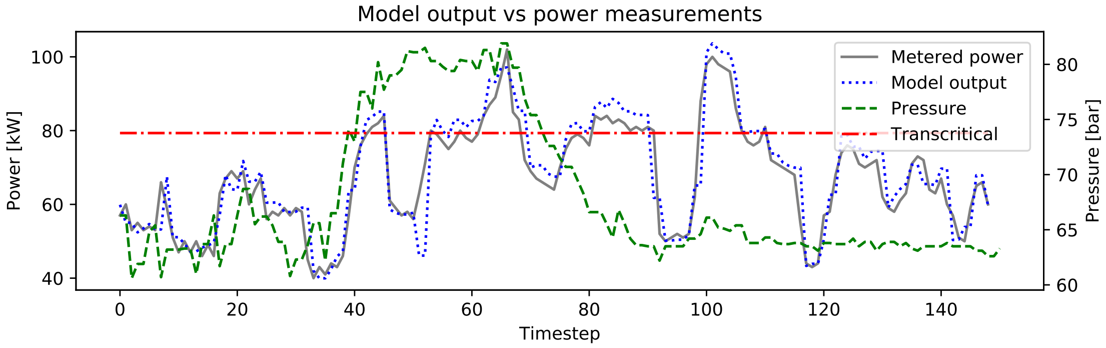


Figure B.8: 2019-04-10, 24 hours - Aggregated model output compared with metered power input to CS, adjusted for $\tau = -2$.

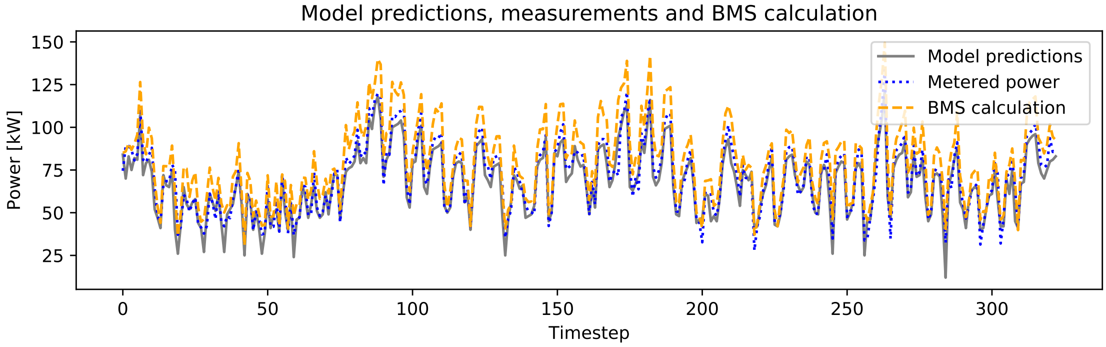


Figure B.9: 2019-08-22 to 2019-08-26 - Aggregated model output compared with metered power input to CS and BMS calculated values.

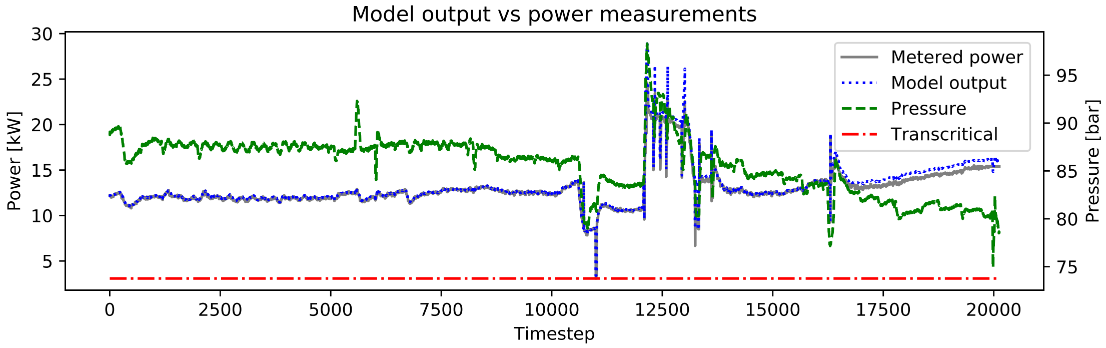


Figure B.10: 2019-08-25 - Aggregated model power calculation compared with metered power input (inverter) at the NTNU laboratory CS.

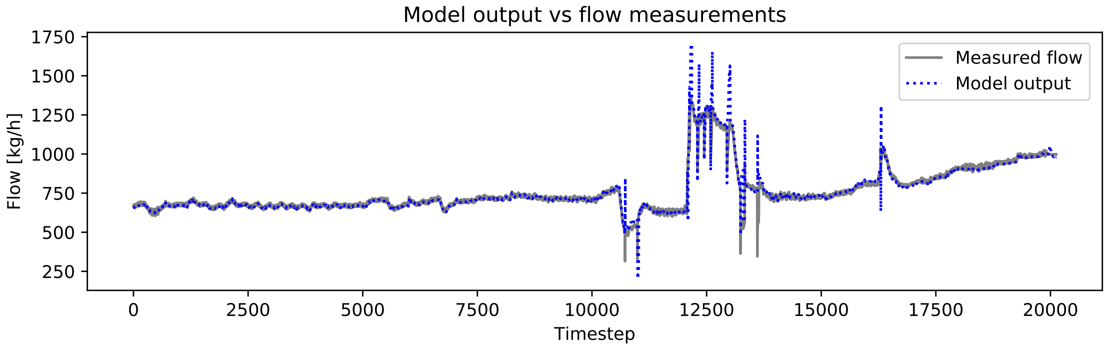


Figure B.11: 2019-08-25 - Aggregated model flow output compared with measured flow at the NTNU laboratory CS.

Due to the jitters in time for the input data, the model output and the power measurements are not precisely temporally aligned. To illustrate and compare the results accordingly, we introduced an offset τ in the time domain to align the two data series. In more detail, we shift $P(t)$ by $\tau \in [-10, 10]$ to find the maximum output of $\max_\tau \sum_t M(t)P(t+\tau)$. In this way, we can probably achieve a more appropriate time alignment. The maximum was found at $\tau = -2$. Adjusting accordingly reduces the MSE in April from 145.7 to 50.5 and the MAPE from 18.3 % to 10.1 %. For 2019-04-10 in Fig. B.6 the MSE was reduced from 133.3 to 29.4 and MAPE from 12.8 % to 5.5 %, results shown in Fig. B.8.

We also compare our aggregated model to BMS calculations. BMS calculation parameters were first adjusted to maximize accuracy on 2019-08-22. Results for 2019-08-22 to 2019-08-26 are plotted in Fig. B.9. Aggregated model calculation MSE on this sample is 41.7, while the MSE for the BMS calculation is 206.2. Similarly, our model calculation MAPE is 8.5 % compared to 20.1 % for the BMS calculation.

Finally, we use data, collected through sensor networks, from an ongoing NTNU CS experiment to validate our approach in a laboratory setting. Measurements of power and flow in the ongoing experiment are compared to the outputs of our aggregated ANN model. The NTNU experiment was conducted in transcritical operating conditions, with pressure ranging from 74.9 bar to 98.3 bar. Results are plotted in Fig. B.10 and B.11. We obtain a MAPE of 3.13 % when comparing the output from the ANNs with measurements from the power meters, whereas using measurements from the inverter for the frequency controlled compressor reduces MAPE to 1.87 %. Measurements from the power meters includes the power consumption of the inverter as well as power conversion losses. The increased accuracy, when using measurements in the inverter, suggests that the aforementioned losses are not included in the Bitzer software (Bitzer-GmbH, 2019) calculations. The result for the ANN flow output compared to NTNU CS measurements is 1.76 % MAPE. These results show that the presented method is accurate, when given synchronized data with a low sampling time period. Our results also suggest that the underlying ground truth mathematical function for each compressor type could possibly be unknown to the compressor manufacturer. The form that the available data is given in, combined our highly accurate results in a laboratory setting, suggest that the values for the constants could be based on empirical testing of each compressor. If this is the case, our approach could also be a useful way for the compressor manufacturer to easily encode all their laboratory data in neural networks that can be employed in their own calculation software.

B.4.2 Implementation in the operational setting

Industrial CSs are very power intensive and produce large amounts of surplus heat that is often discarded. In the case study warehouse, excess heat from the CS can be effectively used or stored in the TES to reduce the need for additional heating supplied by the electrical boiler, as described in Fig. B.1. Chilled water can be produced and stored in the TES during periods of favorable CS operating conditions and low energy prices, or access to surplus solar energy that would otherwise be exported to the main grid at a severely reduced energy price. The IEMS can facilitate energy management and reduction of the operational demands in an intelligent way to reduce energy cost and environmental impact. To optimize CS and TES interaction, the time-varying performance of the CS is required. The presented ANN model is currently being implemented and configured to supply the IEMS with compressor power consumption and refrigerant mass flow. Our software has been installed at a dedicated local server and communicates directly with the BMS through an Application Programming Interface (API) developed by the BMS provider, utilizing the JSON-RPC 2.0 protocol. The IEMS then collects live data as needed from the BMS through a local gateway setup.

Historical data generated with our ANN ensemble has also been supplied to the IEMS provider to allow development of predictive models of CS performance. The performance prediction model is developed with machine learning tools and will be utilized as input to the IEMS optimization algorithm. The output of the presented aggregated ANN model will improve the performance of the smart warehouse IEMS by increasing the quality of its necessary input data. The energy management system operator will also use these measures for quality assurance and performance evaluation through visualization in the Building Energy Management System.

B.5 Conclusions

Industrial cooling systems are responsible for a considerable amount of the buildings total energy use and environmental impact. To reduce energy consumption and conserve the environment, it is recommended to recover and store surplus heat, and optimize system operation for utilizing it in coordination with intermittent renewable energy production. These tasks have to be managed intelligently in a complex energy system with dynamic operation of various sub-systems / components. In this work, we have presented ANN model of an operational CO_2-based industrial cooling sub-system of a complex warehouse energy system. The operating temperature and pressure measurements, as well as the operating frequency of frequency-controlled compressors, are used in developing the operational model. The output of the model is electrical consumption and refrigerant mass flow for the compression process. The presented technique is relatively superior to a general theoretical model, both in terms of accuracy, flexibility, cost effectiveness, and implementability in the real-world application.

The developed model has MAPE in the range of 5 % to 12 % in the operational case-study cooling system. The presented results also indicate that the accuracy can be drastically improved with increased quality of data collection frequency in the operational measurement, supported by a MAPE of 1.87 % and 1.76 % in a comparable laboratory CS, for power and flow respectively. The accuracy of the presented ANN flow calculations is promising from a practical standpoint, and can be implemented through Machine-to-Machine communication using IoT related devices.

The developed modelling of the cooling system is currently being implemented in the case study energy system (Fig. B.1). The energy system operator has already noticed improvement in the performance calculation accuracy. The energy system operator will also use these embedded measures for quality assurance and performance evaluation of the building energy management system. Implementation of our approach in current, and future RL, IEMS solutions should be explored. Additional training of the developed models, based on increasing amounts of operational data, could also be further examined.

Bibliography

Bitzer-GmbH . Bitzer Software v6.10.2 rev2250. 2019.

Chen C., Duan S., Cai T., Liu B., Hu G. Smart energy management system for optimal microgrid economic operation // IET Renewable Power Generation. 2011. 5, 3. 258–267.

Chollet François, others . Keras. 2015.

Chua K.J., Chou S.K., Yang W.M. Advances in heat pump systems: A review // Applied Energy. 2010. 87, 12. 3611 – 3624.

Cybenko George. Approximation by superpositions of a sigmoidal function // Mathematics of control, signals and systems. 1989. 2, 4. 303–314.

Refrigerant compressors - Rating conditions, tolerances and presentation of manufacturer's performance data. 2013. Brussels, Belgium, july 2013.

Esena Hikmet, Inallib Mustafa, Sengurc Abdulkadir, Esena Mehmet. Performance prediction of a ground-coupled heat pump system using artificial neural networks // Expert Systems with Applications. 2008. 35, 4. 1940–1948.

Hakimi Seyed Mehdi, Hasankhani Arezoo. Intelligent energy management in off-grid smart buildings with energy interaction // Journal of Cleaner Production. 2020. 244. 118906.

IEA . Energy Efficiency 2019. 2019a.

IEA . Global Status Report for Buildings and Construction 2019. 2019b.

Kim Dongsu, Cox Sam J., Cho Heejin, Im Piljae. Model calibration of a variable refrigerant flow system with a dedicated outdoor air system: A case study // Energy and Buildings. 2018. 158. 884 – 896.

Kingma Diederik P., Ba Jimmy. Adam: A Method for Stochastic Optimization // CoRR. 2014. abs/1412.6980.

Kow Ken Weng, Wong Yee Wan, Rajkumar Rajprasad, Isa Dino. An intelligent real-time power management system with active learning prediction engine for PV grid-tied systems // Journal of Cleaner Production. 2018. 205. 252 – 265.

Li Ling-Ling, Liu Yu-Wei, Tseng Ming-Lang, Lin Guo-Qian, Ali Mohd Helmi. Reducing environmental pollution and fuel consumption using optimization algorithm to develop combined cooling heating and power system operation strategies // Journal of Cleaner Production. 2020. 247. 119082.

Li Wenhua. Simplified steady-state modeling for variable speed compressor // Applied Thermal Engineering. 2013. 50, 1. 318 – 326.

Manic M., Amarasinghe K., Rodriguez-Andina J. J., Rieger C. Intelligent Buildings of the Future: Cyberaware, Deep Learning Powered, and Human Interacting // IEEE Industrial Electronics Magazine. 2016. 10, 4. 32–49.

Mohammadi Kasra, McGowan Jon G. A thermo-economic analysis of a combined cooling system for air conditioning and low to medium temperature refrigeration // Journal of Cleaner Production. 2019. 206. 580 – 597.

Neksa Petter. CO2 heat pump systems // International Journal of Refrigeration. 2002. 25, 4. 421 – 427.

Neksa Petter, Rekstad Havard, Zakeri G.Reza, Schiefloe Per Arne. CO2-heat pump water heater: characteristics, system design and experimental results // International Journal of Refrigeration. 1998. 21, 3. 172 – 179.

Opalic Sven Myrdahl, Goodwin Morten, Jiao Lei, Nielsen Henrik Kofoed, Kolhe Mohan Lal. Modelling of Compressors in an Industrial CO2-Based Operational Cooling System Using ANN for Energy Management Purposes // Engineering Applications of Neural Networks. Cham: Springer International Publishing, 2019. 43–54.

Sarkar J., Bhattacharyya Souvik, Gopal M.Ram. Optimization of a transcritical CO2 heat pump cycle for simultaneous cooling and heating applications // International Journal of Refrigeration. 2004. 27, 8. 830 – 838.

Schmidt Drew, Singleton Jake, Bradshaw Craig R. Development of a light-commercial compressor load stand to measure compressor performance using low-GWP refrigerants // International Journal of Refrigeration. 2019. 100. 443 – 453.

Schrittwieser Julian, Antonoglou Ioannis, Hubert Thomas, Simonyan Karen, Sifre Laurent, Schmitt Simon, Guez Arthur, Lockhart Edward, Hassabis Demis, Graepel Thore, Lillicrap Timothy, Silver David. Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model. 2019.

Silver David, Hubert Thomas, Schrittwieser Julian, Antonoglou Ioannis, Lai Matthew, Guez Arthur, Lanctot Marc, Sifre Laurent, Kumaran Dharshan, Graepel Thore, Lillicrap Timothy, Simonyan Karen, Hassabis Demis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play // Science. 2018. 362, 6419. 1140–1144.

Širokỳ Jan, Oldewurtel Frauke, Cigler Jiří, Prívara Samuel. Experimental analysis of model predictive control for an energy efficient building heating system // Applied Energy. 2011. 88, 9. 3079 – 3087.

Sun Haoran, Ding Guoliang, Hu Haitao, Ren Tao, Xia Guanghui, Wu Guoming. A general simulation model for variable refrigerant flow multi-split air conditioning system based on graph theory // International Journal of Refrigeration. 2017. 82. 22 – 35.

T R., Jasmin E. A., Ahamed T. P. I. Residential Load Scheduling With Renewable Generation in the Smart Grid: A Reinforcement Learning Approach // IEEE Systems Journal. 2018. 1–12.

Venayagamoorthy G. K., Sharma R. K., Gautam P. K., Ahmadi A. Dynamic Energy Management System for a Smart Microgrid // IEEE Transactions on Neural Networks and Learning Systems. 2016. 27, 8. 1643–1656.

Wen Z., O'Neill D., Maei H. Optimal Demand Response Using Device-Based Reinforcement Learning // IEEE Transactions on Smart Grid. 2015. 6, 5. 2312–2324.

Wu Nan, Wang Honglei. Deep learning adaptive dynamic programming for real time energy management and control strategy of micro-grid // Journal of Cleaner Production. 2018. 204. 1169 – 1177.

Zhao Z., Lee W. C., Shin Y., Song K.-B. An Optimal Power Scheduling Method for Demand Response in Home Energy Management System // IEEE Transactions on Smart Grid. 2013. 4, 3. 1391–1400.

Zhu Xiaochen, Wang Fuli, Niu Dapeng, Guo Yuming, Jia Mingxing. An energy-saving bottleneck diagnosis method for industrial system applied to circulating cooling water system // Journal of Cleaner Production. 2019. 232. 224 – 234.

Zhu Yonghua, Jin Xinqiao, Du Zhimin, Fan Bo, Fu Sijie. Generic simulation model of multi-evaporator variable refrigerant flow air conditioning system for control analysis // International Journal of Refrigeration. 2013. 36, 6. 1602 – 1615.

Zou Shenghua, Xie Xiaokai. Simplified model for coefficient of performance calculation of surface water source heat pump // Applied Thermal Engineering. 2017. 112. 201 – 207.

Appendix C

Paper C

Title:	A Deep Reinforcement Learning scheme for Battery Energy Management
Authors:	Sven Myrdahl Opalic, Morten Goodwin, Lei Jiao, Henrik Kofoed Nielsen, and Mohan Lal Kolhe
Affiliation:	University of Agder, Faculty of Engineering and Science, 4879, Grimstad, Norway
Conference:	*5th International Conference on Smart and Sustainable Technologies (SpliTech)*, Sept., 2020.
DOI:	10.23919/SpliTech49282.2020.9243797.

A Deep Reinforcement Learning scheme for Battery Energy Management

Sven Myrdahl Opalic, Morten Goodwin, Lei Jiao, Henrik Kofoed Nielsen, and Mohan Lal Kolhe
Department of Engineering Sciences
Faculty of Engineering and Science, University of Agder
4879, Grimstad, Norway
E-mails: {sven.opalic, morten.goodwin, lei.jiao, henrik.kofoed.nielsen, mohan.l.kolhe}@uia.no

Abstract — **Deep reinforcement learning is considered promising for many energy cost optimization tasks in smart buildings. However, agent learning, in this context, is sometimes unstable and unpredictable, especially when the environments are complex. In this paper, we examine deep Reinforcement Learning (RL) algorithms developed for game play applied to a battery control task with an energy cost optimization objective. We explore how agent behavior and hyperparameters can be analyzed in a simplified environment with the goal of modifying the algorithms for increased stability. Our modified Deep Deterministic Policy Gradient (DDPG) agent is able to perform consistently close to the optimum over multiple training sessions with a maximum cost reduction of 25 % and an average cost reduction of 99 % of the maximum in the simplified environment. DDPG is an actor-critic RL algorithm consisting of four neural networks - the actor and critic, main and target, networks. When environment complexity is increased, the DDPG agent performance decreases and a modified Twin Delayed DDPG (TD3) agent is utilized to achieve an average of 99.9 % of the optimal result. The TD3 algorithm uses two main critic networks to avoid known value overestimation bias.**

C.1 Introduction

Smart buildings, featuring local energy production, and energy storage, are expected to play an increasingly important role in the strife against global warming through efficient and reduced energy consumption IEA (2019); Manic et al. (2016). Building energy systems and energy price tariffs are, therefore, rapidly evolving, introducing new complexity to building energy use optimization.

