博士論文

Research on operation optimization of building energy systems based on machine learning

機械学習に基づく建物エネルギーシステムの運用最適化に関する研究

北九州市立大学国際環境工学研究科

2023 年 6 月

徐　陽

Xu Yang

Doctor Thesis

RESEARCH ON OPERATION OPTIMIZATION OF BUILDING ENERGY SYSTEMS BASED ON MACHINE LEARNING

June 2023

Xu Yang

2020DBB411

The University of Kitakyushu
Faculty of Environmental Engineering
Department of Architecture
Gao Laboratory

Research on operation optimization of building energy systems based on machine learning

Abstract

Renewable energy has developed steadily in recent years in the context of energy shortages and safe supply requirements. The power sector, in particular, plays a crucial role in energy conservation and emission reduction. Renewable energy development can reduce dependence on fossil fuels and improve energy self-sufficiency rates. Since over 40% of total energy consumption comes from buildings, increasing the self-sufficiency rate of renewable energy in buildings is critical. While Japan's implementation of the feed-in tariff in 2011 led to explosive growth in household renewable energy equipment, the trend slowed as the feed-in tariff price decreased. Therefore, it is urgent to reduce further the cost of running household renewable energy equipment. This research focuses on applying machine learning in optimizing building energy system operations further to reduce the operation cost of building energy systems and increase the self-sufficiency rate of renewable energy.

In Chapter 1, INTRODUCTION AND PURPOSE OF THE RESEARCH. Chapter 1 introduces the background of energy research, including the current situation and bottlenecks of comprehensive energy development, as well as the importance of developing variable renewable energy sources. Additionally, it presents renewable energy's development and current state globally and in Japan. The chapter also highlights recent advances in energy prediction, reinforcement learning control, and related research demonstrating how machine learning technology can address energy security and renewable energy deployment issues in building energy systems. Finally, this chapter outlines the paper's research purpose and logical framework to help reviewers better understand its content.

In Chapter 2, METHODOLOGY. Chapter 2 focuses on the key concepts and methods used in the study, which include machine learning, deep learning, deep reinforcement learning, and energy storage systems. Specifically, the chapter summarizes the fundamental theories and methodologies of deep learning and deep reinforcement learning, which form the foundation of the algorithms utilized in the subsequent research.

In Chapter 3, MATERIALS AND DATA PREPROCESSING. Chapter 3 provides an in-depth analysis of the data resources and this study's preprocessing steps. The measured energy system data from Kitakyushu Science Research Park and Jono Zero Carbon Smart Community were utilized. This section details the system under consideration, the methodology employed for data preprocessing, potential data patterns, and the creation of the training and test sets utilized in the subsequent experiments.

In Chapter 4, POTENTIAL ANALYSIS OF THE ATTENTION-BASED LSTM MODEL IN BUILDING ENERGY SYSTEM. Chapter 4 aimed to evaluate the potential of using an attentional-based LSTM network (A-LSTM) to predict HVAC energy consumption in practical applications. To assess the potential applicability of the A-LSTM model in practical scenarios, the training and testing datasets used in the experiments consist of actual energy consumption data collected from Kitakyushu Science Research Park in Japan. Five baseline models (A-LSTM, LSTM, RNN, DNN, and SVR) were developed, and the Tree-structured Parzen Estimators (TPE) algorithm was introduced to optimize the model's super parameters. The subsequent application of the models on the target database resulted in a comprehensive analysis of the results from multiple perspectives. The results indicate that the A-LSTM model achieved the highest prediction accuracy, surpassing the LSTM model with a 3.06% reduction in overall RMSE, a 6.54% decrease in MSE, and a 0.43% increase in R^2 value. Furthermore, the A-LSTM model performed exceptionally well when the length of the training set was between 4 and 6 years. However, the model's prediction accuracy sharply decreased when the size of the training set was reduced to 2 years, indicating its limitations in predicting small sample data.

In Chapter 5, OPERATIONAL OPTIMIZATION FOR BUILDING ENERGY SYSTEMS USING VALUE-BASED REINFORCEMENT LEARNING. Chapter 5 presented the proposed model-based deep reinforcement learning algorithm called Model-based Double-Dueling Deep Q-Networks (MB-D3QN). This algorithm optimizes the cost-effective operation of a residential house equipped with a grid-connected PV-battery system in Japan. Results compared and analyzed the performance of Q-learning, DQN, and D3QN agents in optimizing the scheduling strategy of the residential PV-battery system based on real-world monitored data and real-time electricity price. The experimental results proved the effectiveness of the reward function design, and both DQN and D3QN algorithms can reduce energy costs. The case analysis based on the measured data also proves that the MB-D3QN algorithm provides a more efficient scheduling strategy. Compared to the baseline model, it reduces the annual electricity cost by 11.27%. According to the analysis of cost-effectiveness and influencing factors, it could be concluded that the optimization effect of the MB-D3QN method was mainly affected by the difference between the average PV generation and average load and then by the average RTP. The analysis of the Soc control effect proves that MB-D3QN can intelligently judge the future load and electricity price peak and take reasonable charge and discharge action. The comparison between the model-based D3QN method and the model-free D3QN method shows that the model-based approach proposed in this study can significantly improve sample utilization and effectively learn empirical knowledge from limited small sample data.

In Chapter 6, OPERATIONAL OPTIMIZATION FOR BUILDING ENERGY SYSTEMS USING ACTOR-CRITIC BASED REINFORCEMENT LEARNING CONSIDERING REAL-TIME ENERGY PREDICTION. Chapter 6 proposed a model-based RL control method considering real-

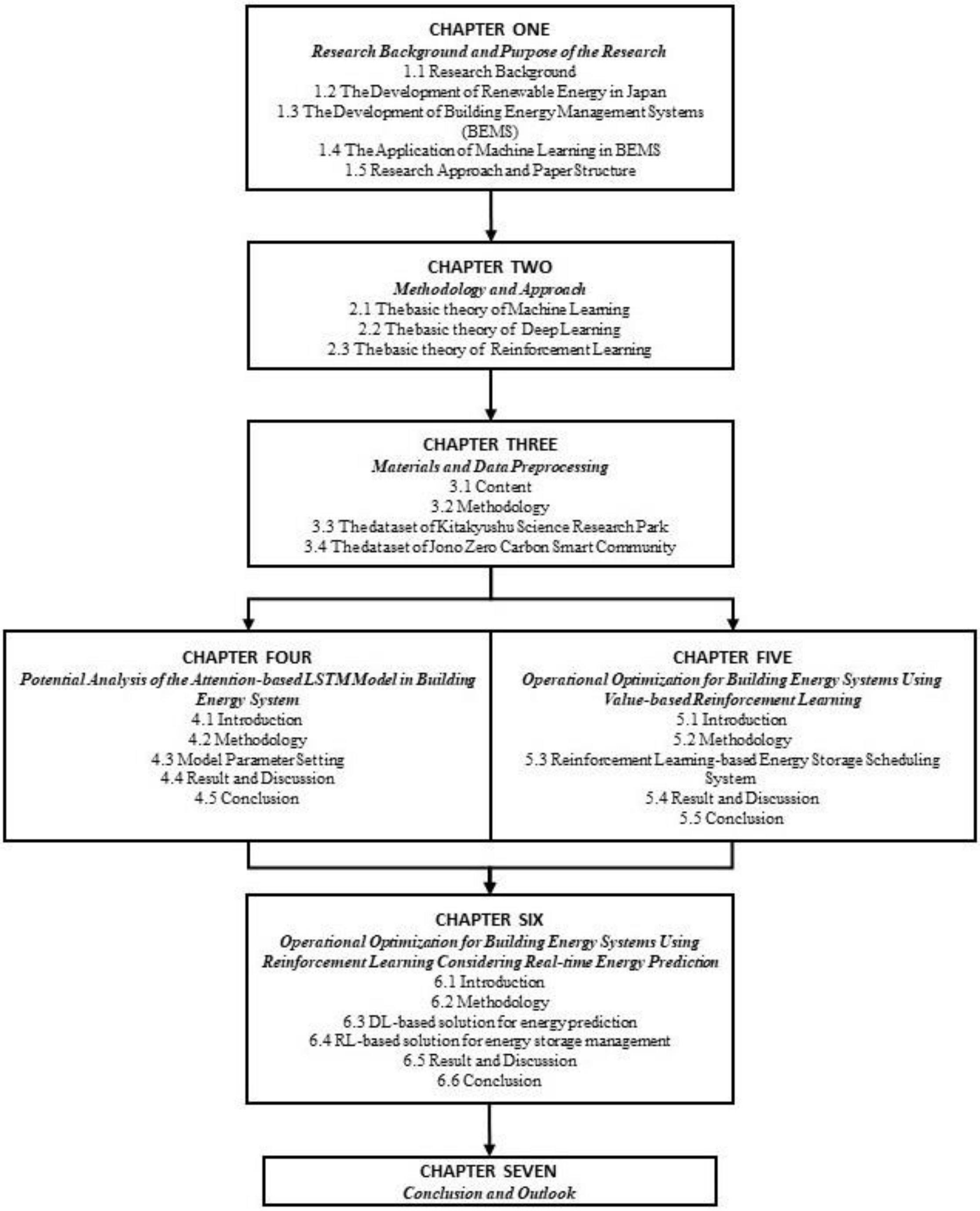
time prediction values for operation optimization of the residential PV-battery system. The optimization goals aim at reducing the energy cost of the microgrid and ensuring that the PV self-consumption ratio is not lower than the baseline model. To achieve this goal, this study designed a new multi-objective optimization reward function, and experimental results proved the effectiveness of the designed reward function. One of the key steps in this study was to develop and evaluate nine different prediction models with varying structures to predict power demand, real-time electricity price, and photovoltaic power generation. The optimal prediction model was selected for each variable through a comparative evaluation process. Subsequently, the predicted value from the selected models was incorporated into the observed state variable of the RL models for the next time step. The experimental results showed that the above four algorithms could achieve the optimization objective by using the designed reward function in this study. The TD3 algorithm had the best performance in each season. It could reduce the annual energy costs by 17.82% and increase the PV self-consumption ratio by 0.86% compared with the baseline model. In addition, the improved method proposed in this chapter is superior to the models proposed in Chapter 5 in terms of cost optimization and PV self-consumption ratio, which indicates that the solution proposed in this chapter is a better approach for this scenario.

In Chapter 7, CONCLUSION AND OUTLOOK. A summary of each Chapter is concluded.

Keywords: Machine Learning, Deep reinforcement learning, Operational optimization, Photovoltaic battery systems

徐 阳 博士論文の構成

Research on Operation Optimization of Building Energy Systems Based on Machine Learning



CHAPTER ONE

Research Background and Purpose of the Research

1.1 Research Background

1.2 The Development of Renewable Energy in Japan

1.3 The Development of Building Energy Management Systems (BEMS)

1.4 The Application of Machine Learning in BEMS

1.5 Research Approach and Paper Structure

CHAPTER TWO

Methodology and Approach

2.1 The basic theory of Machine Learning

2.2 The basic theory of Deep Learning

2.3 The basic theory of Reinforcement Learning

CHAPTER THREE

Materials and Data Preprocessing

3.1 Content

3.2 Methodology

3.3 The dataset of Kitakyushu Science Research Park

3.4 The dataset of Jono Zero Carbon Smart Community

CHAPTER FOUR

Potential Analysis of the Attention-based LSTM Model in Building Energy System

4.1 Introduction

4.2 Methodology

4.3 Model Parameter Setting

4.4 Result and Discussion

4.5 Conclusion

CHAPTER FIVE

Operational Optimization for Building Energy Systems Using Value-based Reinforcement Learning

5.1 Introduction

5.2 Methodology

5.3 Reinforcement Learning-based Energy Storage Scheduling System

5.4 Result and Discussion

5.5 Conclusion

CHAPTER SIX

Operational Optimization for Building Energy Systems Using Reinforcement Learning Considering Real-time Energy Prediction

6.1 Introduction

6.2 Methodology

6.3 DL-based solution for energy prediction

6.4 RL-based solution for energy storage management

6.5 Result and Discussion

6.6 Conclusion

CHAPTER SEVEN

Conclusion and Outlook

TABLE OF CONTENTS

CHAPTER 1: RESEARCH BACKGROUND AND PURPOSE OF THE RESEARCH

CHAPTER 2: METHODOLOGY AND APPROACH

CHAPTER 3: MATERIALS AND DATA PREPROCESSING

CHAPTER 4: POTENTIAL ANALYSIS OF THE ATTENTION-BASED LSTM MODEL IN BUILDING ENERGY SYSTEM

CHAPTER 6: OPERATIONAL OPTIMIZATION FOR BUILDING ENERGY SYSTEMS USING REINFORCEMENT LEARNING CONSIDERING REAL-TIME ENERGY PREDICTION

CHAPTER 7:CONCLUSION AND OUTLOOK

Chapter 1

RESEARCH BACKGROUND AND PURPOSE OF THE RESEARCH

CHAPTER ONE: RESEARCH BACKGROUND AND PURPOSE OF THE RESEARCH

1.1 Research Background

In recent years, the rapid development of industrialization and urbanization has led to a sharp rise in global energy demand, creating severe challenges for mitigating climate change. The energy industry is essential for ensuring the production and development of human society, as it not only guarantees the sustainable growth of the economy but also reflects a nation's strategic competitiveness. This growing energy demand necessitates the implementation of an effective and efficient energy strategy that can address both economic and environmental concerns. Fig. 1-1 demonstrates that global electricity demand has grown significantly over the past 30 years, increasing from 12,000 TWh in 1990 to 25,000 TWh in 2020 - an increase of 52%. Furthermore, the International Energy Agency (IEA) predicts this demand will reach 40,000 TWh by 2040, representing an additional 27.5% increase.

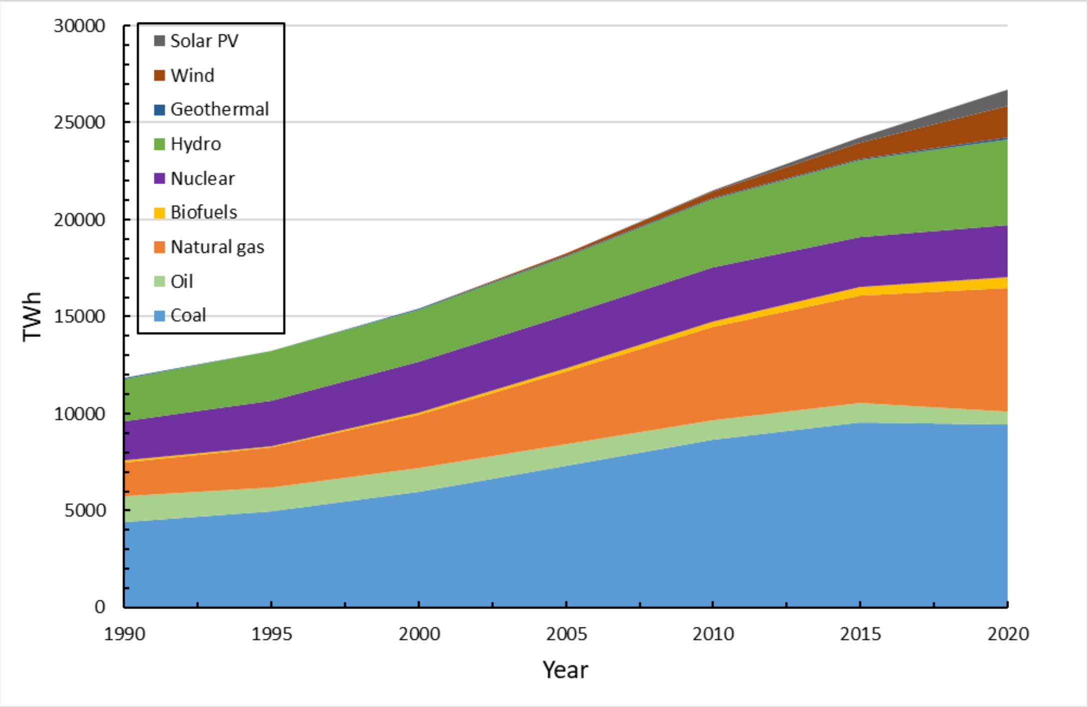


Fig. 1-1 The electricity generation by the different technology of the world (Resource: IEA data, World Energy Outlook 2020)[1]

Fossil fuels have been a cornerstone of economic and social development since the industrial revolution. However, the unsustainable consumption of fossil fuels has led to global climate change and human health problems. Fossil fuels account for approximately 75% of global greenhouse gas emissions, resulting in rising global temperatures and more frequent extreme weather events, negatively impacting the natural environment and human society. Furthermore, fossil fuels are a major contributor to local air pollution. The harmful substances, such as nitrogen oxides, sulfur dioxide, and particulate matter, released during the combustion of fossil fuels not only pollute the air but also negatively impact human health. According to the World Health Organization, at least 5 million people die prematurely each year due to this health issue, making it one of the most significant global health challenges. The world must rapidly transition to low-carbon energy to reduce carbon dioxide emissions and local air pollution. Nuclear power and renewable technologies, including solar, wind, hydro, and biomass energy, are among the most promising options. As an efficient, clean, and reliable energy source, nuclear power can provide significant electricity without producing greenhouse gas emissions or air pollutants and has become an important energy source in many countries worldwide. Renewable energy sources have the advantages of wide distribution, high renewability, and zero emissions and have become an important direction for global energy structure transformation.

According to the International Energy Agency (IEA), CO2 emissions have sharply increased since the 1950s, reaching 2.38 billion tons in 2000. From 2000 to 2010, emissions rose by 32%, reaching 3.16 billion tons in 2010. The IEA predicts total global emissions will reach approximately 3.68 billion tons by 2020, shown in Fig. 1-2. However, due to the suspension of global economic activity and the decrease in energy demand caused by the COVID-19 pandemic, carbon dioxide emissions fell by 5.8% in 2020. Although the decline was temporary, the pandemic has presented an opportunity to recognize the role of human activities in contributing to climate change and to explore ways to reduce carbon emissions in the future. Nevertheless, the IEA anticipates a rapid recovery of CO2 emissions after the outbreak is under control, and economic activity resumes. Therefore, adopting more sustainable and low-carbon energy practices is crucial to reduce carbon emissions and mitigating climate change.

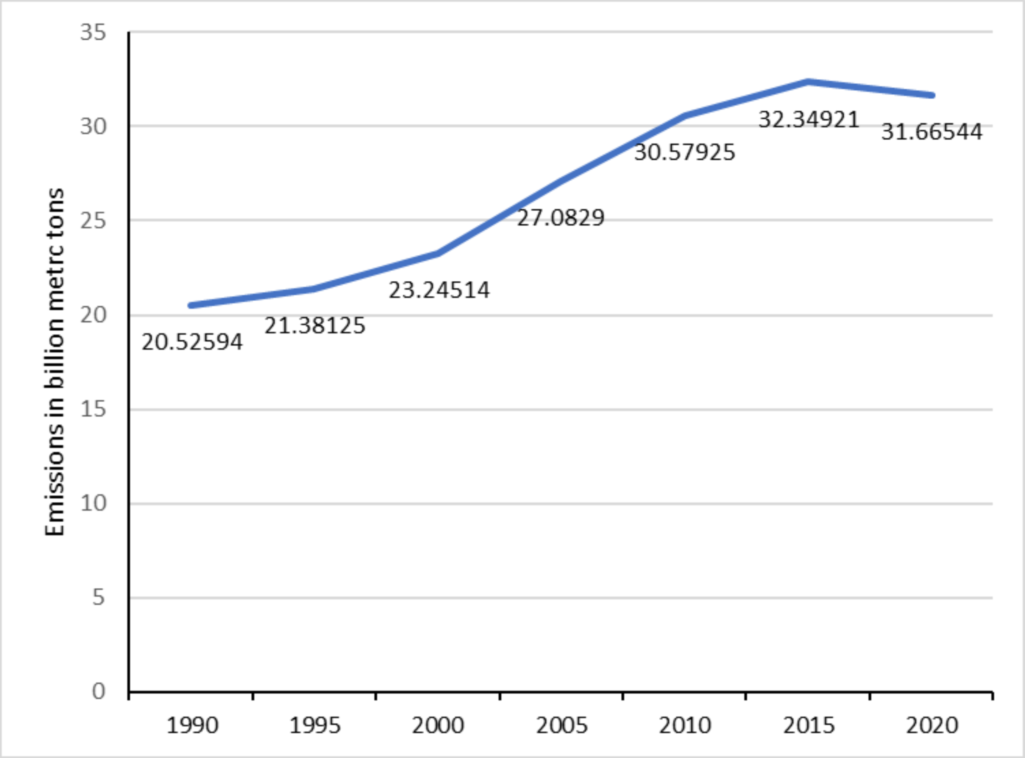


Fig. 1-2 CO2 emissions from global fossil fuel combustion and industrial processes from 1990 to 2020 [1]

According to statistics, building energy consumption accounts for about 40% of global energy consumption[2], and the proportion of building carbon dioxide emissions is as high as 36% of the total emissions[3]. Heating, ventilation, and air-conditioning (HVAC) systems account for 40% (or even higher) of commercial building energy consumption[4]. Within this context, increasing the proportion of renewable energy sources(RES) to reduce building energy consumption has become a research hotspot[5]. Besides, in the context of high building energy consumption, improving energy efficiency and utilizing renewable energy is one of the most efficient routes to achieve 'carbon neutrality' and meet the increasing demand for building energy. This approach helps facilitate the clean transformation of the energy structure, enabling more sustainable development.

Currently, building systems are rapidly evolving and transitioning towards smart grids that are more active, flexible, and intelligent, thus presenting new challenges for traditional energy management. Particularly, energy management and economical operation in high renewable energy penetration require real-time perception, Analysis, and decision-making to ensure optimal system performance. Small energy systems, consisting of multiple distributed generating units, storage systems, and local loads, are also called microgrids. A microgrid can be operated in either grid-connected or isolated island mode, allowing residential users to switch roles from sole consumers purchasing electricity from the public grid to independently producing and consuming electricity. In other words, a microgrid can integrate and utilize various forms of renewable energy to meet the local load demand. Moreover, this approach enables additional income arbitrage in the energy market, as it facilitates the exchange of power with the outside world under different pricing mechanisms.

Since photovoltaic(PV) technology has the advantages of excellent cost and convenient deployment, making it one of the most widely used RES[6]. Consequently, more and more households are opting for the household multi-energy system (HMES)[7], and its structure is shown in Fig. 1-3, which integrates electricity, natural gas, and renewable energy sources (such as photovoltaic and wind power) as energy sources. As a bidirectional grid-connected energy system, HMES can meet multiple load demands of users and sell excess renewable energy to the grid, reducing household energy payment costs[8]. Therefore, the HMES integrating RES undoubtedly has significant research value and application potential. However, due to the multiple uncertainties in the application of the HMES, the energy scheduling of the system faces significant challenges. Firstly, renewable energy production is greatly affected by environmental factors (such as weather conditions)and has strong intermittency and uncertainty. Secondly, with the development of the electricity market, many countries have adopted the real-time electricity price (RTP)[9], which is also highly uncertain due to the fluctuations of the electricity futures trading price. Third, for residential customers, the differences in living habits and rapid electrification will also lead to the uncertainty of electricity demand. The energy storage system (ESS) is an effective approach to deal with these uncertainties[10]. The ESS can not only effectively alleviate the instability caused by the fluctuation of renewable energy but also optimize the economy of the energy system according to the dynamic information of energy prices, and the grid-connected residential photovoltaic-battery system based on HMES has become Japan's fastest-growing renewable energy technology. It should be noted that although the ESS has the above advantages, it also increases the system cost and the complexity of system optimization[11].

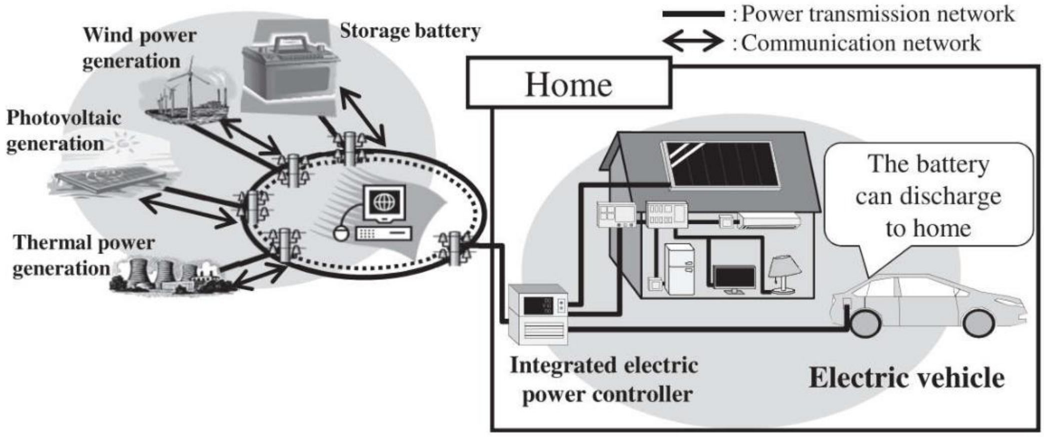


Fig 1-3 Structure of home energy management system (HEMS)[12]

To further improve the economy of ESS and the utilization of renewable energy, intelligent ESS has attracted increasing attention. Intelligent ESS enables more efficient energy management by introducing control systems that formulate optimal control strategies considering renewable energy production, electricity demand, and RTP[13,14]. Most current residential intelligent ESS systems use classical control methods, such as proportional, integral, and derivative controllers or rule-based controllers. However, these controllers cannot predict the many uncertainties in the system because they do not include domain-specific knowledge and cannot use historical data or model predictions. Therefore, traditional control methods can not achieve relatively accurate energy storage control. Model predictive control (MPC) is a popular multi-objective control method, which could formulate these uncertainties as a constrained optimization problem[15–17]. For example, the MPC controller can advance charge or discharge control of ESS based on the forecast of demand, RTP, and renewable energy production to improve renewable energy utilization and save energy costs. However, since the prediction of future data and the setting of constraints are the basis of MPC model implementation, its control effect is greatly affected by the model's prediction accuracy[18]. It can be seen that the load prediction model is the core of MPC, and the commonly used load prediction models are currently based on traditional machine learning techniques[19–21]. In recent years, with the rapid development of deep learning, deep neural networks have been increasingly used in load prediction. Deep learning(DL) is a series of new structures and methods developed based on multi-layer neural networks. DL models have significant advantages over traditional machine learning models in predicting multivariable time series problems.

However, an accurate prediction model often needs a large amount of training data and careful hyperparameter tuning. This implies that knowledge learned by the MPC model is difficult to transfer between different buildings because the historical data of each residential customer is unique and has different requirements and characteristics. Therefore, developing a standard MPC model for different residential customers is a severe challenge. Due to the increasing popularity of Machine Learning (ML) methods, Markov decision process (MDP) theory-based reinforcement learning (RL), which is another important branch of ML, provides an effective solution to solve the operational optimization problem of building energy systems. Compared with MPC, the RL model does not require complex and accurate plant modeling. It can make the RL agent interact with the environment through training data, select the action that maximizes the cumulative reward, and then make an optimal decision, which makes it possible to make a standard model for different buildings[22].

1.2 The Development of Renewable Energy in Japan

1.2.1 The Development of Renewable Energy in the world

The discovery and utilization of fossil energy sources have significantly contributed to economic and social progress. However, it has also resulted in severe environmental and climate problems. Climate change has emerged as a major non-traditional security challenge with far-reaching consequences for humanity. In addition, using fossil fuels leads to air pollution, posing a significant threat to human health and living conditions. Urgent action is needed to transform the global energy structure by reducing our reliance on fossil fuels and promoting the development and deployment of clean energy sources. This transformation must prioritize energy security and sustainable development, aiming for a win-win scenario for economic growth and environmental protection.

Fossil fuels dominate the energy mix, partly because transportation and heating are challenging to decarbonize. The high energy density and portability required for energy demand in these areas make fossil fuels more advantageous than other sources. However, with the development and innovation of technology, cleaner energy alternatives are gradually emerging. For instance, electric vehicle technology has become more mature, and battery technology has enabled longer-range electric vehicles, providing the transportation industry with more alternatives. Additionally, hydrogen fuel cell technology is maturing and offers another alternative for transportation. In the heating sector, some countries and regions have started experimenting with renewable energy sources, such as solar and geothermal. Furthermore, new clean fuels are emerging, such as biomass energy and liquefied petroleum gas.

The power system offers a more diverse energy option than transportation and heating. With the continual progress of technology and decreasing costs, the share of clean energy in the power system is gradually increasing. The diversity of clean energy options in the power system also expands, supporting the energy structure's transformation. As shown in Fig. 1-4, natural gas is the primary fuel for power generation in North America, CIS, the Middle East, and Africa. South and Central America rely on hydroelectricity for over half of their power. In Asia, coal dominates the generation mix with a share of 57% - far higher than any other region. In Europe, renewables (including biopower) have become the largest source of power generation, accounting for 23.8%

for the first time, surpassing nuclear at 21.6%[23]. Generation in Europe is relatively evenly distributed among renewables, nuclear, gas (19.6%), and hydro (16.9%). Globally, coal remains the primary fuel for power generation, though its share fell 1.3 percentage points to 35.1% in 2020, the lowest level in our data series. Renewable energy sources saw a rise to record levels last year (11.7%), and when combined with gas-fired power, their share (35.1%) matched that of coal for the first time. In Europe, renewables comprised 23.8% of power generation, surpassing nuclear energy and establishing Europe as the first region where renewables are the primary source of power generation. The chart also clearly indicates that coal remains the primary fuel for power generation worldwide, although its share decreased by 1.3 percentage points to 35.1% in 2020. Meanwhile, renewables hit a record high last year (11.7%), and when combined with the gas-fired generation, their share (35.1%) was equal to coal for the first time.

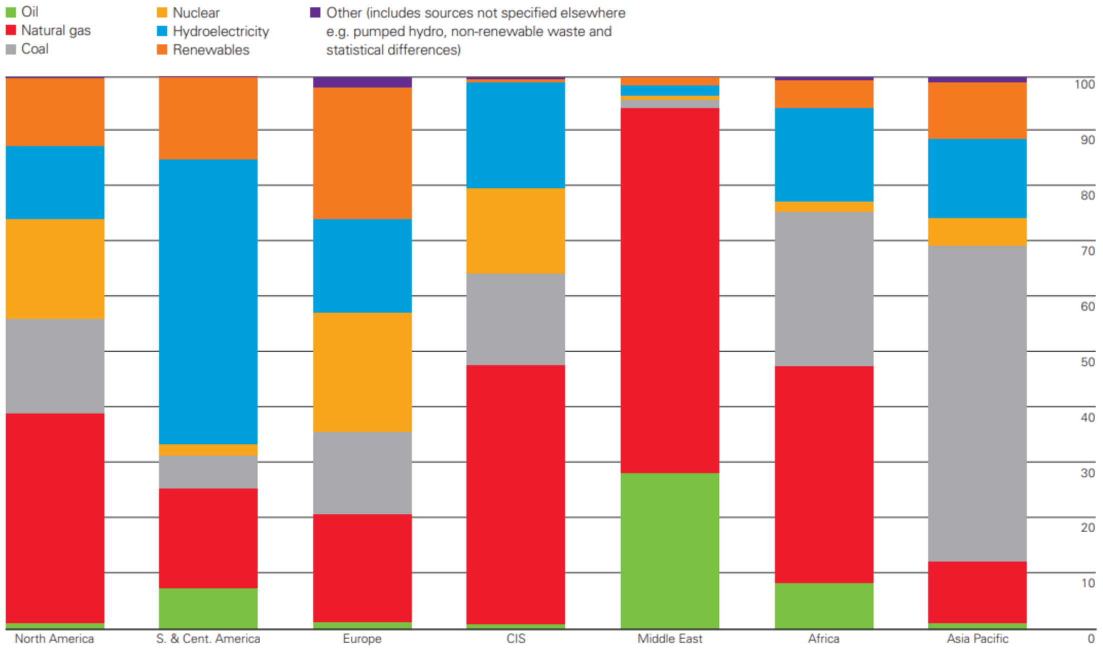


Fig.1-4 Regional electricity generation by fuel 2020[24]

By 2030, according to the IEA, the world's installed capacity for renewable energy will increase by more than 70 percent, with solar and wind power set to dominate. At the same time, the share of renewables in the global electricity market will continue to grow. The application of renewable energy in the European Union, the United States, China, India, and other countries and regions have been widely promoted and has achieved remarkable results. In the European Union, the share of renewable energy in electricity consumption reached 38% in 2020, and the European Commission has set a target of achieving a 55% renewable energy share by 2030. For example, Germany and the UK are experiencing significant growth in renewable energy generation. In Germany, 29% of electricity has been generated from renewable sources since 2015, while the figure for the UK stands at 24.5%. The German government has set an ambitious target of achieving at least 50% of its electricity from renewables by 2030, with the UK similarly requiring 45-55% to meet its 2030 carbon budget. The European Union aims for renewable energy to account for at least 27% of total final energy consumption by 2030, which modeling suggests would translate to around 45-55% of electricity[25]. In the UK, combined wind and solar capacity grew from 5.46GW in 2010 to 27.25GW by the end of 2016. Germany aims to phase out nuclear power plants by 2022 and increase its renewable energy capacity to at least 50% by 2030[26]. In the United States, renewable energy accounted for nearly 21% of the country's electricity generation in 2020. The Biden administration has set a goal of achieving a carbon-free electricity grid by 2035. California has set ambitious targets to reduce greenhouse gas emissions and increase renewable energy penetration levels, intending to raise renewable energy resources to 33% of retail electricity sales by 2020 and 50% by 2030[27]. In 2020, fossil energy power generation still accounted for a significant portion of Asia's total energy production, with renewable energy generation (REG) accounting for only 10.23%, according to BP. In China, the world's largest renewable energy market, the installed renewable energy capacity has exceeded 900 GW as of 2021. The government has set a target of reaching a 50% share of non-fossil fuels in primary energy consumption by 2030. In India, the share of renewable energy in the power sector has increased from 6% in 2014 to over 24% in 2021. The government has set a target of achieving a 450 GW renewable energy capacity by 2030. As Asia continues to face increasing energy demand, there is a need to balance economic growth with carbon emissions through a green transition that heavily relies on renewable energy development[28]. The IEA's World Energy Outlook

2021 report predicts that in a net-zero emission scenario, the Asia-Pacific region will account for 45% of the clean energy market by 2050, indicating significant growth in clean energy deployment potential.

It should be noted that both PV and wind power have seen significant growth in recent years, with solar PV growing at an annual rate of 18% and wind power at an annual rate of 9%. The projected growth trend of each renewable energy source is shown in Fig. 1-5. It can be seen that PV will be the most rapidly growing renewable energy in the future. The increase in PV and wind power adoption can be attributed to two main factors. Firstly, technological advancements and the growing production scale have led to a significant decrease in the manufacturing cost of photovoltaic cells, particularly silicon-based ones. This has resulted in historically low prices for photovoltaic power generation and improved efficiency and reliability of photovoltaic modules, making it a highly competitive energy option. Secondly, governments worldwide have provided support for clean energy, such as subsidies, tax incentives, and energy regulations, to the photovoltaic power generation industry, reducing the investment cost and increasing the return rate of return of photovoltaic power generation, attracting more investors to the field. As more countries and companies commit to reaching net-zero emissions, the demand for solar and wind energy will likely continue to grow, accelerating the transition to a cleaner, more sustainable energy system.

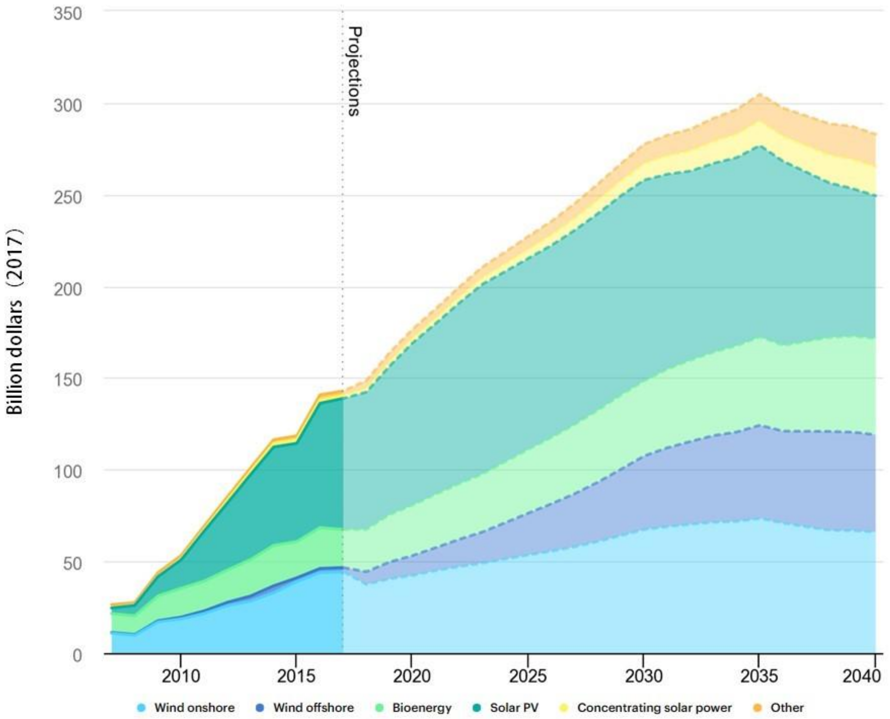


Fig. 1-5 Global renewables-based electricity support in the New Policies Scenario

1.2.2 The Development of Renewable Energy in Japan

According to recent data [29], Japan's final electricity demand in 2020 was largely dominated by the industrial sector, accounting for 35% of the total demand, followed by the commercial sector (34%) and the residential sector (29%) (as shown in Fig. 1-6). The industrial sector, in particular, depends heavily on fossil fuels, accounting for approximately 78% of its energy consumption. Meanwhile, the residential and commercial sectors dedicate a significant portion of their final energy consumption to heating, cooling, and hot water, with 57% and 45% of energy consumption, respectively, being utilized for these purposes. Consequently, it is paramount to increase the rate of renewable energy penetration and consumption in residential buildings if Japan is to realize its ambitious objective of achieving carbon neutrality by 2050.

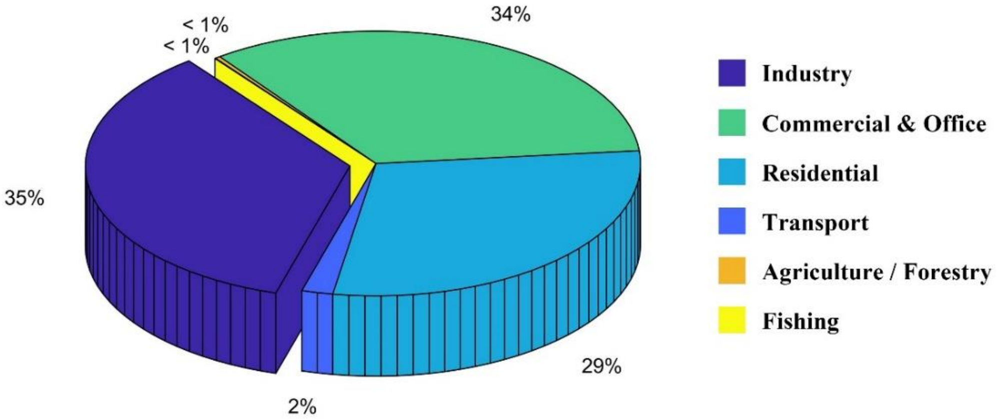


Fig. 1-6 The electricity demand by sector of Japan (Resource: IEA data, Japan Energy Outlook 2022)

To address this situation, the Japanese government has committed to achieving net-zero emissions by 2050. In response to the pledge to achieve carbon neutrality by 2050, the Ministry of Economy, Trade and Industry (METI) has formulated a green growth strategy [7] as the policy framework for this transformation. As per this strategy, Japan aims to derive 50-60% of its electricity from renewable sources by 2050, primarily through offshore wind. The remaining energy needs are anticipated to be met by combining hydrogen (10%), nuclear power, and fossil fuel power plants with carbon capture and storage (CCS), accounting for 30%-40% of the total. It is worth noting that Japan relies heavily on imported fossil and nuclear fuels due to its limited fossil fuel reserves. Promoting domestic renewable energy sources, such as solar and wind power, could help reduce the country's dependence on energy imports. Although the cost of deploying solar photovoltaic (PV) and wind energy in Japan is presently higher than in other countries, as these technologies become more widely adopted globally, costs are likely to converge over time as market competition increases and experience is gained. Consequently, it is anticipated that the cost of solar PV and wind power in Japan will decrease significantly in the future.

Japan is the world's fifth-largest emitter of greenhouse gases (GHG) and possesses a low energy self-sufficiency rate, primarily due to the scarcity of conventional energy sources such as coal, oil, and natural gas. Presently, renewable sources account for 21% of Japan's electricity production, while nuclear energy accounts for 3.95% and fossil fuels for 75.18%, as shown in Fig. 1-3. Since the Fukushima nuclear disaster in 2011, Japan has experienced a marked reduction in nuclear power generation, as depicted in Fig.1-7. However, As reported on Japan's Agency of Natural Resources and Energy website, Japan's energy self-sufficiency rate was merely 9.6% as of 2017 due to heavy reliance on foreign countries for more than 80% of its energy supply. In the same year, Japan's proportion of renewable energy accounted for approximately 16% of its total energy consumption. Comparatively, other countries such as Canada, Italy, Germany, and Spain had significantly higher proportions of renewable energy overseas, with shares of 65.7%, 35.6%, 33.6%, and 32.4%, respectively[30]. These figures indicate that Japan's current energy self-sufficiency rate and renewable energy development still have much room for improvement compared to leading countries.

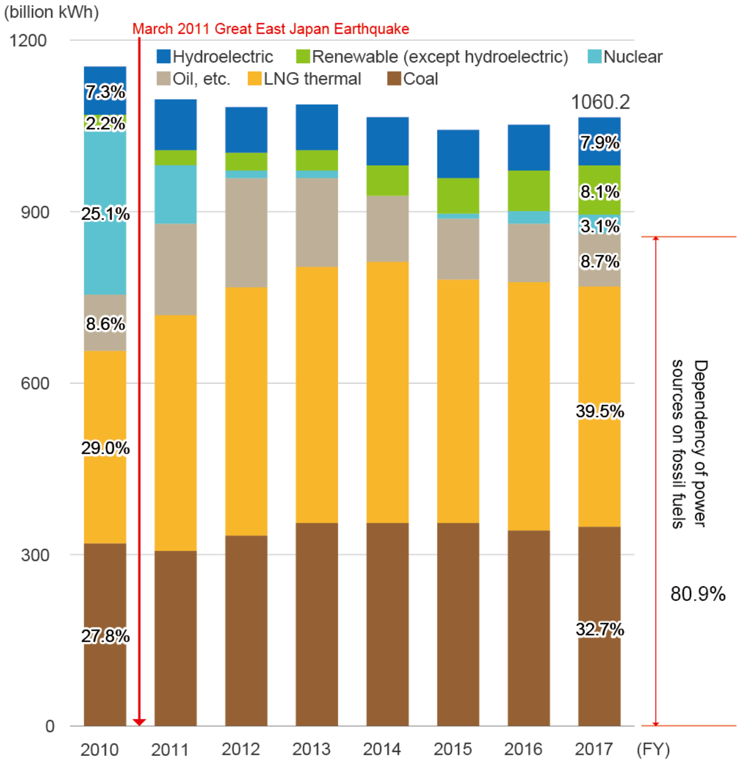


Fig. 1-7 Changes in the Japanese composition of power sources [31]

To address the strain on the supply-demand balance of the grid during peak periods, the Japanese government undertook significant political and technical efforts to compensate for the loss of nuclear power after the Great East Japan earthquake. This was mainly accomplished through a substantial increase in thermal power generation. To effectively implement these principles, renewable energy generation (RES-E) must achieve a 20-22% share of power generation, thereby attaining the status of a "major source of electricity" by the year 2030, as prescribed by the Strategic Energy Plan (SEP) released by the Japanese government in 2018[32]. As a result, promoting the integration of renewable energy sources in their electricity supply mix has become a top priority for the government. Fig.1-8 illustrates that PV power generation in Japan has grown rapidly since 2012, following the implementation of a series of incentive policies by the Japanese government aimed at promoting the use of renewable energy sources[33]. By analyzing the development of renewable energy in Japan between 2010 and 2017, it is evident that the installed capacity of renewable energy increased at an average annual rate of 9% during 2010-2012. However, since implementing the Feed-in Tariff (FIT) program in 2012, renewable energy has experienced a significant surge with an annual growth rate of 22%. Solar photovoltaic installations have been the main contributor to the increase in installed capacity. By 2017, it is expected that more than 6,000 gigawatts of renewable energy capacity will be installed in Japan. Therefore, several studies have been conducted to evaluate the long-term energy mix and basic management of sustainable electricity supply systems from both policy and technical perspectives. These studies focus on maximizing the penetration level of renewable energy, supporting grid integration and dispatching, and reducing the power sector's CO2 intensity.

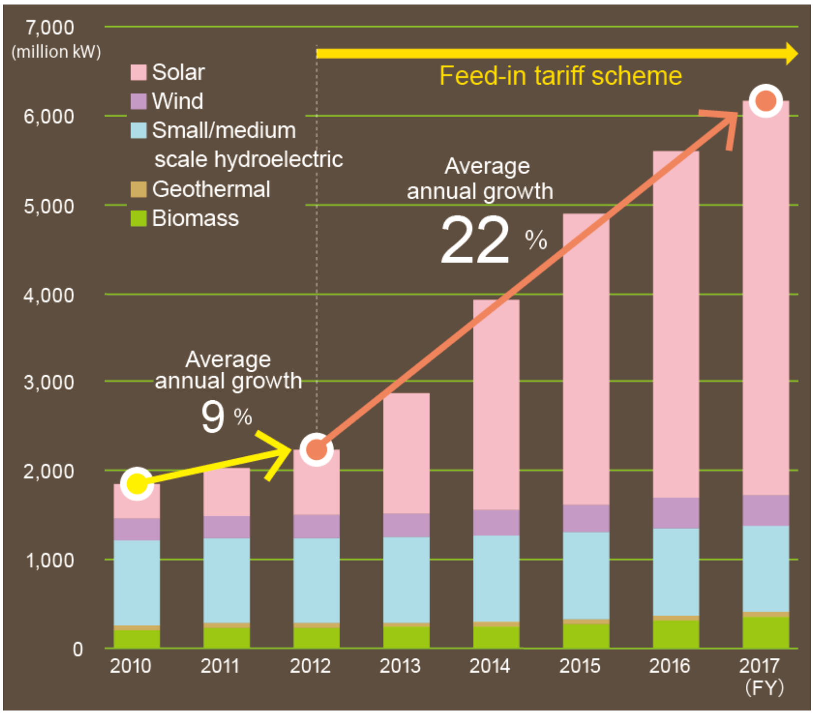


Fig. 1-8 Changes in installed capacity resulting from renewable energy and other factors (Excluding large-scale hydroelectric power)[34]

Fig. 1-9 presents the Kyushu, Shikoku, and Hokkaido islands' detailed regional power supply by shares and the existing fraction of renewable power plants in 2020. Table 1 displays the selected objectives' regional peak hourly load and renewable power capacities in the 2021 scenario. The available data indicates that PV (photovoltaic) is the dominant renewable energy resource in the studied regions. The Kyushu region has the highest integrated PV capacity compared to Shikoku and Hokkaido islands. As of March 2022, the cumulative PV capacity in the Kyushu region was over 10.9 GW. However, the significant installed solar capacity significantly reduces the output of thermal power plants and makes it challenging to integrate intermittent power further. The power utility pays rising attention to handling the rising load volatility.

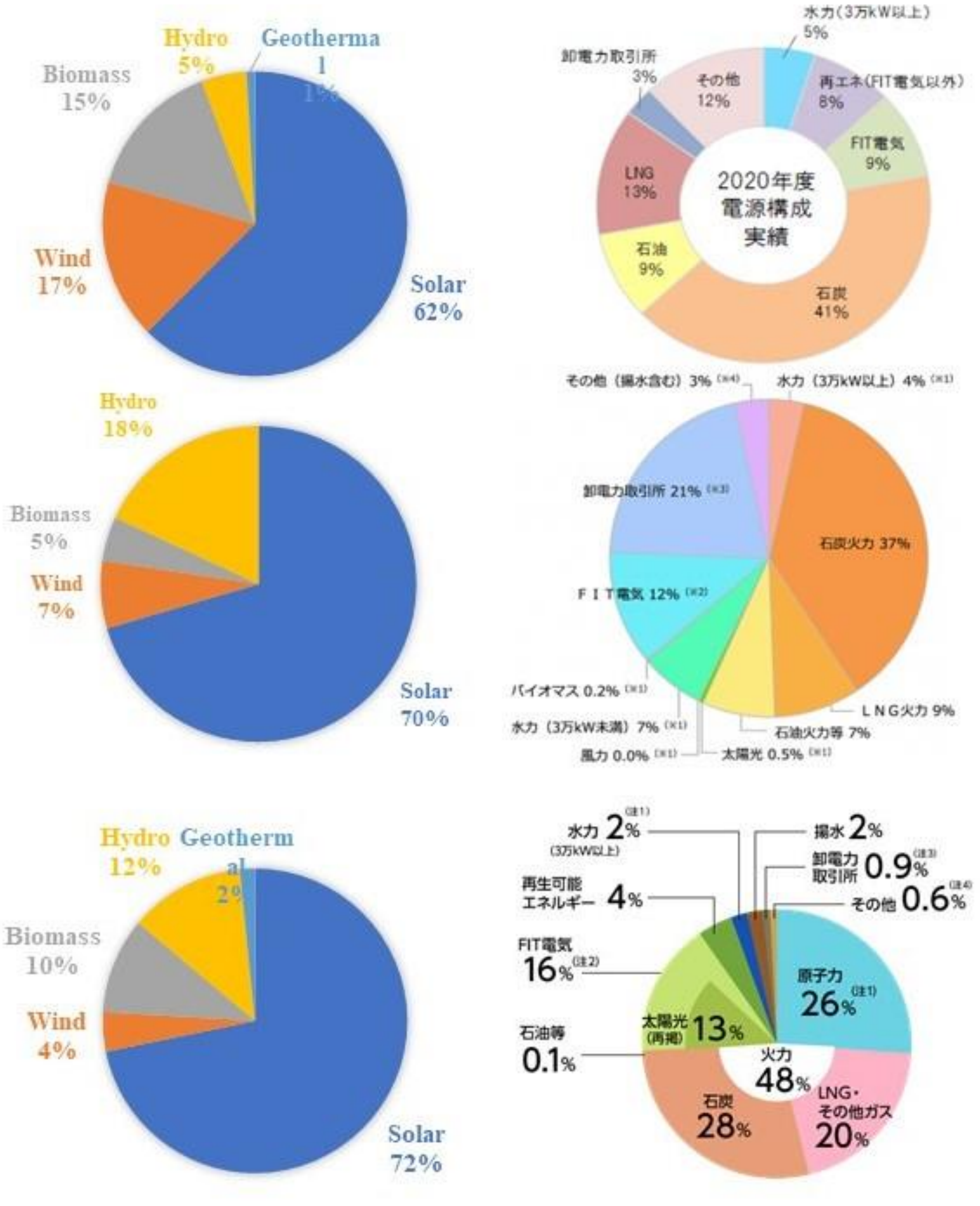


Fig. 1-9 Grid electricity generation mix and renewable power plant capacity in Kyushu, Shikoku, and Hokkaido regions

Table 1 Regional grid peak hourly load and renewable power capacity in 2021, the unit is MW

Variables	Peak load	Solar	Wind	Biomass	Hydro	Geothermal
Kyushu	15592	10850	630	1540	1860	240
Shikoku	5030	3270	320	210	840	0
Hokkaido	5041	2140	580	510	1650	30

Ref[35] examined the potential impact of measures to increase renewable energy generation in Japan while maintaining a reliable energy system. Results show that a merit order dispatch could increase renewable generation by 1.5% while replacing nuclear power with renewables could result in a renewable share of 58.2%. The authors recommend changes to Japan's next Strategic Energy Plan based on these findings. Ref[36] created a high-resolution renewable energy potential map, evaluated the interlinkages with Sustainable Development Goals (SDGs), and discussed issues related to implementing renewable energy systems in Japan. Ref [37] provided an insightful analysis of the impact of green bonds issued in Japan and energy price fluctuations on wind, solar, and hydro energy consumption between 1990 and 2020. The findings of this study highlight the positive long-term impact of green bond issuance on energy prices, with solar and hydro energy consumption being particularly significant beneficiaries. Ref[38] examined the potential of large-scale building-integrated photovoltaic (BIPV) modules for decarbonizing urban building stock in Tokyo, Japan. A model for estimating the hourly PV potential of building surfaces on a regional scale was developed and applied to the commercial building stock. Results indicate that exploiting the PV potential of building facades could satisfy 15%-48% of the annual electricity demand of the building stock in 2050. Ref[39] conducted an economic assessment of residential PV systems integrated with EVs (V2H) in Japan towards 2030. The results show that the PV + EV system is already cost-competitive with grid electricity and a gasoline vehicle in 2018 and could reduce annual energy costs by up to 68% and decarbonize the household energy system by 92% by 2030. Ref [40]proposed a novel approach to enhance the capacity for distributed power generation by combining the design of low-voltage grid systems with subsidies for photovoltaic systems. A case study on rooftop photovoltaic generation in a Japanese town demonstrates the synergistic effect of integrating these two planning issues in facilitating the diffusion of photovoltaic systems. Ref[41] focused on the spread of residential photovoltaic (PV) systems and analyzed their social demand as an external business environment. The study found that electricity and energy conservation awareness has increased, and people have become more interested in renewable energy after the 2011 disaster. Ref[42] analyzed factors affecting the decision-making process of purchasing solar photovoltaic systems in Japan. The survey found that consumers take about four months to make a purchase decision, and Feed-in tariffs correlate highly with purchasing motivation.

In contrast, capital subsidy programs have little impact or even delayed impacts on purchasing timing. Ref[43] examined the potential of PV resources in Japan's power system and analyzed the impact of PV integration on the grid using a high-time-resolution optimal power generation mix model. The simulation results show that while Japan has immense potential for PV capacity, the growth of PV integration slows down when installed PV capacity exceeds the scale of peak demand.

1.3 The Development of Building Energy Management Systems (BEMS)

A distributed energy system is a comprehensive system based on cascade energy utilization. It is usually placed at the user's end to provide cold and hot electricity load. This approach has several advantages, such as increased energy efficiency, reduced energy consumption and emissions, and decreased pressure on the power grid. By integrating wind energy, PV generation, and geothermal energy sources into a traditional distributed energy system, the system's overall performance can be further improved. The adoption of renewable energy in distributed energy systems provides an effective solution to reduce overall energy consumption and emissions. In recent years, integrating distributed energy systems into microgrids has been recognized as an effective approach to enhance the renewable energy consumption capability of the grid. Consequently, various energy management and operation control methods have been developed and deployed, significantly increasing the flexibility of microgrid control. Common BEMS operational optimization schemes include day-ahead optimization, rolling day-to-day optimization, and real-time optimization.

The most conventional building control method is rule-based feedback control, which typically involves two steps. First, pre-determined schedules are used to select setpoints, such as temperature setpoints. Then, Proportional-Integral-Derivative (PID) control techniques are utilized to track these setpoints. The rule-based control (RBC) method is one of the earliest developed energy control strategies, which can intelligently realize the system operation control based on prior knowledge[44]. Authors in[45] examine the feasibility of implementing a rule-based energy management system for a microgrid platform in operation. In literature[46], a rule-based real-time controller is combined with optimization technology based on dynamic programming to manage the microgrid, focusing on minimizing energy costs. The study found that compared to the cost of energy provided by the grid alone, users could reduce their energy costs by 85% daily. Mauricio et al. [47] proposed a rule-based method for the real-time optimization of energy management system sequences. This method has been verified through testing on a real-time hardware-in-the-loop (HIL) platform to demonstrate its performance. However, these solutions typically rely on the current running state of the system and simplify the solution conditions as much as possible to meet the requirements of real-time computing. The decisions made by these methods are often short-sighted and fail to fully consider the opportunity cost, making it challenging for RBC to achieve the optimal solution for the long-term energy management of the system.

To further enhance control accuracy, control methods based on linear programming (LP) have been extensively utilized in energy system management. The basic procedure of LP involves constructing an energy system model based on physical constraints pertinent to the system's operation, followed by utilizing LP technology to acquire the optimal solution by the predefined optimization objectives. The ultimate objective is to attain optimal results for designing and operating the energy system. Authors in[48] developed a model using a mixed-integer programming (MILP) algorithm to determine the optimal size of a PV cell system with the lowest total investment operating cost. This model was implemented in a mountainous building in northern Italy, aiming to replace traditional fossil fuels. Truong Dinh et al.[49] proposed a supervised learning strategy for home energy management systems (HEMS) based on MILP that can proficiently regulate ESS and RES, intending to reduce energy costs. Simian Pang et al.[50] introduced a decomposition algorithm based on mixed-integer linear programming (MILP) to optimize the power scheduling for efficient tracking performance and realize cooperative scheduling of multiple heat loads. The effectiveness and feasibility of the proposed method in enhancing comprehensive benefits were demonstrated using heat load data from a specific region. The authors in [51]proposed a novel Energy Management System (EMS) for battery storage systems, which utilizes a Mixed-Integer Linear Programming (MILP) optimization algorithm implemented in GAMS, with CPLEX as the solver. The proposed EMS strategy guarantees a sustained reduction (around 47%) in the number of long-term (10-year) battery replacements, leading to substantial cost savings.

The control methods based on LP are simple and effective, but they are not optimal for two main reasons. Firstly, these strategies do not consider predictive information, resulting in sub-optimal performance. Secondly, the control sequence, including parameters in PID controllers, is fixed and pre-determined, which means it is not customized for specific building and climatic conditions. As a result, these strategies may not be flexible enough to adapt to changing conditions. The introduction of load prediction into traditional LP methods to improve building control performance constitutes the Model Predictive Control (MPC) method. The core part of the MPC is the load prediction model, which forecasts future load demand within a significantly extended optimization window. Upon the prediction, the MPC controller computes a set of optimal actions and executes them in the next time step, and the detailed process is shown in Fig 1-10. In literature[52], a two-stage robust stochastic programming model was developed for commercial microgrids, which can be adjusted in real-time to minimize the cost of power imbalance while maximizing expected profits in the day-ahead market. In literature[52], a closed-loop distributed model was designed to optimize the energy regulation behavior of multiple participants, thereby reducing potential variations in intra-day economic scheduling. Based on the forecast of wind power generation and electricity price, J.J. Yang et al. proposed a charge-discharge control strategy for energy storage equipment based on the data drive, which realized the maximization of income of energy storage equipment[53].In literature[54], a two-layer energy scheduling strategy based on model predictive control was proposed, which improves the robustness of prediction errors by solving the boundary value problem and adjusting the optimal results of the previous layer. In the study[55], a two-layer coordinated energy management control method was proposed, which generates an economic operation plan during the day-ahead scheduling stage and minimizes the cost of power adjustment by tracking the day-ahead scheduling scheme, thus addressing the deterioration of the optimization effect caused by prediction errors during the day-ahead scheduling process. Sahar Rahim et al.[56] proposed a home energy management system based on the genetic algorithm (GA), which realized the optimization of cost-effectiveness and load peak considering constraints of user comfort satisfaction. Authors in [57] proposed a multi-objective predictive energy management strategy based on the machine learning technique for residential grid-connected hybrid energy. Results showed that electricity costs and carbon dioxide emissions were significantly reduced. A hierarchical two-layer home energy management system controller was presented by Elkazaz et al.[58], which could optimize household energy usage using a mixed-integer linear programming optimization.

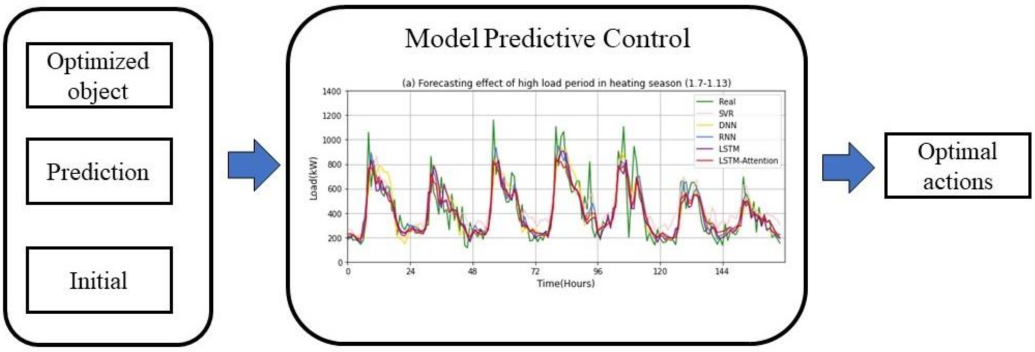


Fig. 1-10 The workflow of MPC

Currently, two limitations are hindering the progress of MPC. Firstly, the stochastic and dynamic nature of energy systems in practical applications tends to augment their complexity, thereby exacerbating the difficulty of energy system modeling. It means every building and its energy systems are unique, so it is difficult to generalize a standard building energy model for various buildings. Secondly, the precision of MPC control relies on the accuracy of the predictive model. However, the randomness and instability of renewable energy sources present significant challenges in developing highly accurate prediction models.

1.4 The Application of Machine Learning in BEMS

In recent years, the advancement of Machine Learning (ML) technology has introduced novel ideas to tackle the issues mentioned above. ML is a model methodology that emulates the human learning process based on statistical principles and enables computers to perform relevant data analysis tasks. ML algorithms can learn implicit knowledge in data directly, without dependence on pre-determined equations or program order. Additionally, these algorithms can adaptively improve their performance as data availability increases. In the last decade, ML has emerged as a specialized field and has made promising contributions in various research and engineering domains, including data mining [59], medical imaging[60], communications [61], multimedia [62], earth sciences[63], remote sensing classification[64], real-time target tracking [65] among others. The rapid development and widespread attention to ML technologies have led to their classification into the following major categories, which are shown in Fig. 1-11:

1) Supervised learning: In supervised learning, the algorithm is trained on labeled data, which means corresponding output labels accompany the input data. The algorithm learns to map the input data to the correct output by adjusting its internal parameters based on the feedback provided by the labeled data. The feedback in supervised learning is immediate and explicit, as the algorithm can compare its output with the known labels and adjust its parameters to improve its accuracy. The ultimate goal of supervised learning is to create a model that can accurately predict output labels for new input data.

2) Unsupervised learning: In unsupervised learning, the algorithm is trained on unlabeled data, which means there are no known output labels. Unsupervised learning aims to find

hidden patterns or structures in the input data. Since there is no labeled data to provide feedback, the algorithm must rely on other techniques to uncover meaningful patterns in the data, such as clustering, dimensionality reduction, or density estimation. Unsupervised learning is often used for exploratory data analysis, data visualization, or anomaly detection.

3) Reinforcement learning (RL): RL is a type of ML in which an agent learns to take actions in an environment to maximize a reward signal. The agent interacts with the environment by acting and receiving feedback through rewards or punishments. The agent's goal is to learn a policy, which is a mapping from states to actions, that maximizes the expected cumulative reward over time. RL can be used for various tasks, such as game playing, robotics, and autonomous driving.

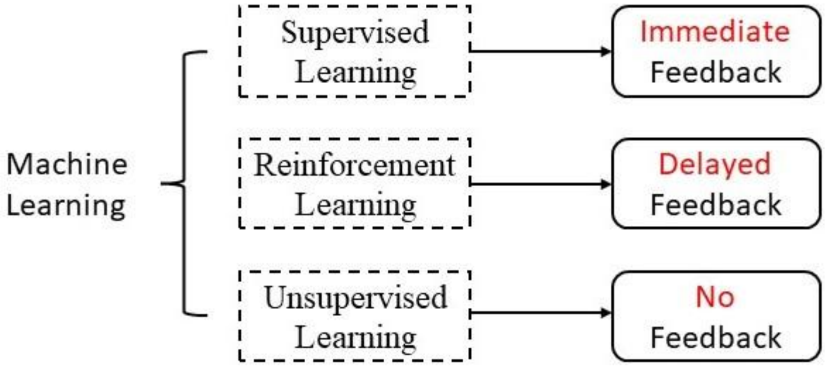


Fig. 1-11 Three types of machine learning problems[66]

The three categories of machine learning problems differ in the feedback the agent/algorithm receives after making a decision/prediction. In supervised learning, the agent immediately knows how accurate its prediction is compared to the ground truth provided by labeled data. This information is then used to update and improve the predictor. In contrast, unsupervised learning involves an unlabeled dataset; therefore, no feedback is provided. RL lies in between these two scenarios as it receives delayed feedback.

Considering the characteristics described above, the following chapters will explore two ML technologies that are most suitable for integrating with energy management systems: load forecasting technology based on supervised learning and control technology based on reinforcement learning.

1.4.1 Load prediction based on Supervised Learning

As previously mentioned, the load prediction model serves as the central component of the MPC controller. The general process for building energy load prediction methods is shown in Fig. 1-12, which typically involves four main steps: data transformation, feature selection, optimization of model parameters, and model training. In the first step of data transformation, the raw historical operation data should be filled with the missing data and normalized to improve the accuracy of the prediction model. Next, feature extraction is performed to identify the most relevant variables affecting the target energy load. These features are then used for model training. The third step, optimization of model parameters, involves optimizing the model's hyper-parameters to obtain the optimal model structure. Finally, the model's coefficients are tuned automatically to obtain the final building energy load prediction model.

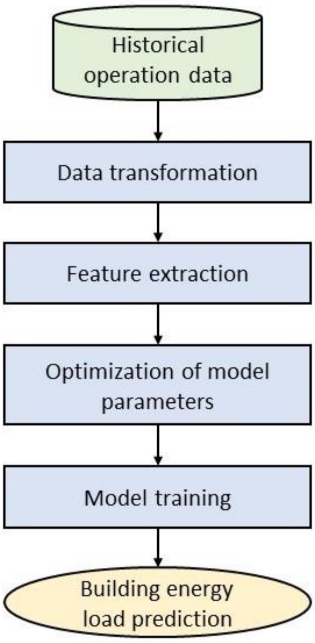

Historical operation data
Data transformation
Feature extraction
Optimization of model parameters
Model training
Building energy load prediction

Fig. 1-12 A general process of building energy load prediction methods[67]

Load prediction can be divided into ultra-short-term, short-term, medium-term, and long-term according to different purposes. The long-term load forecast could predict energy demand for up to several years ahead; The medium-term load forecast refers to the load prediction within several weeks in the future; The short-term load forecast could predict energy demand for the next few hours up to a few days ahead; The ultra-short-term load forecast refers to the load prediction within one hour in the future[44,45]. Currently, the prevailing load forecasting methods are short-term and ultra-short-term load prediction.

Since the behavior of energy demand can be expressed as time-series data with a certain period, the prediction model can learn the load mode of the system from the time-series data and use these modes to make load predictions. After the prediction range is determined, the appropriate algorithm should be selected. The algorithms in the field of time series prediction can be divided into two categories: traditional machine learning algorithms and deep learning algorithms. Traditional machine-learning algorithms are mostly based on statistical models. Current popular algorithms include Autoregressive Moving Average (ARIMA)[70], support vector machine (SVM)[71] , Regression Tree[72],Random Forest[73], and artificial neural networks (ANN)[74,75]. SVR has been widely utilized for building electricity load prediction. For instance, Dong et al. employed a radial basis function-based SVR to forecast the electricity load of commercial buildings in Singapore[76]. Authors in [77]proposed an SVR-based method for predicting the electricity load of public buildings. According to their results, SVR outperformed artificial neural networks (ANN) regarding prediction accuracy. Ref[78] utilized four years of operational data from four commercial buildings to evaluate the performance of SVR in building energy load prediction. The results demonstrated that SVR yielded highly accurate predictions. Ref[79] introduced the support vector regression algorithm for predicting building energy consumption time series, often non-linear and non-stationary. The authors in[78] used three models (MLR, MLP, and SVR) to predict the non-residential electricity load. They tested it using a real case study from the University of Girona. The results show that the

SVR method has high accuracy and low calculation cost.

In recent years, with the rapid development of deep learning, the deep neural network has been more and more applied in load prediction. Deep learning is a series of new structures and new methods evolved based on multi-layer neural networks[79]. Deep learning models have obvious advantages over traditional machine learning models in predicting multivariable time series problems. In the real world, time series prediction presents multiple challenges, such as having multiple input variables, predicting multiple time steps, and performing the same type of prediction for multiple actual observation stations [80]. In particular, a deep learning model can support any number but a fixed number of inputs and outputs. Multivariable time series have multiple time-varying variables, each depending on its past value and other variables. These characteristics are correlated; in this case, multiple variables must be considered to give the best-predicted energy consumption.

The most basic Deep learning model is Deep Neural Networks (DNN), also known as multi-layer perceptron (MLP). DNN has more hidden layers than ordinary artificial neural networks, allowing it to learn complex patterns. Ref [82]presented a load forecasting model based on a DNN that can be easily integrated into a Building Management System or real-time monitoring system. The authors in [83] investigated the potential of DNN in predicting short-term building cooling load profiles. In contrast to conventional physical methods, DNNs can effectively identify nonlinear and complex patterns in big data and offer greater flexibility in model development. Deb proposed a DNN-based model for predicting the daily cooling load of buildings [84]. Waseem Ahmad compared the performance of two widely used energy forecasting models, DNN and Random Forest (RF), in predicting the hourly HVAC energy consumption of a hotel in Madrid. The study found that DNNs performed slightly better, while RF had an advantage in handling complex multidimensional data and sorting variables[85]. Authors in [86] combined the rough set theory and DNN to predict the air conditioning load. The rough set theory was used to identify load-related factors, which were input into the DNN for prediction. Experimental results showed that the RS-DNN model outperformed the single DNN and AMIMA models, with a relative error of less than 4%. Massana proposed a DNN model-based method for predicting short-term electrical loads in non-residential buildings[78]. Zhihan Lv et al. proposed a layered DAE support vector machine (SDAE-SVM) model based on a three-layer neural network and achieved good prediction results[87]. However, the DNN model cannot retain time-series information. It can only be predicted according to the current input and output values. Besides, it cannot learn the time dependence of data, which limits the accuracy of its prediction time series.

Recursive neural network (RNN), as a special deep neural network, can retain and consider the time variation of time series in the training process[88], which makes it very suitable for time series data with periodicity. Huai Su[89] proposed a hybrid method for forecasting gas consumption hours in advance, integrating Wavelet Transform, RNN, and Genetic Algorithm. The results show that the improved RNN model has excellent prediction accuracy. Similarly, Ref[90] proposed a new hybrid model for wind speed forecasting, combining empirical mode decomposition with RNN and linear regression to improve accuracy and stability compared to single RNN models or those with decomposition preprocessing. The authors in [91] propose a method that utilizes Recurrent Neural Networks (RNNs) for accurately simulating loads in distribution systems with high renewable energy penetration and widespread use of electronics. The time series of energy demand is always influenced by environmental variables and human living habits and has a strong periodicity. However, due to the problems of gradient explosion and gradient disappearance in the RNN network, the long-term dependence in time series cannot be retained, which limits the prediction accuracy of the RNN network.

The long-short-term memory (LSTM) network adds a series of multi-threshold gates based on the RNN network, which can deal with a long-term dependency relationship to a certain extent. LSTM networks were first used in natural language processing[92], machine translation[93] and video recognition[94], etc. In recent years, LSTM networks have attracted more and more attention in load prediction. Delu Wang[95] proposed a comprehensive power demand prediction model called CNN-LSTM based on multi-modal information fusion. The results show that the fusion of text and time series data improves prediction performance. The authors of [96] used the LSTM network to build a regional-scale building energy consumption prediction model. Sendra-Arranz proposed various multi-step prediction models based on the LSTM network to predict residential HVAC consumption[97]. Zhe Wang proposed a new method for predicting plug loads using the LSTM network. The data collected from an entire office building in Berkeley, California, verified the prediction accuracy of this method to be better than that of the traditional machine learning algorithm[98,99]. The authors in [100] proposed a novel model combining LSTM and Temporal Convolutional Network (TCN) models for accurate PV power forecasting. The results demonstrated that the LSTM-TCN model outperformed all compared models for PV power forecasting at different time horizons, ranging from 2 to 7 steps. Wei Junqiang proposed a novel method for ultra-short-term wind power prediction combining Maximum Information Coefficient (MIC) with Multi-Task Learning (MTL) and LSTM networks. The results demonstrate that the LSTM-based prediction network achieved high accuracy in this case[101]. Ref[102] proposed an improved LSTM model for predicting PV power, which used the support vector regression (SVR) to analyze the initial time node and reduce the fluctuation error of predicted values. Results show that this model outperforms seven other models when predicting at different intervals.

Since the LSTM network adopts the code-decoding framework, the limitations of the code-decoding framework will lead to information loss when processing long time series. Bahdanau first introduced the attention mechanism into the code-decoding framework in 2014[103]. The attention mechanism can quantitatively assign a weight to each specific time step in the time series feature, which improves the attention distraction defect of traditional LSTM[104]. Many researchers have started experiments in other load fields and achieved some results[105–107]. Such as Heidari used an attention-based LSTM (A-LSTM) model to predict the load of the solar-assisted hot water system and proved that the prediction accuracy of the A-LSTM model was better than that of the traditional LSTM model[108]. Ref[109] presented a Deep LSTM-based Stacked Autoencoder (DLSTM-SAE) model, which incorporates a multi-stage Attention Mechanism (MSAM) for short-term load forecasting. The model outperforms existing models in offline and online load forecasting on actual energy market data. Jince Li proposed an improved attention-based LSTM (A-LSTM) model for multivariate time series of predictions of two process industry cases[110]. Tongguang Yang proposed an attention-based LSTM model to predict the day-ahead PV power output[111]. All these cases show that the A-LSTM model has significant advantages over the traditional LSTM model in dealing with time series problems.

1.4.2 Control techniques based on Reinforcement Learning

Due to the above advantages of the RL method, research on applying RL to the operation optimization of building energy systems has increased significantly over the past decade[66]. As shown in Fig. 1-13, there are five major components in RL settings: the controller, states, actions, rewards, and the environment. Variations in these components, such as using different algorithms or states to represent the environment, can lead to different RL implementations and result in different control performances.