Within Artificial Intelligence (AI) research, Reinforcement Learning (RL) is the process of learning through taking actions in an environment and receiving a numerical reward signal Sutton, Barto (2018). This approach of trial-and-error can be compared to the way a human toddler learns by interacting with the world. However, many iterations are usually required before the actions start to seem coherent to a more knowledgeable observer. RL is applied to many energy related problems. In Kuznetsova et al. (2013), RL is applied to a simulated microgrid featuring a wind turbine, battery storage, the main grid and an energy consumer. Electricity price, battery charge and predictions of wind power output and consumer load are input to a Q-learning RL agent that is tasked to choose actions for the battery. The action space is limited to three actions: charge, discharge, and standby. Charge and discharge power is a fixed value. Demand-response using Q-learning is proposed in Wen et al. (2015) to reschedule user initiated operation of devices in residential and small office buildings.

The developments in RL in recent years, with the introduction of Deep Learning (DL) techniques Lillicrap et al. (2015); Silver et al. (2017, 2018), show the potential for RL to play a major role in real-world energy optimization. Mocanu et al. (2019) explores deep RL for on-line dynamic binary consumer load scheduling in households. Residential battery control with deep RL is explored in Wan et al. (2018). In Wei et al. (2015), the authors propose dual iterative Q-learning neural networks to reduce energy cost with optimal battery control. However, none of the algorithms have been verified in an operational setting.

In this paper, we compare the performance of modified versions of well-known deep RL algorithms applied to a simplified battery control cost optimization task, mainly operating in continuous action space. The aim is to analyze algorithm learning and behaviour as grounds for further modification of the most promising algorithms for real-world application from a practical standpoint.

C.2 Background

Deep learning is a field within AI and machine learning that focuses on extracting patterns from data through a hierarchy of increasingly complex abstractions Goodfellow et al. (2016). The most common implementation of deep learning is practiced by passing data through the hidden layers of an Artificial Neural Network (ANN) and propagating the measured error of the output backwards through the same layers. At each node in the ANN, its numerical activation value consists of an activation function applied to the sum of the weighted input values. Backpropagation facilitates learning by updating the input weights according to the gradient of the error. In recent years, several significant breakthroughs have been made in applying a combination of deep learning and RL to various games. Simulated environments that allow numerous swift learning iterations with clearly defined numerical reward signals have proven to be fertile ground for the exploration of these algorithms. Mnih et al. (2013) introduced a deep ANN adaptation of basic table based Q-Learning called deep Q-learning (also known as Deep Q-Networks, DQN), demonstrating state-of-the-art results in six out of seven Atari 2600 games. The objective was to explore the effects of advances in computing power and deep learning on the common RL benchmarking task of Atari 2600 game play performance. Q-learning is essentially a table-based approach, mapping an environment state to Q-values for each possible action. The Q-table is updated according to

$$\begin{aligned} Q(S_t, A_t) \leftarrow & Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - \\ & Q(S_t, A_t)], \end{aligned} \tag{C.1}$$

where S is the state and A is the action selected. The learning rate α is applied to the sum of the reward R at time t+1 and the discounted (γ) estimated max future Q-value at state S_{t+1}, minus the existing Q-value. In Mnih et al. (2015), introducing DQN, the Q-table is instead encoded into the weight parameters of a deep ANN, specifically a deep Convolutional neural network (CNN), and the weights are updated according to

$$\begin{aligned} \theta_{t+1} = & \theta_t + \alpha[R_{t+1} + \gamma \max_a \hat{Q}(S_{t+1}, a, \theta_t) - \\ & \hat{Q}(S_t, A_t, \theta_t)] \nabla \hat{Q}_\theta(S_t, A_t, \theta_t), \end{aligned} \tag{C.2}$$

where θ_{t+1} are the weights at time t+1, θ_t are the weights at time t and $\nabla \hat{Q}_\theta(S_t, A_t, \theta_t)$ are the partial derivatives of the state-action pair value approximations with respect to the weight vector θ_t. Instead of updating the weights after every action according to the current sequence of actions, the algorithm draws random mini-batch samples from an experience replay memory to update the DQN weights using stochastic gradient descent.

The extremely complex and highly intuitive turn-based game of Go was mastered by AlphaGo through a combination of supervised learning (pre-training with human generated example data) and deep RL Silver et al. (2016), resulting in a 4-1 defeat of 18 time World Champion Lee Sedol. The achievements of AlphaGo has since been surpassed by AlphaGo Zero through *tabula rasa* deep RL without any human knowledge in Silver et al. (2017), where AlphaGo Zero defeated AlphaGo 100-0. Central to these algorithms is the concept of self-play to generate an experience replay database from which random samples are utilized for training. This was further explored in Silver et al. (2018) for the games of Shogi and Chess, leading to similarly impressive results. The AlphaZero algorithm uses neural networks to estimate action probabilities and a monte-carlo tree search algorithm for future move-sequence analysis. The most recent development is the MuZero algorithm introduced in Schrittwieser et al. (2019). Where AlphaZero is informed of the environment dynamics, ie. the rules of the game, MuZero differs by having to learn a model of the environment starting from scratch. This constitutes a significant step toward real-world application of deep RL with stochastic and partially unknown environment dynamics.

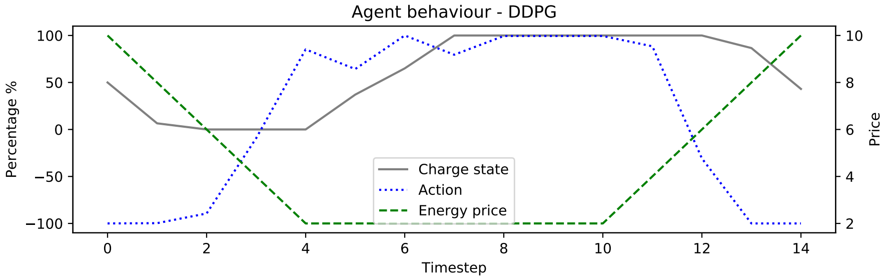


Figure C.1: DDPG agent typical behaviour, ϵ-greedy exploration with $\epsilon_d = 0.99995$.

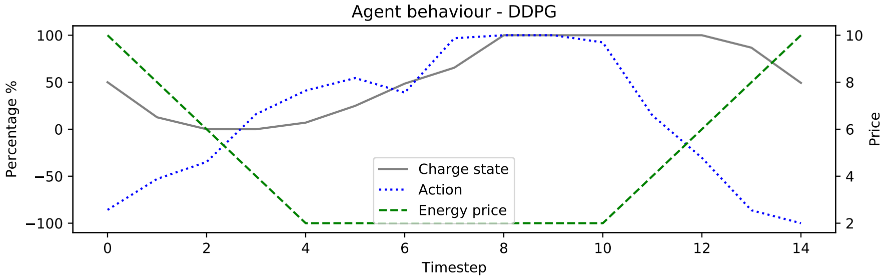


Figure C.2: DDPG agent typical behaviour, Ornstein-Uhlenbeck exploration.

Some efforts to apply state-of-the-art, at the time, deep RL to energy optimization have also been made. Deep RL for on-line dynamic binary consumer load scheduling in households is described in Mocanu et al. (2019). Availability of locally produced solar electricity, energy price and peak shaving are all considered. Data is extracted from the PecanStreet database and used to model households on individual and aggregated levels. The proposed algorithm, Deep Policy Gradient (DPG), replaces the output Q-values in a DQN with an estimated probability of taking action a in state s_t, thus allowing for multiple simultaneous discrete actions to be selected. DPG is found to outperform a DQN modified for simultaneous action selection through action grouping. Wei et al. (2015) proposes dual iterative Q-learning neural networks to reduce energy cost with optimal battery control. The dual iteration relates to an internal iteration j to reduce energy cost for each episode of 24 hours, and an external iteration $i \to \infty$ to update a defined performance index function towards its optimum. The overall claim is that the dual iteration is necessary due to the time dependant nature of the optimal Q-function, $Q^*(S_t, A_t, t)$. Neural networks are used in an actor-critic setup, denoted action and critic networks by the authors. Numerical results show improved performance over particle swarm optimization and time-based ANN Q-learning. Residential battery control with deep RL is explored in Wan et al. (2018). The algorithm can be characterized as Deep Deterministic Policy Gradient (DDPG), first proposed in Lillicrap et al. (2015), and consists of actor-critic deep neural networks, specifically recurrent neural networks using gated recurrent units Cho et al. (2014). The actor network utilizes policy gradient for parameter updates while the critic network utilizes a squared Q-value loss function. Results are compared to the theoretically lowest energy cost calculated by an optimization algorithm and a do-nothing scenario with a clearly favorable, but not optimal outcome.

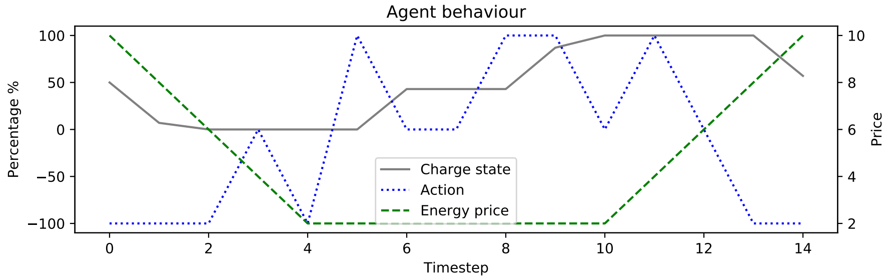


Figure C.3: DQN agent typical behaviour.

C.3 Proposed Deep Reinforcement Learning Approach

To observe what our chosen RL agents are able to learn, we first create a simulated environment consisting of a simple ideal battery, without any losses related to power conversion or storage. Actionable time steps in each episode is set to either 15 or 50 to explore both a basic and a more advanced experiment setting. Allowing the agent to see energy price 6 time steps ahead is also examined in the advanced experiment. The battery storage capacity and inverter power output is set to 460 kWh and 200 kW respectively, in accordance with the battery from our case-study warehouse in Opalic et al. (2019, 2020). Consequently, our agent is allowed to charge or discharge the battery by $B^{kW} \in [-200.0, 200.0]$. We initialize the environment by inputting vectors for hourly energy price P and demand D. Demand load is set to a constant value of 300 kW for all time steps. A baseline energy cost is then calculated as

$$C^{base} = \sum_{t=0}^{T} P_t D_t, \tag{C.3}$$

where T is the terminal time step of each episode. For every non-terminal time step the agent is awarded a numerical reward of 0 by the environment. A reward system where the agent receives the energy cost as a reward signal after each action was also examined. The cost incurred at each time step, where λ is the adjustable time step length in minutes, is calculated by

$$C_t^{agent} = P_t \left(D_t + \frac{B_t^{kW} \lambda}{60} \right), \tag{C.4}$$

and accumulated in

$$C^{agent} = \sum_{t=0}^{T} C_t. \tag{C.5}$$

Finally, the normalized reward for the agent at timestep T is given as

$$r_T = \frac{C^{base} - C^{agent}}{C^{base}}. \tag{C.6}$$

In all the experiments mentioned in this paper, λ was set to 60 minutes. For continuous action space we utilize modified versions of the Deep Deterministic Policy Gradient (DDPG) algorithm, first proposed in Lillicrap et al. (2015) and adopted by Wan et al. (2018). DDPG is an actor-critic RL algorithm with four ANNs – the actor policy network μ,

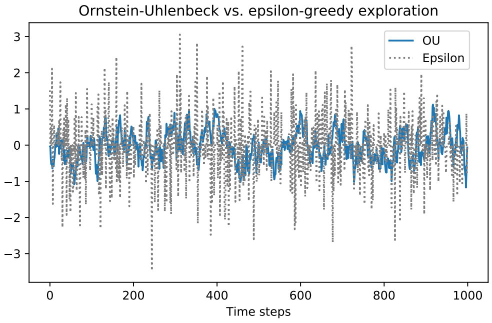


Figure C.4: Random Ornstein-Uhlenbeck and ϵ-greedy plot, 1,000 time steps.

the critic Q network and their respective target networks. Recent improvements suggested in Fujimoto et al. (2018), Twin Delayed DDPG (TD3), include the adoption of clipped dual Q-networks to avoid Q-value overestimation by only considering the most conservative output, delayed updates of the actor networks compared to the critic networks and adding noise to the target network predictions during training. Another development of DDPG is the Soft Actor-Critic (SAC) proposed in Haarnoja et al. (2018), introducing entropy regularization for exploration combined with clipped dual Q-networks. Actor network output layers are configured with a hyperbolic tangent (tanh) activation function, whereas critic network outputs are linear. We use fully connected ANNs for all networks. The target networks weights, θ', trail the main networks weight parameter updates, θ, through

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta', \tag{C.7}$$

with τ set to 0.1, as a mechanism for improving stability. Training the main networks is done through the use of an experience replay database R that holds transitions (s_i, a_i, r_i, s_{i+1}) for each step in every training episode. The algorithm samples a random mini-batch N of non-sequential transitions from R and uses the target actor $\mu'(s|\theta^{\mu'})$ to predict actions a'_{i+1} for every new state s_{i+1} in the mini-batch. A temporary state-action value is then calculated using the target critic as

$$y_i = r_i + \gamma Q'(s_{i+1}, a'_{i+1}) \tag{C.8}$$

and the main critic network updated by minimizing mean squared error between y_i and $Q(s_i, a_i)$ for every transition in the mini-batch. Finally, the main actor network can be updated from the same mini-batch by first calculating new actions a_i from current states s_i with the main actor μ. The gradients for the main Q network weights θ^Q with respect to a_i, and the gradients for the main policy network μ with respect to its parameters θ^μ are then used to approximate the gradient of the policy network cost function J with respect to θ^μ, by sampling as shown in Silver et al. (2014):

$$\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_i \nabla_{a_i} Q(s_i, a_i|\theta^Q)\nabla_{\theta^\mu}\mu(s_i|\theta^\mu). \tag{C.9}$$

The approximated gradients are then applied to θ^μ using an Adam Kingma, Ba (2014) optimizer with a learning rate $\alpha = 0.0001$. We implement an ϵ-greedy strategy for exploration, for comparison with the Ornstein-Uhlenbeck (OU) process Uhlenbeck, Ornstein

(1930) used in Lillicrap et al. (2015); Wan et al. (2018). Fig. C.4 is a comparison of ϵ-greedy and OU for a random 1,000 time step plot. ϵ is initially set to 0.9 (90 % chance of exploration) and then degraded by a factor of 0.99995 (ϵ_d) for every step taken in the environment. If exploration is triggered, a random number is pulled from a mean of 0.0 and a standard deviation of 1 and added to the agent output. The action is multiplied by the action bound of $B^{kW}_m ax = 200$ and clipped to be within the battery inverter range of $[-200.0, 200.0]$. This enables significant exploration, as the agent will still be doing more than 50 % exploratory moves towards the end of each training session. Training sessions with an epsilon decay factor of 0.9995 and 0.9999 were also conducted using identical random seeding settings. Within each training session, we verify each seeming improvement with a test run of the deterministic version of the agent (without any exploration). Exploration with the TD3 algorithm for the advanced environment was conducted with a combination of a clipped epsilon exploration, random action noise of +/- 10 %, and completely random actions. For reference, we also train a Deep Q-Network using a discretized action space of 0,1,2 to either standby, charge or discharge at full capacity.

The algorithms are trained for 10,000 episodes for every epoch with a fixed random seed. For our 15 time step environment, all ANNs have the same hidden layer architecture, with three hidden layers of 64 neurons each using DQN and DDPG.

For the 50 time step experiment, a grid search was conducted to find the optimal network architecture. The grid search resulted in a fourth hidden layer added to the TD3 agent and double hidden layers of 128 neurons for the DDPG agent.

The optimal solutions were found using the GNU Linear Programming Kit (GLPK) through the Pyomo Hart et al. (2017, 2011) programming library in Python.

C.4 Numerical Results

The basic environment consists of 15 time steps, where the state of the environment provided to the agents is limited to the battery charge state and current energy price, and the reward is provided on the conclusion of each 15 time step episode. The advanced environment consists of 50 time steps. In addition to charge state and energy price, the environment can also provide additional information about future energy price (in accordance with common practice in the Norwegian electrical energy market). Utilizing energy cost as a reward signal on every time step yields reduced agent performance and results in slowed learning due to an observed natural preference towards battery discharging.

C.4.1 Basic environment results

Results for the 15 time step experiments are summarized in Table C.1, where the optimal result was found to be a cost reduction of 5,280. Fig. C.1 and C.2 show that the modified DDPG agents are able to learn to charge when the price is low, and discharge when the price is high, achieving a maximum of 25 % cost reduction, which is very close to the optimal result found with GLPK. They are also able to learn to discharge moderately at an intermediary price point so as to make sure that the battery is fully discharged at the last

Table C.1: Basic environment - 15 time step episodes

Algorithm	***Exploration***	***Max. Result******	***Avg. Result***
DDPG-ϵ	$\epsilon = 0.9$, $\epsilon_d = 0.99995$	5,275	5,246
DDPG-OU	$\theta = 0.15$, $\sigma = 0.2$	5,275	4,830
DQN	$\epsilon = 1$, $\epsilon_d = 0.99975$	4,920	4,920

*Cost reduction calculated with optimization algorithm: 5,280.

Table C.2: Results for 50 time step episodes

Algorithm	***Exploration***	***Max. Result******	***Avg. Result***
TD3	$noise = 0.1$	13,999.99	13,998.83
DDPG-ϵ	$\epsilon = 0.9$, $\epsilon_d = 0.99995$	12,997.18	6,979.93
DQN	$\epsilon = 1$, $\epsilon_d = 0.99975$	10,020	9,553.33

*Cost reduction calculated with optimization algorithm: 14,000.

and the most profitable time step. The critical moment for maximum cost reduction is time step 12, as shown in Fig. C.1. Noticeably, the ϵ-greedy agent behaviour tends towards the limits of the action space for the non-critical time steps. The OU agent is more moderate, as expected when analyzing the pattern in C.4. It is likely that changing the parameters of the OU action noise would erase most of the differences between the DDPG agents, although the OU agent would then continue exploration with a similar frequency and range indefinitely. A combination of the two approaches might make an interesting compromise.

The DQN agent is also able to learn to charge when the price is low, and discharge when the price is high (Fig. C.3). Although the discretization of the action space for the DQN agent naturally limits the possible cost reduction to 4,920, it was able to achieve 70 % of the training sessions when trained only once per episode. If the DQN agent was allowed to train after every time step, it converged to maximum performance every time. Naturally, the DQN agent discrete action space could have been increased by introducing actions that give the agent more room to maneuver when charging or discharing.

For the agents operating in continuous action space, it is noted that the algorithms'

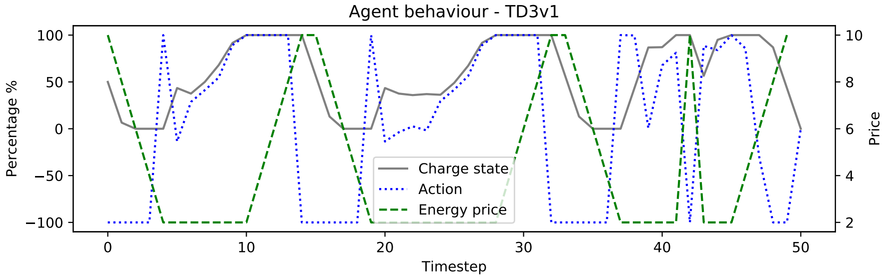


Figure C.5: TD3 agent behaviour in the advanced environment.

performances are volatile, and they are not able to converge towards the optimal behaviour on every training session without hyper-parameter tuning. The DDPG agents require an adjustment to the frequency of network updates. Foregoing network updates after each time step (instead of waiting until the conclusion of each episode) seems to improve average performance over multiple training sessions as well as speeding up the algorithm considerably. For the DDPG agent, delayed training leads to an average cost reduction of 5,246 over multiple training session, with little deviation, using an epsilon decay factor of 0.99995. The maximum cost reduction of 5,275 is also achieved with this setting. Increasing the exploration degradation factor to 0.9995 immediately reduces the performance of the agent to an extent that further training beyond 1,500 episodes appear meaningless. The agent is only occasionally able to approach optimal performance, with numerous training sessions yielding results in the 3,000 range.