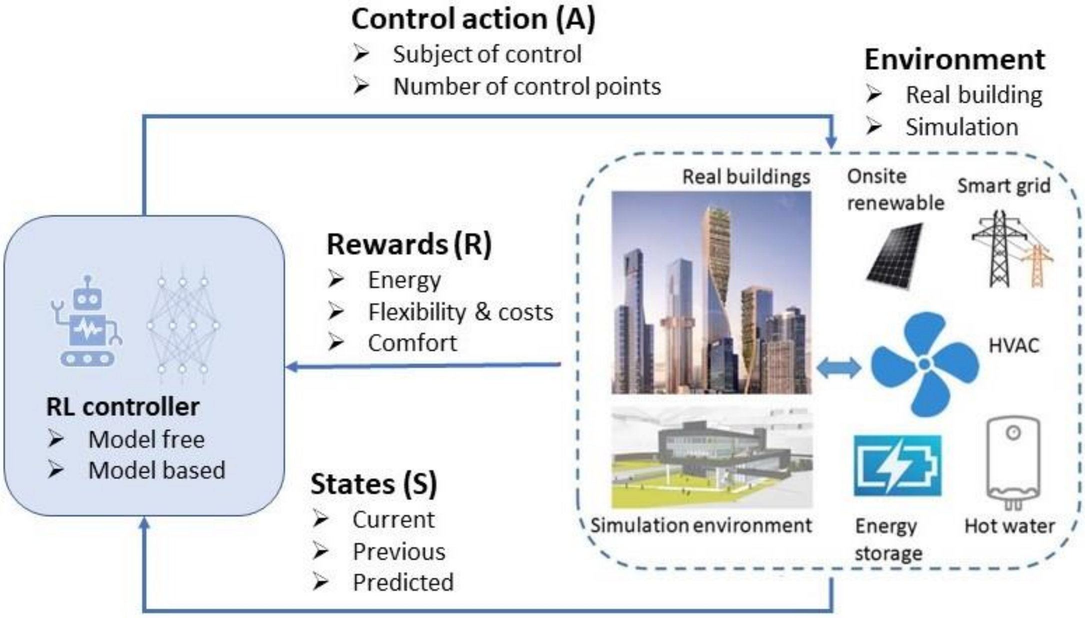


Fig. 1-13 Reinforcement learning for building controls

As the most classical off-policy reinforcement Learning algorithm, the Q-learning algorithm has been widely used in the energy field due to its model-free and easy evaluation strategies, successively implemented for the wind power system[112], electric vehicle[113] and building flexible load control[114]. For example, Waldemar et al. proposed a Q-learning-based method to optimize the non-stationary environment and non-linear storage characteristics of the storage-integrated PV system and verified through simulation experiments that it could reduce the cost of energy purchased on a real-time basis to a minimum[115]. Authors in[116] proposed a model-free Q-learning method that makes optimal control decisions for HVAC and window systems to minimize both energy consumption and thermal discomfort. The work in [117]uses the Q-learning method to optimize a residential RES and reduce energy consumption by improving the utilization rate of renewable energy.

Shunian Qiu et al. [118]proposed to utilize Q-learning for building cooling water systems. The experiment conducted involved a three-month simulation and comparison of controllers using four different optimization methods. The results indicated that the RL controller utilizing Q-learning algorithm could save 11% energy for the system during the first cooling season, compared to the basic controller. Furthermore, the performance was superior to local feedback control (7%) but inferior to the model-based controller (14%). Despite this, the value-based RL control method requires less prior knowledge and precise sensors, allowing the control method to achieve the desired effect of reducing energy consumption, even when accurate models are unavailable. In a similar vein, Ding Zhiliang et al. [119]applied the Q-learning algorithm to the air source heat pump combined electric auxiliary heat system. The simulation results demonstrated that the RL-based operation method of the air conditioning system could effectively reduce the operating costs of the building while meeting the load demand. Additionally, the RL controller responded faster than the MPC method. Additionally, Yuan et al. [120]applied the RL algorithm to the operation optimization of air conditioning systems. The study proposed a modelless control strategy based on reinforcement learning, combining the RBC algorithm with the Q-learning algorithm. The variable air volume air conditioning system of a one-story office building was used as the research object, and the RBC controller and PID controller were compared. The results revealed that the RL controller performed optimally concerning comfort and energy consumption of the air conditioning system when the air was supplied in a single zone. The total energy consumption of the system was reduced by 7.7% and 4.7%, respectively, compared with the RBC and PID strategies. However, the Q-learning method records the optimization knowledge using the Q-value table. When the system's state or action space is too large, it will lead to the curse of dimensionality, which limits the application of Q-learning methods.

With the development of deep learning(DL) technology, deep RL (DRL) has been proposed to solve the above problems. Combining the powerful non-linear fitting ability of deep neural networks with the excellent decision-making ability of RL, the DRL can overcome some previously tricky issues, such as decision problems in continuous action spaces. Harrold et al.[121]adopted the Rainbow

Deep Q-Networks(DQN) method to control batteries in a microgrid for arbitrage. Experimental results show that this method is superior to the actor-critic and linear programming methods, which could effectively carry out arbitrage according to demand, PV generation, and RTP. Authors in [122]proposed a DRL method based on the deep deterministic policy gradient (DDPG) that can minimize the energy cost of smart home energy systems via controlling Heating, Ventilation, and Air Conditioning (HVAC) and ESS. Yang presented a deep RL control strategy combining the double deep Q-networks (DDQN) and prioritized experience replay mechanism, which could optimize the control of the ventilation and heating/cooling systems [123]. Yuan Gao[124] et al. adopted the deep reinforcement learning(DLR) method to optimize a renewable building energy system and the safety of its battery. Experimental results show that the proposed RL design can better achieve these two optimization goals under ordinary and extreme conditions. Authors in[125] proposed a DRL-based scheduling strategy for household multi-energy systems to minimize energy costs while maintaining thermal comfort. Li et al.[126]proposed an end-to-end cooling control algorithm (CCA) based on the DDPG. The results show that the proposed CCA can achieve up to 11% cooling cost reduction on the EnergyPlus simulation platform compared with a manually configured baseline control algorithm. Mocanu et al.[127]compared the operation optimization effects of DQN and DPG algorithms on building energy systems. The experimental results show that the DPG with continuous action space is superior to the DQN with discrete action space, which could reduce the building operation cost by 27.4% and the peak load by 26.3%. Y Du et al.[128] adopted the DDPG algorithm to generate the optimal HVAC control strategy with the minimum energy consumption cost while maintaining the users' comfort. The simulation results show that compared with DQN, the control strategy based on DDPG can reduce the energy consumption cost by 15% and the comfort violation by 79%.

Training RL controllers require much data and time as a data-driven approach. Yang et al.[129] had proposed that three years of training data was sufficient to ensure that the RL controller was superior to the rule-based controller. However, high-quality data for residence energy systems for three years is always difficult to obtain since small changes in users' habits, or household appliances will significantly affect the quality of training data. Using fewer data to achieve high performance is a crucial research question in this field. Much of the current research used a virtual environment to generate infinite simulated data for training RL agents, which was implemented with MATLAB or EnergyPlus for complex modeling [130,131]. Therefore, we used actual data rather than simulated data to evaluate the application potential of the RL method in this study, and reducing the dimension of state-action space was adopted to reduce the training set required to ensure the optimization effect.

While the works mentioned above have contributed to the applications of RL technologies in building energy systems scheduling, there are still two limitations of these approaches. First, the optimization of ESS is mainly focused on a single optimization objective, such as the system's economy. Specifically, it uses ESS to arbitrage under RTP fluctuation while ignoring the local consumption of renewable energy, which contradicts the original intention of improving the renewable energy penetration level of the grid. Second, learning control policies using RL methods require enormous amounts of data. Most of the works mentioned above used infinite simulated data generated by building simulators (such as EnergyPlus) or a large amount of actual data (over three years), which leads to time-consuming training. Model-based RL (MB-RL) is one of the methods to overcome this problem. MBRL can use the domain knowledge of the model to learn the optimal control policy in a data-driven way more effectively. Heeyun et al. [132] established a battery energy consumption model based on the domain knowledge of vehicle powertrain for RL training. The simulation results show that compared with the dynamic programming result, the performance of MBRL reaches 93.8%. Authors in[133] proposed a model-based A3C algorithm to realize the strategic bidding for wind energy. The simulation results show that the strategy generated by MB-A3C is superior to other model-free or model-based RL algorithms.

1.5 Research Approach and Paper Structure

Buildings consume more than 40% of total energy, making it crucial to develop renewable energy sources. However, the economic factor is currently hindering the development of renewable energy. Therefore, improving the economic viability of renewable energy has become a general trend. Building upon the aforementioned brief overview of the research landscape in building energy systems, it is evident that various energy prediction algorithms and RL algorithms have been applied and yielded promising results. However, despite these advancements, there still exist some gaps in the current state of machine learning research in building energy, which includes:

1) The continuous development of building energy management systems has increased system complexity, which presents new challenges in constructing accurate energy prediction models. However, despite these challenges, the field of energy prediction still largely relies on traditional machine learning algorithms. As deep learning algorithms continue to advance, it is becoming increasingly important to evaluate and apply new algorithms that have demonstrated success in other domains to the energy system field in a scientifically rigorous manner.

2) Many studies in the field of building energy focus solely on implementing a single algorithm in a particular case and do not compare the performance of different algorithms in the same scenario. Consequently, it is critical to conduct comprehensive evaluations of the performance of multiple algorithms in specific scenarios. Doing so would provide a theoretical basis for selecting appropriate algorithms in subsequent research, thus ensuring that the most effective and efficient algorithms are employed.

3) Most reinforcement learning research utilizes data generated by simulators or data sets spanning three years or more to train agents. However, such data sets are difficult to obtain in real-world deployments. Therefore, it is essential to evaluate the data efficiency of various algorithms and develop novel methods that can improve the data utilization rate.

Such methods should achieve excellent control ability through small-sample learning, enabling researchers to use limited data to train agents, thus making it more feasible to deploy reinforcement learning algorithms in real-world building energy systems.

4) As Japan currently adopts multi-stage and time-of-use prices, most studies on operational optimization of building energy systems in Japan are based on these pricing models. However, real-time electricity price has advantages in terms of reflecting the actual market price of electricity and promoting the energy-saving behavior of consumers. Therefore, it is necessary to explore the performance of various algorithms under the scenario of real-time electricity price introduction and to provide guidance for the future deployment of building energy systems in Japan.

This paper proposes using machine learning to optimize the operation of building energy systems, to improve the system's economy while ensuring a high local consumption rate of renewable energy，including the following works：

1) First, we introduce the latest attentional mechanisms in deep learning to improve the accuracy of the prediction model and evaluate its potential in the field of energy prediction.

2) Secondly, we propose a model-based RL approach to optimize the operation of residential photovoltaic energy storage systems. The experimental results show that this method has achieved a good adjustment effect using the measured data of one and a half years. Its sample utilization rate is better than the traditional model-free RL method.

3) Finally, based on the verified prediction model and model-based RL method, we propose a multi-objective optimization control method considering real-time prediction values, which can optimize the system's economy while ensuring the high local absorption rate of renewable energy. This method provides the best solution for such scenarios.

The chapter names and basic structure of this paper are shown in Fig. 1-15 Besides, the brief introduction of chapters schematic is shown in Fig.1-16

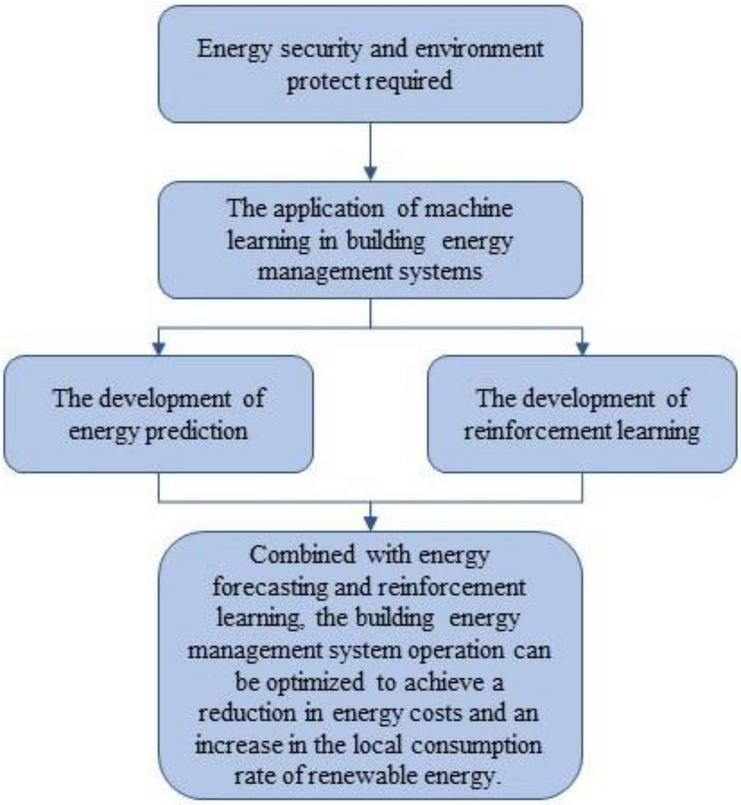

Energy security and environment protect required
The application of machine learning in building energy management systems
The development of energy prediction
The development of reinforcement learning
Combined with energy forecasting and reinforcement learning, the building energy management system operation can be optimized to achieve a reduction in energy costs and an increase in the local consumption rate of renewable energy.

Fig. 1-14 Research logic of the article

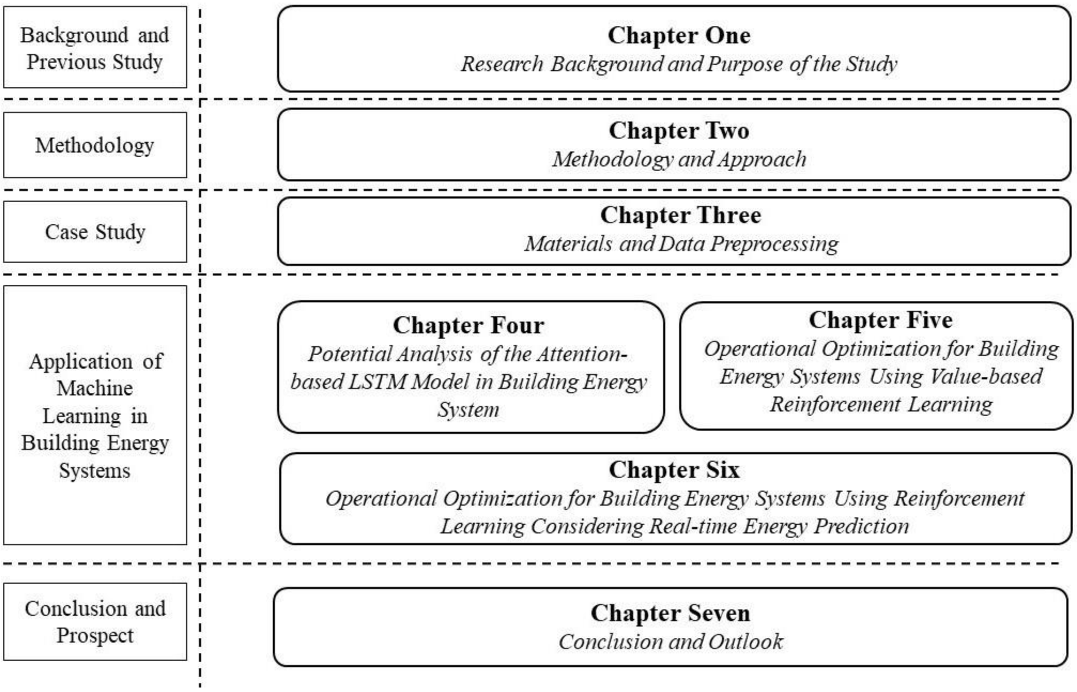

Background and Previous Study
Chapter One
Research Background and Purpose of the Study
Methodology
Chapter Two
Methodology and Approach
Case Study
Chapter Three
Materials and Data Preprocessing
Application of Machine Learning in Building Energy Systems
Chapter Four
Potential Analysis of the Attention-based LSTM Model in Building Energy System
Chapter Five
Operational Optimization for Building Energy Systems Using Value-based Reinforcement Learning
Chapter Six
Operational Optimization for Building Energy Systems Using Reinforcement Learning Considering Real-time Energy Prediction
Conclusion and Prospect
Chapter Seven
Conclusion and Outlook

Fig. 1-15 Chapter name and basic structure

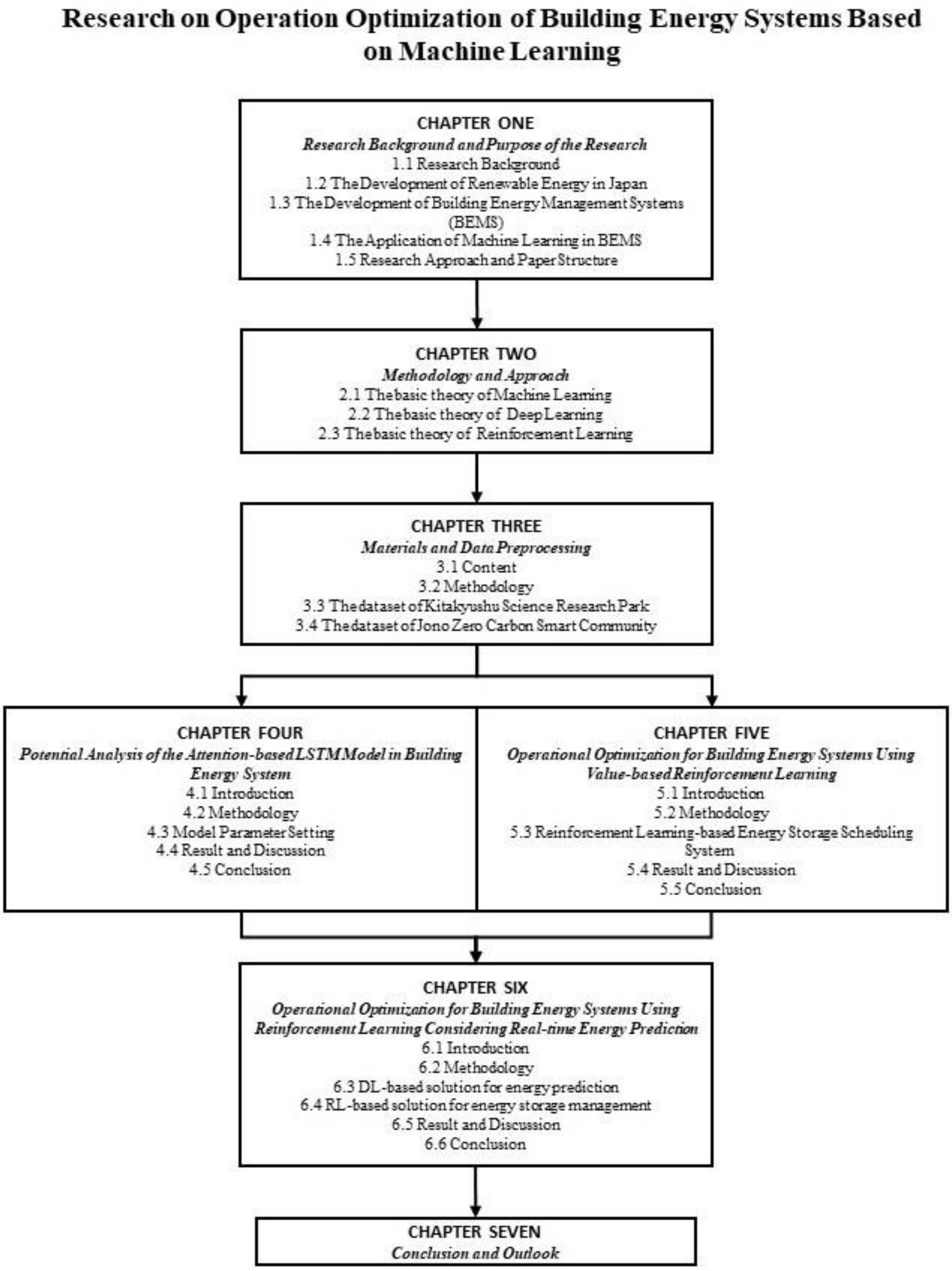

Research on Operation Optimization of Building Energy Systems Based on Machine Learning
CHAPTER ONE
Research Background and Purpose of the Research
1.1 Research Background
1.2 The Development of Renewable Energy in Japan
1.3 The Development of Building Energy Management Systems (BEMS)
1.4 The Application of Machine Learning in BEMS
1.5 Research Approach and Paper Structure
CHAPTER TWO
Methodology and Approach
2.1 The basic theory of Machine Learning
2.2 The basic theory of Deep Learning
2.3 The basic theory of Reinforcement Learning
CHAPTER THREE
Materials and Data Preprocessing
3.1 Content
3.2 Methodology
3.3 The dataset of Kitakyushu Science Research Park
3.4 The dataset of Jono Zero Carbon Smart Community
CHAPTER FOUR
Potential Analysis of the Attention-based LSTM Model in Building Energy System
4.1 Introduction
4.2 Methodology
4.3 Model Parameter Setting
4.4 Result and Discussion
4.5 Conclusion
CHAPTER FIVE
Operational Optimization for Building Energy Systems Using Value-based Reinforcement Learning
5.1 Introduction
5.2 Methodology
5.3 Reinforcement Learning-based Energy Storage Scheduling System
5.4 Result and Discussion
5.5 Conclusion
CHAPTER SIX
Operational Optimization for Building Energy Systems Using Reinforcement Learning Considering Real-time Energy Prediction
6.1 Introduction
6.2 Methodology
6.3 DL-based solution for energy prediction
6.4 RL-based solution for energy storage management
6.5 Result and Discussion
6.6 Conclusion
CHAPTER SEVEN
Conclusion and Outlook

Fig. 1-16 Brief chapter introduction

- Introduction and Purpose of the Research

Chapter 1 first provides an overview of the global energy demand and the current state of renewable energy technology development. In this way, the necessity to develop renewable energy sources becomes apparent. Next, we summarize the development trend of renewable energy in Japan and explain how renewable energy is integrated into the power grid. With the increasing adoption of renewable energy, there are both opportunities and challenges for energy management systems. Since the main objective of this paper is to investigate the role of machine learning in building energy management systems, this chapter also emphasizes recent advancements and related research in energy prediction and reinforcement learning control. Finally, this article's context and chapter structure are described for readers' reference.

➢ Methodology

Chapter 2 summarizes the fundamental theories and methods of deep learning and deep reinforcement learning. Specifically, it covers essential concepts and formulas such as neural networks, reinforcement learning, and gradient descent.

➢ Materials and Data Preprocessing

Chapter 3 provides an in-depth analysis of the data resources and this study's preprocessing steps. The measured energy system data from Kitakyushu Science Research Park and Jono Zero Carbon Smart Community were utilized. This section details the system under consideration, the methodology employed for data preprocessing, potential data patterns, and the creation of the training and test sets utilized in the subsequent experiments.

➢ Potential Analysis of the Attention-based LSTM Model in Building Energy System

Accurately predicting system energy consumption is crucial for implementing model predictive control (MPC). The LSTM network has made remarkable achievements in energy prediction in recent years. This chapter aimed to evaluate the potential of using an attentional-based LSTM network (A-LSTM) to predict HVAC energy consumption in practical applications. To assess the potential applicability of the A-LSTM model in practical scenarios, the training and testing datasets used in the experiments consist of actual energy consumption data collected from Kitakyushu Science Research Park in Japan. Pearce analysis was first carried out on the source data set and built the target database. Then five baseline models (A-LSTM, LSTM, RNN, DNN, and SVR) were built. Besides, to optimize the super parameters of the model, the Tree-structured of Parzen Estimators (TPE) algorithm was introduced. Finally, the applications are performed on the target database, and the results are analyzed from multiple perspectives. The results showed that the performance of the A-LSTM model was better than other baseline models, which could provide accurate and reliable hourly forecasting for HVAC energy consumption. Additionally, we evaluated the performance of the abovementioned algorithms using training sets of varying lengths and analyzed their sensitivity to data, thus providing a solid foundation for future research.

➢ Operational Optimization for Building Energy Systems Using Value-based Reinforcement Learning

With the rapid development of photovoltaic (PV) and energy storage systems, optimization strategies focus more on the cost-effectiveness of energy system management. However, the uncertainty of PV generation and the mismatch with consumer demand have become the major challenges for cost-effective optimization. In this chapter, we propose a model-based deep reinforcement learning algorithm Double-Dueling Deep Q-Networks (D3QN), to optimize the cost-effective operation of a residential house with the grid-connected PV-battery system in Japan and conduct experiments to evaluate three value-based reinforcement learning algorithms in an actual data center. The performance evaluation is based on their ability to improve cost-effectiveness and adaptability to the real-time electricity price (RTP). The results were analyzed and compared in detail, and special attention was paid to the sensitivity of the data features and the feasibility of the scheduling strategy. Besides, this chapter also compared D3QN using a model-free method and the proposed model-based framework to verify its effectiveness.

➢ Operational Optimization for Building Energy Systems Using Actor-Critic based Reinforcement Learning Considering Real-time Energy Prediction

As distributed PV and energy storage devices are widely developed, the uncertainty of on-site generation and the mismatch between local generation and residents' energy demand have become the major challenges for energy management systems. Nowadays, Reinforcement learning (RL) as an advanced control algorithm has gained more and more attention. However, the traditional model-free RL method has demanding requirements for the quality and quantity of data, which limits its application in energy management. Therefore, in this chapter, we applied the energy prediction model proposed in Chapter 2 to RL control and proposed a model-based Actor-Critic RL method to optimize the operation control of the energy storage system (ESS) by taking the measured dataset of an actual existing building in Japan as the research object. With an optimization goal of reducing the microgrid's energy cost and ensuring the PV self-consumption ratio, we designed a new reward function for these goals. We took the benchmark strategy currently used by the target building's energy management system as the baseline model in the experiment. We applied four advanced RL algorithms (PPO, DQN, DDPG, and TD3) to optimize the baseline model. The results show that the proposed RL design can better achieve the two optimization objectives of minimizing energy cost and maximizing the PV self-consumption ratio.

- Conclusion and Outlook

Chapter 7 provides a conclusion for the entire thesis and discusses potential directions for future research.

Reference

[1] IEA (2022), Energy Statistics Data Browser, IEA, Paris https://www.iea.org/data-and-statistics/data-tools/energy-statistics-data-browser[EB].

[2] NIU Z, WU J, LIU X, 等. Understanding energy demand behaviors through spatio-temporal smart meter data analysis[J/OL]. Energy, 2021, 226: 120493. DOI:10.1016/j.energy.2021.120493.

[3] JRADI M, VEJE C, JØRGENSEN B N. Deep energy renovation of the Mærsk office building in Denmark using a holistic design approach[J/OL]. Energy and Buildings, 2017, 151: 306-319. DOI:10.1016/j.enbuild.2017.06.047.

[4] KIM B, YAMAGUCHI Y, KIMURA S, 等. Urban building energy modeling considering the heterogeneity of HVAC system stock: A case study on Japanese office building stock[J/OL]. Energy and Buildings, 2019, 199: 547-561. DOI:10.1016/j.enbuild.2019.07.022.

[5] BOGDANOV D, RAM M, AGHAHOSSEINI A, 等. Low-cost renewable electricity as the key driver of the global energy transition towards sustainability[J/OL]. Energy, 2021, 227: 120467. DOI:10.1016/j.energy.2021.120467.

[6] LI Y, GAO W, RUAN Y. Performance investigation of grid-connected residential PV-battery system focusing on enhancing self-consumption and peak shaving in Kyushu, Japan[J/OL]. Renewable Energy, 2018, 127: 514-523. DOI:https://doi.org/10.1016/j.renene.2018.04.074.

[7] SU Y, ZHOU Y, TAN M. An interval optimization strategy of household multi-energy system considering tolerance degree and integrated demand response[J/OL]. Applied Energy, 2020, 260: 114144. DOI:10.1016/j.apenergy.2019.114144.

[8] LI Y, GAO W, ZHANG X, 等. Techno-economic performance analysis of zero energy house applications with home energy management system in Japan[J/OL]. Energy and Buildings, 2020, 214: 109862. DOI:10.1016/j.enbuild.2020.109862.

[9] ZHAO X, GAO W, QIAN F, 等. Electricity cost comparison of dynamic pricing model based on load forecasting in home energy management system[J/OL]. Energy, 2021, 229: 120538. DOI:10.1016/j.energy.2021.120538.

[10] ULLAH Z, ELKADEEM M R, KOTB K M, 等. Multi-criteria decision-making model for optimal planning of on/off grid hybrid solar, wind, hydro, biomass clean electricity supply[J/OL]. Renewable Energy, 2021, 179: 885-910. DOI:10.1016/j.renene.2021.07.063.

[11] MCILWAINE N, FOLEY A M, MORROW D J, 等. A state-of-the-art techno-economic review of distributed and embedded energy storage for energy systems[J/OL]. Energy, 2021, 229: 120461. DOI:10.1016/j.energy.2021.120461.

[12] KC R, RIJAL H, SHUKUYA M, 等. An in-situ study on occupants’ behaviors for adaptive thermal comfort in a Japanese HEMS condominium[J/OL]. Journal of Building Engineering, 2018, 19: 402-411. DOI:10.1016/j.jobe.2018.05.013.

[13] AL-HINAI A, ALYAMMAHI H, HAES ALHELOU H. Coordinated intelligent frequency control incorporating battery energy storage system, minimum variable contribution of demand response, and variable load damping coefficient in isolated power systems[J/OL]. Energy Reports, 2021, 7: 8030-8041. DOI:10.1016/j.egyr.2021.07.072.

[14] PALLONETTO F, DE ROSA M, FINN D P. Impact of intelligent control algorithms on demand response flexibility and thermal comfort in a smart grid ready residential building[J/OL]. Smart Energy, 2021, 2: 100017. DOI:10.1016/j.segy.2021.100017.

[15] ZHOU Y, RAVEY A, PÉRA M C. Real-time cost-minimization power-allocating strategy via model predictive control for fuel cell hybrid electric vehicles[J/OL]. Energy Conversion and Management, 2021, 229: 113721. DOI:10.1016/j.enconman.2020.113721.

[16] STEBEL K, FRATCZAK M, GRELEWICZ P, 等. Adaptive predictive controller for energy-efficient batch heating process[J/OL]. Applied Thermal Engineering, 2021, 192: 116954. DOI:10.1016/j.applthermaleng.2021.116954.

[17] MASERO E, MAESTRE J M, CAMACHO E F. Market-based clustering of model predictive controllers for maximizing collected energy by parabolic-trough solar collector fields[J/OL]. Applied Energy, 2022, 306: 117936. DOI:10.1016/j.apenergy.2021.117936.

[18] CEUSTERS G, RODRÍGUEZ R C, GARCÍA A B, 等. Model-predictive control and reinforcement learning in multi-energy system case studies[J/OL]. Applied Energy, 2021, 303: 117634. DOI:10.1016/j.apenergy.2021.117634.

[19] MAYNE D Q. Model predictive control: Recent developments and future promise[J/OL]. Automatica, 2014, 50(12): 2967-2986. DOI:10.1016/j.automatica.2014.10.128.

[20] SULTANA W R, SAHOO S K, SUKCHAI S, 等. A review on state of art development of model predictive control for renewable energy applications[J/OL]. Renewable and Sustainable Energy Reviews, 2017, 76: 391-406. DOI:10.1016/j.rser.2017.03.058.

[21] HAZYUK I, GHIAUS C, PENHOUET D. Optimal temperature control of intermittently heated buildings using Model Predictive Control: Part II – Control algorithm[J/OL]. Building and Environment, 2012, 51: 388-394. DOI:10.1016/j.buildenv.2011.11.008.

[22] ZHANG W, WANG J, LIU Y, 等. Reinforcement learning-based intelligent energy management architecture for hybrid construction machinery[J/OL]. Applied Energy, 2020, 275: 115401. DOI:10.1016/j.apenergy.2020.115401.

[23] STRUNZ S, GAWEL E, LEHMANN P. The political economy of renewable energy policies in Germany and the EU[J/OL]. Utilities Policy, 2016, 42: 33-41. DOI:10.1016/j.jup.2016.04.005.

[24] Statistical Review of World Energy 2020. bp Statistical Review of World Energy 2021. Available at: https://www.bp.com/content/dam/bp/business-sites/en/global/corporate/pdfs/energy-economics/statistical-review/bp-stats-review-2021-full-report.pdf[Z].

[25] JOOS M, STAFFELL I. Short-term integration costs of variable renewable energy: Wind

curtailment and balancing in Britain and Germany[J/OL]. Renewable and Sustainable Energy Reviews, 2018, 86: 45-65. DOI:10.1016/j.rser.2018.01.009.

[26] HU H, XIE N, FANG D, 等. The role of renewable energy consumption and commercial services trade in carbon dioxide reduction: Evidence from 25 developing countries[J/OL]. Applied Energy, 2018, 211: 1229-1244. DOI:10.1016/j.apenergy.2017.12.019.

[27] WALMSLEY M R W, WALMSLEY T G, ATKINS M J. Achieving 33% renewable electricity generation by 2020 in California[J/OL]. Sustainable Development of Energy, Water and Environment Systems, 2015, 92: 260-269. DOI:10.1016/j.energy.2015.05.087.

[28] WANG B, WANG J, DONG K, 等. Is the digital economy conducive to the development of renewable energy in Asia?[J/OL]. Energy Policy, 2023, 173: 113381. DOI:10.1016/j.enpol.2022.113381.

[29] Source: IEA World Energy Balances 2022 https://www.iea.org/data-and-statistics/data-product/world-energy-statistics-and-balances[EB].

[30] Ministry of Economy, Trade and Industry, Agency for Natural Resources and Energy. JAP AN'S ENERGY 2019. Available at: https://www.enecho.meti.go.jp/en/category/brochures/pdf/japan_energy_2019.pdf[R].

[31] 2019 – Understanding the current energy situation in Japan (Part 1). Available at: https://www.enecho.meti.go.jp/en/category/special/article/energyissue2019_01.html[R].

[32] JEPIC, 2020. The Electric Power Industry in Japan 2020. Japan Electric Power Information Center, INC. URL: https://www.jepic.or.jp/pub/pdf/epijJepic2020.pdf [accessed: 10.09.2020].[Z].

[33] KOMIYAMA R, FUJII Y. Assessment of post-Fukushima renewable energy policy in Japan's nation-wide power grid[J/OL]. Energy Policy, 2017, 101: 594-611. DOI:10.1016/j.enpol.2016.11.006.

[34] 2019 – Understanding the current energy situation in Japan (Part 2). Available at: https://www.enecho.meti.go.jp/en/category/special/article/energyissue2019_02.html[R].

[35] KNUEPFER K, ROGALSKI N, KNUEPFER A, 等. A reliable energy system for Japan with merit order dispatch, high variable renewable share and no nuclear power[J/OL]. Applied Energy, 2022, 328: 119840. DOI:10.1016/j.apenergy.2022.119840.

[36] CHENG C, BLAKERS A, STOCKS M, 等. 100% renewable energy in Japan[J/OL]. Energy Conversion and Management, 2022, 255: 115299. DOI:10.1016/j.enconman.2022.115299.

[37] TAGHIZADEH-HESARY F, PHOUMIN H, RASOULINEZHAD E. Assessment of role of green bond in renewable energy resource development in Japan[J/OL]. Resources Policy, 2023, 80: 103272. DOI:10.1016/j.resourpol.2022.103272.

[38] SHONO K, YAMAGUCHI Y, PERWEZ U, 等. Large-scale building-integrated photovoltaics installation on building façades: Hourly resolution analysis using commercial building stock in Tokyo, Japan[J/OL]. Solar Energy, 2023, 253: 137-153.

DOI:10.1016/j.solener.2023.02.025.

[39] KOBASHI T, YARIME M. Techno-economic assessment of the residential photovoltaic systems integrated with electric vehicles: A case study of Japanese households towards 2030[J/OL]. Innovative Solutions for Energy Transitions, 2019, 158: 3802-3807. DOI:10.1016/j.egypro.2019.01.873.

[40] NOMAGUCHI Y, TANAKA H, SAKAKIBARA A, 等. Integrated planning of low-voltage power grids and subsidies toward a distributed generation system – Case study of the diffusion of photovoltaics in a Japanese dormitory town[J/OL]. Energy, 2017, 140: 779-793. DOI:10.1016/j.energy.2017.08.114.

[41] FUKUYO K. Change in Energy Consciousness and Spread of Photovoltaic Cells after the Great East Japan Earthquake[J/OL]. The 6th Indonesia International Conference on Innovation, Entrepreneurship, and Small Business (IICIES 2014), 2015, 169: 98-108. DOI:10.1016/j.sbspro.2015.01.290.

[42] TANAKA K, SEKITO M, MANAGI S, 等. Decision-making governance for purchases of solar photovoltaic systems in Japan[J/OL]. Energy Policy, 2017, 111: 75-84. DOI:10.1016/j.enpol.2017.09.012.

[43] KOMIYAMA R, FUJII Y. Assessment of massive integration of photovoltaic system considering rechargeable battery in Japan with high time-resolution optimal power generation mix model[J/OL]. Energy Policy, 2014, 66: 73-89. DOI:10.1016/j.enpol.2013.11.022.

[44] M. YAZDANIAN, A. MEHRIZI-SANI. Distributed Control Techniques in Microgrids[J/OL]. IEEE Transactions on Smart Grid, 2014, 5(6): 2901-2909. DOI:10.1109/TSG.2014.2337838.

[45] RESTREPO M, CAÑIZARES C A, SIMPSON-PORCO J W, 等. Optimization- and Rule-based Energy Management Systems at the Canadian Renewable Energy Laboratory microgrid facility[J/OL]. Applied Energy, 2021, 290: 116760. DOI:10.1016/j.apenergy.2021.116760.

[46] JAFARI M, MALEKJAMSHIDI Z. Optimal energy management of a residential-based hybrid renewable energy system using rule-based real-time control and 2D dynamic programming optimization method[J/OL]. Renewable Energy, 2020, 146: 254-266. DOI:10.1016/j.renene.2019.06.123.

[47] C. M. RANGEL, D. MASCARELLA, G. JOOS. Real-time implementation & evaluation of grid-connected microgrid energy management systems[C/OL]//2016 IEEE Electrical Power and Energy Conference (EPEC). 2016: 1-6. DOI:10.1109/EPEC.2016.7771717.

[48] ALBERIZZI J C, ROSSI M, RENZI M. A MILP algorithm for the optimal sizing of an off-grid hybrid renewable energy system in South Tyrol[J/OL]. The 6th International Conference on Energy and Environment Research - Energy and environment: challenges towards circular economy, 2020, 6: 21-26. DOI:10.1016/j.egyr.2019.08.012.