C.4.2 Advanced environment results

Results for the 50 time step experiments are summarized in Table C.2. These include the following environment modifications that allowed the TD3 agent to achieve optimal results:

- Reward received at every time step.
- Expanded state - where the agent also receives the energy price of the next 5 time steps.

The rationale behind the expanded state for future energy price is that future energy price is often known as far ahead as 24 hours. This is true for energy tariffs in Norway, where day-ahead spot prices are given at the start of each day. Including future energy price as part of the current state is therefore quite reasonable. However, the environment modifications had an inverse effect on the DDPG and DQN agents, causing a decline in performance. Results for DDPG and DQN in the advanced experiment in Table C.2 are therefore excluding environment modifications. Fig. C.5 displays the behaviour of the choicest deterministic version of the TD3 agent. The maximum achievable cost reduction in this environment was found to be 14,000. Achieving stable results across multiple seeded training sessions required an extensive hyperparameter grid search and tuning of the algorithm.

C.5 Conclusions and Future Work

In this paper, we designed a deep Q-learning based algorithm for optimal scheduling of charging and discharging of battery based on the DDPG and TD3 algorithms. Our experiments reveal that the algorithms are quite sensitive to changes in hyperparameter and exploration settings and need to be configured appropriately to deliver consistent performance. This could pose a serious challenge in complex environments where ideal agent behaviour is less transparent. However, our appropriately configured DDPG agent was able to reduce the energy cost by 25 % while maintaining an average of 99 % of maximum over multiple training sessions in the basic environment. In the advanced experiment setting, our TD3 agent was able to achieve optimal results when future energy price for 5 time steps was included as state input variables. Stabilizing results over multiple seeded training sessions required hyperparameter tuning and an additional hidden layer, leading to a virtually optimal average performance.

Algorithm performance stability in multiple training sessions should be further explored in future work, to reduce the need for hyperparameter tuning. The environment complexity needs to be enhanced to approach real world operational settings, introducing constraints for battery health preservation as well as more realistic battery charge cycles. Our environment should also be expanded to include a realistic dynamic load and locally produced solar power. Real energy pricing schemes need to be introduced, including peak power tariffs and price differentiation between import and export energy.

More advanced state-of-the-art deep RL algorithms previously untested for battery control tasks should be explored. If simulated performance is adequately promising, after further development and hyperparameter tuning, our chosen algorithm will be tested in an operational setting in the smart warehouse described in Opalic et al. (2019, 2020).

Bibliography

Cho Kyunghyun, Merrienboer Bart van, Gulcehre Caglar, Bahdanau Dzmitry, Bougares Fethi, Schwenk Holger, Bengio Yoshua. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. 2014.

Fujimoto Scott, Hoof Herke van, Meger David. Addressing Function Approximation Error in Actor-Critic Methods // CoRR. 2018. abs/1802.09477.

Goodfellow Ian, Bengio Yoshua, Courville Aaron. Deep Learning. 2016. http://www.deeplearningbook.org.

Haarnoja Tuomas, Zhou Aurick, Abbeel Pieter, Levine Sergey. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor // CoRR. 2018. abs/1801.01290.

Hart William E., Laird Carl D., Watson Jean-Paul, Woodruff David L., Hackebeil Gabriel A., Nicholson Bethany L., Siirola John D. Pyomo–optimization modeling in python. 67. 2017. Second.

Hart William E, Watson Jean-Paul, Woodruff David L. Pyomo: modeling and solving mathematical programs in Python // Mathematical Programming Computation. 2011. 3, 3. 219–260.

IEA . Renewables 2019. 2019.

Kingma Diederik P., Ba Jimmy. Adam: A Method for Stochastic Optimization // CoRR. 2014. abs/1412.6980.

Kuznetsova Elizaveta, Li Yan-Fu, Ruiz Carlos, Zio Enrico, Ault Graham, Bell Keith. Reinforcement learning for microgrid energy management // Energy. 2013. 59. 133 – 146.

Lillicrap Timothy P., Hunt Jonathan J., Pritzel Alexander, Heess Nicolas, Erez Tom, Tassa Yuval, Silver David, Wierstra Daan. Continuous control with deep reinforcement learning. 2015.

Manic Milos, Amarasinghe Kasun, Rodriguez-Andina Juan J., Rieger Craig. Intelligent Buildings of the Future: Cyberaware, Deep Learning Powered, and Human Interacting // IEEE Industrial Electronics Magazine. 2016. 10, 4. 32–49.

Mnih Volodymyr, Kavukcuoglu Koray, Silver David, Graves Alex, Antonoglou Ioannis, Wierstra Daan, Riedmiller Martin A. Playing Atari with Deep Reinforcement Learning // CoRR. 2013. abs/1312.5602.

Mnih Volodymyr, Kavukcuoglu Koray, Silver David, Rusu Andrei A., Veness Joel, Bellemare Marc G., Graves Alex, Riedmiller Martin, Fidjeland Andreas K., Ostrovski Georg, Petersen Stig, Beattie Charles, Sadik Amir, Antonoglou Ioannis, King Helen, Kumaran Dharshan, Wierstra Daan, Legg Shane, Hassabis Demis. Human-level control through deep reinforcement learning // Nature. II 2015. 518, 7540. 529–533.

Mocanu E., Mocanu D. C., Nguyen P. H., Liotta A., Webber M. E., Gibescu M., Slootweg J. G. On-Line Building Energy Optimization Using Deep Reinforcement Learning // IEEE Transactions on Smart Grid. July 2019. 10, 4. 3698–3708.

Opalic Sven Myrdahl, Goodwin Morten, Jiao Lei, Nielsen Henrik Kofoed, Kolhe Mohan Lal. Modelling of Compressors in an Industrial CO2-Based Operational Cooling System Using ANN for Energy Management Purposes // Engineering Applications of Neural Networks. Cham: Springer International Publishing, 2019. 43–54.

Opalic Sven Myrdahl, Goodwin Morten, Jiao Lei, Nielsen Henrik Kofoed, Pardiñas Ángel Álvarez, Hafner Armin, Kolhe Mohan Lal. ANN modelling of CO2 refrigerant cooling system COP in a smart warehouse // Journal of Cleaner Production. 2020. 120887.

Schrittwieser Julian, Antonoglou Ioannis, Hubert Thomas, Simonyan Karen, Sifre Laurent, Schmitt Simon, Guez Arthur, Lockhart Edward, Hassabis Demis, Graepel Thore, Lillicrap Timothy, Silver David. Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model. 2019.

Silver David, Huang Aja, Maddison Chris J., Guez Arthur, Sifre Laurent, Driessche George van den, Schrittwieser Julian, Antonoglou Ioannis, Panneershelvam Veda, Lanctot Marc, Dieleman Sander, Grewe Dominik, Nham John, Kalchbrenner Nal, Sutskever Ilya, Lillicrap Timothy, Leach Madeleine, Kavukcuoglu Koray, Graepel Thore, Hassabis Demis. Mastering the Game of Go with Deep Neural Networks and Tree Search // Nature. I 2016. 529, 7587. 484–489.

Silver David, Hubert Thomas, Schrittwieser Julian, Antonoglou Ioannis, Lai Matthew, Guez Arthur, Lanctot Marc, Sifre Laurent, Kumaran Dharshan, Graepel Thore, others . A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play // Science. 2018. 362, 6419. 1140–1144.

Deterministic policy gradient algorithms. // . 2014.

Silver David, Schrittwieser Julian, Simonyan Karen, Antonoglou Ioannis, Huang Aja, Guez Arthur, Hubert Thomas, Baker Lucas, Lai Matthew, Bolton Adrian, Chen Yutian, Lillicrap Timothy, Hui Fan, Sifre Laurent, Driessche George van den, Graepel Thore, Hassabis Demis. Mastering the game of Go without human knowledge // Nature. X 2017. 550. 354–359.

Sutton Richard S., Barto Andrew G. Reinforcement Learning. 2018.

Uhlenbeck George E, Ornstein Leonard S. On the theory of the Brownian motion // Physical review. 1930. 36, 5. 823.

Wan Z., Li H., He H. Residential Energy Management with Deep Reinforcement Learning // 2018 International Joint Conference on Neural Networks (IJCNN). July 2018. 1–7.

Wei Q., Liu D., Shi G. A novel dual iterative Q-learning method for optimal battery management in smart residential environments // IEEE Transactions on Industrial Electronics. April 2015. 62, 4. 2509–2518.

Wen Zheng, O'Neill Daniel, Maei Hamid. Optimal Demand Response Using Device-Based Reinforcement Learning // IEEE Transactions on Smart Grid. 2015. 6, 5. 2312–2324.

Appendix D

Paper D

Title:	Augmented Random Search with Artificial Neural Networks for energy cost optimization with battery control
Authors:	Sven Myrdahl Opalic, Morten Goodwin, Lei Jiao, Henrik Kofoed Nielsen, and Mohan Lal Kolhe
Affiliation:	University of Agder, Faculty of Engineering and Science, 4879, Grimstad, Norway
Journal:	*Journal of Cleaner Production, Elsevier*, Accepted Oct., 2022.
DOI:	10.1016/j.jclepro.2022.134676.

Augmented Random Search with Artificial Neural Networks for energy cost optimization with battery control

Sven Myrdahl Opalic, Morten Goodwin, Lei Jiao, Henrik Kofoed Nielsen,
and Mohan Lal Kolhe
Department of Engineering Sciences
Faculty of Engineering and Science, University of Agder
4879, Grimstad, Norway
E-mails: {sven.opalic, morten.goodwin, lei.jiao, henrik.kofoed.nielsen,
mohan.l.kolhe}@uia.no

Abstract — **Intermittent renewable energy production and dynamic load must be balanced through appropriate control of integrated energy storage to account for the temporal discrepancy among power supply and demand. Intelligent control systems are required to anticipate and optimize the charging and discharging of energy storage. In recent years, reinforcement learning based techniques have been applied to a multitude of problems, including building integrated energy storage solutions. In this work, the focus is on the application of reinforcement learning based techniques to the specific energy optimization problem of controlling a battery energy storage system in a smart warehouse. This paper adopts data from a real operational battery energy storage system installed in a smart warehouse, integrated with photovoltaic, for food distribution on the west coast of Norway. In the smart warehouse, an intelligent energy management system controls the on-site battery energy storage using machine learning predictions of load and photovoltaic production, and an optimization algorithm is presented to generate a schedule for effective utilization of battery energy storage in coordination with a thermal storage system. This paper presents the combination of the augmented random search reinforcement algorithm with artificial neural networks as a basis to design an intelligent energy management system for controlling energy flows of battery energy storage systems to minimize the energy cost. The developed algorithm finds very promising solutions in the considered case-study of a smart house for energy cost minimization through a battery energy storage system, achieving an average of 99.2% accuracy across 10 seeded trials.**

D.1 Introduction

The world faces an ongoing increase in energy demand and environmental problems associated with a majority of the existing energy sources, which has forced a shift towards renewable energy sources. Furthermore, with our growing reliance on intermittent energy sources such as solar and wind, we require a critical tool to balance the temporal discrepancy between instant power demand and available production capacity. As fossil-fueled power plants are gradually phased out, wind and solar will increasingly have to rely on energy storage technology with intelligent control systems for such purposes (Hannan et al., 2021; Yang et al., 2009). Integrating renewable energy into smart city solutions is a crucial aspect of sustainable development (Hoang et al., 2021). Buildings, responsible for nearly 40% of global CO_2 emissions (IEA, 2020), are a natural target for deploying distributed renewable energy production and storage systems with intelligent control. Battery Energy Storage Systems (BESS) built with lithium-ion technology are increasingly deployed in both macro and micro scale projects (Stroe et al., 2017). For optimal utilization of the BESS for multiple purposes such as energy cost reduction, reducing peak power demand and frequency regulation, intelligent control systems that balance the need for longer-term planning with immediate response are required. For such systems, many approaches have been suggested, including constrained non-linear programming (CNLP) optimization for aggregated two-stage control in a micro-grid in Long et al. (2018) achieving a 30 % energy cost reduction when combined with peer-to-peer energy sharing, a rule-based approach for many distributed batteries in a data center with a focus on accurate battery health modeling in Aksanli et al. (2013) and a rule-based scheme for Photovoltaic (PV) and wind application in Teleke et al. (2010). When considering the dynamic and ever-changing nature of building-integrated energy systems, it seems unlikely that a rule-based approach can be implemented without extensive follow-up and revision. In related research, Siqueira de, Peng (2021) conducted a review of control strategies for smoothing wind power output, finding Model Predictive Control MPC to be the most common for multi-objective optimization. Lipu et al. (2021) discuss various approaches to intelligent control for battery management in electric vehicles.

As shown in Perera, Kamalaruban (2021), many researchers turn to Reinforcement Learning (RL) as a potentially self-improving and robust approach to intelligent control of building energy systems.

The research gap can be described in two parts. Firstly, according to Perera, Kamalaruban (2021), most of the studies regarding RL application to energy systems are not attempting to implement the state-of-the-art RL algorithms, instead they rely on basic Q-learning. This could limit the application to well-defined and uncomplicated systems and solutions, or lead to sub-optimization through compartmentalization of complex problems into simpler tasks that disregard the intricacies of the energy system.

Secondly, many of the more complicated state-of-the-art algorithms are primarily developed to teach agents to solve benchmark gameplay tasks from the OpenAI Gym (Brockman et al., 2016), or prediction of load forecasting Johannesen et al. (2018). We spent a considerable effort on hyperparameter- and algorithm tuning of well-known RL algorithms in Opalic et al. (2020) to solve a simplified battery control problem. Although the numerical results were satisfying, we concluded that such an approach would not be sufficiently robust and scalable in commercial applications due to the amount of work that was required. Buildings, unlike most other areas of smart technology applications, are mostly unique and different in varying degrees from all other buildings. Each building has a specifically tailored energy system to suit the needs of the building occupants. Implementing such algorithms for energy cost optimization and tuning them for each building lead to questions of applicability, scalability and robustness.

Our approach features the Augmented Random Search (ARS) (Mania et al., 2018), adapted for policy parameterization with Artificial Neural Networks (ANN) instead of the suggested linear function employed in Mania et al. (2018). The method is benchmarked to the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm (Fujimoto et al., 2018), found to be the most promising and able to find an optimal solution to a simpler BESS control problem Opalic et al. (2020). We also compare results with the vanilla ARS. A new simulated environment is introduced, developed for training the agents on historical operational data from a case-study smart warehouse.

In summary, our contributions include:

- Introducing a robust RL algorithm that can handle complex energy optimization problems.

- Combining ARS with ANN for energy optimization of BESS.

- Creating a data-driven simulation environment of a smart warehouse for RL training.

The remainder of the paper is organized as follows: Section D.2 features an overview of related work relevant to our research. Section D.3 briefly introduces the energy system in our case-study smart warehouse. Section D.4 explains our method, and Section D.5 details the experiments we have conducted. Results are also presented and discussed in Section D.5 before we conclude in Section D.6. Abbreviations are included in the appendix.

D.2 Related work

In this paper, we focus on the application of RL to the specific energy optimization problem of controlling a BESS in a smart warehouse. We believe that RL algorithms have the potential to reduce the need for human expert attention, and in therefore the cost of initial investment and maintenance, in real-world implementation due to the self-exploratory nature of such algorithms. Risk associated with this behaviour can be vastly reduced by training RL agents in an off-line data-driven simulated environment. We, therefore, dedicate a section to optimization and RL applications in energy research. In more detail, the first subsection is dedicated to energy cost optimization in general featuring more traditional approaches. Thereafter, the second subsection summarizes RL-related work. The last subsection presents the specific RL algorithm we concentrate on in this paper, i.e., ARS.

D.2.1 Energy optimization in buildings

For optimizing energy cost and power flow in a Direct Current (DC) microgrid Sechilariu et al. (2014) proposed Mixed Integer Linear Programming (MILP). The approach is similar to the Intelligent Energy Management System (IEMS) already implemented in our previously described case-study smart warehouse. It includes load and PV prediction, a human-machine interface, and energy management. Unlike the case-study IEMS, it also features instant power balancing. Huang et al. (2015) proposed a hybrid (MPC) for energy cost optimization in a case-study airport terminal building. The authors introduce Neural Networks as a way to handle non-linearity. Another MPC approach was suggested in Lešić et al. (2017) using hierarchies of multiple MPCs for energy cost optimization and thermal comfort control. A data-driven MPC, i.e., Data Predictive Control (DPC), was proposed in Smarra et al. (2018). The authors suggested using random forests for predictions and argued that intelligent control systems that require physical models of buildings are not practical due to high complexity and variance in building design. Wang et al. (2020) propose MPC for control of a dual BESS connected to a wind power farm. Based on simulations, the authors claim improved wind farm dispatchability, and extended battery life as their results. Barbato, Capone (2014) conducted a survey to describe various optimization techniques designed to solve Demand Side Management (DSM) problems for end-users in smart grid scenarios. They conclude that although researchers had undergone extensive work in this field of research, many research questions remained unanswered. Mariano-Hernández et al. (2021) conducted a review of various strategies for Building Energy Management Systems (BEMS), including MPC, DSM, and optimization. The authors found MPC to be the most used management strategy in non-residential buildings and conclude that the building model will be critical to ensure intelligent control in future research. Rätz et al. (2019) describe a methodology for automated data-driven modeling of energy systems in buildings that could be applicable to MPC and RL.

D.2.2 Reinforcement learning

RL, according to Sutton, Barto (2018), is learning by discovering what actions to take to maximize a reward. Experiments with simulated environments are often designed for agents to learn, by trial and error, how to maximize a numerical reward signal, often binary in nature. It is common for researchers to design a reward function to reward desired behavior and, in some cases, to penalize unwanted behavior. The reward function may be updated if the desired behavior changes over time in an operational scenario, even if the overall goal is unchanged.

RL researchers commonly model the problem as a finite Markov Decision Process (MDP). We included an illustration of the process in Figure D.1. An agent interacting with an environment through taking actions receives feedback from the environment as reward or penalty. The agents' actions may affect the environments' internal state as a direct or partial consequence. The environment also influences which possible actions are available to the agent, and the action space is usually either a constant set of discrete numbers, a continuous range of floats, or decided with each new state. The agent determines what actions to take by following its internal policy π. The policy usually includes a mechanism

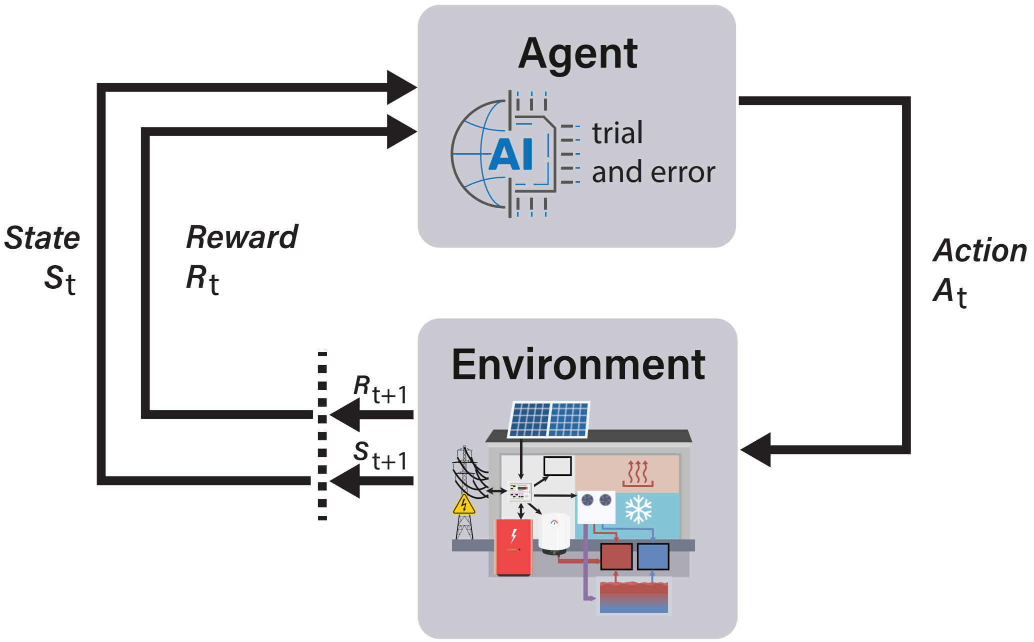


Figure D.1: Markov decision process interaction between agent and environment.

that allows the agent to explore alternative actions outside the most strict interpretation of its policy to be able to discover new states and actions that can potentially generate higher rewards. Upon such discoveries, various methods exist to update the policy accordingly. Finally, a crucial element in RL is the $valuefunction$ that defines the value of a state through probabilities related to actions, rewards, and future states. Sutton, Barto (2018) refer to the Bellman equation as the definition of the value of a state while following the policy π:

$$V_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s'r|s,a)\left[r + \gamma V_\pi(s')\right], \forall s \in S. \tag{A.1}$$

The Bellman equation describes the value of the state s while following the policy π as the sum of the probability of taking action a in the state s, multiplied by the sum of the probability of arriving in each state s' and receiving reward r, multiplied by the sum of r and the discounted (γ) expected value of the future state s'. The Bellman equation is a central part of RL theory and research.

A popular family of RL algorithms is Q-learning (Watkins, 1989) and its younger sibling Deep Q-Network (DQN (Mnih et al., 2013). The original Q-learning algorithm is a table-based mapping of states to the Q-values of all possible actions. The Q-value is a mathematical estimate of the expected discounted future value of the action. The state space and the action space have to be discrete and finite. The agents' policy is encoded in the Q-table, where each state has a corresponding Q-value for each possible action, and the deterministic version of the policy consists of choosing the action with the highest Q-value. The mechanism for exploring actions outside the policy in Q-learning consists of adding a random component to a fraction of the actions taken. DQNs replace the Q-table with an ANN such that the output of the neural network is the Q-values of all possible discrete actions in a given state. ANNs come in many forms, but the most common kinds feature an input layer, one or more hidden layers, and an output layer. Each layer consists of so-called neurons, named after the neurons in the human brain. The neurons in the input layer hold represent the chosen input parameters, passing these directly to the neurons in the first hidden layer. Usually, the hidden layers are fully connected, meaning each neuron in the hidden layers is connected to each neuron in the previous layer. The values are passed along the connections and summed before an activation function is applied to determine the output value from the neuron. Two of the most common activation functions are the hyperbolic Tangent (Tanh) and the Rectified Linear Unit (ReLU). Finally, the output layer neurons process the values according to the desired application, where the number of neurons corresponds to the chosen output values.

Researchers have applied RL to many problems and challenges within renewable energy, energy storage, and complex energy systems. Kuznetsova et al. (2013) simulated a microgrid consisting of a wind turbine and a BESS connected to a power grid. The authors utilized Q-learning with energy price, battery State Of Charge (SOC), wind energy predictions, and energy demand as inputs. The agent can choose between three discrete actions: Battery charging, discharging, or stand-by. Wen et al. (2015) also proposed Q-learning for controlling the temporal shift of flexible loads based on end-user device utilization in small offices and residential buildings. Mbuwir et al. (2017) suggested Fitted Q-iteration as the basis for transfer learning of battery control to and from BESS with similar characteristics. Henze, Schoenmann (2003) examined Q-learning for control of a Thermal Energy Storage (TES) in a simulated environment.

As stated in Perera, Kamalaruban (2021), most of the RL employed in the energy domain uses Q-learning and other simpler algorithms. However, some examples of more advanced state-of-the-art algorithms being tested also exist. Mocanu et al. (2019) used Deep Policy Gradient (DPG), similar to DQN, for binary scheduling of flexible residential consumer loads. Wan et al. (2018) proposed a variant of Deep Deterministic Policy Gradient (DDPG), from Lillicrap et al. (2015), for residential BESS control. DQN, with some proposed improvements, was suggested in Cao et al. (2020) for BESS arbitrage. The algorithm includes a lithium-ion battery degradation model, with discretized action space for full or 50 % capacity charging and discharging in addition to stand-by. Shang et al. (2020) proposed a DQN with bootstrapping combined with monte carlo tree search to control a BESS in a microgrid. In all cases except in Wan et al. (2018), the algorithms work in discrete domain, and therefore limited action space. In addition, in many cases the reward functions are quite sophisticated and tailored to the specific experiment. We hypothesize that most of the above mentioned approaches are not ideal if the goal is to enable large-scale adoption and quick implementation of IEMS.

D.2.3 Augmented random search

ARS is a more efficient version of what the authors (Mania et al., 2018) term basic random search due to the various mechanisms in the algorithm that targets the search towards higher rewards. The authors designed ARS to work with a simple linear policy, unlike the direction that many other RL researchers are taking, and it also operates in continuous action space. Additionally, different from many RL algorithms, exploration with the ARS is done directly in the parameters of the policy function by randomly making minute changes to the parameter weights. In other words, the algorithm directly manipulates the parameters of the linear policy function to search for a policy that generates higher rewards. In contrast, well-known algorithms for continuous action space such as DDPG (Lillicrap et al., 2015), Soft Actor-Critic (SAC) (Haarnoja et al., 2018), Trust-Region Policy Optimization (TRPO) (Schulman et al., 2015) and TD3 (Fujimoto et al., 2018) all add a random component to the agent output action to encourage exploration. For ARS, the parameter space is explored by generating a table of random noise and adding the noise to the policy parameters in both positive and negative directions. The new parameters are tested by running an episode and collecting the reward. N such tests, termed rollouts, are performed and sorted by reward in descending order (Mania et al., 2018). The top b directions are then chosen and used to update the policy according to

$$\theta_{j+1} = \theta_j + \frac{\alpha}{b\sigma_R} \sum_{k=1}^{b} \left[r\left(\pi_{j,(k),+}\right) - r\left(\pi_{j,(k),-}\right) \right] \delta_{(k)}, \tag{A.2}$$

where θ is the parameters of the policy, α is the learning rate, σ_R is the standard deviation of the rewards, $r(\pi_{j,(k),+})$ and $r(\pi_{j,(k),-})$ are the sorted rewards from positive and negative rollouts and $\delta_{(k)}$ is the randomly generated noise of the same size as θ. Mean and standard deviation of input variables are continuously updated and used to normalize input values. The authors demonstrate impressive performance across a wide range of known RL benchmark problems while also vastly decreasing computational resources required for training.

D.3 Energy system

This paper adopts data from an operational BESS installed in a smart warehouse for food distribution on the west coast of Norway. We described the smart warehouse in more detail in Opalic et al. (2020) and its main components are shown in Fig. D.2. We list technical details and component specifications in Tab. D.1. An IEMS currently controls the on-site BESS based on machine learning predictions of load and PV production, and an optimization algorithm that generates a 48-h schedule for the utilization of the BESS and a thermal storage system (Marton, others, 2019). The schedule is automatically implemented through the local Building Management System (BMS) and updated at hourly intervals. The existing system does not react to live operational data but follows the schedule precisely for the next hour. Every hour the system generates another 48-hour schedule and implements the first hour suggested actions. As such, the system relies very heavily on accurate predictions to be able to harness the energy storage options for maximum energy and cost reduction. Preventing excessive peak power load costs in this way is an extremely difficult exercise in practice. The magnitude of the challenge is only amplified when considering the monthly peak power tariff structure utilized by the local grid operator where the single highest monthly peak is the basis for the entire monthly peak power cost. Furthermore, consistently avoiding such peaks relying on predictions would require perfect predictions of each power surge in the demand load. To fully take

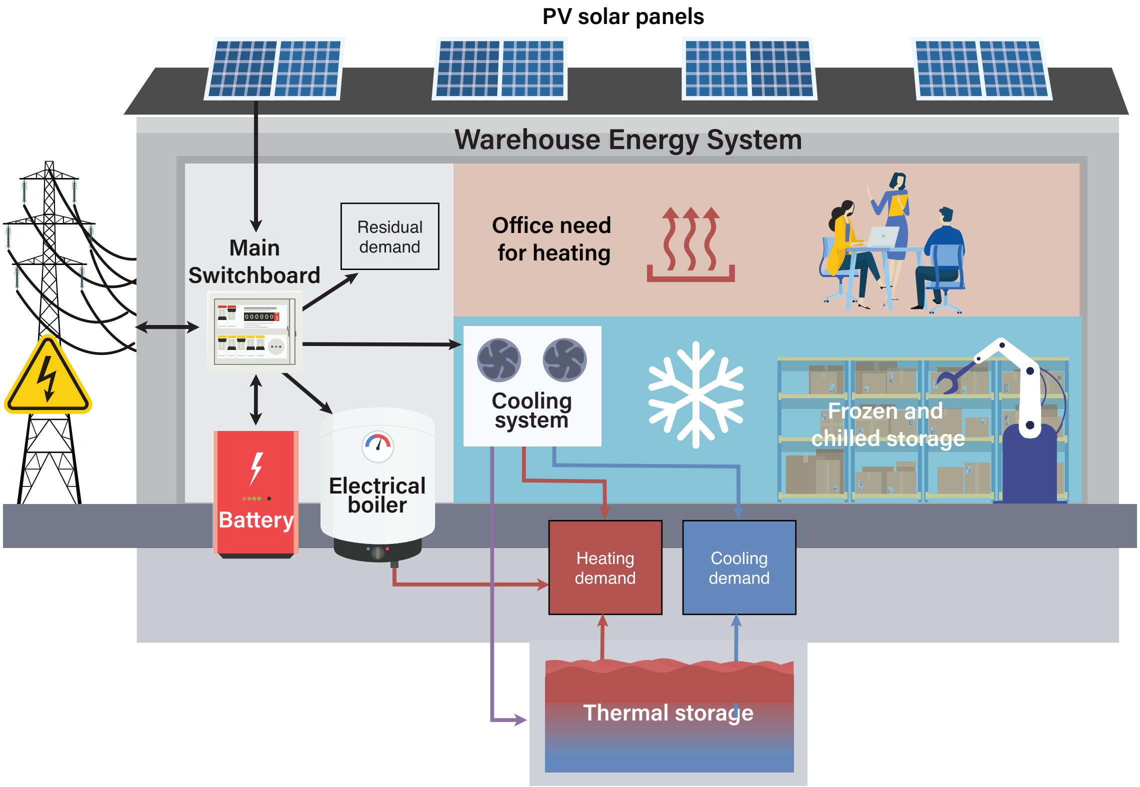


Figure D.2: The smart warehouse energy system.

advantage of the BESS' ability to reduce power peaks, the IEMS needs to be able to combine long-term planning with short-term reactions. This functionality can be achieved in multiple ways, including the introduction of a separate system for online reactive control. However, our ambition is to design a system that can react to important events while maintaining a longer time horizon in an integrated and robust manner. More specifically, our goals include:

1. Online control of battery systems, proactive and reactive. Both long-term planning and instant reactions are necessary.
2. Energy cost reduction.
3. Reducing peak power demand.
4. Scalability through ease of commercial implementation and minimizing necessary human intervention in operation.

D.4 Methodology

In this paper, we suggest an RL-based method for controlling a BESS for energy and peak power cost reduction in a smart warehouse. Our goal is to train an agent to learn intelligent control of a BESS in a simulated environment of the smart warehouse, and thus our research method follows a standard scientific engineering approach where we

Table D.1: Smart warehouse energy system components.

Component	Capacity	Unit of measurement
Photovoltaic solar panels	1,000	[kW$_p$]
Lithium-ion battery energy storage system	460/200	[kWh/kW]
Thermal energy storage system	300/300	[m^3/kW$_{thermal}$]
Cooling system	1,140	[kW$_{thermal}$]

continuously test and verify in simulation. We have designed a simulated environment consisting of a simple battery model and historical data from the smart warehouse. We emphasize that our suggested modelling approach is data-driven, which allows for lower demand on human resources in initial design when compared to purely physical models, as well as automatic adaptation to changes in building occupant behaviour and other operational parameters. These characteristics are crucial for successful adoption of IEMS in smart houses. We define our goal as a problem of energy cost optimization.

$$\begin{aligned}
\min_{\{\xi_t\}} \quad & \sum_{t=1}^{T} \left(E_t^+ \lambda_t^+ - E_t^- \lambda_t^- \right) + \sum_{m=1}^{M} P_m^+ \lambda_m^p && \text{(A.3)} \\
\text{s.t.} \quad & E_t^+ - E_t^- = L_t - S_t + \xi_t, \forall t \in \mathbb{N}, && \text{(A.4)} \\
& \min\{E_t^-, E_t^+\} = 0, \forall t \in \mathbb{N}, && \text{(A.5)} \\
& \Xi_t = \Xi_{t-1} + \xi_{t-1}\eta_\xi && \text{(A.6)} \\
& \Xi \in [32, 445], \xi \in [-200, 200], && \text{(A.7)} \\
& L_t, S_t \geq 0, \forall t \in \mathbb{N}, \\
& E_t^+ \in \mathbb{R}_{\geq 0}, \forall t \in \mathbb{N}, \\
& E_t^- \in \mathbb{R}_{\geq 0}, \forall t \in \mathbb{N}.
\end{aligned}$$

We frame our goal in the form of a cost reducing optimization problem in Eq. (A.3), where we sum over the energy cost for every time step t and sum the cost of the monthly peak power tariff for each month m. Eq. (A.4) dictates the energy balance of imported energy and exported energy related to demand load, solar power production and battery charging or discharging. The variables in Eqs. (A.3) and (A.4 are as follows:

- E_t^+ is energy imported from the grid at time step t,
- E_t^- is energy exported to the grid at t,
- P_m^+ is the peak power load in the month m,
- λ_t^+ is purchase price at t,
- λ_t^- is selling price at t,
- λ_m^p is the peak power tariff in the month m,
- L_t is the consumer load at t,

- S_t is solar PV power production at t,
- ξ_t is the inverter charge or discharge rate at t.

Eq. (A.5) states that energy can only be either imported or exported in a single time step. Eq. (A.6) is related to the BESS SOC, where Ξ_t is the battery charge state at t and η_ξ is the inverter power conversion efficiency. Eq. (A.7) defines the SOC range for the BESS Ξ, and the inverter range for charging and discharging ξ.

D.4.1 ARS with ANN

Our approach features a simple modification of the work on the ARS algorithm introduced in Mania et al. (2018). We adopt ANNs instead of the suggested linear function to parameterize the policy, see Algorithm 1. In our approach, we have changed the mapping of input to output from a linear to a nonlinear function using neural networks already implemented in the RLLIB programming library. The original ARS algorithm suggested the use of a simple linear policy, namely a matrix directly mapping input to output. The strength and simplicity of this algorithm are self-evident when examining the results presented in Mania et al. (2018). However, a linear policy is not always enough when dealing with highly complex environments such as building energy systems featuring local energy production and energy storage.

We utilize the RLLIB programming library Liang et al. (2018) as the main framework for training the agents. In addition, we've built a custom simulated training environment and a custom neural network model using Python 3.7, TensorFlow Abadi et al. (2015), and Pytorch Paszke et al. (2019). We've also utilized the RLLIB Liang et al. (2018) implementation of ARS with ANN. For reference and benchmarking, we calculate near-optimal solutions for the sampled episodes using Pyomo (Hart et al., 2011; Bynum et al., 2021) with a GNU Linear Programming Kit (GLPK) solver. The GLPK solver is given perfect information of the training scenario and attempts to find an optimal solution. We also compare our results to another well-known state-of-the-art RL algorithm, namely the TD3 algorithm utilized in Opalic et al. (2020).

D.4.2 State input

The state S_t, shown in Fig. D.1, is composed of historical operational data as well as parameters calculated by the simulated training environment. Operational data given as current temporal values include time, energy demand load, PV production, energy buy price, and energy sell price. The energy price is composed of all the factors listed in Section D.3, except peak power and reactive power. Although an important factor to total energy cost, peak power cost per kW is a constant and included in the reward signal and therefore it is deemed unnecessary to include as a constant value in the environment state. Input time includes the time of day, week, and year given as a two-component sine and cosine vector decomposition (for a total of 6 values). Future energy buy price is also given for 6 timesteps ahead. The future energy price is freely available 24 hours ahead in the Norwegian energy market, and thus price predictions are currently deemed unnecessary

Algorithm 1 Augmented Random Search with ANN

```
Set hyperparameters:
    • $\alpha$ - learning rate
    • $n$ - number of directions sampled per iteration
    • $v$ - standard deviation of the exploration noise
    • $b$ - number of top-performing directions to use
Run algorithm 2 to initialize policy parameters $\theta_j$, i.e. ANN weights
Initialize:
    • Mean - $\mu_0 = 0 \in \mathbb{R}^{inputs}$
    • Covariance - $\Sigma_0 = \mathbf{I}_n \in \mathbb{R}^{inputsxinputs}$
while ending condition not satisfied do
    Sample $\delta_1, \delta_2, ..., \delta_N$ of the same size as $\theta_j$, with i.i.d. standard normal entries.
    Normalize input values $x$ with $x_{normalized} = diag(\Sigma_j)^{-\frac{1}{2}}(x - \mu_j)$. Collect $2N$ rollouts of horizon $H$ and their corresponding rewards using the ANN policies $\pi_{j,k,+}$ and $\pi_{j,k,-}$, where the exploration noise $v\delta_k$ is added to the ANN weights $\theta_j$ for $\pi_{j,k,+}$ and subtracted from $\theta_j$ for $\pi_{j,k,-}$ with $k \in \{1, 2, ..., N\}$.
    Sort the directions $\delta_k$ by $\max\{r(\pi_{j,k,+}), r(\pi_{j,k,-})\}$, denote by $\delta_{(k)}$ the $k$-th largest direction, and by $\pi_{j,(k),+}$ and $\pi_{j,(k),-}$ the corresponding policies.
    Make the update step for the ANN weights:
    $\theta_{j+1} = \theta_j + \frac{\alpha}{b\sigma_R} \sum_{k=1}^{b} [r(\pi_{j,k,+}) - r(\pi_{j,k,-})]\delta_k$, where $\sigma_R$ is the standard deviation of the $2b$ rewards used in the update step.
    Set $\mu_{j+1}, \Sigma_{j+1}$ to be the mean and covariance of the $2NH(j+1)$ states encountered from the start of training.
    $j \leftarrow j + 1$.
end while
```

Algorithm 2 ANN for ARS in RLLIB

```
Set hyperparameters:
    • $\theta^{hl}$ - the number of hidden layers in ANN
    • $\theta^{nu}$ - the list of neurons in each hidden layer
    • $\theta^{af}$ - activation function for neurons
Initialize: $j = 0$, policy parameters $\theta_j$ of shape defined by $\theta^{hl}$ and $\theta^{nu}$ and random values $X$ pulled from a normal distribution $N(\mu_\theta, \sigma_\theta^2)$ of mean $\mu_\theta = 0$ and variance $\sigma_\theta^2 = 1$, multiplied by standard deviation $\sigma = 1.0$ for the hidden layers and $\sigma = 0.1$ for the output layer, divided by the square root of the random value $X^{hl,nu}$, $\theta_j^{hl,nu} = X^{hl,nu}\frac{\sigma}{\sqrt{X}}$
```

even though other approaches to intelligent BESS control feature energy price predictions. Battery SOC and peak power limit are calculated by the simulated training environment and included in the state.

D.4.3 Reward function

We have attempted to design a reward system following goal number 4, from Section D.3, focusing on scalability. We decided that one practical way to reduce the need for human maintenance of the agent in operation is to engineer a simple reward system that is closely coupled to the actual financial benefit. We suggest that this also potentially enables the use of the same reward function regardless of the system dynamics or degree of complexity, which in turn simplifies implementation and thereby increases scalability.

We, therefore, experiment with a reward system that calculates baseline energy cost C^b for each episode where no actions are taken and compares this to the actual cost C^a after the agent has selected an action,

$$R = C^b - C^a. \tag{A.8}$$

In addition, at the end of each day, the agent is penalized for the fraction that the battery SOC differs from the ideal, from a battery health perspective, SOC of 50% multiplied with the absolute value of the accumulated reward,

$$R^{penalty} = \frac{|\Xi_{50\%} - \Xi_{endofday}|}{\Xi_{max} - \Xi_{min}} \times |R^{accumulated}|. \tag{A.9}$$

We experiment with this mechanism to give the agent incentive to return to 50% charge state as often as possible to preserve battery health. This incentive is quite simple and we could potentially replace it with a much more sophisticated mechanic, but we include this simple version to examine how the agent responds to small changes in the reward signal in the face of the much larger potential for peak power cost reduction. The reward function therefore becomes

$$R_{t,h} = C^b_{t,h} - C^a_{t,h} - R^{penalty}_h. \tag{A.10}$$

where t is the current timestep, h is the hour of day and $R^{penalty}$ is non-zero only at $h = 0$.