[49] DINH H T, LEE K haeng, KIM D. Supervised-learning-based hour-ahead demand response for a behavior-based home energy management system approximating MILP optimization[J/OL]. Applied Energy, 2022, 321: 119382. DOI:10.1016/j.apenergy.2022.119382.

[50] PANG S, ZHENG Z, XIAO X, 等. Collaborative power tracking method of diversified thermal loads for optimal demand response: A MILP-Based decomposition algorithm[J/OL]. Applied Energy, 2022, 327: 120006. DOI:10.1016/j.apenergy.2022.120006.

[51] NEBULONI R, MERALDI L, BOVO C, 等. A hierarchical two-level MILP optimization model for the management of grid-connected BESS considering accurate physical model[J/OL]. Applied Energy, 2023, 334: 120697. DOI:10.1016/j.apenergy.2023.120697.

[52] M. DANESHVAR, B. MOHAMMADI-IVATLOO, K. ZARE, 等. Two-Stage Robust Stochastic Model Scheduling for Transactive Energy Based Renewable Microgrids[J/OL]. IEEE Transactions on Industrial Informatics, 2020, 16(11): 6857-6867. DOI:10.1109/TII.2020.2973740.

[53] YANG J J, YANG M, WANG M X, 等. A deep reinforcement learning method for managing wind farm uncertainties through energy storage system control and external reserve purchasing[J/OL]. International Journal of Electrical Power & Energy Systems, 2020, 119: 105928. DOI:10.1016/j.ijepes.2020.105928.

[54] J. SACHS, O. SAWODNY. A Two-Stage Model Predictive Control Strategy for Economic Diesel-PV-Battery Island Microgrid Operation in Rural Areas[J/OL]. IEEE Transactions on Sustainable Energy, 2016, 7(3): 903-913. DOI:10.1109/TSTE.2015.2509031.

[55] S. D. DWIVEDI, P. K. RAY. Energy Management and control of Grid-connected Microgrid integrated with HESS[C/OL]//2022 International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP). 2022: 1-6. DOI:10.1109/ICICCSP53532.2022.9862374.

[56] RAHIM S, JAVAID N, AHMAD A, 等. Exploiting heuristic algorithms to efficiently utilize energy management controllers with renewable energy sources[J/OL]. Energy and Buildings, 2016, 129: 452-470. DOI:10.1016/j.enbuild.2016.08.008.

[57] SHIVAM K, TZOU J C, WU S C. A multi-objective predictive energy management strategy for residential grid-connected PV-battery hybrid systems based on machine learning technique[J/OL]. Energy Conversion and Management, 2021, 237: 114103. DOI:10.1016/j.enconman.2021.114103.

[58] ELKAZAZ M, SUMNER M, NAGHIYEV E, 等. A hierarchical two-stage energy management for a home microgrid using model predictive and real-time controllers[J/OL]. Applied Energy, 2020, 269: 115118. DOI:10.1016/j.apenergy.2020.115118.

[59] XU Q, NING L, YUAN T, 等. Application of data mining combined with power data in assessment and prevention of regional atmospheric pollution[J/OL]. Energy Reports, 2023, 9: 3397-3405. DOI:10.1016/j.egyr.2023.02.016.

[60] CHEN J, YANG N, PAN Y, 等. Synchronous Medical Image Augmentation framework for deep learning-based image segmentation[J/OL]. Computerized Medical Imaging and Graphics, 2023, 104: 102161. DOI:10.1016/j.compmedimag.2022.102161.

[61] ZHANG Z, CHANG Q, XING J, 等. Deep-learning methods for integrated sensing and communication in vehicular networks[J/OL]. Vehicular Communications, 2023, 40: 100574. DOI:10.1016/j.vehcom.2023.100574.

[62] NOURA H N, AZAR J, SALMAN O, 等. A deep learning scheme for efficient multimedia IoT data compression[J/OL]. Ad Hoc Networks, 2023, 138: 102998. DOI:10.1016/j.adhoc.2022.102998.

[63] AGHAZADEH F, GHASEMI M, KAZEMI GARAJEH M, 等. An integrated approach of deep learning convolutional neural network and google earth engine for salt storm monitoring and mapping[J/OL]. Atmospheric Pollution Research, 2023, 14(3): 101689. DOI:10.1016/j.apr.2023.101689.

[64] LIAO Q. Intelligent classification model of land resource use using deep learning in remote sensing images[J/OL]. Ecological Modelling, 2023, 475: 110231. DOI:10.1016/j.ecolmodel.2022.110231.

[65] HUNT B, GILL G S, ALEXANDER D A, 等. Fast Deformable Image Registration for Real-Time Target Tracking During Radiation Therapy Using Cine MRI and Deep Learning[J/OL]. International Journal of Radiation Oncology*Biology*Physics, 2023, 115(4): 983-993. DOI:10.1016/j.ijrobp.2022.09.086.

[66] WANG Z, HONG T. Reinforcement learning for building controls: The opportunities and challenges[J/OL]. Applied Energy, 2020, 269: 115036. DOI:10.1016/j.apenergy.2020.115036.

[67] ZHAO Y, ZHANG C, ZHANG Y, 等. A review of data mining technologies in building energy systems: Load prediction, pattern identification, fault detection and diagnosis[J/OL]. Energy and Built Environment, 2020, 1(2): 149-164. DOI:10.1016/j.enbenv.2019.11.003.

[68] HUANG G, CHOW T T. Uncertainty shift in robust predictive control design for application in CAV air-conditioning systems[J/OL]. Building Services Engineering Research and Technology, 2011, 32(4): 329-343. DOI:10.1177/0143624411399686.

[69] QIAN F, GAO W, YANG Y, 等. Potential analysis of the transfer learning model in short and medium-term forecasting of building HVAC energy consumption[J/OL]. Energy, 2020, 193: 116724. DOI:10.1016/j.energy.2019.116724.

[70] LIU M D, DING L, BAI Y L. Application of hybrid model based on empirical mode decomposition, novel recurrent neural networks and the ARIMA to wind speed prediction[J/OL]. Energy Conversion and Management, 2021, 233: 113917. DOI:10.1016/j.enconman.2021.113917.

[71] MA Z, YE C, LI H, 等. Applying support vector machines to predict building energy consumption in China[J/OL]. Cleaner Energy for Cleaner Cities, 2018, 152: 780-786.

DOI:10.1016/j.egypro.2018.09.245.

[72] YANG F, WANG D, XU F, 等. Lifespan prediction of lithium-ion batteries based on various extracted features and gradient boosting regression tree model[J/OL]. Journal of Power Sources, 2020, 476: 228654. DOI:10.1016/j.jpowsour.2020.228654.

[73] LI R, XU M, CHEN Z, 等. Phenology-based classification of crop species and rotation types using fused MODIS and Landsat data: The comparison of a random-forest-based model and a decision-rule-based model[J/OL]. Soil and Tillage Research, 2021, 206: 104838. DOI:10.1016/j.still.2020.104838.

[74] WEI Y, XIA L, PAN S, 等. Prediction of occupancy level and energy consumption in office building using blind system identification and neural networks[J/OL]. Applied Energy, 2019, 240: 276-294. DOI:10.1016/j.apenergy.2019.02.056.

[75] BUI D K, NGUYEN T N, NGO T D, 等. An artificial neural network (ANN) expert system enhanced with the electromagnetism-based firefly algorithm (EFA) for predicting the energy consumption in buildings[J/OL]. Energy, 2020, 190: 116370. DOI:10.1016/j.energy.2019.116370.

[76] DONG B, CAO C, LEE S E. Applying support vector machines to predict building energy consumption in tropical region[J/OL]. Energy and Buildings, 2005, 37(5): 545-553. DOI:10.1016/j.enbuild.2004.09.009.

[77] FU Y, LI Z, ZHANG H, 等. Using Support Vector Machine to Predict Next Day Electricity Load of Public Buildings with Sub-metering Devices[J/OL]. The 9th International Symposium on Heating, Ventilation and Air Conditioning (ISHVAC) joint with the 3rd International Conference on Building Energy and Environment (COBEE), 12-15 July 2015, Tianjin, China, 2015, 121: 1016-1022. DOI:10.1016/j.proeng.2015.09.097.

[78] MASSANA J, POUS C, BURGAS L, 等. Short-term load forecasting in a non-residential building contrasting models and attributes[J/OL]. Energy and Buildings, 2015, 92: 322-330. DOI:10.1016/j.enbuild.2015.02.007.

[79] D. LIU, Q. CHEN, K. MORI. Time series forecasting method of building energy consumption using support vector regression[C/OL]//2015 IEEE International Conference on Information and Automation. 2015: 1628-1632. DOI:10.1109/ICInfA.2015.7279546.

[80] LV Z, SINGH A, LI J. Deep Learning for Security Problems in 5G Heterogeneous Networks[J]. IEEE Network, 2021, 35: 67-73.

[81] ASKARI S, MONTAZERIN N, ZARANDI M H F. A clustering based forecasting algorithm for multivariable fuzzy time series using linear combinations of independent variables[J/OL]. Applied Soft Computing, 2015, 35: 151-160. DOI:10.1016/j.asoc.2015.06.028.

[82] BAGNASCO A, FRESI F, SAVIOZZI M, 等. Electrical consumption forecasting in hospital facilities: An application case[J/OL]. Energy and Buildings, 2015, 103: 261-270. DOI:10.1016/j.enbuild.2015.05.056.

[83] FAN C, XIAO F, ZHAO Y. A short-term building cooling load prediction method using deep

learning algorithms[J/OL]. Applied Energy, 2017, 195: 222-233. DOI:10.1016/j.apenergy.2017.03.064.

[84] DEB C, EANG L S, YANG J, 等. Forecasting diurnal cooling energy load for institutional buildings using Artificial Neural Networks[J/OL]. Energy and Buildings, 2016, 121: 284-297. DOI:10.1016/j.enbuild.2015.12.050.

[85] AHMAD M W, MOURSHED M, REZGUI Y. Trees vs Neurons: Comparison between random forest and ANN for high-resolution prediction of building energy consumption[J/OL]. Energy and Buildings, 2017, 147: 77-89. DOI:10.1016/j.enbuild.2017.04.038.

[86] HOU Z, LIAN Z, YAO Y, 等. Cooling-load prediction by the combination of rough set theory and an artificial neural-network based on data-fusion technique[J/OL]. Applied Energy, 2006, 83(9): 1033-1046. DOI:10.1016/j.apenergy.2005.08.006.

[87] LV Z, QIAO L, LI J, 等. Deep-Learning-Enabled Security Issues in the Internet of Things[J/OL]. IEEE Internet of Things Journal, 2021, 8(12): 9531-9538. DOI:10.1109/JIOT.2020.3007130.

[88] HOCHREITER S, SCHMIDHUBER J. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8): 1735-1780.

[89] SU H, ZIO E, ZHANG J, 等. A hybrid hourly natural gas demand forecasting method based on the integration of wavelet transform and enhanced Deep-RNN model[J/OL]. Energy, 2019, 178: 585-597. DOI:10.1016/j.energy.2019.04.167.

[90] DUAN J, CHANG M, CHEN X, 等. A combined short-term wind speed forecasting model based on CNN–RNN and linear regression optimization considering error[J/OL]. Renewable Energy, 2022, 200: 788-808. DOI:10.1016/j.renene.2022.09.114.

[91] ZHENG C, WANG S, LIU Y, 等. A novel RNN based load modelling method with measurement data in active distribution system[J/OL]. Electric Power Systems Research, 2019, 166: 112-124. DOI:10.1016/j.epsr.2018.09.006.

[92] VERWIMP L, VAN HAMME H, WAMBACQ P. State gradients for analyzing memory in LSTM language models[J/OL]. Computer Speech & Language, 2020, 61: 101034. DOI:10.1016/j.csl.2019.101034.

[93] SU C, HUANG H, SHI S, 等. Neural machine translation with Gumbel Tree-LSTM based encoder[J/OL]. Journal of Visual Communication and Image Representation, 2020, 71: 102811. DOI:10.1016/j.jvcir.2020.102811.

[94] LV Z, QIAO L, SINGH A K, 等. Fine-Grained Visual Computing Based on Deep Learning[J]. ACM Transactions on Multimidia Computing Communications and Applications, 2021.

[95] WANG D, GAN J, MAO J, 等. Forecasting power demand in China with a CNN-LSTM model including multimodal information[J/OL]. Energy, 2023, 263: 126012. DOI:10.1016/j.energy.2022.126012.

[96] WANG W, HONG T, XU X, 等. Forecasting district-scale energy dynamics through integrating building network and long short-term memory learning algorithm[J/OL]. Applied Energy, 2019, 248: 217-230. DOI:10.1016/j.apenergy.2019.04.085.

[97] SENDRA-ARRANZ R, GUTIÉRREZ A. A long short-term memory artificial neural network to predict daily HVAC consumption in buildings[J/OL]. Energy and Buildings, 2020, 216: 109952. DOI:10.1016/j.enbuild.2020.109952.

[98] WANG Z, HONG T, PIETTE M A. Predicting plug loads with occupant count data through a deep learning approach[J/OL]. Energy, 2019, 181: 29-42. DOI:10.1016/j.energy.2019.05.138.

[99] WANG Z, HONG T, PIETTE M A. Building thermal load prediction through shallow machine learning and deep learning[J/OL]. Applied Energy, 2020, 263: 114683. DOI:10.1016/j.apenergy.2020.114683.

[100] LIMOUNI T, YAAGOUBI R, BOUZIANE K, 等. Accurate one step and multistep forecasting of very short-term PV power using LSTM-TCN model[J/OL]. Renewable Energy, 2023, 205: 1010-1024. DOI:10.1016/j.renene.2023.01.118.

[101] WEI J, WU X, YANG T, 等. Ultra-short-term forecasting of wind power based on multi-task learning and LSTM[J/OL]. International Journal of Electrical Power & Energy Systems, 2023, 149: 109073. DOI:10.1016/j.ijepes.2023.109073.

[102] WANG L, MAO M, XIE J, 等. Accurate solar PV power prediction interval method based on frequency-domain decomposition and LSTM model[J/OL]. Energy, 2023, 262: 125592. DOI:10.1016/j.energy.2022.125592.

[103] BAHDANAU D, CHO K, BENGIO Y. Neural Machine Translation by Jointly Learning to Align and Translate[J]. Computer ence, 2014.

[104] LI Y, ZHU Z, KONG D, 等. EA-LSTM: Evolutionary attention-based LSTM for time series prediction[J]. Knowledge-Based Systems, 2019, 181(Oct.1): 104785.1-104785.8.

[105] LU J, XIONG C, PARIKH D, 等. Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017.

[106] YU Z, YU J, FAN J, 等. Multi-modal Factorized Bilinear Pooling with Co-attention Learning for Visual Question Answering[C]//2017 IEEE International Conference on Computer Vision (ICCV). 2017.

[107] LIANG Y, KE S, ZHANG J, 等. GeoMAN: Multi-level Attention Networks for Geo-sensory Time Series Prediction[C]//Twenty-Seventh International Joint Conference on Artificial Intelligence IJCAI-18. 2018.

[108] HEIDARI A, KHOVALYG D. Short-term energy use prediction of solar-assisted water heating system: Application case of combined attention-based LSTM and time-series decomposition[J]. Solar Energy, 2020, 207: 626-639.

[109] FAZLIPOUR Z, MASHHOUR E, JOORABIAN M. A deep model for short-term load

forecasting applying a stacked autoencoder based on LSTM supported by a multi-stage attention mechanism[J/OL]. Applied Energy, 2022, 327: 120063. DOI:10.1016/j.apenergy.2022.120063.

[110] LI J, YANG B, LI H, 等. DTDR–ALSTM: Extracting dynamic time-delays to reconstruct multivariate data for improving attention-based LSTM industrial time series prediction models[J]. Knowledge-Based Systems, 211.

[111] YANG T, LI B, XUN Q. LSTM-Attention-Embedding Model-based Day-Ahead Prediction of Photovoltaic Power Output Us-ing Bayesian Optimization[J]. IEEE Access, 2019, PP(99): 1-1.

[112] KOSANA V, TEEPARTHI K, MADASTHU S, 等. A novel reinforced online model selection using Q-learning technique for wind speed prediction[J/OL]. Sustainable Energy Technologies and Assessments, 2022, 49: 101780. DOI:10.1016/j.seta.2021.101780.

[113] XU B, SHI J, LI S, 等. Energy consumption and battery aging minimization using a Q-learning strategy for a battery/ultracapacitor electric vehicle[J/OL]. Energy, 2021, 229: 120705. DOI:10.1016/j.energy.2021.120705.

[114] BRANDI S, PISCITELLI M S, MARTELLACCI M, 等. Deep reinforcement learning to optimise indoor temperature control and heating energy consumption in buildings[J/OL]. Energy and Buildings, 2020, 224: 110225. DOI:10.1016/j.enbuild.2020.110225.

[115] KOLODZIEJCZYK W, ZOLTOWSKA I, CICHOSZ P. Real-time energy purchase optimization for a storage-integrated photovoltaic system by deep reinforcement learning[J/OL]. Control Engineering Practice, 2021, 106: 104598. DOI:10.1016/j.conengprac.2020.104598.

[116] CHEN Y, NORFORD L K, SAMUELSON H W, 等. Optimal control of HVAC and window systems for natural ventilation through reinforcement learning[J/OL]. Energy and Buildings, 2018, 169: 195-205. DOI:10.1016/j.enbuild.2018.03.051.

[117] HAQ E U, LYU C, XIE P, 等. Implementation of home energy management system based on reinforcement learning[J/OL]. 2021 The 8th International Conference on Power and Energy Systems Engineering, 2022, 8: 560-566. DOI:10.1016/j.egyr.2021.11.170.

[118] QIU S, LI Z, LI Z, 等. Model-free control method based on reinforcement learning for building cooling water systems: Validation by measured data-based simulation[J/OL]. Energy and Buildings, 2020, 218: 110055. DOI:10.1016/j.enbuild.2020.110055.

[119] 丁志梁, 潘毅群, 谢建彤, 等. 强化学习算法在空调系统运行优化中的应用研究[J]. 建筑节能(7): 7.

[120] YUAN X, PAN Y, YANG J, 等. Study on the application of reinforcement learning in the operation optimization of HVAC system[J]. 建筑模拟(英文), 2021.

[121] HARROLD D J B, CAO J, FAN Z. Data-driven battery operation for energy arbitrage using rainbow deep reinforcement learning[J/OL]. Energy, 2022, 238: 121958. DOI:10.1016/j.energy.2021.121958.

[122] L. YU, W. XIE, D. XIE, 等. Deep Reinforcement Learning for Smart Home Energy Management[J/OL]. IEEE Internet of Things Journal, 2020, 7(4): 2751-2762. DOI:10.1109/JIOT.2019.2957289.

[123] YANG T, ZHAO L, LI W, 等. Towards healthy and cost-effective indoor environment management in smart homes: A deep reinforcement learning approach[J/OL]. Applied Energy, 2021, 300: 117335. DOI:10.1016/j.apenergy.2021.117335.

[124] GAO Y, MATSUNAMI Y, MIYATA S, 等. Operational optimization for off-grid renewable building energy system using deep reinforcement learning[J/OL]. Applied Energy, 2022, 325: 119783. DOI:10.1016/j.apenergy.2022.119783.

[125] ZHAO L, YANG T, LI W, 等. Deep reinforcement learning-based joint load scheduling for household multi-energy system[J/OL]. Applied Energy, 2022, 324: 119346. DOI:10.1016/j.apenergy.2022.119346.

[126] Y. LI, Y. WEN, D. TAO, 等. Transforming Cooling Optimization for Green Data Center via Deep Reinforcement Learning[J/OL]. IEEE Transactions on Cybernetics, 2020, 50(5): 2002-2013. DOI:10.1109/TCYB.2019.2927410.

[127] E. MOCANU, D. C. MOCANU, P. H. NGUYEN, 等. On-Line Building Energy Optimization Using Deep Reinforcement Learning[J/OL]. IEEE Transactions on Smart Grid, 2019, 10(4): 3698-3708. DOI:10.1109/TSG.2018.2834219.

[128] DU Y, ZANDI H, KOTEVSKA O, 等. Intelligent multi-zone residential HVAC control strategy based on deep reinforcement learning[J/OL]. Applied Energy, 2021, 281: 116117. DOI:10.1016/j.apenergy.2020.116117.

[129] YANG L, NAGY Z, GOFFIN P, 等. Reinforcement learning for optimal control of low exergy buildings[J/OL]. Applied Energy, 2015, 156: 577-586. DOI:10.1016/j.apenergy.2015.07.050.

[130] AZUATALAM D, LEE W L, DE NIJS F, 等. Reinforcement learning for whole-building HVAC control and demand response[J/OL]. Energy and AI, 2020, 2: 100020. DOI:10.1016/j.egyai.2020.100020.

[131] PINTO G, DELTETTO D, CAPOZZOLI A. Data-driven district energy management with surrogate models and deep reinforcement learning[J/OL]. Applied Energy, 2021, 304: 117642. DOI:10.1016/j.apenergy.2021.117642.

[132] LEE H, KIM K, KIM N, 等. Energy efficient speed planning of electric vehicles for car-following scenario using model-based reinforcement learning[J/OL]. Applied Energy, 2022, 313: 118460. DOI:10.1016/j.apenergy.2021.118460.

[133] SANAYHA M, VATEEKUL P. Model-based deep reinforcement learning for wind energy bidding[J/OL]. International Journal of Electrical Power & Energy Systems, 2022, 136:

107625. DOI:10.1016/j.ijepes.2021.107625.

Chapter 2

METHODOLOGY AND APPROACH

CHAPTER TWO: METHODOLOGY AND APPROACH

2.1 The Basic Theory of Machine Learning

As a subfield of artificial intelligence (AI), machine learning (ML) develops algorithms and statistical models that enhance computer systems' performance at specific tasks through experience. In contrast to traditional programming, where programmers write code to address specific problems, ML involves programs that learn to solve problems independently by analyzing data and making predictions or decisions based on the available data, enabling systems to adapt to new situations and improve their performance over time[1,2]. As new ML algorithms and theories continue to emerge and online data and computing resources expand, it has become a shared goal of academia and industry to design algorithms suitable for specific problem scenarios and enhance the efficiency of data analysis. This section will provide an in-depth exploration of the fundamental theories of machine learning, which aims to offer fresh perspectives and ideas for the future design of algorithms.

2.1.1 Definition of the ML Model

To create an ML model, it is necessary to define its input space x and output space y. The input and output spaces can be either a finite set or a whole Euclidean space, although the input space is generally much larger than the output space. A feature vector represents each specific input called an instance, and all feature vectors constitute the feature space. Each feature corresponds to a dimension of the feature space. The output space of different machine learning tasks varies, with finite and discrete spaces for classification problems and continuous spaces for regression problems.

The input space x and output space y jointly constitute the sample space. Any given sample in the sample space is assumed to be independently and randomly generated according to some unknown joint distribution $p_r(x, y)$. The relationship between x and y can be described by a true mapping function $C(x)$, such that $C(x) = y$ holds for any sample x. This function $C(x)$ is referred to as the target function, and the set of all functions that can be learned to approximate the target function is known as the hypothesis set, denoted by C. As the true mapping function $C(x)$ is unknown, the hypothesis set F is designed based on empirical knowledge to include all possible mappings. Through learning on a training set, the best hypothesis f can be found from hypothesis F. The hypothesis set is defined as follows:

$$F = \{f(x;\theta), \theta \in R^d\} \tag{2-1}$$

Where $f(x;\theta)$ is called a function or model with parameter θ, and R^d is called the parameter space. Generally, the hypothesis space can be divided into linear and nonlinear spaces, with the corresponding models being linear and nonlinear. The hypothesis space for the most basic linear model consists of parameterized linear functions:

$$f(x;\theta) = w^T x + b \tag{2-2}$$

Where w is the weighting parameter, and b is the offset parameter. The generalized nonlinear model can be expressed as follows:

$$f(x;\theta) = w^T \phi(x) + b \tag{2-3}$$

Where $\phi(x) = [\phi_1(x), \phi_2(x) \dots \phi_k(x),]^T$ is a vector composed of K nonlinear basis functions. Besides.

2.1.2 Learning Criterion

After constructing the hypothesis space of the model, machine learning must consider which criteria to use for learning or selecting the optimal model. Typically, a loss function is used to measure the accuracy of a prediction, while a risk function is used to measure the accuracy of a model in meeting expectations. The loss function is a non-negative, real-valued function that measures the discrepancy between the predicted value and the actual value of a model. It serves as a critical component of many machine learning algorithms, enabling the optimization of model parameters by minimizing the difference between predictions and actual outcomes. By quantifying the error of a model's predictions, the loss function provides a mechanism for evaluating and improving the performance of machine learning models.

Two commonly used loss functions are the mean squared error loss function and the cross-entropy loss function. The mean squared error loss function is typically used for regression problems and is defined in Eq. (2-4). In contrast, the cross-entropy loss function is used for classification problems and is defined as shown in Eq. (2-5).

$$L\big(y, f(x;\theta)\big) = \frac{1}{2}\big(y - f(x;\theta)\big)^2 \qquad (2\text{-}4)$$

$$L\big(y, f(x;\theta)\big) = -y^T \log\big(f(x;\theta)\big) \qquad (2\text{-}5)$$

where $L\big(y, f(x;\theta)\big)$ denotes the mean squared error loss function, y denotes the actual value, $f(x;\theta)$ denotes the predicted value. A smaller value of the loss function indicates better model performance. This is because the loss function quantifies the difference between the predicted and actual values of the model, and minimizing this difference is a key objective of many machine learning algorithms. Minimizing the loss function optimizes a model to predict more accurate outcomes and improve its overall performance.

The quality of an ML model can be measured using a risk function (expected risk). The risk function measures the model's expected error on a given dataset, while the expected risk is the average risk over all possible datasets, expressed as follows:

$$R(\theta) = E_{(x,y)\sim p_r(x,y)}\big[L\big(y, f(x;\theta)\big)\big] \qquad (2\text{-}6)$$

In general, a smaller expected risk indicates better performance of a machine learning model. The expected risk measures the average risk of the model over all possible datasets and provides a more reliable estimate of the model's performance on unseen data. However, it is important to note that minimizing the expected risk may not always be the best approach. It can lead to underfitting, where the model is too simple and performs poorly on training and test data. Empirical risk, also known as the empirical error or training error, is a measure of the error of an ML model on a given dataset. It is calculated as the average loss function value over the training set and is used to evaluate the model's performance on the training data, expressed as follows:

$$R_D^{emp}(\theta) = \frac{1}{n} \Sigma_{n=1}^{N} L\big(y, f(x; \theta)\big) \qquad (2\text{-}7)$$

Minimizing the empirical risk is a common approach in machine learning, leading to optimized models for the specific training data. However, minimizing the empirical risk alone may lead to overfitting, where the model becomes overly complex and performs poorly on new, unseen data. To avoid overfitting, various regularization techniques are used to balance the minimization of the empirical risk with the complexity of the model. These techniques prevent the model from fitting the noise in the training data and encourage it to learn the underlying patterns and generalize well to new, unseen data. In practice, the trade-off between minimizing the expected risk and avoiding underfitting and overfitting is carefully balanced to achieve the best possible performance of the machine learning model.

In summary, the learning principles of ML require not only fitting the available data in the training set well but also minimizing the prediction error on unknown test data. The success of a machine learning model depends on its ability to balance the trade-off between fitting the training data well and avoiding overfitting while minimizing the prediction error on new, unseen data, which requires careful consideration and evaluation of various learning principles, techniques and performance metrics throughout the ML pipeline.

2.1.3 Optimization of Algorithm

In machine learning, optimization can be divided into two main categories: parameter optimization and hyperparameter optimization. Since θ in model $f(x; \theta)$ is called the parameter of the model. Parameter optimization involves finding the model parameters' optimal values that minimize the training data's loss function. Parameter optimization aims to make the model fit the training data as closely as possible while avoiding overfitting. This can be achieved using various optimization algorithms such as gradient descent, stochastic gradient descent, and Adam.

Hyperparameter optimization involves finding the optimal values of the model hyperparameters that minimize the expected risk on the test data. Hyperparameters are the configuration parameters of the machine learning algorithm that are not learned from the data, such as the learning rate, regularization parameter, and number of hidden layers in a neural network. Hyperparameter optimization is a challenging task in machine learning as it involves searching a large hyperparameter space and requires a lot of computational resources. Various techniques, such as grid search, random search, and Bayesian optimization, are used to automate and speed up the hyperparameter optimization process.

In ML, the gradient descent algorithm is one of the most common and simplest optimization algorithms, which is an iterative optimization algorithm that aims to find the optimal values of the model parameters that minimize the loss function on the training data[3]. The gradient of the objective function L for the parametric variable $W^{[i]}$can be denoted by Eq. (2-8).

$$\nabla_{W^{[i]}} L\left(W^{[1]}, \dots, W^{[n]}\right) = \frac{\partial L(W^{[1]}, \dots, W^{[n]})}{\partial W^{[i]}}, \forall i = 1, \dots, n \qquad (2\text{-}8)$$

One of the fundamental concepts in gradient descent optimization is that the gradient points in the direction of the steepest ascent of the function. Therefore, moving along the gradient direction would lead to an increase in the function value. Conversely, moving in the opposite direction of the gradient leads to a decrease in the function value. Hence, the function value decreases the fastest in the opposite direction of the gradient. This property is exploited in gradient descent optimization algorithms, where the model parameters are updated iteratively in the opposite direction of the gradient until a minimum of the loss function is reached. By taking small steps in the opposite direction of the gradient, the algorithm gradually approaches the optimal values of the model parameters that minimize the loss function on the training data. Assume that the optimization variable at the current time t is $W_t^{[1]}, \dots, W_t^{[n]}$, and the gradient descent method is shown in Eq.(2-9) :

$$W_{t\_new}^{[i]} \leftarrow W_t^{[i]} - \alpha * \nabla_{W^{[i]}} L\left(W_{t\_new}^{[1]}, \ldots, W_{t\_new}^{[n]}\right), \forall i = 1, \ldots, n \qquad (2\text{-}9)$$

In Eq(2-9), the learning rate $\alpha \in [0,1]$ is a hyperparameter that determines the step size of each iteration. The value of the learning rate is manually set and tuned during the training of the neural network, as it directly affects the convergence rate of the gradient descent algorithm and ultimately affects the final performance of the model. A learning rate that is too small would lead to slow convergence and longer training times, while a learning rate that is too large may cause the algorithm to overshoot the minimum of the loss function and diverge. Therefore, selecting an appropriate learning rate that balances the trade-off between convergence speed and accuracy is essential. To determine the optimal learning rate, a common practice is to start with a relatively large value and gradually reduce it over time, monitoring the model's performance at each step. Alternatively, adaptive learning rates methods such as AdaGrad, RMSProp, and Adam can automatically adjust the learning rate during training based on historical gradient information.

When the training data is too large to fit into memory, stochastic gradient descent (SGD) is a widely used optimization algorithm that randomly extracts mini-batches of training data to compute the gradient of the loss function[4]. The formula is shown below:

$$L\left(W^{[1]}, \ldots, W^{[n]}\right) = \frac{1}{f} \sum_{j=1}^{f} F_j\left(W^{[1]}, \ldots, W^{[n]}\right) \qquad (2\text{-}10)$$

In summary, SGD can help the optimization process escape from local minima and converge to a good solution. Despite its popularity, SGD has some limitations. For example, it can be sensitive to the choice of the learning rate, which controls the step size in each update. If the learning rate is too low, the algorithm may converge too slowly, while it may fail to converge if it is too large. Several variants of SGD have been proposed to address these limitations, including momentum-based and adaptive learning rate methods such as AdaGrad and Adam. These variants can help accelerate convergence and improve performance on certain problems.

2.2 The Basic Theory of Deep Learning

Deep learning (DL) has become increasingly popular in recent years due to several factors, including the availability of large amounts of data, the development of powerful computing hardware, and advances in optimization algorithms. DL is a general term for a class of pattern analysis methods that process various simple features into more complex features to form high-level representations of objects, which can then be used to solve complex tasks. In this way, DL models can learn hierarchical representations of objects, enabling them to capture more abstract and nuanced concepts. This approach allows DL models to learn from large datasets without human intervention, making it particularly useful in areas where manual feature engineering is challenging or impossible. Several types of deep learning models are commonly used, including:

- Deep Neural Networks (DNN): DNNs are the most basic DL model, also known as multi-layer perceptron (MLP). DNN has more hidden layers than ordinary artificial neural networks, allowing it to learn complex patterns.

- Convolutional Neural Networks (CNNs): CNNs are commonly used in computer vision tasks such as image and video recognition. They are designed to recognize spatial patterns in the data by processing it through layers of convolutional filters.

- Recurrent Neural Networks (RNNs): RNNs are designed to handle sequential data, such as text and time series data. They can learn from the temporal dependencies in the data by using feedback loops to pass information from one time step to another.

- Long Short-Term Memory Networks (LSTM): LSTM is an improved cyclic neural network that deals with sequential data with long-term dependencies. LSTM has been successfully applied to a wide range of applications that require processing long-term dependencies and modeling complex sequential patterns.

- Generative Adversarial Networks (GANs): GANs are a type of deep learning model used for generating new data similar to the training data. They consist of two neural networks, a generator, and a discriminator, trained in a game-like setup.

This section will provide a detailed overview of the deep learning algorithms employed in this study, which includes an introduction to the fundamental principles of neural networks and the relevant mathematical formulas.

2.2.1 Deep Neural Networks (DNNs)

Artificial neural networks (ANNs), comprised of neurons, are information processing units that abstractly, simplistically, and systematically map to neurons in the human brain. Neurons are the core components of ANNs. Through interconnection, ANNs can mathematically simulate the activity of neurons in the human brain, allowing for efficient computation and other data processing. A typical artificial neural network is shown in Fig.2-1:

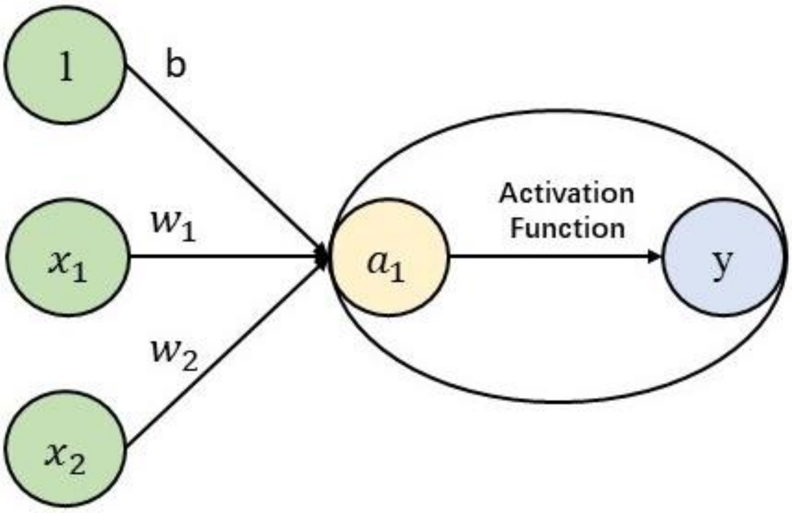


Fig. 2-1 The structure of the ANNs

The entire process of the fully connected layer can be defined by Eq.(2-11) and (2-12). In these equations, 1、x_1、x_2 represent the input signals to the neurons, while b is a bias parameter that controls the ease of neuron activation. Additionally, w_1 and w_2 represent the parameters that denote the weights of the respective signals. After multiplying each weight by its corresponding signal, the resulting values are transmitted to the next neuron a_1, where they are added together, as shown in Eq. (2-11). The activation function $h(x)$ then transforms the output of this neuron to produce the final output signal y, as shown in Eq. (2-12).

$$a = b + w_1x_1 + w_2x_2 \qquad (2\text{-}11)$$

$$y = h(a) \qquad (2\text{-}12)$$

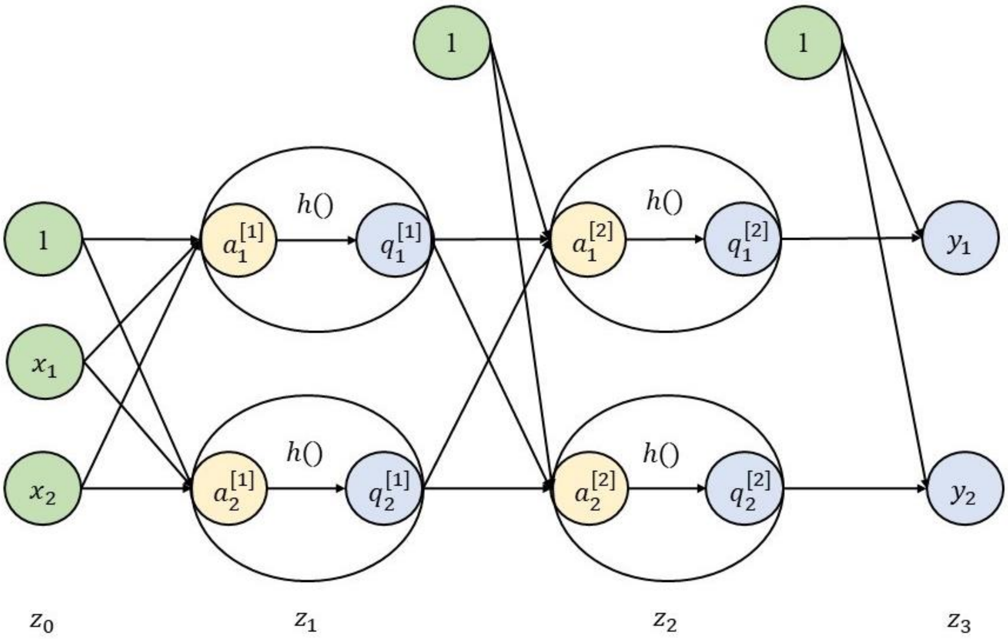


Fig. 2-2 The structure of the DNN (MLP)

The fully connected layer can be used as a fundamental component to construct a Deep Neural Network (DNN), also known as a Multi-Layer Perceptron (MLP). In contrast to a single fully connected layer, a DNN contains multiple hidden layers with numerous neuron nodes, which enhances the network's ability to represent and process complex multi-input, multi-output problems. As a result, the DNN architecture can effectively improve the network's capacity for information processing. Fig.2-2 depicts a three-layer DNN that implements the mapping of input vectors $z^{[0]}$ to $z^{[3]}$. Based on this principle, an n-layer DNN can be represented by Eq. (2-13) to (2-15):

The first layer： $z^{[1]} = h_1(W^{[1]}z^{[0]} + b^{[1]})$ (2-13)

The second layer: $z^{[2]} = h_2(W^{[2]}z^{[1]} + b^{[2]})$ (2-14)

$$\vdots \qquad \vdots$$

The nth layer： $z^{[n]} = h_n(W^{[n]}z^{[n-1]} + b^{[n]})$ (2-15)

The parameters $W^{[1]}, \dots, W^{[n]}$ and $b^{[1]}, \dots, b^{[n]}$ in neural networks require constant optimization from training data to achieve optimal model performance. It should be noted that the parameters differ between layers. $h_1, \dots, h_n$ are the activation functions that can introduce nonlinearity into the network's output. The choice of activation function can have a significant impact on the performance of the neural network. The Rectified Linear Unit (ReLU) function is commonly used to activate hidden layers. However, selecting the activation function for the output layer depends on the task. The Sigmoid function is often used for binary classification problems, while for multi-class classification problems, the Softmax function is a better choice. For regression problems, the output layer is typically activated by a linear function. The major deep learning libraries in Python, such as TensorFlow, PyTorch, and Keras, all provide tools for building deep neural networks.