D.4.4 Simulated training environment

We have built a simulated training environment that uses the simplified BESS model from (Opalic et al., 2020) combined with operational data from our case-study warehouse to create training scenarios where the agent is tasked to reduce the energy cost by controlling the BESS. We explain environment state values in Subsection D.4.2. A randomly seeded episode of consecutive timesteps is pulled from the data source period of March and April 2020. We have trained the agents on one or more samples for 100 million timesteps. The length of the training session was decided through an empirical investigation of a reasonable time frame to allow the models to arrive at a reasonable solution. Most of the ARS agents can converge on a stable solution within this time frame. The length of the episode can be freely chosen but is naturally limited by the size of the available data set. When initialized, the environment calculates a baseline energy cost for the selected data when no action is taken. The energy cost calculated by the training environment is structured according to the actual energy pricing scheme utilized by the grid operator in the case-study location, consisting of the following parts:

- spot price per kWh
- a fixed annual fee
- a fixed rate per kWh consumed (summer/winter)
- monthly peak power
- monthly peak reactive power above a certain threshold

The most significant contributor to the total energy cost is the monthly peak power tariff of 80 NOK/kW (Lnett, 2022) during the winter months. We have implemented an adjustable target value for peak power tariff inclusion in the simulated training environment such that the tariff only contributes to the total cost and reward signal when exceeded. Setting the target value to 0 will reflect how the tariff is calculated at the start of each month in reality, but will fail to take into account the fact that the tariff is monthly and in most cases will have incurred a certain cost level before the start of each new episode. For experiments with shorter temporal horizons than one month, it is more realistic to set the initial value to an achievable monthly goal. In our case, an examination of the data set revealed that around 500 kW maximal power consumption would be a realistic goal for the BESS.

Reactive power is not an issue for the local power grid and our case-study smart warehouse. We, therefore, disregard it in the simulated training environment.

D.5 Experiments and results

We explore RL algorithms applied to cost optimization of energy storage in BESS, based on operational data from a case-study warehouse. We initially tested all relevant RL algorithms included in the RLLIB library, but the vast majority showed little promise and were discarded from further testing. The ARS and ARS-ANN algorithms showed the most promising results. The RLLIB version of the TD3 algorithm was also included in our first experiment for benchmarking purposes.

We conducted multiple experiments to analyze and document the behavior of the agents in various random scenarios. A total of three experiment setups were defined after the initial testing, each consisting of multiple training sessions and various agents and algorithms. The RL agents were trained for 100 million timesteps each in a simulated environment in our three main experiments. Our first experiment was a randomly seeded 48-hour period, where we compare results with the well-known RL algorithm TD3 and the near-optimal solution found with the GLPK solver given perfect information. The second experiment consisted of 10 randomly seeded 48 hour periods and features comparison between ARS-ANN, original ARS, and the GLPK solver. For our third experiment, we expanded the episode to nearly include the entire dataset and compared our ARS-ANN

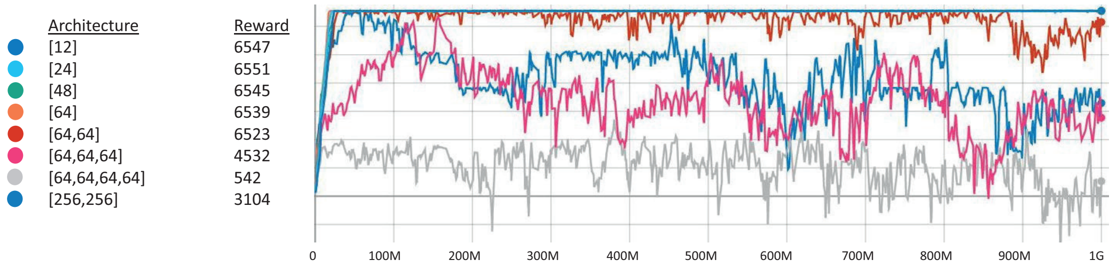


Figure D.3: Hyperparameter search 1: Testing different ANN architectures with a tanh activation function for a seeded 48 hour episode. Timesteps = 1G.

agent performance to the original ARS and the GLPK solver solution. Unlike experiments one and three, our second experiment also includes the reward penalty for deviation from 50% SOC at midnight.

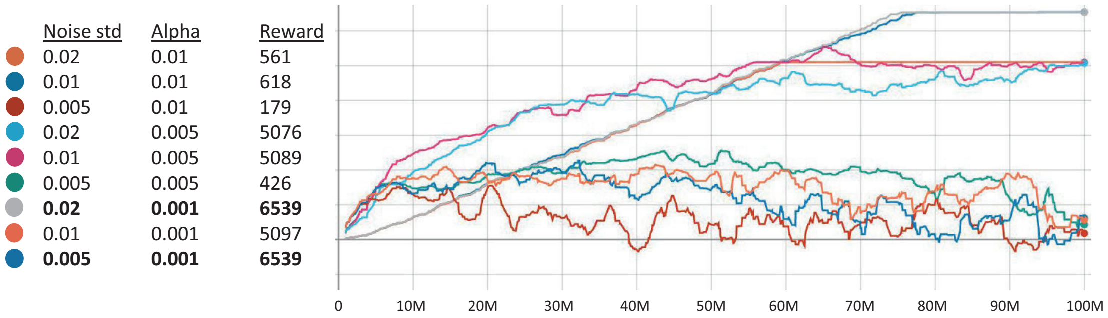
D.5.1 Hyperparameter searches

To find the best configuration of the network, we conducted so-called hyperparameter searches, i.e., exploration of hyperparameters and empirical verification of which of these parameters yielded the best result. Our chosen method for the hyperparameter search was a grid search over a table of predefined hyperparameter values. We ran the first grid search with different neural network architectures and activation functions to observe agent learning in a seeded 48-hour episode. Tanh activation function showed a tendency to destabilize as the size of the network increases, shown in Fig. D.3. Relu activation becomes stuck in zero for multiple architectures but is quicker to train in some cases than Tanh, although it converges at a level further from a near-optimal solution. The 24 neuron, a single hidden layer with Tanh activation quickly reaches a performance level very close to the GLPK solver solution and remains stable and even slightly increased performance after 1 billion timesteps. When network size increased from a single 64 neuron hidden layer to double 64 neuron hidden layers a notable decrease in stability occurs, shown in red in Fig. D.3. As can be observed, this destabilizing effect continues to increase with increasing in ANN size.

Seeing the destabilizing effect of increasing ANN complexity, and observing the results from experiment three presented further down, it led to the second 48h episode hyperparameter search featuring 4 hidden layers with 64 neurons each which consisted of a grid search with different learning rates α and noise standard deviations υ. Results show that learning and validation performance for deeper neural network architectures can be stabilized by decreasing the learning rate, shown in Fig. D.4. Reducing the learning rate from 0.01 to 0.001 significantly increases algorithm stability and performance.

D.5.2 Experiment one - Proof of concept

Our first full experiment was a proof of concept with the simple research goal of finding a solution for a single instance of our simulated environment. The experiment was conducted as a single randomly seeded 48-hour episode, initialized with a 450kW initial peak limit.



	Noise std	Alpha	Reward
●	0.02	0.01	561
●	0.01	0.01	618
●	0.005	0.01	179
●	0.02	0.005	5076
●	0.01	0.005	5089
●	0.005	0.005	426
●	**0.02**	**0.001**	**6539**
●	0.01	0.001	5097
●	**0.005**	**0.001**	**6539**

Figure D.4: Hyperparameter search 2: Testing impact of reducing learning rate and noise standard deviation for 4 hidden layer with 64 neuron architectures with a tanh activation function for a seeded 48 hour episode. Timesteps = 100M.

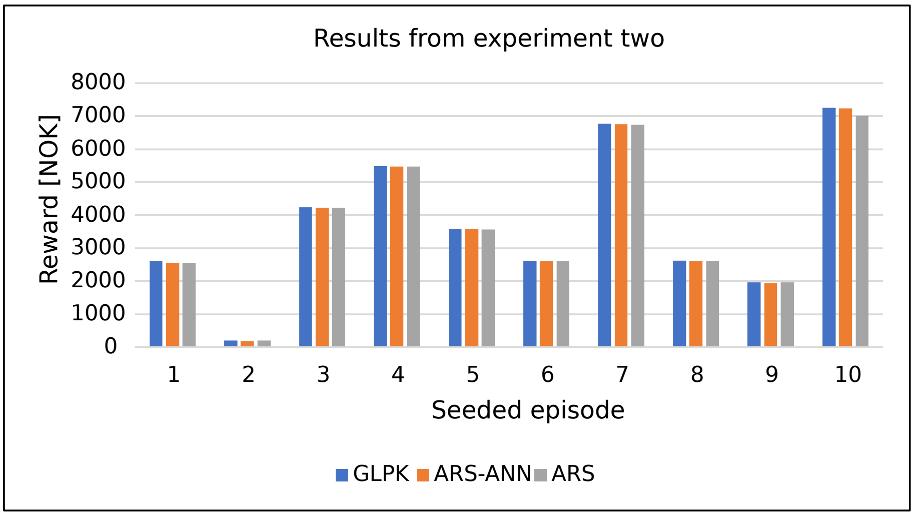


Figure D.5: Results for experiment two - 48 h trials with 50% SOC reward incentive. ARS-ANN architecture ”24 tanh”.

Table D.2: Results for experiment one.

Algorithm	Architecture	Score
Pyomo	GLPK Solver	6554
ARS-ANN	24-tanh	6543
ARS	Linear	6536
TD3	64x4-relu-torch (grid search)	1488

Table D.3: Results for experiment two - 48 h trials with 50% SOC reward incentive. ARS-ANN architecture ”24 tanh”.

Episode	GLPK	ARS-ANN			ARS-Original		
	Reduction	Reward	Reduction	Penalty	Reward	Reduction	Penalty
1	2609	2547	2556	8.9	2555	2556	1.2
2	206	185	187	2.7	193	193	0.6
3	4247	4211	4223	10.9	4204	4216	11.8
4	5497	5455	5481	26.6	5464	5474	10.7
5	3581	3576	3576	0.1	3573	3573	0.5
6	2611	2568	2605	36.2	2566	2601	35.4
7	6777	6728	6757	29.6	6727	6744	17.1
8	2613	2600	2605	5.0	2602	2606	4.5
9	1970	1933	1953	19.8	1955	1965	9.3
10	7247	7242	7241	0.1	6965	7020	56.0
Average	**3736**	**3705**	**3718**	**14.0**	**3680**	**3695**	**14.7**

The agent is given the energy price 6 timesteps ahead and the initial BESS SOC is set to 32kWh, which is the lower SOC limit. A grid search of ANN hyperparameters was conducted to find a fitting architecture. The grid search included the number of hidden layers and neurons, as well as activation functions ReLU and Tanh. The following architectures were attempted with both activation functions (the number of neurons in each hidden layer was always identical):

- One hidden layer: [12, 24, 48, 64]
- Two hidden layers: [24, 64, 256, 512]
- Three hidden layers: [64]
- Four hidden layers: [64]

Results for our ARS-ANN agent compared to benchmark algorithms in a 48-hour episode (experiment one) are shown in Table D.2. As shown in our initial trial, the ARS-ANN architecture with 24 neurons in a single hidden layer with a Tanh activation function shows the best performance behind the near-optimal solution found by the GLPK solver. The original ARS also performs quite well, scoring slightly below ARS-ANN at 99,7 compared to 99,8 % of the GLPK solution. We observe that the ARS algorithms seem to be very well

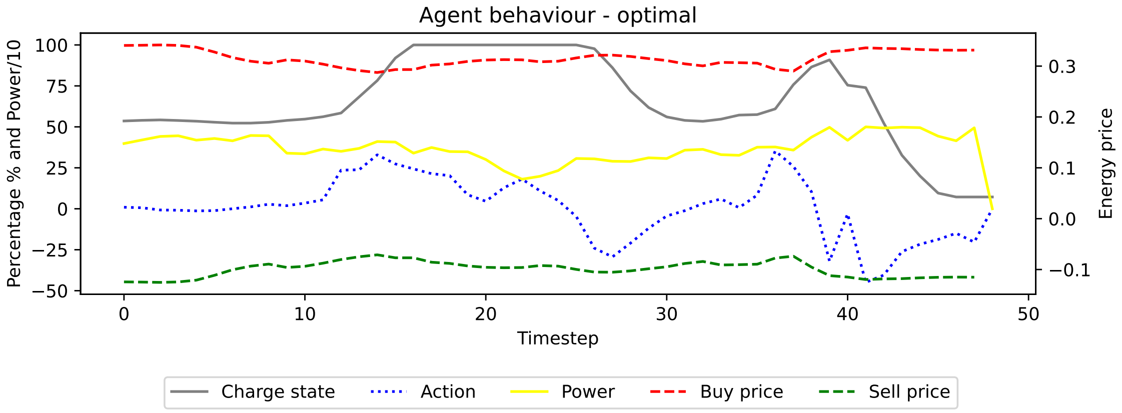


Figure D.6: Experiment two - Agent behaviour in 10th seeded episode.

suited for our energy cost optimization problem. Although the ARS-ANN algorithm only slightly improves the results in this experiment, we know that using an ANN means that we can find nonlinear solutions when problem complexity increases.

D.5.3 Experiment two - Verification through seeded multiple trials

As stated in Mania et al. (2018), too few experiments in RL verify results across multiple seeds, thus shedding some doubt on whether the reported performance is a result of algorithm ingenuity or extensive hyperparameter tuning to a single instance of the RL problem. To verify our results across multiple trials, we conducted an experiment with 10 randomly seeded time periods pulled from our data set. We set the initial peak power limit to 500kW. Initial battery SOC is set to 50% of available capacity between the operational capacity limits given in Equation A.7, therefore initiating at 238.5kWh. To incentivize an average SOC around the healthy 50%, the reward function was tuned to punish the agent at 00:00 each day by a fraction of the accumulated rewards proportional to the absolute difference between the SOC and the desired 50%. Results from experiment two can be observed in Table D.3 and Fig. D.5. We observe in Fig. D.5 that both the ARS and ARS-ANN algorithms are achieving results that are very close to the GLPK solver. When comparing numerical results in Table D.3 we observe that the ARS-ANN has a slight increase in performance when compared to the original ARS. In addition to peak shaving, the ARS-ANN can extract some values from energy price differentiation even though the reward increase from this behavior is an almost inconsequential due to an exceptionally low energy cost at around 0.3 NOK/kWh. The agent behavior in the 10th seeded trial is shown in Fig. D.6. The agent, actions shown in the blue dotted line, chose to charge when the energy price, red dashed line, was high. Similarly, it chose to discharge when the price was low while simultaneously avoiding the 500kW power limit and arriving at around 50% SOC at midnight. The ANN architecture in this experiment was 24 neurons in the hidden layer with a Tanh activation function.

We observe that the performance of the ARS algorithms is still very high, with the original ARS achieving an average of 98.5% and the ARS-ANN achieving 99.2% of the GLPK solver solution. The gap between the original ARS and ARS-ANN has also increased slightly. The same ARS-ANN architecture and hyperparameters were used for all the seeded trials, indicating that the performance is the result of a well-designed algorithm.

D.5.4 Experiment three - Longer episode

As a basis for future research, we also wanted to examine how our algorithm performs in a more complex scenario. A simple way to do this is to increase the number of timesteps taken in a single episode. We therefore finally conducted an experiment that uses almost the entire dataset as a single training episode. The episode was set to 2000 hours pulled sequentially from the operational data. The agent now faces a wide range of operational states, such that the problem can only be solved by learning a general and robust solution. The initial peak power limit was set to 550kW. Future energy buy price was set to 6 timesteps and the initial SOC was set to 32kWh. The near-optimal solution found with Pyomo and GLPK was a cost reduction of 7760NOK. The learning rate was not adjusted for the ARS-ANN agent, but a simple grid-search was performed to select the most appropriate ANN architecture. The grid search included the following architectures: [12], [24], [48], [64], [64,64], [64,64,64], [64,64,64,64], [256,256], as well as both Tanh and Relu activation functions. Results are shown in Table D.4. Although we can observe that the ARS results were quite far from optimal, Table D.4 shows that the ARS-ANN agent clearly outperformed the original ARS agent with a 21 % increase in performance. We interpret this as an indication that the original ARS with its linear policy will be too limited to handle the full complexity of our case-study warehouse energy system. We also believe that results for ARS-ANN can be greatly improved in complex scenarios by lowering the learning rate when ANN architecture is increased, as indicated by our previous discussion on Fig. D.4, combined with finding the most suitable architecture.

Table D.4: Results for experiment three - 2000 h episode.

Algorithm	Architecture	Score
Pyomo	GLPK	7761
ARS-ANN	64x2	4472
ARS	Linear	3689

D.6 Conclusions

This work is presenting the application of reinforcement learning based techniques to the specific energy optimization problem for controlling the battery energy storage system in a smart warehouse for minimizing the energy bill. This paper has adopted data from a real operational smart warehouse, integrated with a photovoltaic and battery energy storage system, for food distribution on the west coast of Norway. Multiple experiments have been conducted within a simulated training environment built with operational data from a case-study of a smart warehouse, featuring a 460kWh lithium-ion battery energy storage system. In this work, an RL agent and specifically the proposed ARS-ANN agent is trained and used for controlling the battery energy storage system's charging and discharging for minimizing the energy costs. Obtained results show that both the ARS and ARS-ANN algorithms have performed very well on 48-hour episodes, achieving an average of 98.5 and 99.2% accuracy respectively across 10 seeded trials. Also, ARS-ANN has shown promising results on a longer time horizon, outperforming original ARS by 21 %. As seen in the initial experiment on ARS-ANN with reduced learning rates, learning for deeper neural network architectures can be stabilized by lowering the learning rate α. We should therefore further explore if the ARS-ANN algorithm can be used to solve highly complex and realistic operational scenarios with longer time frames by increasing the depth of the architecture and reducing the learning rate. Adding multiple controllable energy storage solutions with different and more realistic dynamics, including thermal energy production and storage, should also be explored. The developed algorithm finds very promising solutions in the considered case-study of a smart house for energy cost minimization through a battery energy storage system. The presented methodology can be implemented in a wider range of smart energy-efficient buildings (e.g. smart warehouse) with less engineering detail for a reduction in energy bills.

Bibliography

Abadi Martín, Agarwal Ashish, Barham Paul, Brevdo Eugene, Chen Zhifeng, Citro Craig, Corrado Greg S., Davis Andy, Dean Jeffrey, Devin Matthieu, Ghemawat Sanjay, Goodfellow Ian, Harp Andrew, Irving Geoffrey, Isard Michael, Jia Yangqing, Jozefowicz Rafal, Kaiser Lukasz, Kudlur Manjunath, Levenberg Josh, Mané Dandelion, Monga Rajat, Moore Sherry, Murray Derek, Olah Chris, Schuster Mike, Shlens Jonathon, Steiner Benoit, Sutskever Ilya, Talwar Kunal, Tucker Paul, Vanhoucke Vincent, Vasudevan Vijay, Viégas Fernanda, Vinyals Oriol, Warden Pete, Wattenberg Martin, Wicke Martin, Yu Yuan, Zheng Xiaoqiang. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. Software available from tensorflow.org.

Aksanli Baris, Rosing Tajana, Pettis Eddie. Distributed battery control for peak power shaving in datacenters // 2013 International Green Computing Conference Proceedings. 2013. 1–8.

Barbato Antimo, Capone Antonio. Optimization models and methods for demand-side management of residential users: A survey // Energies. 2014. 7, 9. 5787–5824.

Brockman Greg, Cheung Vicki, Pettersson Ludwig, Schneider Jonas, Schulman John, Tang Jie, Zaremba Wojciech. OpenAI Gym. 2016.

Bynum Michael L., Hackebeil Gabriel A., Hart William E., Laird Carl D., Nicholson Bethany L., Siirola John D., Watson Jean-Paul, Woodruff David L. Pyomo–optimization modeling in python. 67. 2021. Third.

Cao Jun, Harrold Dan, Fan Zhong, Morstyn Thomas, Healey David, Li Kang. Deep Reinforcement Learning-Based Energy Storage Arbitrage With Accurate Lithium-Ion Battery Degradation Model // IEEE Transactions on Smart Grid. 2020. 11, 5. 4513–4521.

Fujimoto Scott, Hoof Herke van, Meger David. Addressing Function Approximation Error in Actor-Critic Methods // CoRR. 2018. abs/1802.09477.

Haarnoja Tuomas, Zhou Aurick, Abbeel Pieter, Levine Sergey. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor // CoRR. 2018. abs/1801.01290.

Hannan M.A., Wali S.B., Ker P.J., Rahman M.S. Abd, Mansor M., Ramachandaramurthy V.K., Muttaqi K.M., Mahlia T.M.I., Dong Z.Y. Battery energy-storage system: A review of technologies, optimization objectives, constraints, approaches, and outstanding issues // Journal of Energy Storage. 2021. 42. 103023.

Hart William E, Watson Jean-Paul, Woodruff David L. Pyomo: modeling and solving mathematical programs in Python // Mathematical Programming Computation. 2011. 3, 3. 219–260.

Henze Gregor P., Schoenmann Jobst. Evaluation of Reinforcement Learning Control for Thermal Energy Storage Systems // HVAC&R Research. 2003. 9, 3. 259–275.

Hoang Anh Tuan, Pham Van Viet, Nguyen Xuan Phuong. Integrating renewable sources into energy system for smart city as a sagacious strategy towards clean and sustainable process // Journal of Cleaner Production. 2021. 305. 127161.

Huang Hao, Chen Lei, Hu Eric. A new model predictive control scheme for energy and cost savings in commercial buildings: An airport terminal building case study // Building and Environment. 2015. 89. 203–216.

IEA . Energy Efficiency 2020. 2020.

Johannesen Nils Jakob, Kolhe Mohan, Goodwin Morten. Comparison of regression tools for regional electric load forecasting // 2018 3rd International Conference on Smart and Sustainable Technologies (SpliTech). 2018. 1–6.

Kuznetsova Elizaveta, Li Yan-Fu, Ruiz Carlos, Zio Enrico, Ault Graham, Bell Keith. Reinforcement learning for microgrid energy management // Energy. 2013. 59. 133 – 146.

Lešić Vinko, Martinčević Anita, Vašak Mario. Modular energy cost optimization for buildings with integrated microgrid // Applied Energy. 2017. 197. 14–28.

Liang Eric, Liaw Richard, Moritz Philipp, Nishihara Robert, Fox Roy, Goldberg Ken, Gonzalez Joseph E., Jordan Michael I., Stoica Ion. RLlib: Abstractions for Distributed Reinforcement Learning. 2018.

Lillicrap Timothy P., Hunt Jonathan J., Pritzel Alexander, Heess Nicolas, Erez Tom, Tassa Yuval, Silver David, Wierstra Daan. Continuous control with deep reinforcement learning. 2015.

Lipu MS Hossain, Hannan MA, Karim Tahia F, Hussain Aini, Saad Mohamad Hanif Md, Ayob Afida, Miah Md Sazal, Mahlia TM Indra. Intelligent algorithms and control strategies for battery management system in electric vehicles: Progress, challenges and future outlook // Journal of Cleaner Production. 2021. 292. 126044.

Lnett . Priser og vilkår bedrift. 2022.

Long Chao, Wu Jianzhong, Zhou Yue, Jenkins Nick. Peer-to-peer energy sharing through a two-stage aggregated battery control in a community Microgrid // Applied Energy. 2018. 226. 261–276.

Mania Horia, Guy Aurelia, Recht Benjamin. Simple random search provides a competitive approach to reinforcement learning. 2018.

Mariano-Hernández D., Hernández-Callejo L., Zorita-Lamadrid A., Duque-Pérez O., Santos García F. A review of strategies for building energy management system: Model predictive control, demand side management, optimization, and fault detect & diagnosis // Journal of Building Engineering. 2021. 33. 101692.

MIP in demand side response. // . 2019.

Mbuwir Brida V, Ruelens Frederik, Spiessens Fred, Deconinck Geert. Battery energy management in a microgrid using batch reinforcement learning // Energies. 2017. 10, 11. 1846.

Mnih Volodymyr, Kavukcuoglu Koray, Silver David, Graves Alex, Antonoglou Ioannis, Wierstra Daan, Riedmiller Martin A. Playing Atari with Deep Reinforcement Learning // CoRR. 2013. abs/1312.5602.

Mocanu E., Mocanu D. C., Nguyen P. H., Liotta A., Webber M. E., Gibescu M., Slootweg J. G. On-Line Building Energy Optimization Using Deep Reinforcement Learning // IEEE Transactions on Smart Grid. July 2019. 10, 4. 3698–3708.

Opalic S. M., Goodwin M., Jiao L., Nielsen H. K., Lal Kolhe M. A Deep Reinforcement Learning scheme for Battery Energy Management // 2020 5th International Conference on Smart and Sustainable Technologies (SpliTech). 2020. 1–6.

Opalic Sven Myrdahl, Goodwin Morten, Jiao Lei, Nielsen Henrik Kofoed, Pardiñas Ángel Álvarez, Hafner Armin, Kolhe Mohan Lal. ANN modelling of CO2 refrigerant cooling system COP in a smart warehouse // Journal of Cleaner Production. 2020. 260. 120887.

Paszke Adam, Gross Sam, Massa Francisco, Lerer Adam, Bradbury James, Chanan Gregory, Killeen Trevor, Lin Zeming, Gimelshein Natalia, Antiga Luca, Desmaison Alban, Kopf Andreas, Yang Edward, DeVito Zachary, Raison Martin, Tejani Alykhan, Chilamkurthy Sasank, Steiner Benoit, Fang Lu, Bai Junjie, Chintala Soumith. PyTorch: An Imperative Style, High-Performance Deep Learning Library // Advances in Neural Information Processing Systems 32. 2019. 8024–8035.

Perera A.T.D., Kamalaruban Parameswaran. Applications of reinforcement learning in energy systems // Renewable and Sustainable Energy Reviews. 2021. 137. 110618.

Rätz Martin, Javadi Amir Pasha, Baranski Marc, Finkbeiner Konstantin, Müller Dirk. Automated data-driven modeling of building energy systems via machine learning algorithms // Energy and Buildings. 2019. 202. 109384.

Schulman John, Levine Sergey, Moritz Philipp, Jordan Michael I., Abbeel Pieter. Trust Region Policy Optimization // CoRR. 2015. abs/1502.05477.

Sechilariu Manuela, Wang Bao Chao, Locment Fabrice. Supervision control for optimal energy cost management in DC microgrid: Design and simulation // International Journal of Electrical Power & Energy Systems. 2014. 58. 140–149.

Shang Yuwei, Wu Wenchuan, Guo Jianbo, Ma Zhao, Sheng Wanxing, Lv Zhe, Fu Chenran. Stochastic dispatch of energy storage in microgrids: An augmented reinforcement learning approach // Applied Energy. 2020. 261. 114423.

Siqueira Luanna Maria Silva de, Peng Wei. Control strategy to smooth wind power output using battery energy storage system: A review // Journal of Energy Storage. 2021. 35. 102252.

Smarra Francesco, Jain Achin, de Rubeis Tullio, Ambrosini Dario, D'Innocenzo Alessandro, Mangharam Rahul. Data-driven model predictive control using random forests for building energy optimization and climate control // Applied Energy. 2018. 226. 1252–1272.

Stroe D., Knap V., Swierczynski M., Stroe A., Teodorescu R. Operation of a Grid-Connected Lithium-Ion Battery Energy Storage System for Primary Frequency Regulation: A Battery Lifetime Perspective // IEEE Transactions on Industry Applications. 2017. 53, 1. 430–438.

Sutton Richard S., Barto Andrew G. Reinforcement Learning: An Introduction. 2018. Second.

Teleke Sercan, Baran Mesut E., Bhattacharya Subhashish, Huang Alex Q. Rule-Based Control of Battery Energy Storage for Dispatching Intermittent Renewable Sources // IEEE Transactions on Sustainable Energy. 2010. 1, 3. 117–124.

Wan Z., Li H., He H. Residential Energy Management with Deep Reinforcement Learning // 2018 International Joint Conference on Neural Networks (IJCNN). July 2018. 1–7.

Wang Bo, Cai Guowei, Yang Deyou. Dispatching of a Wind Farm Incorporated With Dual-Battery Energy Storage System Using Model Predictive Control // IEEE Access. 2020. 8. 144442–144452.

Learning from delayed rewards. // . 1989.

Wen Zheng, O'Neill Daniel, Maei Hamid. Optimal Demand Response Using Device-Based Reinforcement Learning // IEEE Transactions on Smart Grid. 2015. 6, 5. 2312–2324.

Yang Hongxing, Wei Zhou, Chengzhi Lou. Optimal design and techno-economic analysis of a hybrid solar–wind power generation system // Applied Energy. 2009. 86, 2. 163 – 169. IGEC III.

Appendix E

Paper E

Title:	COST-WINNERS: COST reduction WIth ANN-ARS for simultaneous thermal and electrical energy storage control
Authors:	Sven Myrdahl Opalic, Fabrizio Palumbo, Morten Goodwin, Lei Jiao, Henrik Kofoed Nielsen, and Mohan Lal Kolhe
Affiliation:	University of Agder, Faculty of Engineering and Science, 4879, Grimstad, Norway and Oslo Metropolitan University, P.O. Box 4 St. Olavs plass, 0130, Oslo, Norway
Journal:	*Journal of Energy Storage, Science Direct*, Accepted June, 2023.
DOI:	10.1016/j.est.2023.108202

E

COST-WINNERS: COST reduction WIth ANN-ARS for simultaneous thermal and electrical energy storage control

Sven Myrdahl Opalic, Fabrizio Palumbo, Morten Goodwin, Lei Jiao,
Henrik Kofoed Nielsen, and Mohan Lal Kolhe
Department of Engineering Sciences
Faculty of Engineering and Science, University of Agder
4879, Grimstad, Norway
E-mails: {sven.opalic, morten.goodwin, lei.jiao, henrik.kofoed.nielsen,
mohan.l.kolhe}@uia.no
Oslo Metropolitan University
P.O. Box 4 St. Olavs plass
0130, Oslo, Norway E-mails: {fabrizio}@oslomet.no

Abstract — **The combination of local renewable energy production, dynamic loads, and multiple energy storage systems with different dynamics requires sophisticated control systems to maximize the energy cost efficiency of the combined energy system. Battery and thermal energy storage systems can be combined to increase the local use of on-site renewable energy, reduce peak power demand, and exploit time-of-use energy pricing. In this paper, we focus on how the augmented random search algorithm and artificial neural networks can be used together to solve an energy cost optimization problem involving the control of a battery energy storage system and a thermal energy storage system at the same time in a smart warehouse. As part of this work, a simulated training environment made using the data from the smart warehouse's operations. In addition to the energy storage systems, the warehouse energy system has integrated a large roof mounted photovoltaic power plant and an industrial-scale cooling system.**

The developed solution is able to minimize the energy costs by modulating both energy systems, depending on the situation. Additionally, when it is tested against the state-of-the-art solutions, our developed solution at worst matches performance when the alternative algorithm is allowed to increase training time by a factor of nearly three. On average, our presented solution doubles the performance of the benchmark algorithm with much less computational resource expenditure.

E.1 Introduction

The current state of global energy supply and demand highlights the need for controllable energy production and storage (IEA, 2021). There is an increasing demand for robust and responsive electrical and thermal Energy Storage Systems (ESS) (Kebede et al., 2022) as an increasing fraction of the world's energy demand is met by wind and solar power at the expense of fossil-fueled and nuclear power (Buongiorno et al., 2019). The building sector represents a natural candidate to deploy an algorithm controlling renewable energy production and storage systems, as buildings are responsible for nearly 40% of global CO_2 emissions (IEA, 2020).

Energy Storage Systems (ESS) can consist of various technologies and be applied in a multitude of ways (Palizban, Kauhaniemi, 2016). From the perspective of the main electrical distribution grid, an important distinction exists between centralized and decentralized ESS. As opposed to decentralized ESS, centralized systems can be directly controlled by the grid operator. However, decentralized ESSs are seen as an important component of a more environmentally friendly energy system, but they come with a new set of challenges (Bögel et al., 2021). The decentralized systems should monitor the energy market, integrate the control algorithm with market dynamics, and use it to reduce the peak load of the system while also minimizing the costs. In the case of multiple ESSs with different dynamics, such as a combination of a Battery Energy Storage System (BESS) and Thermal Energy Storage (TES), the complexity of the optimization problem further increases.

One approach that is recently gaining a lot of interest in the scientific community as a robust and self-improving method to control building energy systems is Reinforcement Learning (RL) algorithms (Perera, Kamalaruban, 2021). RL algorithms can reduce costs by reducing necessary human resource expenditure, and risks associated with their behavior can be managed through off-line, data-driven training. Newer RL algorithms often include training Artificial Neural Networks (ANN) to output desired actions or action values, showing improved performance (Lillicrap et al., 2015; Cao et al., 2020; Shang et al., 2020). In contrast, Mania et al. (2018) showed that the Augmented Random Search (ARS) algorithm could achieve high performance with very little computational resource expenditure by training a simple linear function for action selection with their proposed search algorithm.

In this article we build on the work published in Opalic et al. (2022) where we showed that using ANNs for action selection together with the ARS search algorithm improved the agent performance on a BESS control problem. We now propose COST-WINNERS - a novel approach to control, for the first time, both the BESS and TES of a smart warehouse.

Specifically, our contributions in this paper are:

- We implement the ARS (Mania et al., 2018) RL algorithm, modified with ANNs to encode the agent policy, to simultaneously control TES and BESS energy storage systems.

- We build a data-driven simulated training environment, also modeling the dynamics of the TES.

- Overall, we introduce a novel approach to control both the BESS and TES of a smart warehouse simultaneously to reduce total energy cost. This is important because combining different energy storage systems can lead to improved performance and cost savings but also introduces new challenges due to each system's different dynamics and control requirements.

E.2 Related Work

It was suggested in Xu, Shen (2018) an algorithm for optimal control of multiple ESSs using individual custom defined boundaries for energy price. However, the study only features Battery Energy Storage Systems (BESSs) and does not specify how to determine the price boundaries for each system. Zhu et al. (2020) examines decentralized ESSs in urban railway applications and suggests multiagent deep Reinforcement Learning (RL) for cooperative control using Q-learning with recurrent ANNs. ANNs are also at the core of Model Predictive Control (MPC) of TES developed by Cox et al. (2019). Zhang et al. (2021) propose Soft Actor-Critic (SAC, (Haarnoja et al., 2018)) to optimize BESS control with multiple energy production facilities. However, the authors have not clarified if the experiment is based on more than a single 24-hour episode and results are only compared with other simpler RL algorithms. Goldsworthy et al. (2022) have implemented a cloud-based Model Predictive Control (MPC) battery control algorithm for energy cost reduction at an office building. The system has been operational for a year and achieved an energy cost reduction of 5.5%. Although some of the related work show promising results, we were unable to find any related work that examines advanced control algorithms for energy cost optimization with multiple ESSs with different dynamics, such as the BESS and TES in our smart warehouse.

E.2.1 Energy optimization in buildings

Similar to the Intelligent Energy Management System (IEMS) implemented in the warehouse and described in our previous work Opalic et al. (2022), Sechilariu et al. (2014) proposed Mixed Integer Linear Programming (MILP) to optimize energy cost and power flow in a Direct Current (DC) microgrid. Unlike the implemented smart warehouse IEMS, it also features instant power balancing. A hybrid Model, suggested by Huang et al. (2015), uses MPC for energy cost optimization in a case-study of an airport terminal. The authors suggest ANNs to account for non-linearity. MPC using hierarchical MPCs to provide thermal comfort and reduce energy cost was suggested in Lešić et al. (2017). Smarra et al. (2018) propose a data-driven MPC, i.e., Data Predictive Control (DPC), using a random forest algorithm for predictions, claiming that physical models are impractical when considering the unique character and complexity of building-related control systems. On the same line, Rätz et al. (2019) also explore data-driven energy system modeling for buildings using RL and MPC. For a twin BESS connected to a wind turbine power plant, Wang et al. (2020) suggest MPC. The authors assert greater production dispatchability and increased battery life. To conclude, a review study by Mariano-Hernández et al. (2021) determined that the most popular management technique in non-residential buildings is

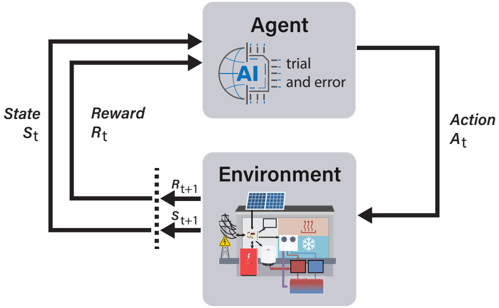


Figure E.1: Interaction between agent and environment.

MPC. They come to the conclusion that enabling intelligent control will depend on the building modeling methodology.

E.2.2 Reinforcement learning

According to Sutton, Barto (2018), RL is learning through discovering which actions that increase a reward.

The operational concept of RL is often described as shown in Figure E.1. An agent interacts with an environment, following its internal policy π, by taking actions and receiving feedback from it as reward or penalty. The policy typically gives the agent certain degrees of freedom to choose actions that deviate from the strictest application of the policy. This allows the agent to discover new states and actions that generate higher reward, consequently updating its policy.

Well-known RL algorithms include Q-learning (Watkins, 1989) and Deep Q-Networks (DQN) (Mnih et al., 2013). The Q-learning algorithm maps each state to the expected discounted future value of all the possible actions (Q-value). In Q-learning, the agents' policy is encoded in the Q-table, and the deterministic version of it maximizes Q-value. DQN deploys a deep ANN to compute the Q-values of available discrete actions given an environment state.

The application of RL has been conducted in many different fields, ranging from renewable energy to energy storage, and complex energy systems. An example can be seen in Kuznetsova et al. (2013). The authors developed a simulated microgrid, including a BESS and a wind turbine. The methodology is based on Q-learning taking as inputs the BESS State Of Charge (SOC), energy price, predictions of wind power production, and energy consumption demand. The discrete action space includes three possible BESS actions: charging, discharging, or none. Mbuwir et al. (2017) proposed fitted Q-iteration for transfer learning of BESS control to and from systems with comparable properties. Wen et al. (2015) suggest adopting Q-learning and end-user device utilization for controlling load shifting in modest office and apartment buildings. Additionally, Henze, Schoenmann (2003) also used Q-learning for TES control.

Perera, Kamalaruban (2021) found that Q-learning is the most common use of RL techniques in the energy research area, even if simpler algorithms are still deployed. Importantly, there are also attempts at exploring state-of-the-art algorithms in the literature. Mocanu et al. (2019) propose Deep Policy Gradient (DPG), similar to DQN, for on-off load shifting in the residential sector. Focusing on residential BESS control, a variant of Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2015) is developed by Wan et al. (2018). Moreover, an improved DQN was implemented also by Cao et al. (2020) for BESS arbitrage. This algorithm takes into account a lithium-ion battery degradation model, with discretized action space for full or 50% capacity dynamics together with the stand-by state. Shang et al. (2020) combines DQN with bootstrapping and a monte carlo tree search for BESS control in a microgrid. However, in all cases except Wan et al. (2018), the algorithms work in a discrete domain, having limited action space. In addition, the reward functions are generally complicated and experiment specific. Therefore, most of the approaches mentioned are not ideal for large-scale implementation of IEMS in a multitude of sites using RL.

Brandi et al. (2022) explored control of a TES using online deep RL, MPC and offline deep RL. For the online RL controller, energy cost was increased by 160% for a four week period before it converged to comparable behaviour to the top performing MPC and offline RL controllers. The study is limited to optimizing electricity cost incurred by the chiller while disregarding overall building energy cost and potential peak power cost.