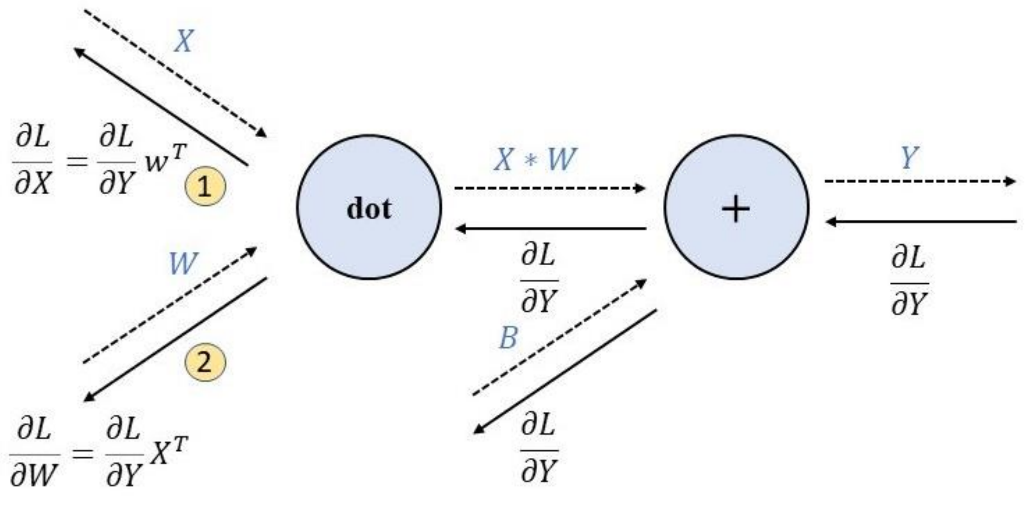

X
$\frac{\partial L}{\partial X} = \frac{\partial L}{\partial Y} w^T$
1
W
2
$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial Y} X^T$
dot
$X * W$
$\frac{\partial L}{\partial Y}$
B
$\frac{\partial L}{\partial Y}$
+
Y
$\frac{\partial L}{\partial Y}$

Fig. 2-3 Schematic diagram of forward and backward propagation

During the training of DNN, it is common practice to select a mini-batch of data from the training set randomly. The goal of training is to minimize the loss function associated with the mini-batch, which requires calculating the gradients of the weights using backpropagation. These gradients are then used to update the weights using a variant of stochastic gradient descent (SGD), which involves making small updates to the weights in the direction of the negative gradient. This process is repeated for a specified number of epochs or until convergence. Fig. 2-3 illustrates the forward and backward propagation of a neural network. The dashed lines represent the forward propagation of the network, while the solid lines represent the backward propagation. The essence of backpropagation is to use the chain rule of calculus to calculate the partial derivatives of the loss function L concerning the weights and biases of the neural network, which is represented by Eq (2-16) and (2-17):

$$\frac{\partial L}{\partial X} = \frac{\partial L}{\partial Y} * \frac{\partial Y}{\partial X} = \frac{\partial L}{\partial Y} w^T \tag{2-16}$$

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial Y} * \frac{\partial Y}{\partial W} = \frac{\partial L}{\partial Y} X^T \tag{2-17}$$

The network structure for the fully connected layer is shown in Fig 2-2, and the neural network's output $z^{[i]}$ is the neural network's prediction. Let L be the loss function, and the gradient of L for the parameter $W^{[i]}$ can be obtained based on the chain rule, as shown in Eq. (2-18):

$$\frac{\partial L}{\partial W^{[i]}} = \frac{\partial z^{[i]}}{\partial W^{[i]}} * \frac{\partial L}{\partial z^{[i]}} \tag{2-18}$$

After obtaining the gradient $\frac{\partial L}{\partial W^{[n]}}$, iterate through $i = n, \ldots, 1$ and update the parameter $W^{[i]}$ using this gradient. Similarly, the gradient of L for the parameter $z^{[i-1]}$can be obtained based on the chain rule, as shown in Eq.(2-19):

$$\frac{\partial L}{\partial z^{[i-1]}} = \frac{\partial z^{[i]}}{\partial z^{[i-1]}} * \frac{\partial L}{\partial z^{[i]}} \tag{2-19}$$

The backpropagation path of the entire neural network is shown in Fig. 2-4. When the gradient of the loss function L for $z^{[i]}$ is obtained, the gradient of L for both $z^{[i-1]}$ and $W^{[i]}$ can be calculated.

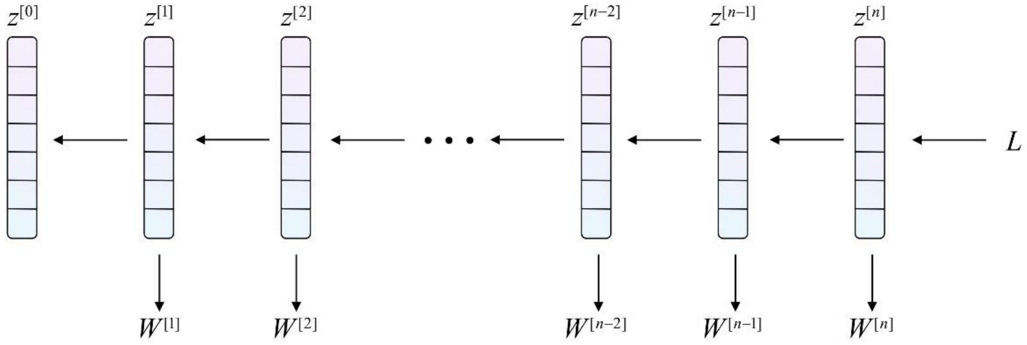


Fig. 2-4 The backpropagation path of the entire neural network

2.2.2 Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) is a neural networks designed to process time series data. In contrast to traditional Deep Neural Networks (DNNs) and Convolutional Neural Networks (CNNs), RNNs employ a cyclic structure to replace the hidden layer of the feedforward neural network. During the process of information transmission, a portion of the information is retained in the current neuron during each cycle, and this retained information is used as input for the next neural unit with new information. This unique mechanism enables RNNs to effectively "remember" past inputs and employ them to influence the processing of subsequent inputs[5]. Through this cyclic structure, RNNs exhibit the capacity for dynamic temporal behavior and are particularly useful for tasks involving sequential data, such as speech recognition, natural language processing, and time series prediction. By enabling the network to maintain a form of internal memory, RNNs can identify temporal patterns in the data and use these patterns to make informed predictions[6]. A typical RNN architecture is shown in Fig.2-5:

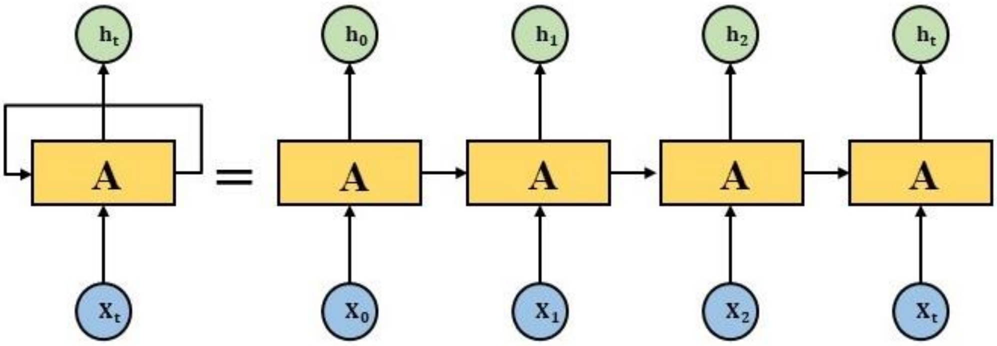


Fig. 2-5 The structure of the RNNs

In Fig.2-5, the input vector at time step t is denoted as x_t, and the output vector is represented as h_t. RNNs allow the network to pass the output from the one-time step as input to the next time step. From a mathematical perspective, the RNNs can be viewed as a function f with weights w, and the network becomes a recursive function. The mathematical formulation of this recursive function is shown in Eq.(2-20):

$$h_{(t)} = f(h_{t-1}, x_t, w) \tag{2-20}$$

While RNNs have proven effective in handling sequential data, they can suffer from the problem of vanishing gradients when attempting to learn long-term dependencies. This problem arises when the relevant information for a prediction is located several time steps back in the sequence, beyond the reach of the network's short-term memory. As a result, gradients that are backpropagated through time can become very small, leading to slow learning or even complete stalling of the training process[7].

2.2.3 Long Short-Term Memory (LSTM)

In 1997, Hochreiter et al. proposed improvements to the RNN named Long Short-Term Memory (LSTM)[8]. LSTM is a type of RNN designed to address the vanishing and exploding gradient problems when training traditional RNNs. The architecture of an LSTM is shown in Fig 2-6, including a memory cell, which can selectively forget or remember information from previous time steps, as well as an input gate, an output gate, and a forget gate. The input gate controls the flow of new input into the memory cell, while the output gate controls the flow of information out of the cell. The forget gate determines which information from the previous time step should be forgotten or retained.

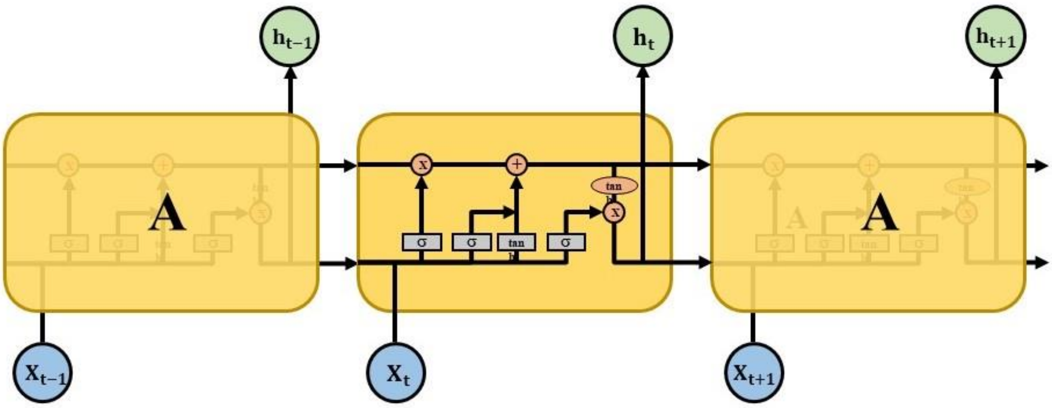


Fig. 2-6 The structure of the LSTM

Based on the RNN unit, the input gate i_t, output gate o_t, forgetting gate f_t and cell state C_t are added to the LSTM unit to control the inheritance and abandonment of information. There are three inputs of the LSTM unit: the input vector x_t at the current time slot t, the unit state C_{t-1} at the time slot t-1, and the state of the hidden layer h_{t-1} at the time slot t-1. The final output of the LSTM unit is the cell state C_t at the current time slot t and the state of the hidden layer at the current time h_t To figure out the h_t, we first let the W_i, W_o and W_f be the weight matrix of the input gate, the output gate, and the forgetting gate, and let the $[h_{t-1}, x_t]$ represent the combination of the hidden state at the moment t-1 and the unit's input at the time slot t into a new vector. Besides, let the b_i, b_o, and b_f be their bias vectors, and let the σ represent the Sigmoid activation function.

The formulas of the i_t, o_t, and f_t are shown below:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2-21}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{1-22}$$

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{2-23}$$

Finally, let $tanh$ represent the activation function, $*$ represent the Hadamard product, and let the $\tilde{C}_t$ be the state of the intermediate unit input at time slot t; then we can calculate the h_t.as follows:

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_c) \tag{2-24}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{2-25}$$

$$\hbar_t = o_t * tanh(C_t) \tag{2-26}$$

Thus, it can be seen that the cell state C_t in LSTM can be propagated through the hidden layer solely using linear summation, averting the issue of gradient attenuation. Furthermore, LSTMs allow the neural network to alternate between recollecting recent and remote information, empowering the data to decide which information to retain and discard[9]. Based on the advantages mentioned above, LSTM is currently the most widely used algorithm in load forecasting.

2.3 The Basic Theory of Reinforcement Learning (RL)

RL, as a branch of machine learning, is a computational method that can solve sequential decision problems[10,11]. All the RL problems can be defined as MDP, which represents the process by which an agent guides its behavior by obtaining rewards from interaction with the environment. Formally, an MDP is a five-tuple (S, A, P, R, R), where:

- S is the state space, which represents the available information that the RL agents use to make decisions.
- A is the action space, which means the RL agents make different decisions when interacting with the environment.
- P is the state-transition probability, which describes the probability distribution of going from state s to state s' when action a is taken.
- R is the reward (or cost) function, usually the objective function in a control problem.
- γ is the discount factor. The discount factor is used to overcome the feedback delay in the interaction between the agent and the environment. By discounting the rewards for multiple steps, the sum of the accumulated rewards for numerous steps in the future can be obtained. Then the short-term optimization objective and long-term optimization objective can be balanced.

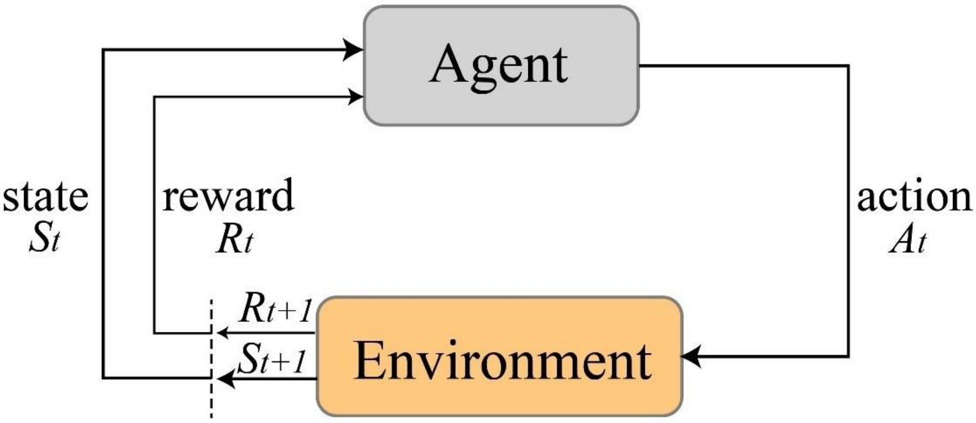


Fig. 2-7 The basic principles of RL

The basic principles of RL are illustrated in Fig. 2-7, where the intelligent agent adjusts its policy based on the feedback it receives from the environment through rewards or punishments. At each time step t, the agent observes the environmental state $s_t \in S$ and chooses an action $a_t \in A$ based on policy π. Then the agent will receive the reward $r(s_t, a_t)$, and the system will evolve into another state $s_{t+1} \in S$ The *policy* π is the probability distribution for each possible action $a \in A$ been selected in a state $s \in S$, and it can be performed deterministic or stochastic based on the specific algorithm. In this way, the agent utilizes the experiential data it collects to improve its performance, ultimately achieving either the maximization of cumulative rewards or the attainment of specific objectives.

2.3.1 State-value function and State-action value function (Q-function)

To solve sequential decision questions, the purpose of RL algorithms is to learn a value function $v_\pi(s)$ or state-action value function $Q^\pi(s,a)$, it is also called Q-function. The first refers to the value of a state s under a policy π, which indicates the expected return when starting in the state s and continuing with policy π. While the second refers to the expected return when starting in the state s with action a according to policy π, the *state-value function* is defined as Eq. (1):

$$V_\pi(s) = E_\pi\left[\sum_{\mathrm{k}=0}^{N} \gamma^k R_{t+k+1} | s_t = s\right] \tag{2-27}$$

Where E_π denotes the expected value under a policy π in the state s_t . N is the final step in an episode and t is any time step. The value of the discount factor γ needs to be tuned to balance the future and immediate reward.

The action-value function is a strong correlation with the state-value function, which can be defined as follows:

$$Q_\pi(s,a) = E_\pi\left[\sum_{k=0}^{N} \gamma^k r_{t+k+1} | s_t = s, a_t = a\right] \tag{2-28}$$

Where E_π is the expected value that the agent chooses an action a_t based on the policy π in the state s_t.

Given the MDP problem, the learning agent must find optimal policies π^* and value functions. Value functions are different according to the various policies. The optimal value function is the one that gets the maximum value compared to all the other value functions. It can be easily computed by taking the maximum of the Q-function as follows Eq. (2-29):

$$Q_{\pi}^{*}(s,a)=\left[r(s,a)+\gamma \max Q^{\pi}(s_{t+1},a_{t+1})\right] \tag{2-29}$$

The above equation is called a *Bellman optimality equation*, and it indicates the recursive relation between a value of the state s_t performed an action a_t under the policy π and its subsequent state and the average overall possibilities[12].

2.3.2 Q-learning Algorithm

Q-learning, first proposed by Watkins[13], is a value-based and off-policy RL method and is currently the most widely used RL method. The Q value is the $Q(s,a)$, which is the maximum expectation value that taking actions can obtain benefits under the state at a certain moment. The environment will calculate the corresponding reward according to the agent's action, so the main idea of the algorithm is to build a Q-table to store the Q value and then select the action that can obtain the maximum benefit according to the Q value. The update process of the Q-table can be summarized as follows:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha\left[r_i + \gamma max_{\alpha_{t+1}} Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)\right]$$

(2-30)

Where the s_{t+1} is the next state, the α is the learning rate, r_i denotes the real reward, and the γ means the discount factor that influences the current value of the future rewards. In addition, $r_i + \gamma max_{\alpha_{t+1}} Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)$ is the calculation of the time difference error (TD-Error), which denotes the change value of the Q value during the iteration.

However, When tabular features are used to represent the $Q(s_t, a_t)$, the state-space tends to grow exponentially with the dimension of the state, which is called the dimensional disaster. Q-learning usually uses linear function approximation or fixed sparse representation to represent the $Q(s_t, a_t)$ to overcome the dimensional disaster.

As the most basic RL algorithm, Q-learning offers the advantage of a simple and fast-running algorithm. It is primarily used to solve small action-space or state-space optimization control problems. For instance, in a study on the energy system optimization of renewable energy buildings, a Markov decision process was employed to model the problem, where energy cost minimization was set as the reinforcement learning objective. The decision variable was the time series of battery charging and discharging action. The optimal energy system optimization strategy with cumulative rewards was obtained through continuous interaction between the intelligent agent and the zero-energy residential energy system environment.

2.3.3 Classification of Reinforcement Learning

The above summarized the basic principles of RL. Next, we will discuss the classification of RL algorithms. The overview of the most popular algorithms is summarized in Table 1. According to the different action selection strategies, RL algorithms can be divided into two branches: value-based algorithms and policy-based algorithms, as shown in Fig. 2-8. The value-based algorithm calculates the expectation of reward through the potential reward as the basis for selecting actions. Examples of value-based methods include Q-learning and SARSA. The policy-based algorithm trains a probability distribution through policy sampling and enhances the probability of the desired action with a high return value[10]. Therefore, value-based algorithms can only be used for discrete action space, while policy-based algorithms have more advantages in continuous action space control. Examples of policy-based methods include the REINFORCE algorithm and the Policy Gradient method. Currently, the most popular actor-critic method combines the benefits of these two branches. Specifically, the actor-network will take actions based on the probability distribution of policies. The critic-network will give the value of actions to the actions, making it more convenient for the latter to deal with continuous control. Examples of actor-critic methods include Advantage Actor-Critic (A2C) and Deep Deterministic Policy Gradient (DDPG).

The RL algorithm can be off-policy or on-policy according to the interaction between the RL agent and the environment. On-policy methods learn the value function or policy based on the agent's current policy. In contrast, off-policy methods learn the value function or policy based on a different policy that generates the behavior data. For the off-policy method, the agent can learn by interacting with the environment in person or through accumulated experience (such as experience replay or replay buffer mechanism)[14]. In contrast, for the on-policy method, the agent can only interact with the environment to update the network. On-policy methods, such as SARSA, update their policy at every time step based on the current experience. Off-policy methods, such as Q-learning, update the value function based on the optimal policy. The choice of on-policy or off-policy depends on the specific requirements of the problem being solved. As the research object of this study is the measurement data collected by the actual HMES, the amount of data is limited, and the data collection is slow, so we would prefer to choose the off-policy method because they are more sample-efficient. In contrast, the on-policy method is more suitable for scenarios where data is generated using simulators.

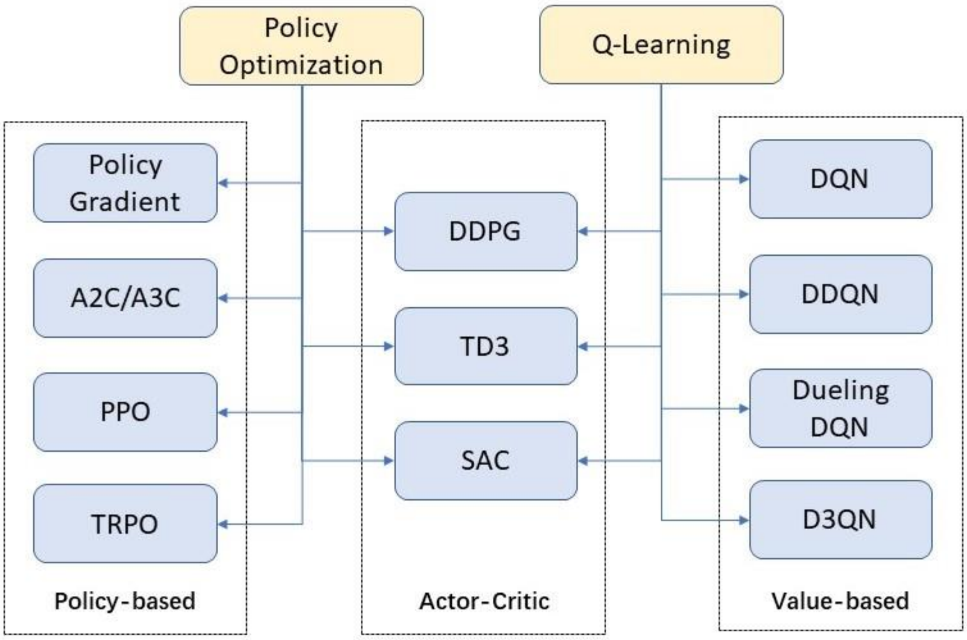


Fig. 2-8 The classification of reinforcement learning

Table 2-1 Common properties of the popular RL algorithms.

Algorithm	Type	Data usage	Action space
DQN	value-based	Off-policy	Discrete
DDQN	value-based	Off-policy	Discrete
Dueling DQN	value-based	Off-policy	Discrete
D3QN	value-based	Off-policy	Discrete
DPG	policy-based	Off-policy	Continuous
DDPG	actor-critic	Off-policy	Continuous
TD3	actor-critic	Off-policy	Continuous
SAC	actor-critic	Off-policy	Discrete/Continuous
TRPO	policy-based	On-policy	Discrete/Continuous
PPO	actor-critic	On-policy	Discrete/Continuous

Reference

[1] Jordan M I, Mitchell T M. Machine learning: Trends, perspectives, and prospects[J]. Science, 2015, 349(6245): 255–260.

[2] Li F, Du Y. From AlphaGo to Power System AI: What Engineers Can Learn from Solving the Most Complex Board Game[J]. IEEE Power and Energy Magazine, 2018, 16(2): 76–84.

[3] Bengio Y. Learning long-term dependencies with gradient descent is difficult[J]. IEEE Trans Neural Netw, 2002, 5.

[4] Niu F, Recht B, Re C, Wright S J. HOGWILD!: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent[J]. Advances in Neural Information Processing Systems, 2011, 24: 693–701.

[5] Shi Y, Zhao W, Li S, Li B, Sun X. Direct derivation scheme of DT-RNN algorithm for discrete time-variant matrix pseudo-inversion with application to robotic manipulator[J]. Applied Soft Computing, 2023, 133: 109861.

[6] Arriandiaga A, Portillo E, Sánchez J A, Cabanes I, Zubizarreta A. Downsizing training data with weighted FCM for predicting the evolution of specific grinding energy with RNNs[J]. Applied Soft Computing, 2017, 61: 211–221.

[7] Dudukcu H V, Taskiran M, Cam Taskiran Z G, Yildirim T. Temporal Convolutional Networks with RNN approach for chaotic time series prediction[J]. Applied Soft Computing, 2023, 133: 109945.

[8] Hochreiter S, Schmidhuber J. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8): 1735–1780.

[9] Verwimp L, Van Hamme H, Wambacq P. State gradients for analyzing memory in LSTM language models[J]. Computer Speech & Language, 2020, 61: 101034.

[10] Wang R. Reinforcement Learning: An Introduction[C]//2006 International Conference on Artificial Intelligence: 50 Years' Achievements, Future Directions and Social Impacts.

[11] Sutton R S, Precup D, Singh S. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning[J]. Artificial Intelligence, 1999, 112(1): 181–211.

[12] Christopher J. Q-learning. machine learning[J]. Machine Learning, 1992, 3.

[13] Watkins C J C Hellaby. Learning From Delayed Rewards[J]. Ph.d.thesis Kings College University of Cambridge, 1989.

[14] Lehna M, Hoppmann B, Heinrich R, Scholz C. A Reinforcement Learning Approach for the Continuous Electricity Market of Germany: Trading from the Perspective of a Wind Park Operator[J]. 2021. ,2021.

Chapter 3

MATERIALS AND DATA PREPROCESSING

CHAPTER THREE: MATERIALS AND DATA PREPROCESSING

3.1 Content

This study focuses on two sets of data samples: Kitakyushu Science Research Park (KSRP) and Jono Zero Carbon Smart Community (JZCSC). We selected these buildings due to their renewable energy facilities and complete energy management systems (EMS), which enable the collection of real-time and high-quality data related to system operation. The operational data for KSRP spans from January 1, 2002, to December 31, 2010, while data for JZCSC was collected from April 1, 2017, to September 30, 2019. The KSRP dataset will be utilized to verify the proposed load prediction model, and the JZCSC dataset will be used for future verification of the RL control model. Before constructing the prediction and control models, the collected data must be processed. The necessary data analysis must be conducted to determine relevant model parameters and prepare the model's training and test sets. Thus, this section provides an overview of the systems, describes the data preprocessing methods, and conducts potential law analysis.

3.2 Methodology

3.2.1 K-Nearest Neighbors (KNN)

In reality, force majeure factors may result in a small amount of missing sample data collected by EMS. As a data-driven method, machine learning (ML) is sensitive to missing data. The absence of sample data can often lead to inaccurate results in model-fitting calculations. However, blindly ensuring the integrity and accuracy of information by discarding data with missing values may result in an insufficient training dataset, causing an underfitting phenomenon. Therefore, selecting an appropriate method for dealing with missing values is crucial.

In this study, we use the K-Nearest Neighbors (KNN) algorithm to implement the interpolation of missing values[1]. The KNN algorithm is a popular ML technique for solving classification and regression problems, which also be used to impute missing values in a dataset[2]. The KNN algorithm identifies k samples that are spatially similar or exhibit similar characteristics in the dataset, determined through distance measurement. These k samples are then utilized to estimate the value of missing data points. Specifically, the missing values of each sample are interpolated by calculating the mean value of the k nearest neighbors found in the dataset. In this step, KNN always uses a distance metric to find the K nearest neighbors of the missing value. The most commonly used distance metric is the Euclidean distance, as shown in Eq.3-1[3]:

$$D(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad (3\text{-}1)$$

Where (x_i, y_i) are input samples. Once the K nearest neighbors are identified, their values are used to compute the average (or median) value, which is then used to replace the missing value. In this study, we used the scikit-learn package in python to implement the KNN algorithm.

3.2.2 Pearson Correlation

Before constructing the training and test sets, selecting the features of the data sample is typically necessary based on the results of correlation analysis between the features. The Pearson correlation coefficient is always used to measure the strength and direction of the relationship between two continuous variables[4]. The Pearson correlation coefficient is calculated by dividing the covariance of the two variables by the product of their standard deviations, as shown as follows[5]:

$$r = \frac{N\Sigma x_i y_i - \Sigma x_i \Sigma y_i}{\sqrt{N\Sigma x_i^2 - (\Sigma x_i)^2}\sqrt{N\Sigma y_i^2 - (\Sigma y_i)^2}} \tag{3-2}$$

where x_i and y_i are the two variables being correlated. A positive Pearson correlation coefficient (r > 0) indicates a positive relationship between the two variables, and a negative Pearson correlation coefficient (r < 0) indicates a negative relationship between the two variables. In this study, we defined features that have a correlation coefficient less than 0.01 with the target variable as not correlated, and we excluded them from the data set construction.

3.2.3 Data Normalization

Data normalization is a crucial step in data preparation that aims to transform data into a common scale, ensuring that every feature is given equal weight when analyzing the dataset. The normalization process not only enhances the efficiency of data processing but also contributes to the accuracy and effectiveness of the predictive models[6]. In particular, when using neural networks for modeling and forecasting, standardizing the sample data can significantly enhance the convergence speed of the model, as well as its prediction accuracy and learning efficiency. Among various normalization techniques in machine learning, the most commonly utilized method is the minimum-maximum normalization, which scales the feature values to a range between 0 and 1, as shown in Eq.(3-3):

$$y = \frac{x - min}{max - min} \tag{3-3}$$

Where y is the result after the max-min conversion function, x is the sample data, max is the maximum value of the original data, and min is the minimum value of the original data. However, it is worth noting that if the input of the neural network model is standardized data, the output result is also standardized data. The original data's corresponding output value can be obtained only after the inverse operation of the standardization function is processed. The inverse function of the standardization function is the inverse of the normalization function, which can transform the standardized data back to the original scale, as shown in Eq.(3-4):

$$x = (max - min) * y + min \tag{3-4}$$

3.3 The dataset of Kitakyushu Science Research Park

3.3.1 Case Introduction

This study utilized data from the Combined Cooling, Heating, and Power (CCHP) system at the Kitakyushu Science Research Park (KSRP) in Japan. The KSRP system is a distributed energy system consisting of a gas engine (160 kW), a fuel cell (200 kW), and a photovoltaic system (150 kW). The primary purpose of this system is to provide energy to the main teaching building of The University of Kitakyushu, which can accommodate the teaching and office needs of more than 3000 individuals. The target building in this study was divided into four floors, with the first floor comprising the student center, meeting rooms, and classrooms, and the second to fourth floors consisting of teachers' offices and student research rooms.

The energy supply for the system at KSRP was sourced from the gas engine, fuel cells, photovoltaic system, and the utility grid. The absorption chiller was primarily responsible for cooling, heating, and hot water loads. The gas engine and fuel cell also contributed to the cooling and heating load during electricity generation. The schematic diagram of the CCHP system at KSRP has presented in Fig.3-1. The system included a detailed data acquisition system that recorded operational data for each device and environmental data of the target building. By utilizing the temperature and flow data collected by the data acquisition system for hot and cold water supply and recovery, it was possible to calculate the target building's hot and cold load requirements in real-time. The KSRP cogeneration system was established in 2001. To ensure that the model accurately reflected the actual operating state of the system, data from 2002 to 2010 (78,820 data points) were selected as the research object. During this period, only three days of system failure occurred, which minimized the impact of missing data on the modeling process.

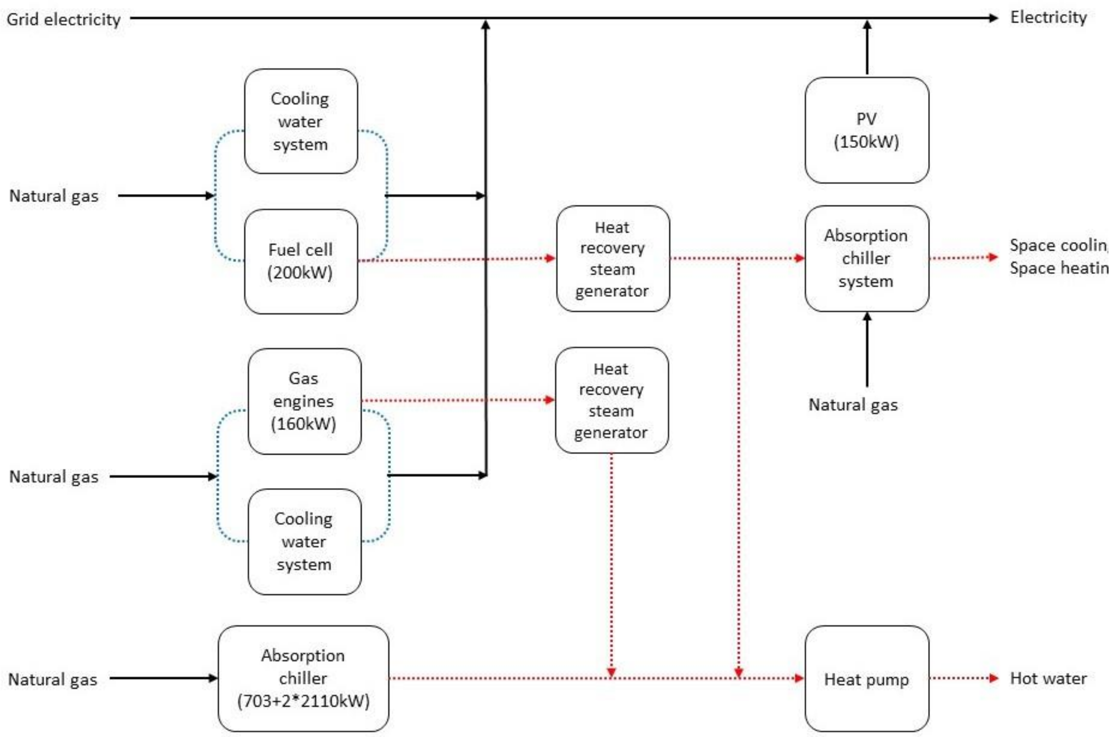


Fig. 1-1 The basic schematic diagram of the CCHP system at KSRP

3.3.2 Potential Analysis of Input Data Set

The cooling and heating output of the equipment from January 1, 2002, to December 31, 2010, was initially computed based on the gas consumption of the equipment and the annual average COP (cooling: 1.00, heating: 0.85). To verify the accuracy of the data, we also calculated the cooling and heating output using the temperature and flow rate of cold and hot water supply and recovery, which were collected by the system. This study used eight years of data from January 1, 2002, to December 31, 2010, for the target building. Each time step represented one hour. The first-eight data were used for training the model, and the last year's data were used for evaluating the models. Fig. 3-2 shows the time-series changes in HVAC load, temperature, and lux on a 1-hour basis used by the test set in this experiment; it should be noted that the positive value of Load represents heating, and the negative value represents cooling. It indicates that the target building's HVAC load, temperature, and lux had stochastic characteristics, and their values varied significantly in different months.