Wang, Hong (2020) conducted a survey of RL application to control technical systems in buildings. The authors argue that established techniques such as MPC requires extensive domain knowledge to properly design and implement, making it less applicable in the building control domain compared with mass production domains such as the automobile industry. Furthermore, Wang, Hong (2020) state that RL combined with transfer learning should be further explored for building control.

The authors in Xu et al. (2021) propose a combination of RL with differential evolution to reduce energy cost for industrial users with solar power and thermal energy production, as well as BESS and TES, while satisfying local energy demand and trading energy in an energy trading platform.

E.2.3 Augmented random search

ARS is an optimization of what was named basic random search by Mania et al. (2018). ARS is designed for continuous action space and works with a strictly linear policy matrix, as opposed to other current RL approaches. Moreover, exploration with the ARS is done directly in the parameters of the policy function. In comparison, algorithms such as SAC (Haarnoja et al., 2018), DDPG (Lillicrap et al., 2015), TD3 (Fujimoto et al., 2018), and Trust-Region Policy Optimization (TRPO) (Schulman et al., 2015), also operating in continuous action space, promote action exploration with random noise added to the agents selected action. In the ARS algorithm, random noise is generated and added directly to the policy parameters and tested in the environment. The rewards from N such tests, or rollouts, are then sorted in descending order (Mania et al., 2018). The top b directions are used to update the policy according to

$$\theta_{j+1} = \theta_j + \frac{\alpha}{b\sigma_R} \sum_{k=1}^{b} \left[r\left(\pi_{j,(k),+}\right) - r\left(\pi_{j,(k),-}\right) \right] \delta_{(k)}, \tag{A.1}$$

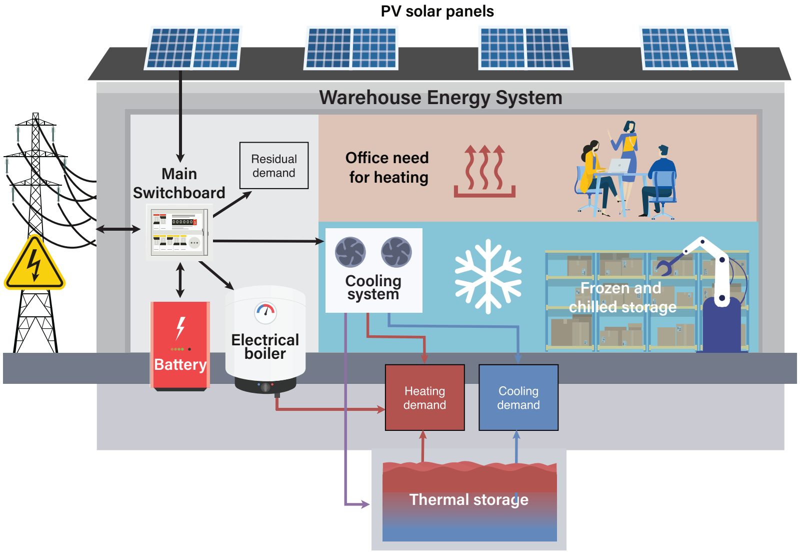


Figure E.2: The smart warehouse energy system with BESS, TES, cooling system and PV power plant. Arrows indicate the direction of energy flow.

where θ represents the policy parameters, α represents the learning rate, σ_R is the reward standard deviation, $r(\pi_{j,(k),+})$ and $r(\pi_{j,(k),-})$ are the rewards from rollouts and $\delta_{(k)}$ is the random noise fitted in size to θ. Continuously updated mean and standard deviation of input variables are used to normalize the inputs. Mania et al. (2018) managed to achieve outstanding performance while also drastically using less computational resources when tested in a variety of well-known RL benchmark problems.

E.3 Smart warehouse energy system

Table E.1: Main components of the smart warehouse energy system. * *At 10 °K temperature difference.*

System	Characteristic value	Unit of measurement
Solar power plant	1,000	[kW$_p$]
BESS	460/200	[kWh/kW]
TES	300/300*	[m^3/kW$_{thermal}$]
Cooling plant	1,140	[kW$_{thermal}$]
Electric boiler	500	[kW]

The energy system in the smart warehouse has previously been described in detail in Opalic et al. (2020), Opalic et al. (2020) and Opalic et al. (2022). Table E.1 lists its main components and a scheme of it is visualized in Fig. E.2. In this work we focus mainly on describing the thermal components of the energy system, and specifically the TES.
The main thermal components of the energy system are:

- the cooling plant with cooling energy distribution through evaporators based on direct expansion of carbon dioxide

Table E.2: Thermal energy storage system characteristics.

Attribute	Values	Unit of measurement
Measurements	LxWxH - 12x10x2,5	[m]
Volume	300	[m^3]
Average U-value	0.20	$[\frac{W}{m^2K}]$
Storage medium	Water	N/A
Heat exchanger max flow	25	$[\frac{m^3}{h}]$
Heat exchanger temperature loss	2	$[^oK]$

- heat recovery from the cooling plant with hydronic heating energy distribution
- TES in an insulated firewater tank submerged in the ground.
- cooling for ventilation and server rooms with hydronic cooling energy distribution

The physical characteristics of the tank are listed in Table E.2. TES specifications and model parameters are listed in Table E.2. Additionally, the energy system also features a BESS, described in detail in Opalic et al. (2022), that is controlled simultaneously with the TES.

The TES is used to store both heating and cooling energy. Switching between heating and cooling storage, on the other hand, incurs a significant cost due to the difference in operational temperature levels of the heating and cooling distribution systems at 50°C and 25°C, and 9°C and 15°C, respectively. Therefore, the TES is used only for heat storage in winter and for cooling storage during summer. For the remainder of this paper, we focus on the TES in heat storage mode. Since the TES is located underground, the ambient temperature also remains relatively stable and is modelled as a constant temperature.

Excess heat is recovered from the cooling plant and can either be directly distributed to cover the warehouse heating demand or stored in the TES, or both. Available excess heat depends on the cooling demand of the refrigerated areas in the building and will vary proportionally to the cooling work done by the cooling plant. If available heat is not sufficient to cover the heating demand, the remaining demand can either be met by discharging stored energy from the TES or by producing heat with an electrical boiler. The boiler can produce heat at an efficiency of around 0.9, whereas using excess heat from the cooling plant only incurs a small cost based on various operating conditions such as internal operating pressure, operational temperature, external cooling demand, and ambient temperature. Recovering and storing excess heat for later discharge can therefore be defined as a time-dependent optimization problem for energy cost reduction.

An IEMS currently controls the on-site ESSs by applying machine learning to predict load and PV solar panel production (Marton, others, 2019). Additionally, an optimization algorithm calculates a two-day plan for the the BESS and a TES deployment. The local Building Management System (BMS) implements the schedule and updates it hourly. The current IEMS system does not react to live operational data. Every hour the system calculates another two-day schedule, implementing the first hour's actions. Therefore, the system is very dependent on accurate predictions for maximum energy storage and cost reduction. Furthermore, in this scenery, it is challenging to prevent excessive peak power load costs. The magnitude of the challenge is only amplified if we take into account the structure used for the monthly peak power tariff, by the local grid operator: the entire monthly peak power cost is dependent on the highest hourly peak of that month. It is clear then that combining long-term planning with short-term reactions is a key strategy to benefit from the ESS' capability for peak power shaving.

E.4 Methodology

In this paper, we examine the applicability of the ARS-ANN RL algorithm to a complex energy cost reduction problem through direct control of BESS and TES charging and discharging setpoints in a simulated case-study smart warehouse. Our main research goal is to examine if the ARS-ANN algorithm can efficiently control multiple ESSs with different dynamics and substantially varying degrees of impact on energy cost. The agent is trained in a simulated environment of the smart warehouse, which we mainly designed through the use of data-driven techniques. We have emphasized the use of data-driven techniques as a way to reduce the need for human expertise to design the simulated environment and increase the practical utility of our approach.

E.4.1 Simulated environment

We have built the simulated environment on operational data using linear and polynomial regression in order to make the simulated environment accessible for result analysis. As this potentially decreases the accuracy of the system model, one could consider building a more accurate model of the environment using deep learning neural networks in an operational scenario. The methodology described in (Rätz et al., 2019) or similar approaches would then be considered. The current version of the simulated environment features an ensemble of models of energy system components and dynamics.

We use a model for the thermal energy storage, production and distribution featuring:

- The heat exchanger temperature loss.
- Temperature loss through heat conduction to surroundings.
- 4 vertical internal temperature levels.

Important components and dynamics of the models for the TES, production, and distribution are the following:

- Operational data of TES charging and discharging compared to setpoint.
- TES storage loss and internal temperature levels.
- Cooling plant electrical power consumption and recoverable excess heat.

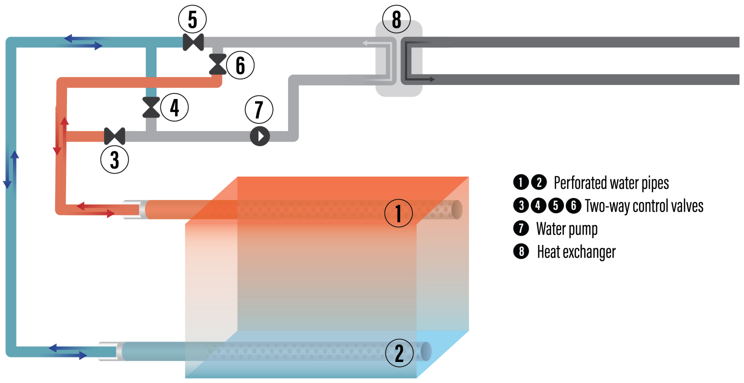


Figure E.3: Thermal energy storage with valves for reversing direction of water flow.

A schematic of the TES is included in Fig. E.3. The schematic shows the TES in the bottom visualized as a rectangular prism. Physically, the TES is a subterranean concrete basin, insulated on all sides with perforated water pipes (1, 2) placed diagonally along opposite walls within. This allows for an even distribution of water flowing into and out of the thermal storage, consistent with a strategy of maintaining water temperature layering inside the tank. The direction of water flowing through the tank can be reversed using an arrangement of four two-way valves (3-6). The TES is physically separated from the main hydronic energy distribution systems by a heat exchanger (8). The flow volume on the TES side of the heat exchanger is automatically balanced with the main hydronic energy distribution system using flow measurements and a frequency controlled pump (7). Our model of the TES includes the ability to reverse the direction of the flow of water such that hotter water is always added to or extracted from the top of the tank and vice versa for colder water. We have not included a model of the heat exchanger due to the physical system automatically balancing volume flow on both sides of the heat exchanger and the observed temperature loss in the heat exchanger is minimal. Modelling the heat exchanger could possibly be considered for future work.

On the secondary side of the heat exchanger, the TES is connected to the hydronic distribution system in two ways (not shown in the figure). Firstly, the TES is connected in parallell with all the thermal heat loads with a modulating two-way control valve that controls the charging according to an external thermal power setpoint. Secondly, the TES can be discharged by circulating the combined return flow through a modulating three-way valve that also responds to an external thermal power setpoint.

However, the dynamics of the hydronic heating system is complicated. We have therefore examined TES operational data in response to charging and discharging set points. The examination shows a high degree of variation between the actual delivered and the requested charge, as well as a non-linear relationship between charging and discharging dynamics. Therefore, we chose to model charging and discharging dynamics with two different functions, using more recent operational data. Charging dynamic is shown in Fig. E.4, while the discharging dynamic is illustrated in Fig. E.5. However, we provide a TES action space balanced around the origin of [-100, +100] to the agent interacting with

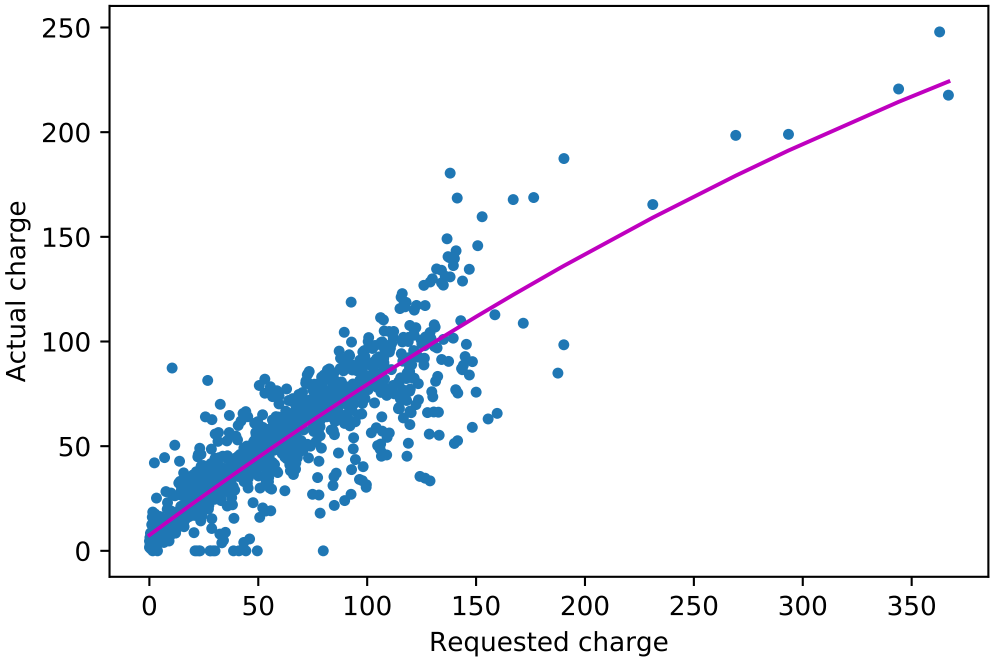


Figure E.4: Requested TES charging vs. actual charging.

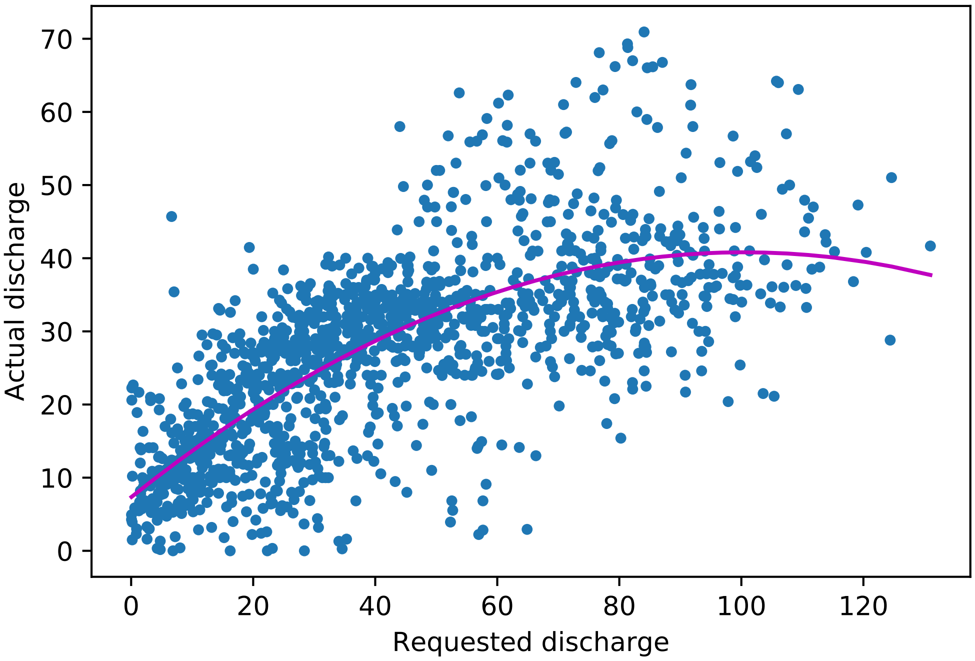


Figure E.5: Requested TES discharging vs. actual discharging.

the environment. Actions below 0.2 and above -0.2 are regarded as standby, or no-action. The R^2 score for the charging and discharging functions is 0.83 and 0.53, respectively. A qualitative analysis of the figures highlight a larger spread in the data point for the discharge function. Importantly, although the R^2 for the discharge function is rather low, the goal of this function is to have a simple and explainable model of the TES while discharging. The variation in TES discharging, related to the setpoint, is known to depend on a multitude of other variables when considering a priori and empirical knowledge of the hydronic heating system and is beyond the scope of this paper. A more practical way to model the TES dynamic, with a higher degree of accuracy, is likely through the use of ANN and multiple input variables. However, this would reduce model explainability, and it is not desirable at the current stage.

Algorithm 3 Augmented Random Search with ANN

```
Set hyperparameters:
    • $\alpha$ - learning rate
    • $n$ - number of directions sampled per iteration
    • $v$ - exploration noise standard deviation
    • $b$ - number of top-performing directions to use
Run algorithm 4 to initialize policy parameters $\theta_j$, i.e. ANN weights
Initialize:
    • Mean - $\mu_0 = 0 \in \mathbb{R}^{inputs}$
    • Covariance - $\Sigma_0 = \mathbf{I}_n \in \mathbb{R}^{inputsxinputs}$
while ending condition not satisfied do
    Sample $\delta_1, \delta_2, ..., \delta_N$ of the same size as $\theta_j$, with i.i.d. standard normal entries.
    Normalize input values $x$ with $x_{normalized} = diag(\Sigma_j)^{-\frac{1}{2}}(x - \mu_j)$. Collect $2N$ rollouts of horizon $H$ and their corresponding rewards using noise modified ANN policies $\pi_{j,k,+}$ and $\pi_{j,k,-}$, where the $v\delta_k$ exploration noise is added to the weight parameters $\theta_j$ of the ANN for $\pi_{j,k,+}$ and subtracted from $\theta_j$ for $\pi_{j,k,-}$ with $k \in \{1, 2, ..., N\}$.
    Sort the directions $\delta_k$ by $\max\{r(\pi_{j,k,+}), r(\pi_{j,k,-})\}$, denote by $\delta_{(k)}$ the $k$-th largest direction, and by $\pi_{j,(k),+}$ and $\pi_{j,(k),-}$ the corresponding policies.
    Make the update step for the ANN weights:
    $\theta_{j+1} = \theta_j + \frac{\alpha}{b\sigma_R} \sum_{k=1}^{b} [r(\pi_{j,k,+}) - r(\pi_{j,k,-})]\delta_k$, where the standard deviation of the $2b$ rewards for the policy update is $\sigma_R$.
    Set the mean and covariance, $\mu_{j+1}, \Sigma_{j+1}$, of the $2NH(j+1)$ training states encountered.
    $j \leftarrow j + 1$.
end while
```

In this article we have implemented the warehouse model described in (Opalic et al., 2020), and configured it to continuously calculate the refrigerant mass flow in the cooling plants. We have fitted a linear regression model, using pressure and mass flow of the refrigerant as inputs and recoverable heat as output. Consequently, this model can be used to find the recoverable heat upper bound at the maximum pressure of 80 bar and at any given refrigerant mass flow.

Moreover, we also model the electric consumption of the cooling plant as a second order polynomial, using refrigerant mass flow and heat recovered as inputs, and the electric consumption as output. The R^2 (R-Squared) score of the electric consumption function is 0.87, while the RMSE is 11.17.

The cooling work, expressed as the refrigerant mass flow, represents the limiting factor for the maximum heat that can be recovered. We model this dynamic with a simple linear function, using as input the refrigerant mass flow and returning as an output the maximum recoverable heat.

Finally, there is a minimum amount of electrical energy required by the cooling plant to keep the storage areas refrigerated. Also in this case we chose a linear model using as input the refrigerant mass flow and returning as an output the least required energy.

The following historical data sources were examined and used as input for the smart warehouse model:

- Total power consumption and local power production.
- Cooling plant power consumption.
- Cooling plant mass flow (Opalic et al., 2020).
- Heating demand.
- TES charging and discharging.
- Energy price for electrical energy bought from and sold to the grid.