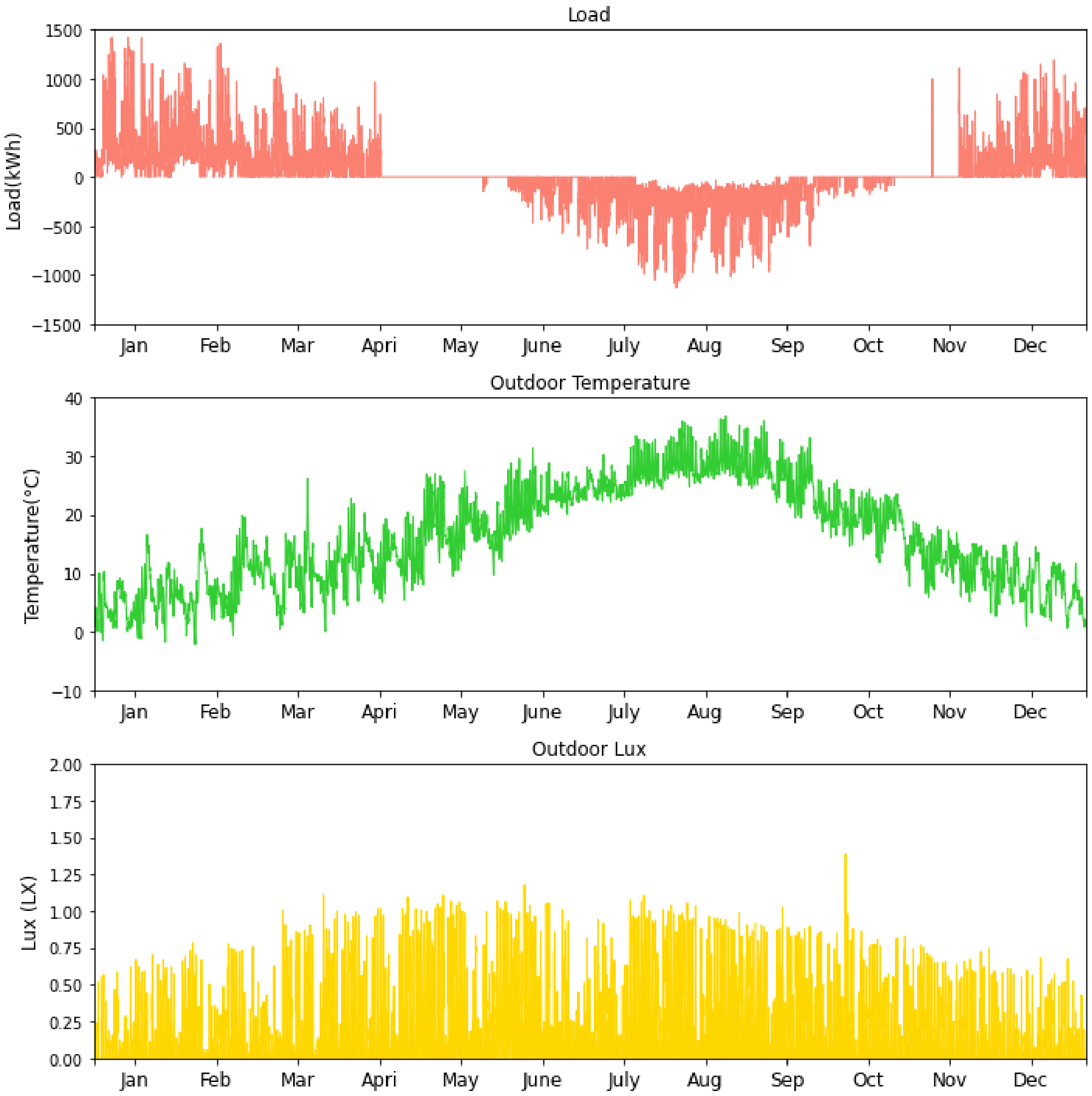


Fig. 3-2 Test Dataset: hourly Load, temperature, and lux from January 2010 to December 2010

To examine the distribution patterns of these data in a time series, we calculated the average Load in units of the month, week, and hour, respectively. The results are presented in Fig.3-3. As indicated in Fig.3-3 (a), the average load varies significantly from month to month. The annual peak value of total heating load output occurs in January, while that of total cooling load output occurs in August. As a result, December, January, February, and March were designated as the heating season; July, August, and September as the cooling season; and April, May, June, October, and November as the low-load season. The model's prediction performance was evaluated separately based on this categorization. Fig.3-3 (b) indicates that the cooling and heating loads are higher on weekdays than on weekends. Moreover, Fig.3-3 (c) demonstrates that the average daily load distribution in the heating and cooling seasons is significantly different. All of the aforementioned temporal information reflects the impact of human activities on Load, thus making them viable characteristic factors for database construction.

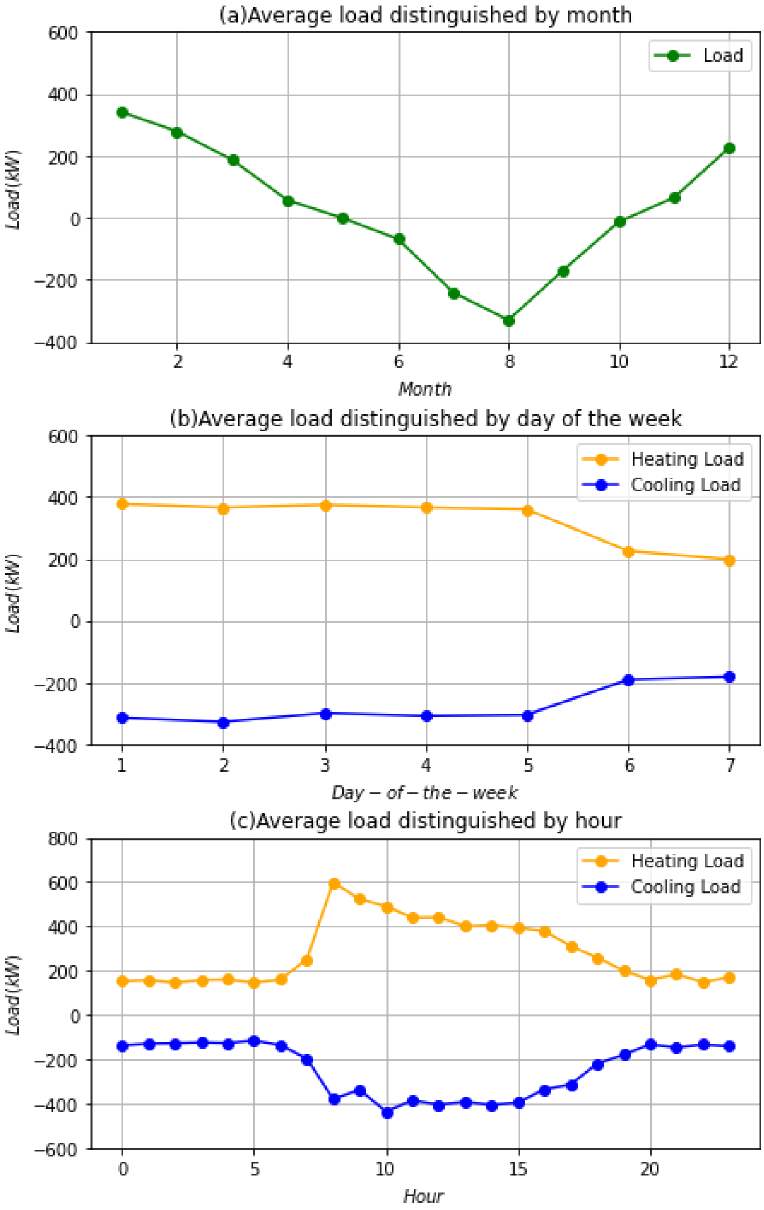


Fig. 3-3 The diagram of average load distributed by the time

We also selected other environmental factors that might affect the heat and cold output to build the initial database, including data collected by the Energy Center every hour from January 1, 2002, to December 31, 2010, a total of 78,820 pieces of data. Each group of data includes time information, outdoor temperature [◦C], relative humidity [%], irradiance [$W\ /\ m^2$], wind speed [m/s], wind direction, and load output (The positive load indicates the heating load, and the negative load indicates the refrigeration load). In addition, we conducted a Pearson correlation analysis for all features to identify and remove features with a small correlation that provide unnecessary information in the model training process, which could affect the model's robustness. The results are presented in Fig.3-4, which indicates that the outdoor temperature has the biggest impact on the load, followed by the time serial number. On the other hand, the correlation between wind direction and the load was too small (-0.0087), leading us to exclude this feature from the subsequent modeling. Examples of these data are presented in Table 3-1. For the following experiments, we divided the data into a training set comprising data from 2002 to 2009, a test set comprising data from 2010, and a validation set comprising 20% of the selected training set. All data with missing values will be interpolated using the KNN algorithm. As the numerical dimensions of different variables varied greatly, we normalized the data to map it uniformly onto the interval [0,1]. In the subsequent section, we will utilize this data set to construct a prediction model capable of forecasting the load for the next hour based on the input data of the previous N hours.

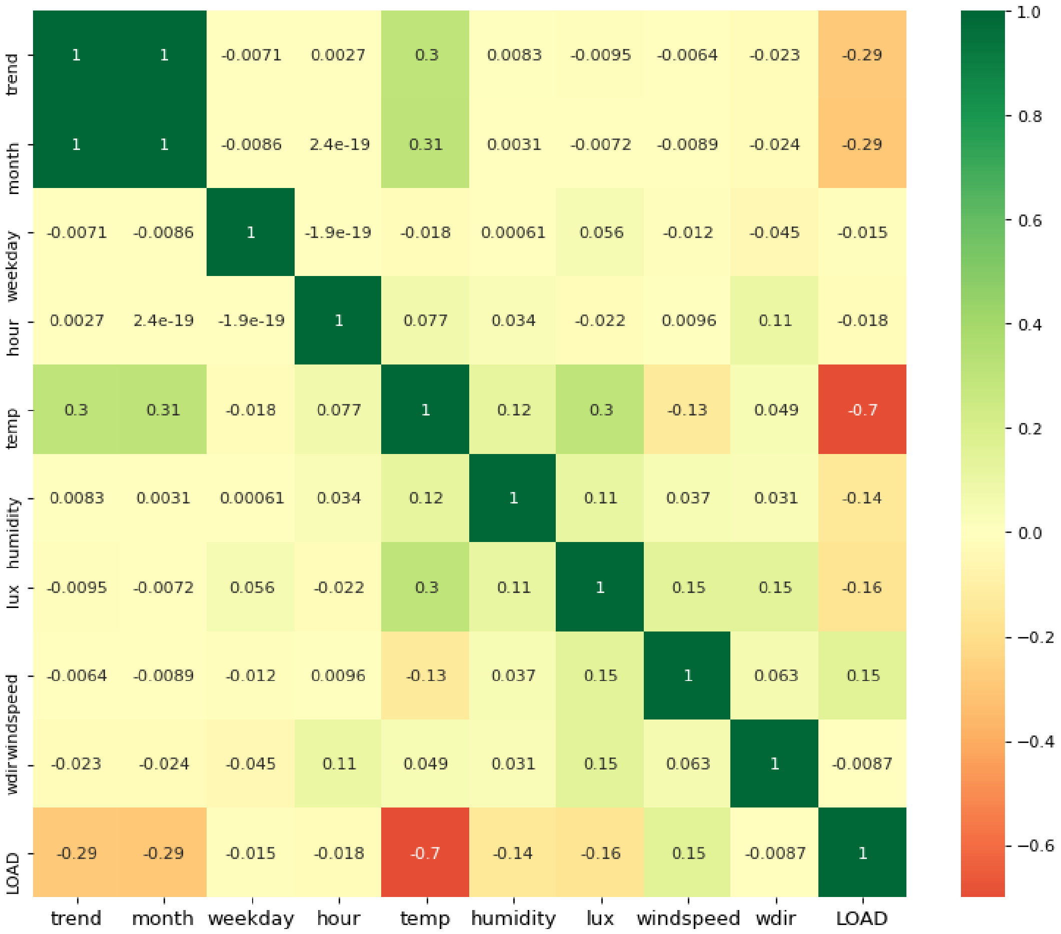


Fig. 3-4 Correlation analysis between the available features

Table 3-1 Example of the database

Trend	Month	Weekday	Hour	Temperature (◦C)	Humidity (%)	Illuminance $(\frac{W}{m_2})$	Windspeed (m/s)	Load (kW)
0	1	5	1	1.7	18	0	3.6	0
1	1	5	2	1.4	19	0	9.1	0
2	1	5	3	1.4	19	0	5.5	147.208
3	1	5	4	1.2	20	0	6.9	165.156
4	1	5	5	1.1	20	0	2.2	111.105
5	1	5	6	1.3	19	0	5.7	79.517
6	1	5	7	1.6	19	0	6.4	127.376

3.4 The dataset of Jono Zero Carbon Smart Community

3.4.1 Case Introduction

This study aims to train an RL agent using energy system data from a specific building to enable intelligent battery charging and discharging power regulation. The agent will automatically choose the optimal battery charging and discharging power based on energy demand, renewable energy generation, and real-time electricity price (RTP) trends. Through exploration of the optimal energy management strategy, the agent aims to reduce energy costs for users, increase the local consumption of renewable energy, and enable accurate energy use management of the building throughout the given period. The RL models proposed in this study were verified using the dataset of an actual Japanese house located in the "Jono Zero Carbon Smart Community" in Kitakyushu. The building has an energy system with PV panels (the capacity is 4.18 kW and the conversion efficiency is 19.6%), a storage battery (the capacity is 5.6 kW and the conversion efficiency is 90%), and connected to the public grid. Since the microgrid is a hybrid AC/DC network, inverters are used on the power lines of the battery and PV arrays for AC-DC conversion (The inverter's efficiency is 95%). Fig. 3-5 illustrates the concept of the PV-battery system. To ensure that the user's load demand is always met, the system is designed to adjust the discharge of the energy storage system or purchase electricity from the grid when the photovoltaic power is insufficient. Conversely, when the photovoltaic power generation exceeds the load demand, the excess power can be stored in the batteries or sold to the grid to maintain the load power balance of the user. Since there is no economic incentive to optimize battery scheduling and load transfer under a fixed electricity price. Therefore, this study's energy system regulation model is based on a scenario with RTP.

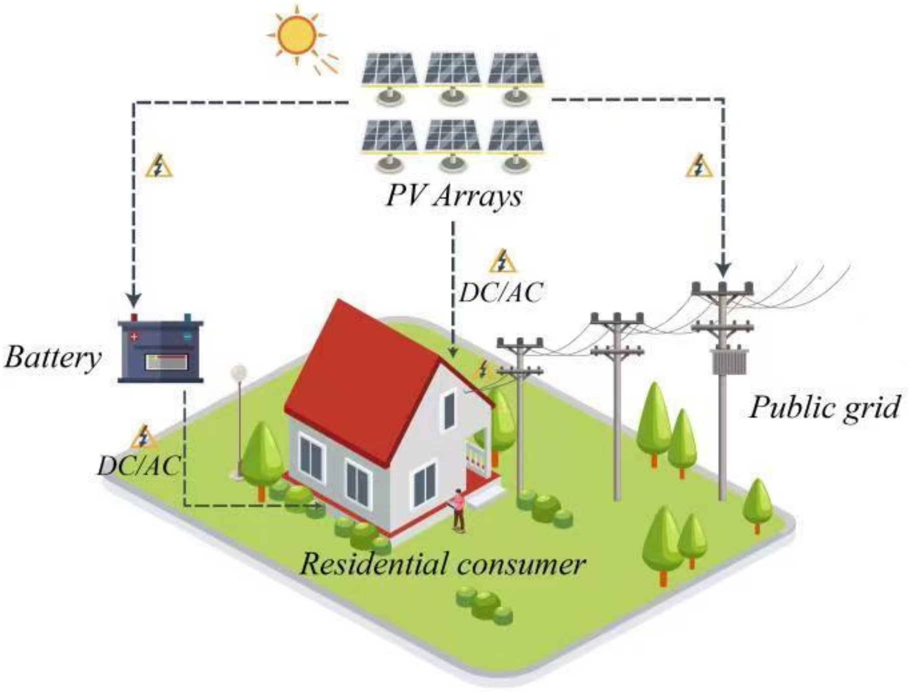


Fig. 3-5 Structure of the residential PV-battery system

3.4.2 Potential Analysis of Input Data Set

The study dataset was collected by the HEMS implemented in the target house, which includes about thirty months of hourly data from April 1, 2017, to September 30, 2019. These hourly data contain eight components: PV generation (kWh), power demand (kWh), electricity price (Yen), month, hour of day, outdoor temperature, illumination intensity, and humidity. In this study, the data collected from April 1, 2017, to September 30, 2018, will be utilized as the training set, whereas the data from October 1, 2018, to September 30, 2019, will be used as the test set. To ensure uniformity across the dataset, all data will undergo normalization during the preprocessing stage.

Fig. 3-6 shows the time-series changes in PV generation, electricity demand, and real-time electricity price on a 1-hour basis used by the training set in this experiment. It indicates that the target house's PV generation, electricity demand, and real-time electricity price had stochastic characteristics, and their values varied significantly in different months.

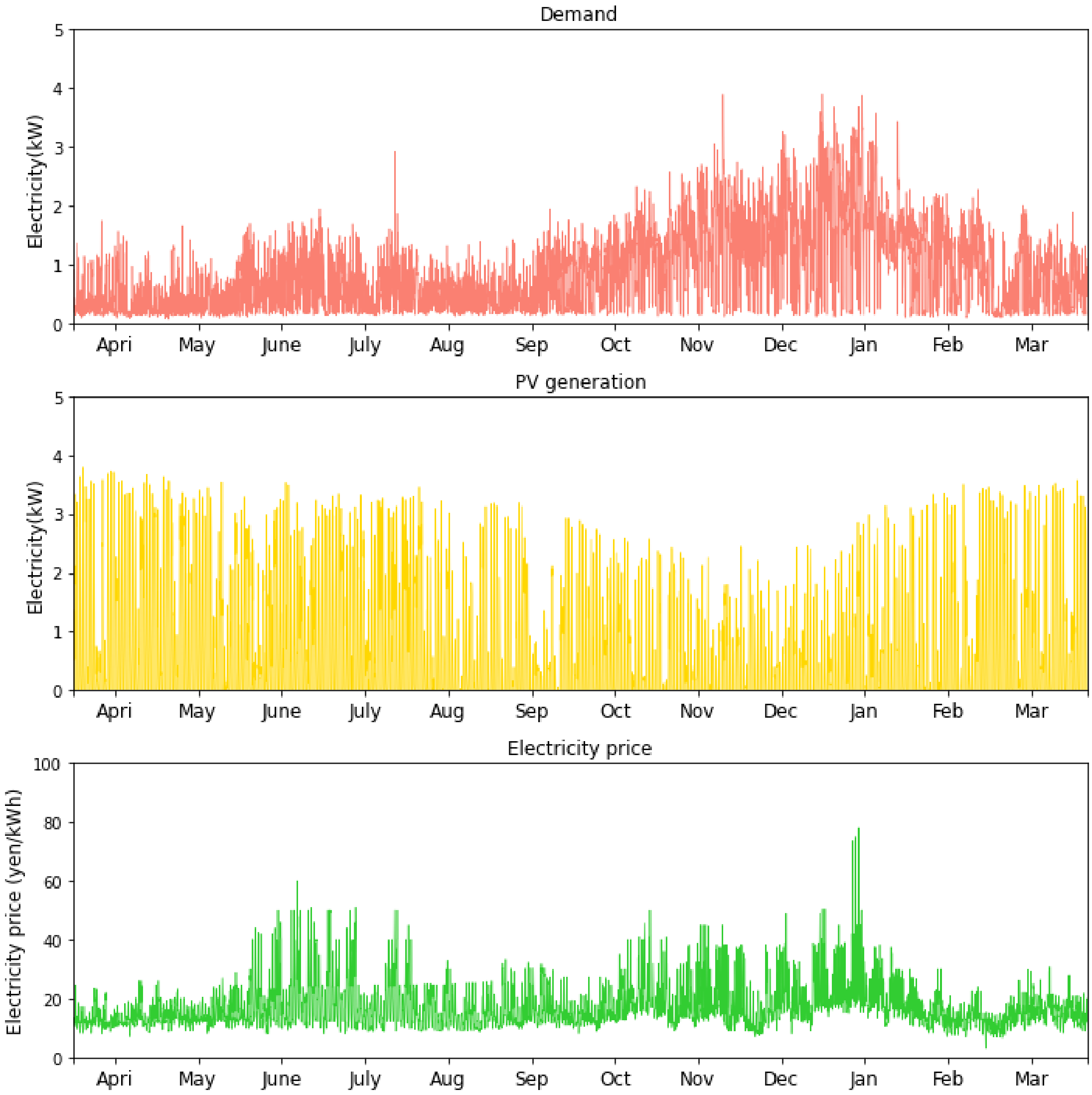


Fig. 3-6 Training Dataset: hourly PV generation, grid demand, and real-time electricity from April 2017 to March 2018

The essential characteristics of the dataset are always the basis for experimental design. Fig.3-7 is the overview of the target dataset. As shown in Fig.3-7(a), the electricity demand is significantly higher in winter than in other seasons (Heat pump heating is used in winter), and RTP has two significant peaks in winter and summer. Since demand, PV, and RTP have strong seasonal characteristics, evaluating the model over a short test set is not comprehensive. To overcome this, we took one year's data as the test set and divided it into three periods: cooling season , heating season, and transition season, to evaluate the model's performance separately. It can be seen from Fig.3-7 (b) that the distribution of PV generation and RTP has evident periodicity. For example, the peak of RTP usually occurs between 17:00 and 21:00, which is not coincide with the peak period of PV generation. It also indicates a vast optimization space for energy storage systems. In addition, it should be noted that since the nighttime electricity price of the ladder electricity price is low, the heat pump of this house is set to operate at night, so the mean load fluctuation is slight

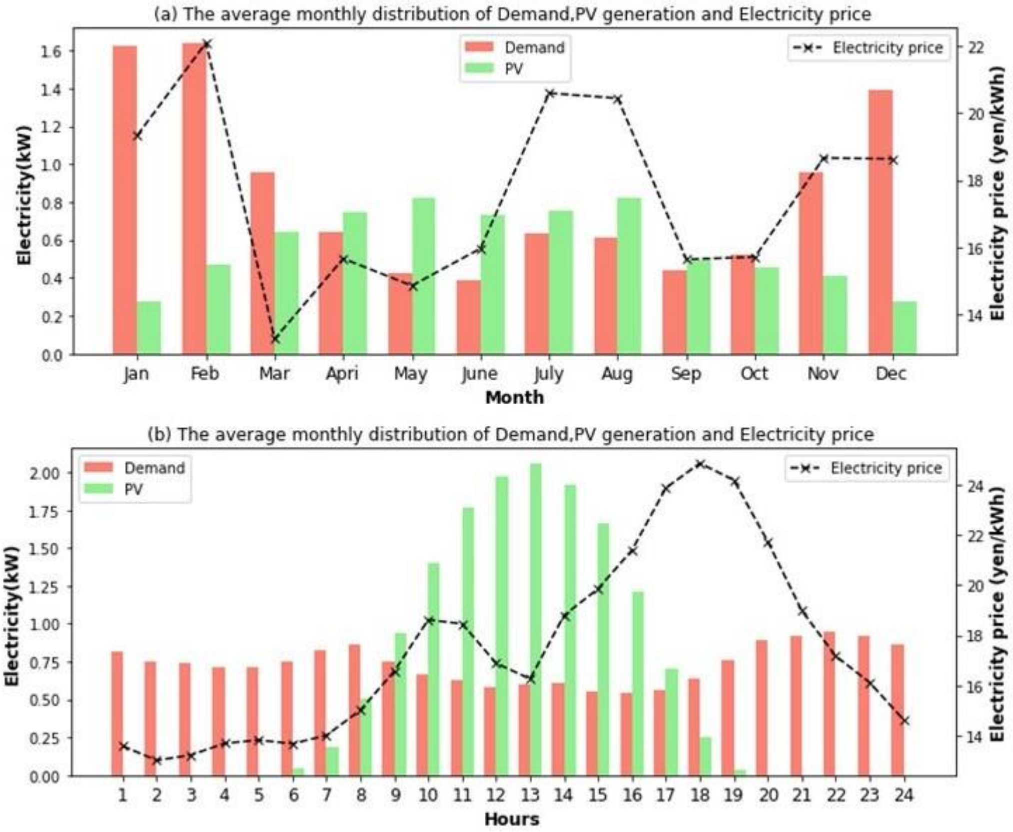


Fig. 3-7 Overview of the dataset: (a)The average monthly distribution of Demand, PV generation, and RTP, (b) The average hourly distribution of Load, PV generation, and RTP

Fig. 3-8 illustrates the distribution of electricity demand, PV generation, and RTP in different seasons, highlighting their significant differences. As shown in Fig. 3-8(a), the battery is charged between 11:00 and 15:00, while the power demand and RTP peak from 16:00 to 19:00 (with two RTP peaks at 10:00 and 18:00), indicating that there is a large optimization space for battery operation during this period. Similarly, Fig.3-8(b) indicates that the battery is charged from 8:00 to 17:00, coinciding with the RTP peak period. Therefore, full power discharge of the battery is preferred after 18:00, implying that the optimization space for battery operation is limited during this period. Finally, as shown in Fig. 3-8(c), there is a clear mismatch between the battery charging period (from 8:00 to 18:00) and the RTP peak period, suggesting a significant optimization space for battery operation during this period.

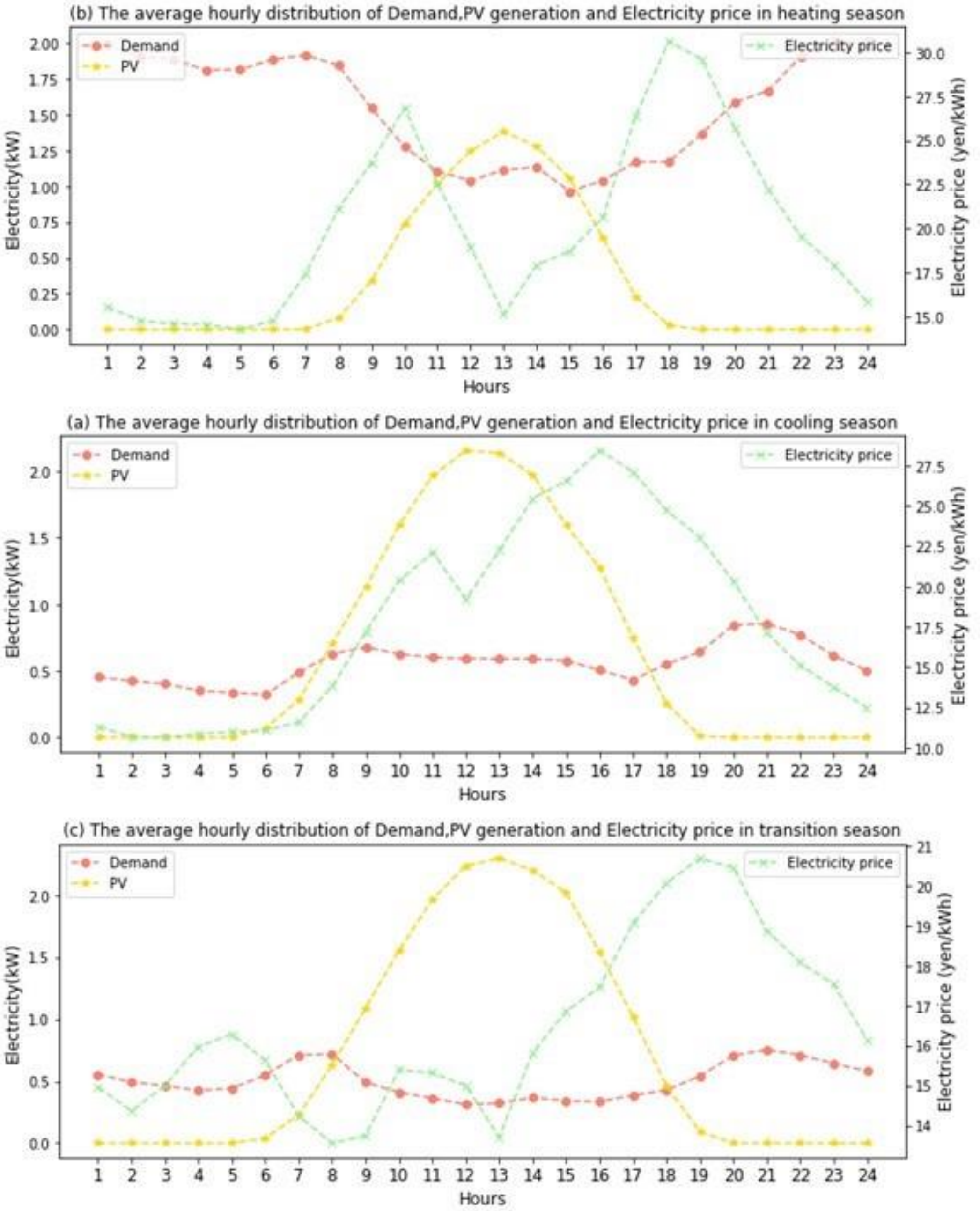


Fig. 3-8 Overview of the dataset: (a) The average hourly distribution of Demand, PV generation, and RTP in the heating season, (b) The average hourly distribution of Demand, PV generation, and RTP in the cooling season, (c) The average hourly distribution of Demand, PV generation, and RTP in the transition season

In this study, our first step will be to predict the target building's electricity demand, PV generation, and RTP. Subsequently, we will integrate these predicted results as new features into the training set of the reinforcement learning model. The process of constructing the datasets for these three prediction models will be explained in detail in this section. In this experiment, a Pearson correlation analysis was conducted on all features to identify and remove those with low correlation. Features with a correlation coefficient of less than 0.01 were considered low correlation features, which were excluded from the training set of the prediction model to avoid unnecessary information that might affect the model's robustness. Fig. 3-9 shows the results of the correlation analysis, where we can observe that power demand has the highest correlation with outdoor temperature due to the heat pump's function. The trend with a correlation coefficient lower than 0.01 was excluded. On the other hand, PV has the highest positive correlation with illuminance and the highest negative correlation with humidity, which is expected as high humidity is typically associated with rainy weather that affects PV generation. Furthermore, RTP has the highest correlation with hours, while the month feature was excluded as it had a low correlation.

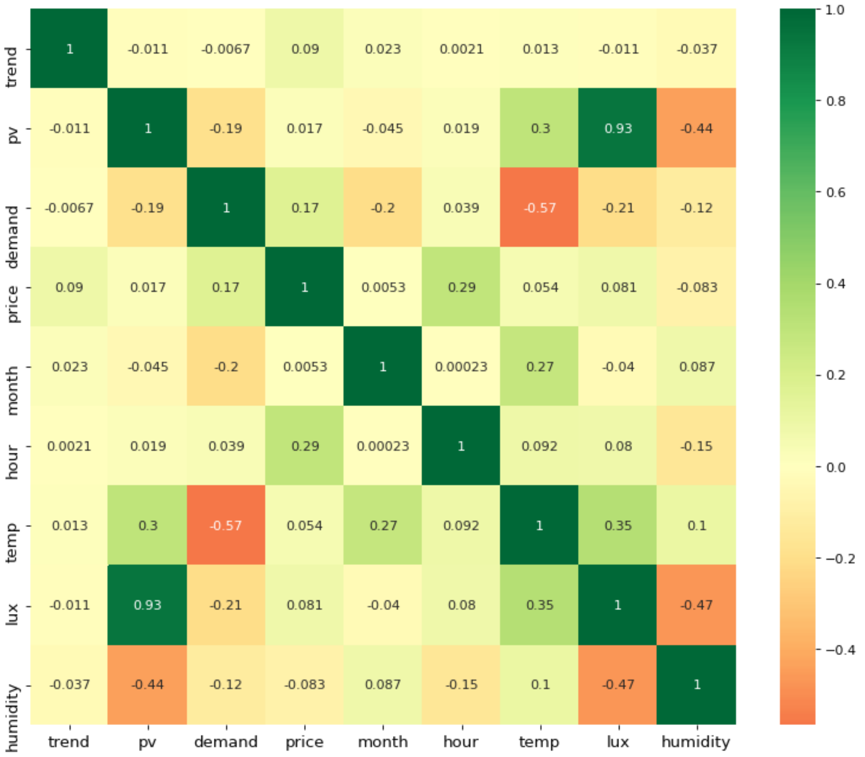


Fig. 3-9 Correlation analysis between the available features

In this study, we set the data collected from April 1, 2017, to September 30, 2018, as the training set, whereas the data from October 1, 2018, to September 30, 2019, will be used as the test set. All data with missing values will be interpolated using the KNN algorithm and normalized to a range of [0,1]. Samples of the dataset for electricity demand prediction, PV generation prediction, and real-time electricity price prediction are presented in Tables 3-3, 3-3, and 3-4, respectively:

Table 3-2 Samples of the dataset for electricity demand prediction

PV (kW)	Demand (kW)	Price (yen)	Month	Hour	Temperature (∘C)	Illuminance ($\frac{W}{m_2}$)	Humidity (%)
0.158	0.558	12.98	5	5	17.4	0.02	68
0.657	0.295	12.4	5	6	18.6	0.41	63
1.361	0.287	11.86	5	7	20.7	1.04	52
1.339	0.273	12.4	5	8	22.4	1.79	46
0.528	0.434	12.4	5	9	22.4	2.39	51

Table 3-3 Samples of the dataset for PV generation prediction

Trend	PV (kW)	Demand (kW)	Price (yen)	Month	Hour	Temperature (∘C)	Illuminance ($\frac{W}{m_2}$)	Humidity (%)
6	0.158	0.558	12.98	5	5	17.4	0.02	68
7	0.657	0.295	12.4	5	6	18.6	0.41	63
8	1.361	0.287	11.86	5	7	20.7	1.04	52
9	1.339	0.273	12.4	5	8	22.4	1.79	46
10	0.528	0.434	12.4	5	9	22.4	2.39	51

Table 3-4 Samples of the dataset for RTP prediction

Trend	PV (kW)	Demand (kW)	Price (yen)	Hour	Temperature (∘C)	Illuminance ($\frac{W}{m_2}$)	Humidity (%)
6	0.158	0.558	12.98	5	17.4	0.02	68
7	0.657	0.295	12.4	6	18.6	0.41	63
8	1.361	0.287	11.86	7	20.7	1.04	52
9	1.339	0.273	12.4	8	22.4	1.79	46
10	0.528	0.434	12.4	9	22.4	2.39	51

Reference

[1] Gangwar A K, Shaik A G. k-Nearest neighbour based approach for the protection of distribution network with renewable energy integration[J]. Electric Power Systems Research, 2023, 220: 109301.

[2] Li L, Chen X, Song C. A robust clustering method with noise identification based on directed K-nearest neighbor graph[J]. Neurocomputing, 2022, 508: 19–35.

[3] Wang Y, Pang W, Jiao Z. An adaptive mutual K-nearest neighbors clustering algorithm based on maximizing mutual information[J]. Pattern Recognition, 2023, 137: 109273.

[4] Peng S, Han W, Jia G. Pearson correlation and transfer entropy in the Chinese stock market with time delay[J]. Data Science and Management, 2022, 5(3): 117–123.

[5] Zhang M, Li W, Zhang L, Jin H, Mu Y, Wang L. A Pearson correlation-based adaptive variable grouping method for large-scale multi-objective optimization[J]. Information Sciences, 2023.

[6] Singh D, Singh B. Investigating the impact of data normalization on classification performance[J]. Applied Soft Computing, 2020, 97: 105524.

Chapter 4

POTENTIAL ANALYSIS OF THE ATTENTION-BASED LSTM MODEL IN BUILDING ENERGY SYSTEM

CHAPTER FOUR: POTENTIAL ANALYSIS OF THE ATTENTION-BASED LSTM MODEL IN BUILDING ENERGY SYSTEM

4.1 Introduction

According to statistics, building energy consumption comprises approximately 40% of the total energy consumption [1]. Additionally, the proportion of building carbon dioxide emissions is alarmingly high, accounting for 36% of the total emissions [2,3]. Among the various contributors to commercial building energy consumption, heating, ventilation, and air-conditioning (HVAC) systems are responsible for 40% (or higher) [4,5]. Given the correlation between energy consumption and energy-saving potential, research on energy-saving technology that combines big data and artificial intelligence has become one of the hotspots in recent years[6].

The energy consumption prediction during the HVAC design stage is accomplished through simulation modeling, considering several factors, such as building physical parameters, outdoor meteorological parameters, indoor environmental parameters, and room utilization rate[7]. However, the model-building process involves varying degrees of assumptions and simplifications, which may result in significant errors between predicted and actual energy consumption during operation. Therefore, it is challenging to achieve accurate load prediction through simulation modeling. Besides, building HVAC system operating parameters are primarily determined based on load predictions during the design stage. However, this can lead to low efficiency of the HVAC system during operation, resulting in significant energy consumption.

To achieve accurate load prediction, data-driven models are widely used in HVAC operation stage load prediction. In practical applications, HVAC load is affected by various factors, including the building itself, meteorological conditions (such as outdoor temperature and lighting), internal personnel activities (such as occupancy and power consumption), time lag, and actual use of air conditioning (such as control deviation and operation plan adjustment) [8]. Due to the influence of these factors, the HVAC load curve exhibits strong volatility, greater randomness, and less periodicity than a power load curve, posing a significant challenge for designing data-driven HVAC models. To address this problem, Wasim Iqbal et al. proposed the Negative Binomial Regression (NBR) model analysis method[9], and BinbinYu et al. proposed the Dynamic Spatial Panel Model (DSPM) [9]. Weiqing Li et al. proposed the Data Envelopment Analysis (DEA) and Entropy method [10] to analyze the interaction among various factors.

Compared to traditional building controllers, Model Predictive Control (MPC) can consider the current state of the building and HVAC system and incorporate predicted states and events into the control loop, which makes MPC a superior choice over other multivariate control methods in terms of improving building efficiency and providing greater comfort. As a result of their high performance, MPC is gaining more attention from researchers and operators and has become increasingly popular in the control of HVAC systems[11–13] [14] [15] [16] 18].

As illustrated in the terminology, prediction is the prerequisite for predictive control. Therefore, This paper aims to develop a new HVAC ultra-short-term energy consumption prediction model. To achieve this goal, we first conducted potential rule analysis and feature engineering for nine years of operation data of Kitakyushu Science and Research Park's (KSRP) Energy Center and constructed a data set for modeling. Then, we developed the LSTM model based on this data set and the A-LSTM model by adding the attention layer to the LSTM model. Besides, the RNN model, DNN model, and SVR model were also developed to compare performance. The TPE algorithm optimized the hyper-parameters of the above models to ensure prediction accuracy. Next, we used the data from the Energy Center from 2002 to 2009 as the training set and the data from 2010 as the test set to conduct experiments. We gradually reduced the size of the data sets to evaluate the performance of the above five models in different training sets. Finally, we also evaluated the small-scale prediction effects of the above five models under four typical operating modes. The core contribution of this work is the development of the ultra-short-term energy consumption prediction model of HVAC based on the attention mechanism. More specifically, the main achievements of this paper are listed below:

1) We adopted an existing distributed energy system's data as the research object and utilized the LSTM method with an attention mechanism to predict the HVAC load of the system. The advantage of this method is that it can quantitatively assign a weight to each specific time step in the time series feature, which improves the attention distraction defect of traditional LSTM. In addition, to verify this method's efficiency, A detailed case analysis and comparison of four advanced deep learning algorithms are presented, in which the baseline model is DNN, RNN, LSTM, and A-LSTMM. We used RMSE, MAE, and R^2 to evaluate these algorithms and proved that the efficiency of A-LSTM is optimal.