E.4.2 ARS with ANN

In (Opalic et al., 2022), we implement a modified version of the ARS algorithm (Mania et al., 2018). We deploy an ANN for policy parametrization in place of the linear function proposed by Mania et al. (2018), see Algorithm 3. We thereby modify the processing of inputs to output from a linear to a nonlinear function. More specifically, the ARS algorithm is used to train an ANN to output actions for the TES and BESS with the input being the current state of the environment. We take advantage of the functionality for neural networks already implemented in the RLLIB programming library. Refer to (Opalic et al., 2022) for a detailed explanation of the implemented solution. The algorithm in this article is based on the previously suggested approach.

We use Pyomo (Hart et al., 2011; Bynum et al., 2021), an open-source Python tool for optimization modeling, with a GNU Linear Programming Kit (GLPK) solver to calculate near-optimal solutions for performance comparisons and benchmarking. We feed the GLPK solver with all the information about the training scenario and it attempts to find an optimal solution. However, due to the complex nature of our energy system, we did not attempt to implement the TES in the GLPK solver solution. We examined the operational data and found that the electrical boiler had contributed very little to satisfying the heating

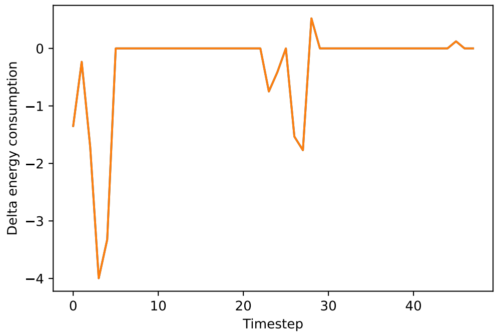


Figure E.6: Change in electrical energy consumption for the cooling system due to ARS-ANN agent TES control.

demand in the selected time period due to the fact that available excess heat from the cooling system seemed to be sufficient. Reducing energy consumption on the boiler is the main way that the TES can contribute to lower electrical energy consumption during winter operation. We argue that the impact of the TES on the energy cost in the time period we pulled our operational data from is very limited. Adopting the performance of the GLPK solver's control of the BESS as a benchmark is therefore still valid and useful.

Algorithm 4 ANN for ARS in RLLIB

```
Set hyperparameters:
    • $\theta^{hl}$ - ANN hidden layers.
    • $\theta^{nu}$ - number of neurons in each hidden layer.
    • $\theta^{af}$ - list of activation function for each layer.
Initialize: $j = 0$, policy parameters $\theta_j$ of shape defined by $\theta^{hl}$ and $\theta^{nu}$ and random
    values $X$ from $N(\mu_\theta, \sigma_\theta^2)$ normal distribution of mean $\mu_\theta = 0$ and variance $\sigma_\theta^2 = 1$,
    multiplied by standard deviation $\sigma = 1.0$ for the hidden layers and $\sigma = 0.1$ for the
    output, divided by the square root of the random value $X^{hl,nu}$, $\theta_j^{hl,nu} = X^{hl,nu}\frac{\sigma}{\sqrt{X}}$.
```

E.5 Scenarios: Results and discussions

In this section, we investigate the application of the ARS-ANN algorithm in a case-study smart warehouse, featuring both electrical (BESS) and thermal (TES) energy storage systems. Therefore, we have the opportunity of analysing algorithm performance on a complex temporal energy optimization problem. The objective of the algorithm is to reduce energy cost by controlling charging and discharging setpoints of both energy storage systems, BESS and TES.

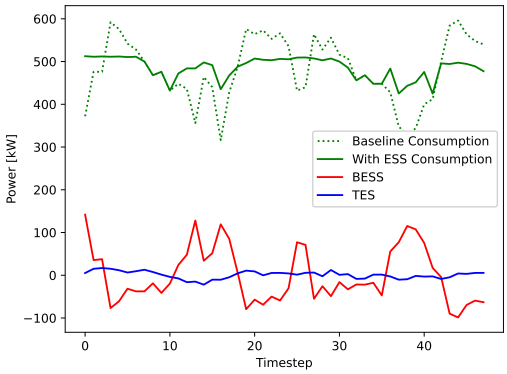


Figure E.7: Total energy consumption and ARS-ANN agent ESS utilization in experiment one.

Table E.3: Results for 10 seeded trials for ARS-ANN vs GLPK - battery only.

Trial	GLPK - Battery only	ARS-ANN Result	Percent of GLPK
1	4910	5046	103%
2	7115	7106	100%
3	7540	7498	99%
4	298	361	121%
5	643	639	100%
6	7117	7109	100%
7	5861	5864	100%
8	3771	3780	100%
9	640	641	100%
10	6652	3233	49%

Table E.4: Results for 10 seeded trials with state-of-the-art RL algorithms.

Trial	SAC		TD3	
	Reward	Percentage ARS-ANN	Reward	Percentage ARS-ANN
1	13	0.3 %	346	7 %
2	7083	99.7 %	290	4 %
3	7147	95.3 %	43	1 %
4	141	39.1 %	-62	-1 7%
5	86	13.4 %	76	12 %
6	1246	17.5 %	305	4 %
7	133	2.3 %	55	1 %
8	3772	99.8 %	21	1 %
9	728	113.5 %	232	36 %
10	691	21.4 %	-338	-10 %

E.5.1 Scenario I - Proof of concept

In scenario I (i.e. first experiment), we apply the ARS-ANN agent to control both BESS and TES for a random 48-hour episode. Our results clearly indicate that the agent is able to find a near-optimal value for BESS charging such that the peak power cost is reduced to a minimum. Change in cooling plant electrical consumption due to control of the TES is shown in Fig. E.6, whereas total energy consumption and ESS actions performed by the ARS-ANN agent are shown in Fig. E.7. As we can observe in Fig. E.7 the maximum hourly energy consumption is flattended, by utilization of the ESS, to around 512 kW, compensating for the consumption peak at almost 600 kW that would occur in the baseline consumption and contributing maunly in the reducing peak power tariff cost. The agent took advantage of the TES, when heating was required, to reduce the electrical energy required by the cooling plant. It is relevant to mention that the heating demand was very low during the random episode used for experiment one. However, the ARS-ANN agent was still able to find and store excess heat when there was no cost induced, and then in turn used this to partially reduce electrical consumption by discharging when necessary. Doing this, the agent was able to minimise cooling system energy demand when heat was in demand.

E.5.2 Scenario II - Seeded trials and benchmarking

To better quantify the performance of the ARS-ANN agent, we compare it with a GLPK optimization solver in multiple seeded trials, as well as benchmark it with other state-of-the-art RL algorithms. The GLPK will be controlling solely the BESS, with perfect information, and the comparison will be done for 10 seeded trials. Opposed to the GLPK, the ARS-ANN agent will have control of both BESS and TES. We have decided that comparing performance to an optimization algorithm, with perfect information, of simultaneous BESS and TES control is out of the scope of this paper due to the complexity. Additionally, the operational data used to pull random seeded trials is from early winter where the potential cost reduction of optimal TES control is minor compared with BESS control. There are two main reasons behind this choice of time period. Firstly, this was the time period with the most available data requiring minimal amounts of data cleaning. Secondly, we decided that observing how the algorithm performs in controlling multiple systems with vastly different impact on the result would be of interest.

The results of the simulation are displayed in Table E.3. We observe that for the majority of the trials, the energy cost reduction of the ARS-ANN with both BESS and TES control either meets or exceeds the cost reduction of the GLPK with BESS control only. For trial 10, the algorithm seems to get stuck in a local optima where it charges the battery too agressively on the first timestep. Additional research is required to explore why this happens and how it can be avoided in the future. In the 4th seeded trial we observe that the ARS-ANN outperforms GLPK by 21%. In this trial, the potential of cost reduction using the BESS is quite low due to a relatively low baseline peak power cost. Finally, we compare results for the SAC and TD3 RL algorithms to the ARS-ANN algorithm solution, shown in Table E.4. In Table E.4 the results for SAC and TD3 are compared to the results for ARS-ANN from Table E.3. Here, we can observe that TD3 seems to get stuck around origo while SAC actually performs reasonably well and even exceeds ARS-ANN in a single trial, finally achieving an average performance of 50% compared with ARS-ANN. However, on a reasonable time frame of running the algorithms for about a week of training time on 6 GPU's and 96 CPU's, both SAC and TD3 achieved similar results. It was only after increasing SAC training, by a factor of 3, to a total of more than 3 weeks that these results could be achieved. Also, the SAC algorithm results were not stable in the sense that the performance does not stabilize at a high performance. In fact, it drops off entirely in most cases. The results in Table E.4 include the maximum award achieved during each training session.

We also ran the seeded trials for the original ARS algorithm to quantify the improvement represented by ARS-ANN. The results showed that ARS performed at an average of 68 % compared to GLPK over the 10 trials and hence was outperformed by ARS-ANN by almost 30 percentage points.

E.5.3 Discussion

In this paper, we examine the applicability of the ARS-ANN RL algorithm to a complex energy cost reduction problem by direct control of BESS and TES charging and discharging setpoints in a simulated environment of an operational smart warehouse.

To evaluate our solution, we use a GLPK optimization solver, controlling only a BESS, as a benchmark. We have decided not to include the TES in the GLPK solver for two main reasons: (i) our initial data analysis demonstrated a marginal impact of the TES, and (ii) its complex thermal dynamics. We argue that for this work a GLPK solver with BESS represents a sufficient approximation to a good solution. We show that for nine out of ten of our seeded trials, the algorithm meets or exceeds the performance of a GLPK optimization solver controlling the BESS only, while given perfect information. For the single trial where it only performs at around 50% of the GLPK, the algorithm seems to get stuck in a local optimum which is to be further explored in future research.

We also compare our solution to state-of-the-art RL algorithms, showing an average of 100% performance increase compared to the SAC algorithm. However, the SAC algorithm was able to match or slightly exceed the performance of ARS-ANN in a few seeded trials when SAC training time was increased by a factor of 3. Further, the best results for SAC were not maintained as the training progressed, meaning that the performance declined after briefly achieving the highest performance for each training session. These "sparks of brilliance" could perhaps be leveraged in some way in future research. It would be of interest, for future work, to investigate possible solutions combining ARS-ANN and SAC for managing BESS and TES.

It is essential to mention that, due to time constraints and a lack of additional data, we only tested our approach in scenarios in which the heating demand was limited. It would be of interest, in future studies, to explore a broader landscape of scenarios, with higher heating demand, to evaluate the general efficacy of the method.

E.6 Conclusions

We demonstrate that we are able to minimize energy cost in the considered warehouse. We are able to model the dynamics of the TES and to use it in combination with BESS, controlled simultaneously by the ARS-ANN agent.

We demonstrate that by combining BESS and TES with the presented ARS-ANN agent, the agent was able to stabilize maximum energy consumption and thereby reducing the peak power cost. Additionally, the agent was able to exploit the TES when the heat was in demand to reduce the required electrical energy consumption by the cooling plant and electrical boiler.

To conclude, we propose a novel approach to control both the BESS and TES of a smart warehouse simultaneously to reduce total energy cost. This is important because combining different energy storage systems can lead to improved performance and cost savings but also introduces new challenges due to each system's different dynamics and control requirements. The results conclusively show that ARS-ANN outperforms comparable RL algorithms, achieving similar performance to an optimization algorithm controlling the BESS with perfect information.

Bibliography

Bögel Paula Maria, Upham Paul, Shahrokni Hossein, Kordas Olga. What is needed for citizen-centered urban energy transitions: Insights on attitudes towards decentralized energy storage // Energy Policy. 2021. 149. 112032.

Brandi Silvio, Fiorentini Massimo, Capozzoli Alfonso. Comparison of online and offline deep reinforcement learning with model predictive control for thermal energy management // Automation in Construction. 2022. 135. 104128.

Buongiorno Jacopo, Parsons John E, Petti David A, Parsons John. The future of nuclear energy in a carbon-constrained world // Massachusetts Institute of Technology Energy Initiative (MITEI). 2019.

Bynum Michael L., Hackebeil Gabriel A., Hart William E., Laird Carl D., Nicholson Bethany L., Siirola John D., Watson Jean-Paul, Woodruff David L. Pyomo–optimization modeling in python. 67. 2021. Third.

Cao Jun, Harrold Dan, Fan Zhong, Morstyn Thomas, Healey David, Li Kang. Deep Reinforcement Learning-Based Energy Storage Arbitrage With Accurate Lithium-Ion Battery Degradation Model // IEEE Transactions on Smart Grid. 2020. 11, 5. 4513–4521.

Cox Sam J., Kim Dongsu, Cho Heejin, Mago Pedro. Real time optimal control of district cooling system with thermal energy storage using neural networks // Applied Energy. 2019. 238. 466–480.

Fujimoto Scott, Hoof Herke van, Meger David. Addressing Function Approximation Error in Actor-Critic Methods // CoRR. 2018. abs/1802.09477.

Goldsworthy M., Moore T., Peristy M., Grimeland M. Cloud-based model-predictive-control of a battery storage system at a commercial site // Applied Energy. 2022. 327. 120038.

Haarnoja Tuomas, Zhou Aurick, Abbeel Pieter, Levine Sergey. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor // CoRR. 2018. abs/1801.01290.

Hart William E, Watson Jean-Paul, Woodruff David L. Pyomo: modeling and solving mathematical programs in Python // Mathematical Programming Computation. 2011. 3, 3. 219–260.

Henze Gregor P., Schoenmann Jobst. Evaluation of Reinforcement Learning Control for Thermal Energy Storage Systems // HVAC&R Research. 2003. 9, 3. 259–275.

Huang Hao, Chen Lei, Hu Eric. A new model predictive control scheme for energy and cost savings in commercial buildings: An airport terminal building case study // Building and Environment. 2015. 89. 203–216.

IEA . Energy Efficiency 2020. 2020.

IEA . World Energy Outlook 2021. 2021.

Kebede Abraham Alem, Kalogiannis Theodoros, Van Mierlo Joeri, Berecibar Maitane. A comprehensive review of stationary energy storage devices for large scale renewable energy sources grid integration // Renewable and Sustainable Energy Reviews. 2022. 159. 112213.

Kuznetsova Elizaveta, Li Yan-Fu, Ruiz Carlos, Zio Enrico, Ault Graham, Bell Keith. Reinforcement learning for microgrid energy management // Energy. 2013. 59. 133 – 146.

Lešić Vinko, Martinčević Anita, Vašak Mario. Modular energy cost optimization for buildings with integrated microgrid // Applied Energy. 2017. 197. 14–28.

Lillicrap Timothy P., Hunt Jonathan J., Pritzel Alexander, Heess Nicolas, Erez Tom, Tassa Yuval, Silver David, Wierstra Daan. Continuous control with deep reinforcement learning. 2015.

Mania Horia, Guy Aurelia, Recht Benjamin. Simple random search provides a competitive approach to reinforcement learning. 2018.

Mariano-Hernández D., Hernández-Callejo L., Zorita-Lamadrid A., Duque-Pérez O., Santos García F. A review of strategies for building energy management system: Model predictive control, demand side management, optimization, and fault detect & diagnosis // Journal of Building Engineering. 2021. 33. 101692.

Marton Gergely, others . MIP in demand side response // Master thesis. 2019.

Mbuwir Brida V, Ruelens Frederik, Spiessens Fred, Deconinck Geert. Battery energy management in a microgrid using batch reinforcement learning // Energies. 2017. 10, 11. 1846.

Mnih Volodymyr, Kavukcuoglu Koray, Silver David, Graves Alex, Antonoglou Ioannis, Wierstra Daan, Riedmiller Martin A. Playing Atari with Deep Reinforcement Learning // CoRR. 2013. abs/1312.5602.

Mocanu E., Mocanu D. C., Nguyen P. H., Liotta A., Webber M. E., Gibescu M., Slootweg J. G. On-Line Building Energy Optimization Using Deep Reinforcement Learning // IEEE Transactions on Smart Grid. July 2019. 10, 4. 3698–3708.

Opalic S. M., Goodwin M., Jiao L., Nielsen H. K., Lal Kolhe M. A Deep Reinforcement Learning scheme for Battery Energy Management // 2020 5th International Conference on Smart and Sustainable Technologies (SpliTech). 2020. 1–6.

Opalic Sven Myrdahl, Goodwin Morten, Jiao Lei, Nielsen Henrik Kofoed, Kolhe Mohan Lal. Augmented Random Search with Artificial Neural Networks for energy cost optimization with battery control // Journal of Cleaner Production. 2022. 134676.

Opalic Sven Myrdahl, Goodwin Morten, Jiao Lei, Nielsen Henrik Kofoed, Pardiñas Ángel Álvarez, Hafner Armin, Kolhe Mohan Lal. ANN modelling of CO2 refrigerant cooling system COP in a smart warehouse // Journal of Cleaner Production. 2020. 260. 120887.

Palizban Omid, Kauhaniemi Kimmo. Energy storage systems in modern grids—Matrix of technologies and applications // Journal of Energy Storage. 2016. 6. 248–259.

Perera A.T.D., Kamalaruban Parameswaran. Applications of reinforcement learning in energy systems // Renewable and Sustainable Energy Reviews. 2021. 137. 110618.

Rätz Martin, Javadi Amir Pasha, Baranski Marc, Finkbeiner Konstantin, Müller Dirk. Automated data-driven modeling of building energy systems via machine learning algorithms // Energy and Buildings. 2019. 202. 109384.

Schulman John, Levine Sergey, Moritz Philipp, Jordan Michael I., Abbeel Pieter. Trust Region Policy Optimization // CoRR. 2015. abs/1502.05477.

Sechilariu Manuela, Wang Bao Chao, Locment Fabrice. Supervision control for optimal energy cost management in DC microgrid: Design and simulation // International Journal of Electrical Power & Energy Systems. 2014. 58. 140–149.

Shang Yuwei, Wu Wenchuan, Guo Jianbo, Ma Zhao, Sheng Wanxing, Lv Zhe, Fu Chenran. Stochastic dispatch of energy storage in microgrids: An augmented reinforcement learning approach // Applied Energy. 2020. 261. 114423.

Smarra Francesco, Jain Achin, de Rubeis Tullio, Ambrosini Dario, D'Innocenzo Alessandro, Mangharam Rahul. Data-driven model predictive control using random forests for building energy optimization and climate control // Applied Energy. 2018. 226. 1252–1272.

Sutton Richard S., Barto Andrew G. Reinforcement Learning: An Introduction. 2018. Second.

Wan Z., Li H., He H. Residential Energy Management with Deep Reinforcement Learning // 2018 International Joint Conference on Neural Networks (IJCNN). July 2018. 1–7.

Wang Bo, Cai Guowei, Yang Deyou. Dispatching of a Wind Farm Incorporated With Dual-Battery Energy Storage System Using Model Predictive Control // IEEE Access. 2020. 8. 144442–144452.

Wang Zhe, Hong Tianzhen. Reinforcement learning for building controls: The opportunities and challenges // Applied Energy. 2020. 269. 115036.

Watkins Christopher John Cornish Hellaby. Learning from delayed rewards // PhD thesis, Cambridge University, Cambridge, England. 1989.

Wen Zheng, O'Neill Daniel, Maei Hamid. Optimal Demand Response Using Device-Based Reinforcement Learning // IEEE Transactions on Smart Grid. 2015. 6, 5. 2312–2324.

Xu Yinliang, Shen Xinwei. Optimal Control Based Energy Management of Multiple Energy Storage Systems in a Microgrid // IEEE Access. 2018. 6. 32925–32934.

Xu Zhengwei, Han Guangjie, Liu Li, Martínez-García Miguel, Wang Zhijian. Multi-energy scheduling of an industrial integrated energy system by reinforcement learning-based differential evolution // IEEE Transactions on Green Communications and Networking. 2021. 5, 3. 1077–1090.

Zhang Bin, Hu Weihao, Cao Di, Li Tao, Zhang Zhenyuan, Chen Zhe, Blaabjerg Frede. Soft actor-critic–based multi-objective optimized energy conversion and management strategy for integrated energy systems with renewable energy // Energy Conversion and Management. 2021. 243. 114381.

Zhu Feiqin, Yang Zhongping, Lin Fei, Xin Yue. Decentralized Cooperative Control of Multiple Energy Storage Systems in Urban Railway Based on Multiagent Deep Reinforcement Learning // IEEE Transactions on Power Electronics. 2020. 35, 9. 9368–9379.