2) To comprehensively assess the potential of A-LSTM in HVAC load forecasting, we conducted a comparative evaluation that included seasonal energy efficiency assessment and data sensitivity analysis. Our results demonstrate that even when trained on two years of data, A-LSTM maintains high prediction accuracy.

3) We devised a hyperparameter optimization method using the TPE algorithm to optimize the hyperparameters of each baseline model training stage. This emphasizes these deep learning models to obtain optimal hyperparameters more efficiently, providing valuable insights for real-world implementations.

The organization of work is as follows. Section 4.2 describes the algorithm details of this paper. Section 4.3 outlines the detailed design of the experiments. Section 4.4 discusses the results of the case study. Finally, section 4.5 provides this paper's conclusions and future outlook.

4.2 Methodology

4.2.1 Attention Mechanism

Although the LSTM model has a memory function, it can save some time-series information. However, since the standard LSTM model uses the traditional encoder-decoder structure, it still has some limitations. When processing the time series x_t, the Encoder will first encode the input sequence into a fixed-length implicit vector h and give the same weight to the implicit vector. However, when the length of x_t increases, the average weight distribution will reduce the discrimination of x_t, and some important time-series information will be ignored in the process of training the model, thus affecting the prediction accuracy of the model.

The Attention mechanism is a deep learning method developed to overcome challenges in modeling complex data sequences. It was inspired by the human brain's ability to focus on important information and ignore distractions selectively. The mechanism enables the model to concentrate on relevant information at a given time while filtering out irrelevant data. Doing so can improve the model's performance on various tasks. The attention mechanism achieves its function by optimizing the encoder-decoder structure of a model. Depending on the specific application, it can be used alone or in combination with other models. The unit structure of the Attention mechanism is depicted in Fig. 4-1:

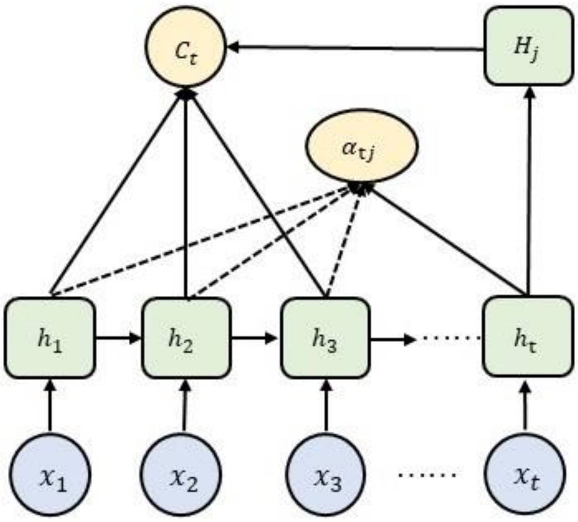


Fig. 4-1 The unit structure of the Attention mechanism

In Fig.3-1, $x_1, x_2, \ldots, x_t$ denotes the input sequence, T is the length of the input sequence, $h_1, h_2, \ldots, h_t$ represents the hidden state values of the corresponding input sequence $x_1, x_2, \ldots, x_t$, and α_{tj} denotes the attention weight of the historical input hidden states for the current input, which calculation formula is as follows:

$$e_{tj} = a\left(h_{i-1}, h_j\right) \tag{3-1}$$

$$\alpha_{tj} = \frac{exp(e_{tj})}{\Sigma_{k=1}^{t} exp(e_{tj})} \tag{3-2}$$

In the above Equations, i denotes the moment; j denotes the j element in the sequence; e_{tj} is the matching degree between the element to be encoded and other elements. We can obtain the feature vector C_t by computing the product of α_{tj}(attention probability weight) and h_i(historical input node's hidden state) and then summing over all historical inputs. The calculation formula for C_t is:

$$C_t = \sum_{j=1}^{T} \alpha_{tj} h_i \tag{3-3}$$

H_j denotes the true hidden state value of the final output node, and its calculation formula is:

$$H_j = H\left(C_t, h_j, x_j\right) \tag{3-4}$$

An encoder-decoder model with an Attention mechanism first learns the weight of each element from the sequence and then recombines the elements by weight. By assigning different weight parameters to each input element, the Attention mechanism can focus more on the relevant parts of the input element, thereby suppressing other useless information. Its biggest advantage is that it can consider global and local connections in one step and realize parallel computation, which is particularly important for big data computation. In this paper, the Bahdanau algorithm [17]is adopted to realize the Attention mechanism, and the structure of the A-LSTM model adopted is shown in Fig. 4-2:

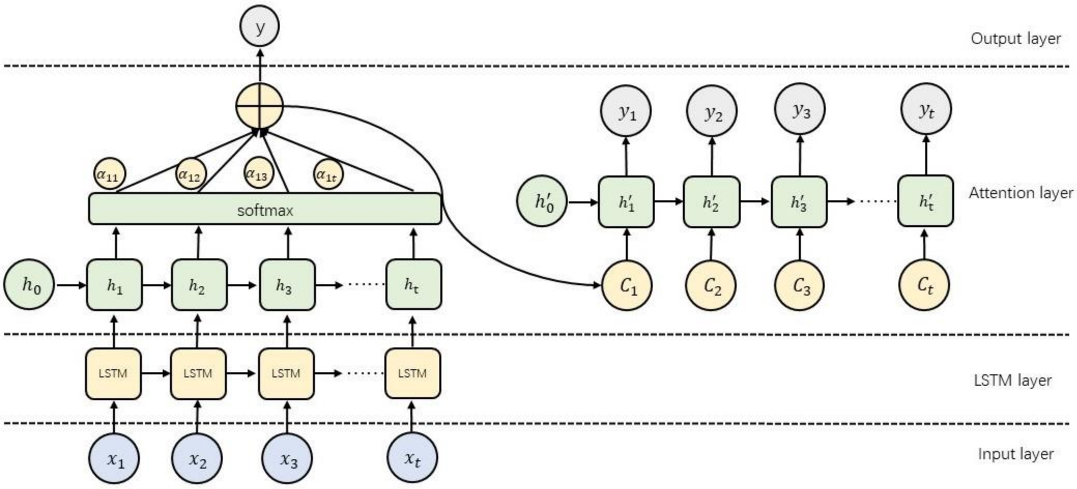


Fig. 4-2 The structure of the A-LSTM model

4.2.2 Tree-structured of Parzen Estimators (TPE)

The performance of machine learning models largely depends on the selection of hyperparameters. With the increase in model complexity and training data, automatic hyperparameter optimization is increasingly important in development [18]. Compared with traditional manual parameter adjustment, automatic parameter optimization has the following advantages :

- It reduces the workforce of development work.
- Improve the performance of machine learning models.
- Improve the reproducibility of the results[19].

This study used the Tree-Structured of Parzen Estimators (TPE) algorithm to optimize the model's hyperparameters. The TPE algorithm is an improved algorithm of the Bayesian optimization algorithm (BO). Since the TPE is based on a tree-structured representation of the hyperparameter space, enabling it efficiently explores many hyperparameters and identifies promising regions of the search space. One of the key advantages of the TPE algorithm is its ability to balance exploration and exploitation of the hyperparameter space. The tree-structured representation of the search space allows the algorithm to identify promising regions and focus its search on those areas while still exploring other regions to ensure that it does not miss potentially good solutions. Besides, TPE also solves the limitation of the traditional BO algorithm in dealing with classification and conditional parameters, so it has higher efficiency.

The main process of the TPE algorithm is to convert the hyperparametric space into the nonparametric density distribution first and then model the process $p(x|y)$. As shown in Equation (3-5), TPE uses two density distributions of Equation to define $p(x|y)$， $y < y^*$ indicates that the value of the objective function is less than the threshold, and $y \geq y^*$ denotes that the value of the objective function is greater than or equal to the threshold.

$$p(x|y) = \begin{cases} l(x) & \text{i}f \;\; y < y^* \\ g(x) & \text{i}f \;\; y \geq y^* \end{cases} \tag{3-5}$$

The calculation of Expected Improvement (EI) is shown in Equations (3-6)(3-7)(3-8).

$$E(x) = \int_{-\infty}^{y^*} (y^* - y) \frac{p(x|y)p(y)}{p(x)} \mathrm{d}y \tag{3-6}$$

$$\gamma = p(y < y^*) \tag{3-7}$$

$$P(x) = \int_R P(x|y)P(y)\,\mathrm{d}y \tag{3-8}$$

Substitute Eq. (3-7)(3-8) into Eq. (3-6) to get the final formula(3-9).

$$EI_{y^*}(x) = \left(\gamma + \frac{g(x)}{l(x)}(1 - \gamma)\right)^{-1} \tag{3-9}$$

It can be seen from formula (3-8) that point x^* with the largest Ei is the point with the smallest $\frac{g(x)}{l(x)}$. The TPE algorithm evaluates the improvement points according to $\frac{g(x)}{l(x)}$ in each iteration, and finally returns a point x^* with the largest EI. The corresponding process is shown in Fig. 4-3.

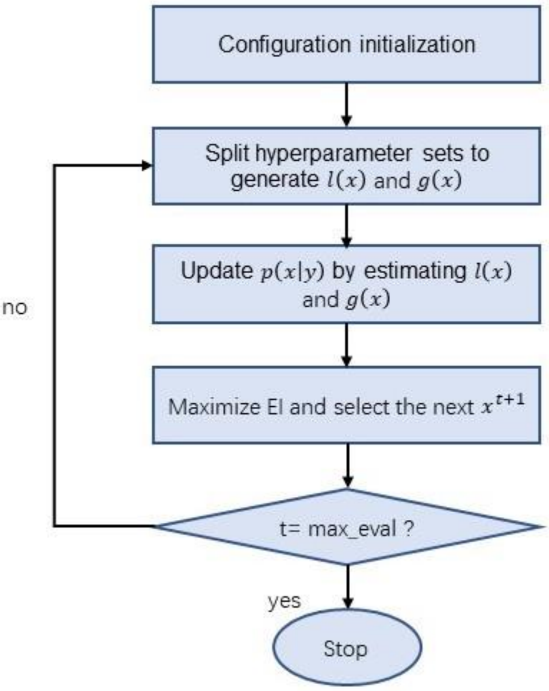

Configuration initialization
Split hyperparameter sets to generate $l(x)$ and $g(x)$
Update $p(x|y)$ by estimating $l(x)$ and $g(x)$
Maximize EI and select the next x^{t+1}
t= max_eval ?
no
yes
Stop

Fig. 4-3 Flowchart of the TPE algorithm

4.2.3 Adam Optimizer

As mentioned in Section 2.1.3, traditional Stochastic Gradient Descent (SGD) optimization algorithms have been known to be sensitive to the learning rate. To address this issue, Diederik Kingma of OpenAI and Jimmy Ba of the University of Toronto introduced a new optimization algorithm called Adam in 2014[20]. The Adam algorithm is designed to optimize the objective function of a deep neural network by adjusting the learning rate based on the gradient of the loss function, which maintains an exponentially decaying average of past gradients and their squared values. These estimates are then used to update the model parameters at each iteration. The Adam algorithm utilizes the first-order moment estimation of the gradient of each parameter for the loss function and the second-order moment estimation of each parameter's gradient to adjust each parameter's learning rate dynamically, and its calculation formula is given as follows:

$$m_t = \mu m_{t-1} + (1 - \mu) g_t \qquad (3\text{-}10)$$

$$n_t = v m_{t-1} + (1 - v) g_t^2 \qquad (3\text{-}11)$$

$$\widehat{m}_t = \frac{m_t}{1-\mu^t} \qquad (3\text{-}12)$$

$$\hat{n}_t = \frac{n_t}{1-v^t} \qquad (3\text{-}13)$$

$$\Delta\theta_t = -\frac{1}{\sqrt{\hat{n}_t+\varepsilon}} * \eta * m_t \qquad (3\text{-}14)$$

m_t and n_t denote the first and second-moment estimations of the gradient, respectively. The parameters $\widehat{m}_t$ and $\hat{n}_t$ are the correction terms applied to the first and second moment estimates. The bias-correction step in Eq.(3-12, 3-13) is necessary because the moment estimates are biased towards zero in the early stages of training. The variable $-\frac{1}{\sqrt{\hat{n}_t+\varepsilon}}$ is a constraint term that ensures that the variation of the learning rate is within a specified range.

Based on the analysis presented above, it is evident that the Adam algorithm offers a unique combination of the advantages provided by the Adagrad and RMSprop algorithms. Adam can obtain the adaptive learning rate required for each parameter by performing calculations on the parameters, thus effectively reducing memory consumption. This approach is particularly beneficial in scenarios with high model complexity, such as those with many layers in the target neural network. Consequently, Adam has become the most widely used optimization algorithm in practical applications. In the experiments conducted for this paper, the neural network training process employed the Adam algorithm to optimize the parameters.

4.3 Model Parameter Setting

In this section, we will use the LSTM model as an example to outline the model's construction methodology. After completing the parameter optimization of the LSTM baseline model, we incorporated an attention layer following the hidden layer of the LSTM model to build the A-LSTM model. We utilized the Hyperopt framework to implement the Tree-structured Parzen Estimator (TPE) algorithm, which automatically optimized the hyperparameters of all baseline models. Our programming language was Python, and we utilized TensorFlow 2.0 as our deep learning framework. Hyperopt is a Python library for hyperparameter optimization based on Bayesian optimization. It supports the optimization of continuous, discrete, and conditional variables. To use the Hyperopt framework, four parameters must be specified: the objective function to be optimized, the search space with super parameters, the Trials Database, and the search algorithm.

The LSTM baseline model requires optimization of four parameters: the time step L of each LSTM layer (determined by the length of previous data), the size of the hidden unit m of each layer, the size of the batch processing b during training (with default same hidden unit for each layer in the two-layer LSTM structure), and the drop rate of the Dropout layer. To establish the range of L, we first conducted autocorrelation analysis on load data to detect data cycle patterns, as depicted in Fig.3-9. The X-axis and Y-axis of Fig. 4-4 represent "hours" and "autocorrelation coefficient," respectively. Our analysis revealed that the overall autocorrelation of the load showed a cycle of decline, with autocorrelation peak every 24 hours. As such, we defined the conditional parameters of L as {12,24,36,48}.

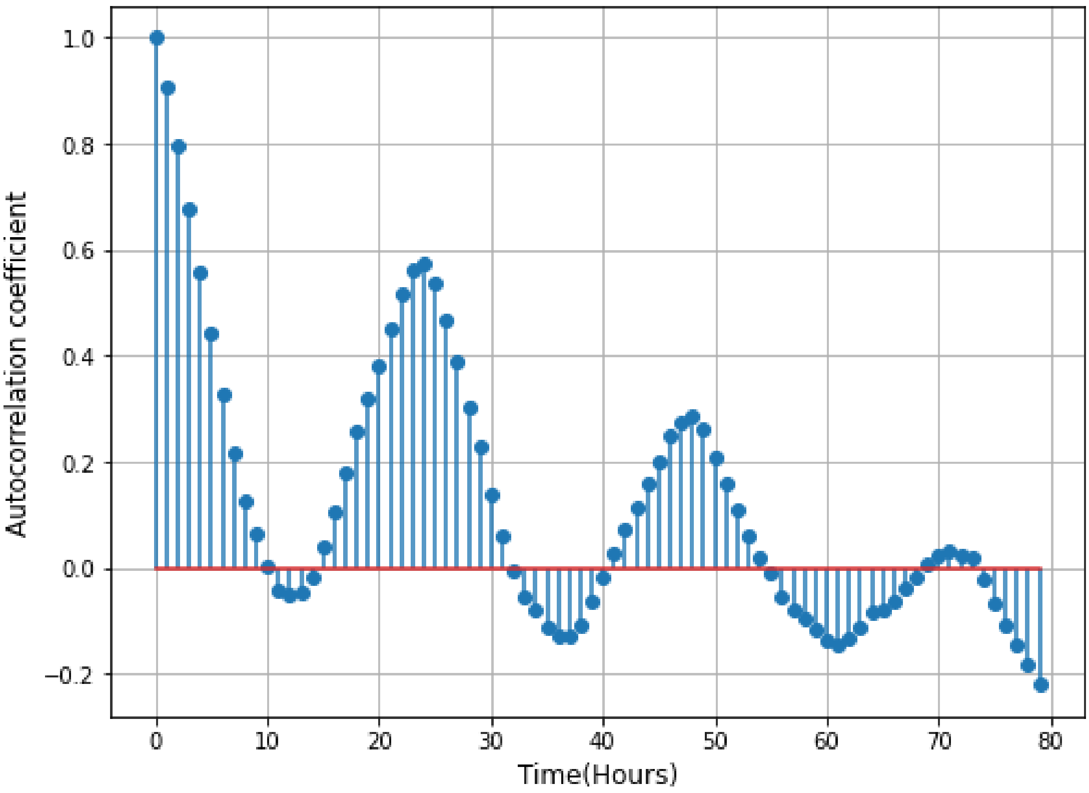


Fig. 4-4 Load autocorrelation analysis results

To avoid overfitting, we added a dropout layer after each LSTM layer, with the conditional parameters of drop rate set as {0.2, 0.3, 0.4, 0.5}. Due to limited computational resources, we set the conditional parameter sets of m and b based on empirical methods to ensure prediction accuracy: m ∈{32,64,128,256} and b ∈{32,64,128,256}[21]. We input the aforementioned conditional parameters into the Hyperopt framework and utilized the TPE algorithm to optimize the model's super parameters. Fig. 4-5 shows the optimized RNN, LSTM, and A-LSTM model structure, with the hyperparameters of these models determined by the TPE algorithm.

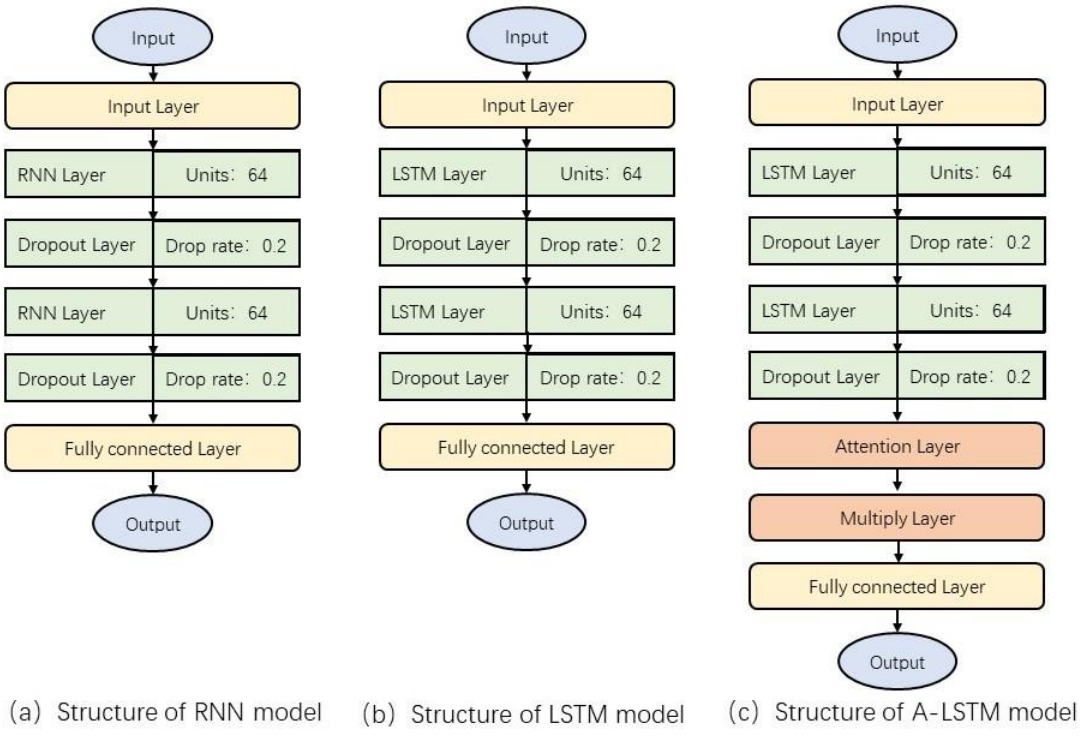
In addition to the three recurrent neural network models, we also included the DNN and SVR models for horizontal comparison. Like the three models, the DNN and SVR models take all the data 24 hours before the time slot t and the time data at the time slot t as input to predict an output the load data at the time slot t. Table 4-1 displays the optimal hyperparameters of the DNN model optimized by the TPE algorithm, while Table 4-2 shows the optimal hyperparameters of the SVR model optimized by the TPE algorithm. The topology of the five models above depends on the characteristics of the KSRP dataset. The model's structure and hyperparameters should be adjusted for other datasets according to the data. As the primary focus of this study was to investigate the potential of the LSTM model with the attention mechanism in the field of load prediction, we aimed to simplify the model's topological structure and input characteristics as much as possible to improve the model's generalization ability and reduce the required computational force while ensuring prediction accuracy.

Table 4-1 Hyperparameters for the DNN model

Model	Layer1 Units	Layer2 Units	Layer3 Units	Batch size	Drop rate
DNN	128	64	32	64	0.2

Table 4-2 Hyperparamers for the SVR model

Model	Kernel	C	gamma
SVR	RBF	97.227588	0.001032



(a) Structure of RNN model (b) Structure of LSTM model (c) Structure of A-LSTM model

Fig. 4-5 Structure of RNN, LSTM, and A-LSTM model

4.4 Result and Discussion

To evaluate the time series prediction effect of the A-LSTM model on this data set, we compared it with the same type of RNN, LSTM model, and DNN model without memory function in this experiment. All models have been trained and tested five times, and the final data used for comparison is the average of the 5 test results to reduce the errors caused by random numbers. Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R-Square Value (R2_SCORE) were used as indicators of the evaluation model, which were calculated according to Eq. (3-1), (3-2), and(3-3). The y_i denotes the real observations, $\bar{y}_i$ denotes the average of the observed value, $\tilde{y}_i$ denotes the predicted value, N denotes the number of test samples.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2} \tag{3-1}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \tilde{y}_i| \tag{3-2}$$

$$R^2 = 1 - \frac{\Sigma_{i=1}^{n}(y_i - \tilde{y}_i)^2}{\Sigma_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{3-3}$$

4.4.1 Annual Prediction Performance Comparison

We trained the models using eight years of data (from 2002 to 2009) as the training set and evaluated their effectiveness in load forecasting for 2010. Table 4-3 displays the results of the five models in predicting the annual data for 2010. The results demonstrate that the four deep learning models have significantly improved prediction accuracy, particularly compared to the SVR algorithm. This indicates that deep learning models offer clear performance advantages over traditional machine learning models when tackling complex load prediction scenarios. Although the four deep learning model's prediction results were close, the A-LSTM model had the highest accuracy. Compared to the second-best predicted LSTM, A-LSTM showed a 3.06% decrease in RMSE, a 6.54% decrease in MSE, and a 0.43% increase in R^2 value. It's worth noting that the system operates under low or zero load for a significant amount of time in a year, resulting in very small prediction errors during these periods, which may reduce the overall average prediction error. We will delve deeper into this phenomenon in the following section.

Table 4-3 Comparison of prediction errors between different models

	SVR	DNN	RNN	LSTM	A-LSTM
RMSE(kW)	106.490	78.788	80.638	77.340	74.977
MAE(kW)	85.632	48.752	48.979	47.929	44.793
R²	0.854	0.922	0.918	0.925	0.929

4.4.2 Prediction Performance Comparison at High and Low Loads

To provide a more intuitive evaluation of the prediction accuracy of the A-LSTM model, we have selected the prediction results for four typical periods in 2010 for comparison. These periods include two high-load periods (heating and cooling) and two low-load periods (heating and cooling), each consisting of two weeks of experimental results. The detailed results can be found in Table 4-4.

Table 4-4 Performance of A-LSTM models compared to the baseline model.

	The high-load period in the heating season (2010.1.7~2010.1.13 and 2010.1.31~2010.2.6)				
	SVR	DNN	RNN	LSTM	A-LSTM
RMSE(kW)	123.508	103.035	93.796	96.558	86.876
MAE (kW)	102.083	72.996	66.898	66.192	62.262
R²	0.737	0.817	0.839	0.848	0.870
	The low-load period in the heating season (2010.3.1~2010.3.7 and 2010.3.8~2010.3.14)				
	SVR	DNN	RNN	LSTM	A-LSTM
RMSE(kW)	99.075	79.973	81.230	78.325	73.427
MAE (kW)	78.165	48.027	53.816	49.141	47.206
R²	0.521	0.687	0.678	0.701	0.737
	The high-load period in the cooling season (2010.7.15~2010.7.21 and 2010.8.1~2010.8.7)				
	SVR	DNN	RNN	LSTM	A-LSTM
RMSE(kW)	108.849	78.129	79.959	75.294	68.361
MAE (kW)	88.199	57.575	54.966	50.127	46.529
R²	0.800	0.896	0.892	0.904	0.921

	The low-load period in the cooling season (2010.5.15~2010.5.21 and 2010.5.21~2010.5.27)				
	SVR	DNN	RNN	LSTM	A-LSTM
RMSE(kW)	68.277	54.021	50.452	53.515	47.805
MAE (kW)	56.184	35.185	31.627	38.351	29.855
R²	0.642	0.776	0.805	0.781	0.824

Table 4-4 shows that the A-LSTM model has the best prediction performance during the high-load heating period, followed by the RNN model. Specifically, the RMSE of A-LSTM decreases by 10.02%, MSE decreases by 5.93%, and the R² value increases by 2.59% compared to the second-best RNN model. Fig. 4-6 presents the complex load prediction curve during this period, showing that the heat load fluctuates sharply, with a peak load at around 9 a.m. every day, followed by a load decline trend. However, the prediction results indicate that none of the five models can accurately predict this morning's peak load, with DNN being the best for predicting the peak load. Conversely, during the period of medium and low load, the fitting effect of the A-LSTM prediction curve is the best, while the fitting effect of the DNN model is poor in this stage.

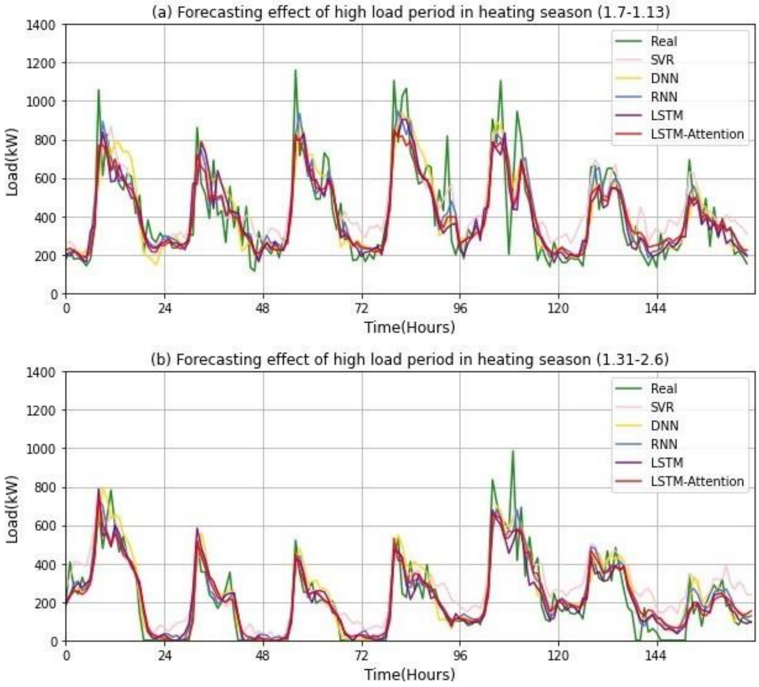


Fig. 4-6 Forecasting effect of the high-load period in the heating season

As shown in Table 4-4, the A-LSTM model has the best prediction performance during the low-load heating period, followed by the LSTM model. Specifically, during this period, the RMSE of A-LSTM decreased by 6.25%, MSE decreased by 3.94%, and the R^2 value increased by 5.14% compared to the sub-optimal LSTM prediction. Fig.4-7 presents the detailed forecast data during this period, showing that the heat load fluctuation is relatively slow relative to the high load period. In addition to the first load peak at around 9 a.m. every day, there will also be a second load peak at a random time in the afternoon. While all five load prediction models accurately predict the time point of the two load peaks, none of them can accurately fit the peak. Among the models, DNN is relatively the best fit for peak load, but its fit for the low load stage is poor, while A-LSTM is more balanced in overall fitting performance.

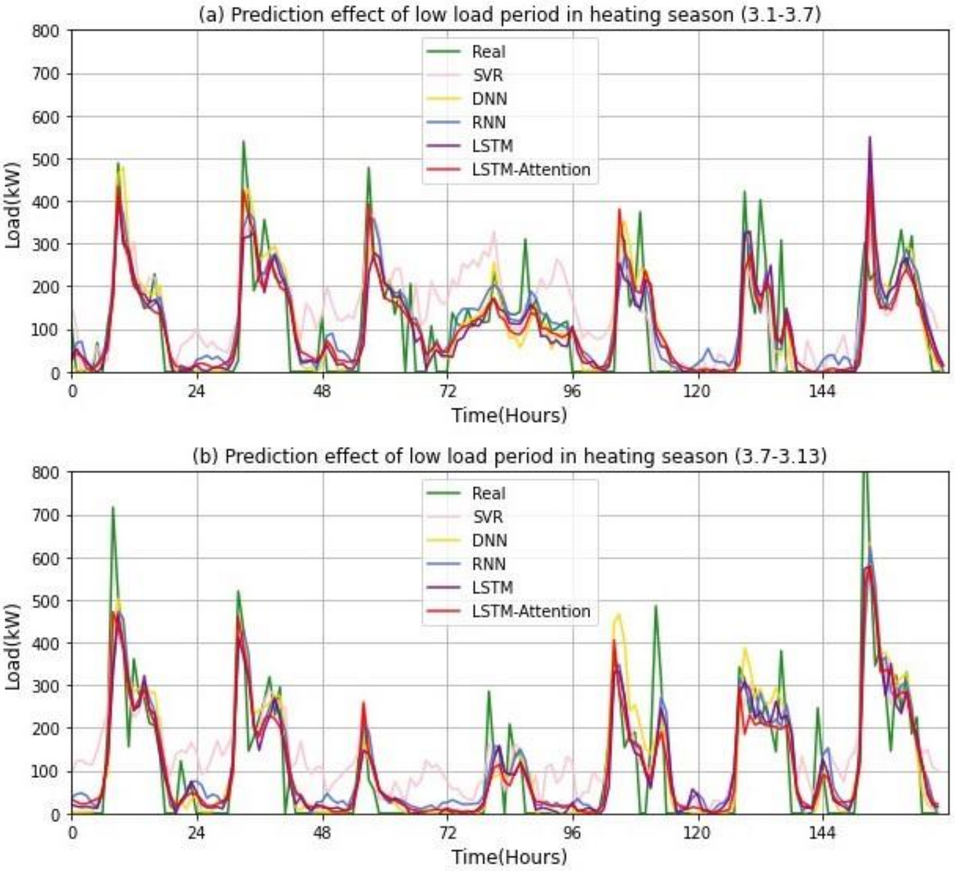


Fig. 4-7 Forecasting effect of the low-load period in the heating season

Table 4-4 shows that A-LSTM has the best predictive performance during high-load cooling periods, with LSTM coming in second. During this period, A-LSTM exhibited a 9.21% decrease in RMSE, an 8.80% decrease in MSE, and a 1.88% increase in R^2 compared to sub-optimal predictions. Fig. 4-8 shows detailed forecast data for this period, revealing that the cooling load curve fluctuates relatively steadily compared to the heating load curve. The load peak occurs between approximately 8 and 2 o'clock daily, with increasing or decreasing load peaks during other periods. This characteristic results in higher accuracy for the five prediction models in the cooling season than in the heating season. For example, using the A-LSTM model, the RMSE during the peak cooling period decreased by 18.515 kW, the MAE decreased by 15.733 kW, and the R^2 value increased by 0.051 compared to the peak heating period. Furthermore, A-LSTM exhibited better fitting for peak load during this period than in the heating season.

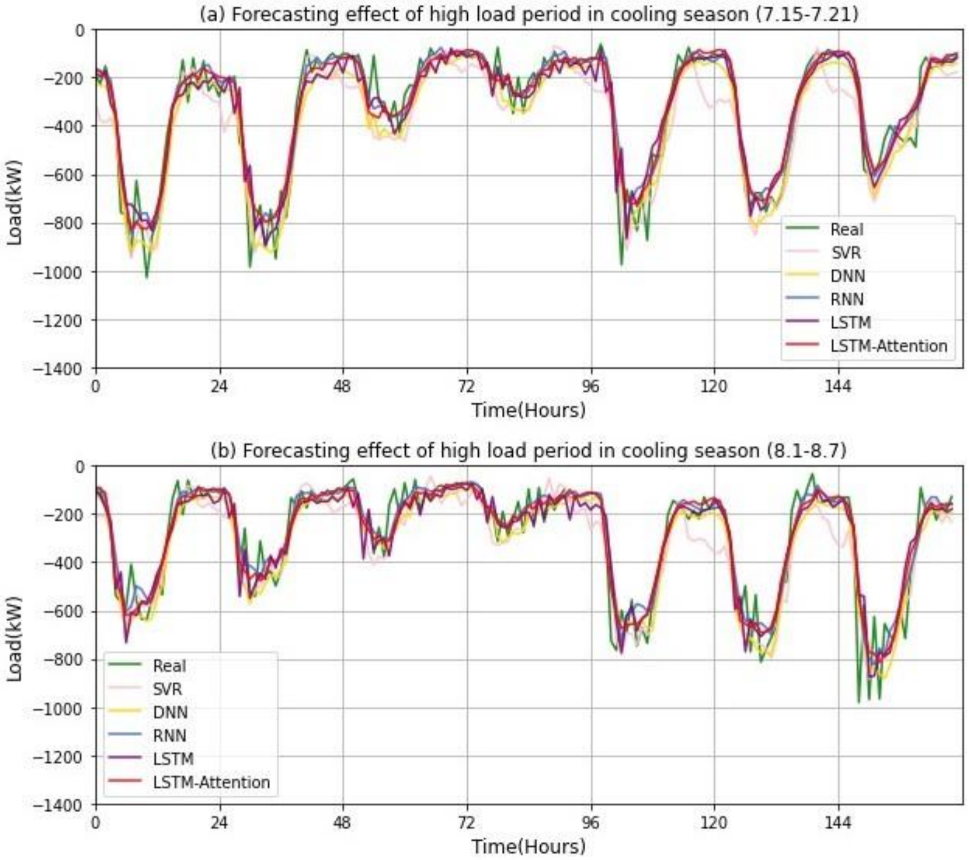


Fig. 4-8 Forecasting effect of the high-load period in the cooling season

Table 4-4 demonstrates that A-LSTM exhibits the most accurate predictions during the high-load cooling period, followed by RNN. Specifically, the RMSE of A-LSTM decreases by 5.24%, MSE decreases by 4.47%, and the R^2 value increases by 2.36% compared to RNN's suboptimal prediction. The detailed forecast data for this period is illustrated in Fig. 4-9. Compared to the high cooling load period, this period's load fluctuation is more severe, with multiple peaks appearing randomly in the morning. Despite this challenge, all five prediction models can predict the timing of the load peaks, but none can accurately predict the peak itself. DNN, in particular, exhibits a better prediction effect on the random load peaks. A-LSTM's R^2 value reveals that the prediction accuracy decreases by 0.097 compared to the high cooling load period, indicating a decrease in prediction accuracy. It can be concluded that all five models have higher prediction accuracy for cooling load than for heat load, especially during the high cooling load period, where the prediction effect is the best.

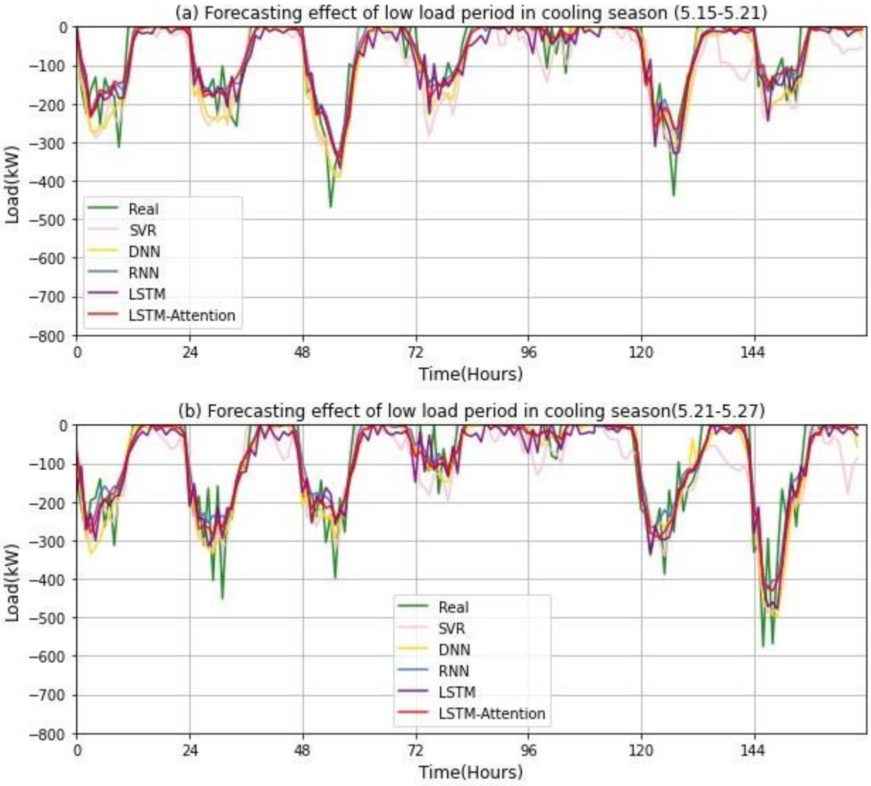


Fig. 4-9 Forecasting effect of the low-load period in the cooling season

We performed a statistical analysis of the characteristic correlation coefficients during the cooling and heating seasons to elucidate this phenomenon. Fig.4-10 provides valuable insights into the reasons for the observations, as mentioned in section 3.5.1. As depicted in Fig. 4-10, environmental factors correlate more with a load than time factors. Furthermore, the correlation between load and environmental factors varies significantly across different seasons, particularly concerning the correlation between outdoor temperature and humidity. Notably, the load during the refrigeration season displays the highest correlation with environmental factors, which enhances the ability of the prediction model to acquire knowledge from existing data. This could explain why the model's prediction accuracy is superior during the refrigeration season compared to the heating season. Specifically, for the data during the cooling season, the correlation during high load periods is stronger than that during low load periods. In contrast, the opposite is true for the data during the heating season. These differences ultimately manifest in the prediction accuracy of the model. This suggests that environmental factors have a greater impact on the output of cooling load, while the laws of human production and life influence the output of heat load. While the existing data may not fully reflect the laws of human production and life, it comprehensively captures the impact of environmental factors, which could account for this phenomenon.

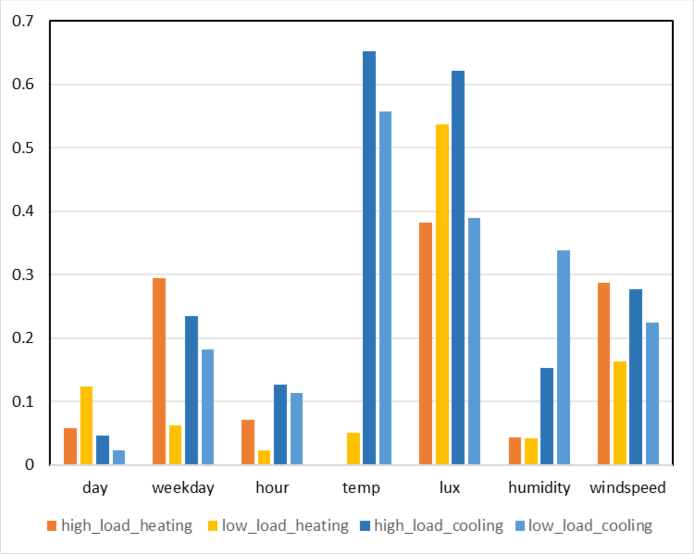


Fig. 4-10 Absolute value of correlation coefficients of load and other features in the database between heating season and cooling season

4.4.3 Data Sensitivity Analysis

To explore the influence of the size of the training set on the prediction accuracy of the model, we also conducted the following experiments: the topological structure of the five models used in the previous experiments was kept unchanged, and the training set was gradually reduced in a unit of 2 years. The 2010 data were used as the test set to evaluate each model separately. The experimental results are shown in Fig.4-11, from which we can see that each model's prediction accuracy decreases with the training set's reduction. The result showed that the prediction accuracy of the A-LSTM model was the best when the data from 8, 6, and 4 years were used as the training set. Compared with the suboptimal LSTM model, its RMSE decreased by 3.06%, 10.86%, and 11.29%, respectively. R^2 value increased by 0.43%, 2.21%, and 2.57%, respectively. However, when two years of data were used as the training set, the prediction accuracy of the A-LSTM model decreased significantly, and its prediction accuracy was only better than that of the SVR model. This indicates that the prediction accuracy of the A-LSTM model will increase with the length of the training set, and the prediction accuracy of 4-year or 6-year data sets of the A-LSTM model has obvious advantages compared with other models.

It is worth emphasizing that the LSTM model outperforms the other models when trained on a 2-year data set. Even when the training set is reduced from 8 years to 2 years, the RMSE of the LSTM model only increases by 17.6%, and the R^2 value decreases by 4.1%, which is the smallest change compared to the other models, which indicates that the LSTM model exhibits strong robustness and effective learning ability with limited training data. Such a finding provides valuable theoretical guidance for constructing accurate prediction models in future studies.

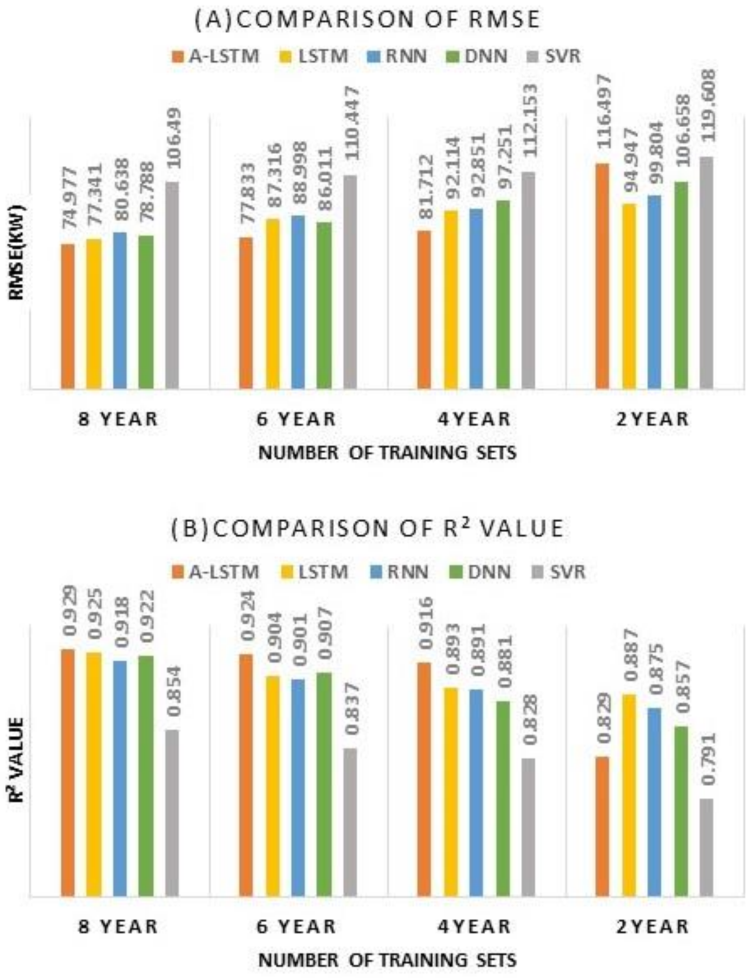


Fig.4-11 The prediction accuracy of each model under different lengths of the training set

There are two limitations in the current study. First, due to the limited computational force, the search method adopted in the hyperparameter optimization in this paper is based on conditional parameters rather than the search based on the assignment interval. Although the conditional parameters based on the empirical method can ensure the prediction's accuracy, there is undeniable room for further optimization of the super parameters of the models. Secondly, the Bahdanau algorithm adopted is the classical gradient-based method to obtain the optimal solution. The gradient-based method has the advantage of easy implementation, but at the same time, it will bring premature convergence and the problem of falling into a locally optimal solution. Therefore, there is room for further optimization at the algorithm level of this study.

4.5 Conclusion

Predictive control has become an increasingly popular approach to enhancing building energy efficiency. Prior studies have highlighted the complex structure of HVAC systems in large buildings, which are susceptible to random environmental factors and human activities, posing significant challenges in predicting short-term HVAC loads. In this study, we analyzed nine-year operational data from the KSRP Energy Center to identify underlying patterns and factors influencing HVAC load. The factors used to establish the model were determined based on the results of Pearce correlation calculations. The findings revealed that outdoor temperature significantly influenced the cooling and heating load of HVAC systems. In contrast, the daily peak load was concentrated in a specific period.

To address these challenges, this study proposed a novel model that combines the attention mechanism with the LSTM neural network. The model was implemented in the following steps: First, we used autocorrelation analysis to determine the previous 24 hours of data as the time step to predict the load for the next hour. Second, we employed the TPE optimization method to optimize the hyperparameters of the baseline LSTM model. The test results demonstrated that the LSTM model with two layers of 64 neurons exhibited the best prediction performance. Third, we integrated the attention layer into the baseline LSTM model to develop the A-LSTM model. Finally, we established RNN, DNN, and SVR models as horizontal comparison objects.

Finally, we used data from the KSRP Energy Center between 2002 and 2009 as the training set and the data from 2010 as the test set to evaluate the performance of the five models mentioned above. The results demonstrated that the A-LSTM model had the highest prediction accuracy. Compared to the LSTM model, the A-LSTM model achieved a 3.06% reduction in overall RMSE, a 6.54% decrease in MSE, and an increase of 0.43% in R^2 value. Furthermore, we observed that the advantage of the A-LSTM model was most significant when the length of the training set was between 4 and 6 years. However, when the size of the training set was reduced to 2 years, the prediction accuracy of the A-LSTM model decreased sharply, indicating that it has limitations in predicting small sample data. To validate the impact of low-load and zero-load samples on the experimental results, we evaluated four typical operating mode samples in 2010. We drew a graph to show the predicted results. The results showed that the A-LSTM model had better prediction accuracy for refrigeration load than heating load. Its performance was better during high load periods compared to low load periods.

In summary, the proposed A-LSTM model combining attention mechanism and LSTM neural network demonstrated improved accuracy in predicting large buildings' cooling and heating loads and peak load. However, further analysis is needed to understand the model's performance under different operating modes, and the attention mechanism algorithm can be further optimized.

Future work can focus on applying the A-LSTM model to real-time HVAC energy-saving control using a model-free deep reinforcement learning algorithm. By using the predicted value as the agent's observed state, the RL model's control accuracy can be improved. This approach can potentially overcome the limitations of model-based control systems, which require accurate modeling of controlled objects.

Reference

[1] Mohsin M, Taghizadeh-Hesary F, Panthamit N, Anwar S, Abbas Q, Vo X V. Developing Low Carbon Finance Index: Evidence From Developed and Developing Economies[J]. Finance Research Letters, 2020: 101520.

[2] Jradi M, Veje C, Jørgensen B N. Deep energy renovation of the Mærsk office building in Denmark using a holistic design approach[J]. Energy and Buildings, 2017, 151: 306–319.

[3] Mohsin M, Hanif I, Taghizadeh-Hesary F, Abbas Q, Iqbal W. Nexus between energy efficiency and electricity reforms: A DEA-Based way forward for clean power development[J]. Energy Policy, 2021, 149: 112052.

[4] Kim B, Yamaguchi Y, Kimura S, Ko Y, Ikeda K, Shimoda Y. Urban building energy modeling considering the heterogeneity of HVAC system stock: A case study on Japanese office building stock[J]. Energy and Buildings, 2019, 199: 547–561.

[5] Wang Z, Liu J, Zhang Y, Yuan H, Zhang R, Srinivasan R S. Practical issues in implementing machine-learning models for building energy efficiency: Moving beyond obstacles[J]. Renewable and Sustainable Energy Reviews, 2021, 143: 110929.

[6] Sun H, Awan R U, Nawaz M A, Mohsin M, Rasheed A K, Iqbal N. Assessing the socio-economic viability of solar commercialization and electrification in south Asian countries[J]. Environment, Development and Sustainability, 2021, 23(7): 9875–9897.

[7] Ma W, Fang S, Liu G, Zhou R. Modeling of district load forecasting for distributed energy system[J]. Applied Energy, 2017, 204: 181–205.

[8] Wang J, Huang G, Sun Y, Liu X. Event-driven optimization of complex HVAC systems[J]. Energy and Buildings, 2016, 133: 79–87.

[9] Iqbal W, Tang Y M, Chau K Y, Irfan M, Mohsin M. Nexus between air pollution and NCOV-2019 in China: Application of negative binomial regression analysis[J]. Process Safety and Environmental Protection, 2021, 150: 557–565.

[10] Li W, Chien F, Hsu C-C, Zhang Y, Nawaz M A, Iqbal S, Mohsin M. Nexus between energy poverty and energy efficiency: Estimating the long-run dynamics[J]. Resources Policy, 2021, 72: 102063.

[11] Mayne D Q. Model predictive control: Recent developments and future promise[J]. Automatica, 2014, 50(12): 2967–2986.

[12] Sultana W R, Sahoo S K, Sukchai S, Yamuna S, Venkatesh D. A review on state of art development of model predictive control for renewable energy applications[J]. Renewable and Sustainable Energy Reviews, 2017, 76: 391–406.

[13] Zhan S, Chong A. Data requirements and performance evaluation of model predictive control in buildings: A modeling perspective[J]. Renewable and Sustainable Energy Reviews, 2021, 142: 110835.

[14] Zhao X, Gao W, Qian F, Ge J. Electricity cost comparison of dynamic pricing model based on load forecasting in home energy management system[J]. Energy, 2021, 229: 120538.

[15] Yang J J, Yang M, Wang M X, Du P J, Yu Y X. A deep reinforcement learning method for managing wind farm uncertainties through energy storage system control and external reserve purchasing[J]. International Journal of Electrical Power & Energy Systems, 2020, 119: 105928.

[16] Chang F, Chen T, Su W, Alsafasfeh Q. Control of battery charging based on reinforcement learning and long short-term memory networks[J]. Computers & Electrical Engineering, 2020, 85: 106670.

[17] Bahdanau D, Cho K, Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate[J]. Computer ence, 2014.

[18] Nguyen H-P, Liu J, Zio E. A long-term prediction approach based on long short-term memory neural networks with automatic parameter optimization by Tree-structured Parzen Estimator and applied to time-series data of NPP steam generators[J]. Applied Soft Computing, 2020, 89: 106116.

[19] Luo H, Cai J, Zhang K, Xie R, Zheng L. A multi-task deep learning model for short-term taxi demand forecasting considering spatiotemporal dependences[J]. Journal of Traffic and Transportation Engineering (English Edition), 2021, 8(1): 83–94.

[20] Kingma D P, Ba J. Adam: A Method for Stochastic Optimization[J]. 2014. arXiv,2014[2023-03-07].

[21] Wang Z, Hong T, Piette M A. Predicting plug loads with occupant count data through a deep learning approach[J]. Energy, 2019, 181: 29–42.

Chapter 5

OPERATIONAL OPTIMIZATION FOR BUILDING ENERGY SYSTEMS USING VALUE-BASED REINFORCEMENT LEARNING

CHAPTER FIVE: OPERATIONAL OPTIMIZATION FOR BUILDING ENERGY SYSTEMS USING VALUE-BASED REINFORCEMENT LEARNING

5.1 Introduction

Japanese electricity market achieved a full retail market liberalization in April 2016, and regional power companies and a rising number of retailer electrical operators entered the wholesale electricity market. Japan Electric Power Exchange (JEPE) is a voluntary platform for companies to buy and sell electricity. JEPX provides a day-ahead wholesale market to trade electricity through a price auction mechanism. Since 2017, the volume of gross bidding has increased continuously. JEPX volume shared more than 40% of Japan's electricity demand by October 2021. Under the promotion of this policy, the grid-connected residential PV-battery system has become Japan's fastest-growing renewable energy technology. By 2021, the total installed PV capacity in the residential field had reached 50,000 MW[1].

However, despite the economic advantages of RES, both PV and wind power generation are susceptible to environmental factors in the form of high randomness and intermittency, which creates a significant challenge to power utilities [2]. For example, many new PV capacities had led to PV generation curtailment in certain seasons; thus, Kyushu and Tohoku Electric Power Companies had reduced the feed-in tariff to suppress PV capacity, which was contrary to the original intention of improving the PV penetration level of the grid. Implementing energy storage systems (ESS) effectively solves this problem, which can positively impact power balance and grid reliability[3]. The smart ESS has attracted rising attention to deal with the imbalance of energy supply and demand. An optimized smart storage strategy can be more efficient in energy management by optimizing the scheduling strategy of variable renewable energy generation, energy demand, and real-time electricity price [4–6]. It can not only increase the utilization of RES and use the ESS for energy arbitrage but also achieve load balancing, peak shaving, and frequency regulation[7].

The electricity pricing strategy can play an essential role in the power market mechanism and significantly impact the application and promotion of intelligent ESS. Currently, the commonly used pricing strategies in the Japanese electricity market include multistage price (MEP) and time of use price (TOU). MEP is a charge based on monthly electricity consumption and the unit price increases as the electricity consumption increases. TOU determines the electricity price according to different periods of electricity consumption. Currently, second-order electricity price is widely used[8]. With the continuous development of the electricity market and the close combination of electricity futures trading, the real-time electricity price (RTP) mechanism has been adopted by more and more regions and countries. The RTP is the development trend of the future power market, which has several advantages compared with MEP and TOU, such as adjusting the load curve, improving the utilization rate of grid equipment and generation efficiency, and guiding consumers to form reasonable electricity consumption patterns [9,10]. Therefore, whether it can be compatible with the RTP mechanism to be used in the future is also an essential factor in evaluating an intelligent ESS.

To sum up, there are the following challenges in applying an energy control system (EMS) to the existing Grid-Connected Residential PV-battery system in Japan: Firstly, the electricity demand of each residence is affected by the personal living habits of the users. It is impossible to control the ESS of each house through a unified model, and independent modeling would significantly increase the cost. Secondly, The uncertainty of real-time electricity prices poses a new challenge to the scheduling strategy of the intelligent EMS. Since RES's equipment cost is also an essential factor hindering its promotion, users' willingness to comply may be further reduced if it can't arbitrage.

Based on the above challenges, we propose a reinforcement learning (RL) approach to managing intelligent EMS. RL is an essential branch of machine learning. It uses an agent to interact with the environment constantly, learn valuable knowledge from mass data, and obtain the optimal control strategy[11]. In recent years, deep learning technology has injected new vitality into RL. With the help of the deep neural network (DNN)[12,13], the RL agent can identify sample features more efficiently, which can help to solve complex problems[14]. In addition, traditional control methods, such as mixed-integer linear programming, require building the objective function and the mathematical equation of the system's dynamic process. In contrast, the RL avoids the complex manual modeling process as a model-free method based on data-driven. Therefore, it is more suitable for optimizing the residential hybrid energy system operation.

The core contribution of this work is the development of state of an art RL agent-based controller for cost-effective energy management of the grid-connected residential PV-battery system. More specifically, the main achievements of this paper are listed below:

- Although a growing number of RL-based applications have emerged for microgrid management, only a few studies have aimed to compare different RL algorithms based on actual data. To fill this gap, all data used in the simulations were collected from a real Japanese house equipped with a home energy management system (HEMS). We conduct experiments to evaluate and compare the performance of three value-based RL algorithms. The baseline model uses the strategy used in actual buildings. The result shows that all algorithms can reliably reduce the electricity cost, and the D3QN method could outperform the other two methods by learning only an 18-month sample.
- To evaluate the optimization effect of these methods in a natural microgrid environment, we created a new simulation environment based on a real residential microgrid, including all details, such as the charge and discharge efficiency of lithium-ion batteries and the energy conversion of transformers and inverters.
- A model-based algorithm is developed for the operational optimization of residential ESS. The advantage of this method is that domain knowledge is used in RL model configuration to improve data utilization, the exploration scope of the agent is reduced, and the data utilization rate is improved. In addition, the RL agent is also used to explore the systems that are difficult to model, giving full play to the advantages of the model-free method. Detailed case analysis and comparison of model-free and model-based

implementations of three advanced value-based RL algorithms are presented. We evaluated the efficiency of these algorithms in terms of energy cost and proved that the efficiency of MB-D3QN is optimal.

The organization of work is as follows. Section 2 describes the algorithm details of this paper. Section 3 outlines the case study, including the detailed models and parameters design. Section 4 discusses the results of the case study. Finally, section 5 provides this paper's conclusions and future outlook.

5.2 Methodology

As mentioned in Chapter 2, Reinforcement Learning (RL) algorithms can be classified based on various criteria. Firstly, an agent can be categorized as a model-based or a model-free algorithm. The model-free approach learns the policy from historical data. At the same time, the model-based method requires estimating the environment's transition model to learn the policy based on this model [15].

Secondly, RL can be divided into two categories: on-policy and off-policy. The on-policy approach selects the best option from the current policy, even if it may not be the best choice in the overall phase. On the other hand, the off-policy method employs two policies: the behavior policy and the target policy. In real-world scenarios where data is difficult to collect and slow to generate, such as building energy management systems, the off-policy method can leverage historical data to generate an optimal policy to achieve specific goals. In contrast, the on-policy approach has a high learning cost characteristic in control systems. Therefore, we choose the off-policy method as the main research method in this paper.

5.2.1 Deep Q Networks (DQN)

Deep Reinforcement Learning (DRL) is a machine learning technique that combines the strengths of deep learning and reinforcement learning. The integration of neural networks enables learning feature representations from deep learning while using reinforcement learning to leverage rewards based on interactions with an environment. In environments with finite sets of states and actions, a Q-table is commonly used to store the value of states and actions. However, the Q-table becomes impractical for infinite sets of states and actions due to the exponential growth of dimensions, a phenomenon commonly called the "dimension disaster."

In 2013, Google Deepmind introduced a variant of the Q-learning algorithm, which was the first to successfully apply deep learning to learn control policies from high-dimensional sensory inputs. Subsequently, the DQN algorithm was further improved in 2015 by incorporating a replay buffer and two neural networks, which helped overcome instability issues observed in previous approaches[16]. This advancement in DRL allowed for the achievement of human-level performance in six Atari games, demonstrating the potential for DRL in solving complex problems in various domains[14]

The pseudo-code of the DQN algorithm is presented in Algorithm 1. The key idea behind the algorithm is to use a replay buffer to save each transition information (such as (s_t, a_t, s_{t+1}, r_t)) in a buffer B and train the deep Q network using a random batch of m transitions sampled from the buffer rather than the latest transition to reduce overfitting due to correlated experiences. The neural network's memory R, weight ω, and bias parameters are initialized to initiate training. The agent obtains the initial observation of the state s_1 and preprocesses the sequence $\psi_1 = \psi(s_1)$. The ε-greedy policy is used for trial and error. The agent selects the action a_t with the maximum Q-value output by the current Q network and randomly chooses an action within the $(1 - \varepsilon)$ possibility range. The selected action is then executed, and the reward r_t is computed, followed by observing the next state. The squared error between the target and the predicted value of the neural network is then calculated in the loss function, and the parameters of the current neural network are updated using the gradient descent method. Following this, the weight is frozen for several time C steps and replaced by copying the actual Q network weight ω to stabilize training.

```
Algorithm 1: Deep Q-learning
Initialize replay memory $\mathcal{R}$ to capacity $\mathcal{B}$
Initialize current network $Q_c(\chi(s_t), a; \omega)$ and target network $Q_t(\chi(s_t); \omega^-)$ with random weights $\omega$
  and $\omega^-$
for $episode = 1, M$ do
    Obtain initial observation of state $s_1$ and preprocessed sequenced $\psi_1 = \psi(s_1)$
    for $t = 1, T$ do
        With probability $\epsilon$ select a random action $a_t$
        therwise select $a_t = \max_a Q_c(\psi(s_t), a; \omega)$
        Execute action $a_t$ and compute reward $r_t$ and observe the new state $s_{t+1}$
        Store transition $(\psi_t, a_t, r_t, \psi_{t+1})$ in $\mathcal{B}$
        Randomly sample a mini-batch of $m$ transitions $(\psi_i, a_i, r_i, \psi_{i+1})$ from $\mathcal{R}$
        Set $y_i = \begin{cases} r_i & \text{for terminal } \psi_{i+1} \\ r_i + \gamma \max_{a'} Q_t(\psi_{i+1}, a'; \omega^-) & \text{otherwise} \end{cases}$
        Update $\omega$ by minimizing the loss:
        $L_i(\omega_i) = E\left[(y_i - Q_c(\psi_i, a_i; \omega_i))^2\right]$
        Update $\omega^-$ using the sampled policy gradient:
        $\nabla_{\omega_i} L_i(\omega_i) = E\left[(r_i + \gamma \max_{a'} Q_t(\psi_{i+1}, a_{i+1}; \omega^-) - Q_c(\psi_i, a_i; \omega_i)) \nabla_{\omega_i} Q_c(\psi_i, a_i; \omega_i)\right]$
        Every $C$ steps reset $\omega^- = \omega$
    end
end
```

Despite its success in many domains, the Deep Q-Network (DQN) algorithm has limitations, and one of its primary drawbacks is the overestimation of the Q-value. This overestimation can significantly impact the decision accuracy of the DQN algorithm. To address these challenges, researchers have proposed a series of improved algorithms for DQN, including Double DQN, Dueling DQN, and D3QN. In the next section, we will provide a detailed explanation of how these algorithms function.

5.2.2 Double-Deep Q Networks (DDQN)

The standard DQN algorithm utilizes the max operator to select and evaluate actions using the same Q network. However, this approach is susceptible to selecting overestimated values, as described in Ref [14]. To address this issue, Ref [17] proposes the Double DQN (DDQN), which decouples the action selected from the activity evaluation. Specifically, the main neural network selects the best next action among all available next actions, and the target neural network evaluates this action to determine its Q-value. The target Q value is defined as Eq. (5-1).

$$Target\ Q = r(s_t, a_t) + \gamma Q\big(s_{t+1}, max_{\alpha_{t+1}} Q(s_{t+1}, a_{t+1}; \theta_t); \theta'\big) \qquad (5\text{-}1)$$

Where, $r(s_t, a_t)$ is reward; γ is the attenuation coefficient and two Q functions, each with different weights, a Q function with weights θ_t to select the action in the argmax while the other function with a set of weights θ' to evaluate the action.

5.2.3 Dueling Deep Q Networks (Dueling DQN)

The network architecture of the Dueling DQN is similar to that of the DQN. However, the output is computed differently. Specifically, while the DQN outputs the Q-value directly, the Dueling DQN network generates two separate functions: the predictive state value function $V(s_t; \theta_t, \beta)$ and the relative action advantage function $A(s_t, a_t; \theta_t, \alpha)$. In this context, which denotes the value of a state and represents the advantage of action a in a given state s at time t.

The architecture of a Dueling DQN consists of two streams: a value stream and an advantage stream. The value stream computes the state-value function, while the advantage stream computes the advantage function. These two streams are combined to obtain the final Q-values for each action in a given state. Thus, the Q-function can be described as:

$$Q(s_t, a_t) = A_\pi(s_t, a_t) + V_\pi(s_t) \qquad (5\text{-}2)$$

For the optimal policy $a_{t+1} \epsilon Q(s_t, a_t)$, $A(s_t, a_t) = 0$, then $Q(s_t, a_t) = V_\pi(s_t)$, and Dueling DQN network outputs are described by Eq.(5-3):

$$Q(s_t, a_t; \theta_t, \alpha, \beta) = V(s_t; \theta_t, \beta) + A(s_t, a_t; \theta_t, \alpha) \qquad (5\text{-}3)$$

Where Q is the value of the current network, θ_t is Q network parameters, s is the current state, a is the current action, while α and β the fully connected layer parameters of the two streams.

To solve the problem that is difficult to map from Q values to unique $V(s_t; \theta_t, \beta)$ and $A(s_t, a_t; \theta_t, \alpha)$ values, in [59] proposed an approach for making the advantage function estimator has zero advantage while making a decision, which can be achieved by subtracting the average $\bar{A}$

from the action value, through training V and A is more effective and robust than the standard DQN network structure, the Q function can be expressed as Eq. (5-4):

$$Q(s_t, a_t; \theta_t, \alpha, \beta) = V(s_t; \theta_t, \beta) + [A(s_t, a_t; \theta_t, \alpha) - \frac{1}{|A|}\sum_{a'} A(s_t, a'; \theta_t, \alpha)] \qquad (5\text{-}4)$$

5.2.4 Dueling Double-Deep Q Networks (D3QN)

Dueling Double-Deep Q Networks (D3QN) is an extension of the Dueling DQN algorithm that combines the advantages of Dueling DQN and Double DQN[18]. In D3QNs, the value stream and advantage stream of the Dueling DQN architecture are each implemented with two separate networks, resulting in four networks, which allows for a more stable and accurate estimation of the Q-values for each action in a given state. This technique can effectively address several shortcomings of the DQNs, such as the overestimation problem. Thus, the target Q value of Q network Y_t^{D3QN} is the same as DDQN, which can be denoted by Eq. (5-1) before and appropriate parameters should be trained by minimizing the loss function, which can be expressed as follows:

$$L^{D3QN}(\theta_t) = E[\left(Y_t^{D3QN} - Q_t(s_t, a'; \theta_t, \alpha, \beta)\right)^2] \qquad (5\text{-}5)$$

D3QN updates the training parameters θ_t of the Q network with stochastic gradient descent and copies θ_t to the target network's parameters at every fixed step. Update parameters in the training process can be formulated as follows:

$$\theta_{t+1} = \theta_t + \alpha * E[\left(Y_t^{D3QN} - Q_t(s_t, a'; \theta_t, \alpha, \beta)\right) * \frac{\partial Q_t(s_t, a'; \theta_t, \alpha, \beta)}{\partial \theta_t}] \qquad (5\text{-}6)$$

Additionally, D3QNs use prioritized experience replay, which prioritizes important experiences for replay based on their TD error, improving the algorithm's learning efficiency and convergence speed.

5.3 Reinforcement Learning-based Energy Storage Scheduling System

This section formulates the optimal energy storage scheduling problem of the grid-Connected Residential PV-battery system based on MDP(Markov decision process), including the complete method and parameters design used in the simulation experiment. Fig. 5-1 shows the microgrid's energy storage scheduling decision control process based on reinforcement learning. The RL objective of this experiment is to obtain the optimal energy storage scheduling strategy. The actions of the energy storage system will be the decision variables, and the agent will constantly learn interactively with its environment to adjust and improve the agent's behavior during this process. Table 1 summarizes the RL terminology that will be used in subsequent chapters.

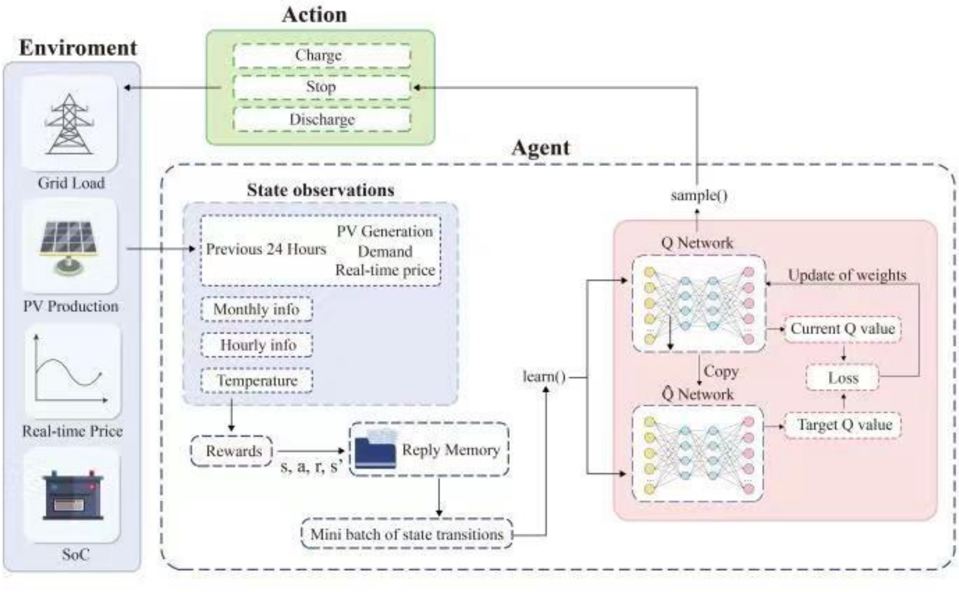


Fig. 5-1 The energy storage scheduling decision control process

5.3.1 Baseline control model

To ensure the stability and security of the energy storage system, the energy storage scheduling strategy made by the RL agent must meet the physical constraints of batteries. The battery model parameters used in the simulation are shown in Table 5-1, which are based on the user questionnaire statistics. Battery operation modes include charging and discharging activities, which can be expressed by a mathematical equation as follows:

$$E_{battery}^{t+1} = \begin{cases} E_{battery}^{t} + \eta_{cha} * P_{battery}^{t} * \Delta t & if\ charge \\ E_{battery}^{t} + \frac{P_{battery}^{t} * \Delta t}{\eta_{disscha}} & if\ discharge \\ E_{battery}^{t} & otherwise \end{cases} \tag{5-7}$$

Where, $E_{battery}^{t}$ denotes the state of battery storage; t denotes the current time point; $P_{battery}^{t}$ >0 refers to charge capacity, $P_{battery}^{t}$<0 refers to discharge capacity; η_{cha} denotes the charge efficiency and $\eta_{disscha}$ denotes the discharge efficiency.

Table 5-1 The battery parameters in simulation.

Parameter	Values
Battery storage capacity	5.6 kWh
Battery efficiency	90%
Battery charging/discharging rate	2kW
SoC (State of charge)	20%~95%

In addition, the battery model must observe the battery capacity constraints and power charge/ discharge rate constraints. The constraint values in this simulation are all taken from the actual customer data, and to simplify the model, a fixed charge-discharge rate is used in this simulation. If the maximum battery capacity is $E_{battery}^{max}$ and the minimum is $E_{battery}^{min}$, the battery capacity constraint can be expressed as :

$$E_{battery}^{min} \leq E_{battery}^{t} \leq E_{battery}^{max} \tag{5-8}$$

The scheduling model of the microgrid is established based on the power balance formula, which can be expressed as:

$$P_{gird}^{t} = P_{load}^{t} - P_{PV}^{t} + P_{battery}^{t} \tag{5-9}$$

Where P_{gird}^{t} is the amount of electricity purchased or sold to the public grid at the time t; P_{load}^{t} is the electricity consumption at time t; P_{PV}^{t} is the PV generation at time T; $P_{battery}^{t}$ is the charge or discharge amount of the battery at time t. According to the power balance formula, the baseline control strategy of the microgrid can be summarized as follows: if the amount of PV generated exceeds the user's electricity demand, the system preferentially stores excess power in the battery, and the remaining PV will be sold back to the public grid, thereby reducing the cost of electricity. Conversely, if the PV generation fails to meet the user's electricity demand, the battery will discharge, providing additional power to reduce the amount of electricity purchased from the public grid. Therefore, the battery is not only used to improve the PV dissipation rate but also can effectively reduce the grid's peak load in this system. It should be noted that to ensure the local consumption of PV generation, the scheduling model had forbidden the battery from actively selling electricity to the grid for arbitrage. In other words, the cost-effective optimization in this paper is mainly based on the accurate allocation of the PV generation rather than simple arbitrage.

The model-based RL method adopted in this study aims to optimize the baseline model using the strategies learned from the data rather than learning a new set of scheduling rules. It means all the proposed RL models will also interact with the environment under the rule of the Baseline control model. That is, the battery performs the action selected by the agent only after the agent determines whether the above conditions for charging and discharging are met. In this model-based RL approach, the agent can use the known rules of the baseline control model for fast and efficient learning, avoiding many unnecessary exploration actions, such as exceeding the battery capacity constraints, frequent selling PV generation to the public grid in pursuit of arbitrage, or other idle behaviors. It means that we can limit and narrow the exploration scope of agents according to the baseline model, thus reducing the number of trials and errors of agents and improving the utilization of training samples.

5.3.2 Model-based RL application

To solve the sequential decision-making problem with RL, we need to ensure that the decision process meets the Markov feature, and the decision process must be modeled as MDP. The MDP can be represented by a tuple (S, A, p, R, γ), where S denotes a state-space, A denotes an action-space, p denotes the state transition probability, R denotes the reward function, and γ denotes the discount factor which is used to calculate the cumulative reward. The rest of this section will expound on How the states, actions, and rewards are set up for this study.

5.3.2.1 States

The states in the proposed system not only represent the current state of the microgrid but also provide a mathematical description of the environment. The state variables also offer valuable information for scheduling and managing the microgrid at each time slot. Given the clear periodicity of PV generation, electricity load, and real-time electricity price in time series data, a 24-step (24h) sliding time window was designed to enable the agent to learn their inherent potential rules. Specifically, the values of these three features within 24 hours were observed respectively, and the list is padded with zeros if there are less than 24 hours of data available. As a result, the agent has seventy-six observations available in the proposed system: s_t^{pv}(the PV generation data within 24 hours), s_t^{load}(the electricity load data within 24 hours), s_t^{price}(the real-time electricity price data within 24 hours), as well as s_t^{hour}(the day's hour), s_t^{month}(the year's month), and s_t^{temp}(the outdoor temperature). The state space of proxy observation can be expressed as:

$$S=[\ s_{t-23}^{pv},\dots s_t^{pv},\ s_{t-23}^{load},\dots s_t^{load},\ s_{t-23}^{price},\dots s_t^{price},\ s_t^{hour},\ s_t^{month},\ s_t^{temp}] \tag{5-10}$$

To improve the training stability of the neural network, we normalized the observed data in the preprocessing stage and normalized the value of each variable to the range of [0,1]

5.3.2.2 Actions

The RL agent in the proposed system takes discrete actions at each time step. The reason for adopting discrete action is that all the reinforcement learning algorithms used in this study are value-based, which only support the discrete action spaces[19]. As a result, we discretize the battery charge and discharge power control actions. The battery control employs a proportional switch control in the range of -1 to 1, where a negative sign indicates battery discharge and a positive sign indicates charging. Subsequently, the continuous action space of [-1,1] will be discrete as [- 1, 0.8, 0.6, 0.4, 0.2, 0,0.2, 0.4, 0.6, 0.8, 1], then it will be remapped into eleven discrete actions of 0 to 10 in the reinforcement learning environment. The action space used in this study can be expressed as:

$$A\ =[-1,-0.8,-0.6,-0.4,-0.2,0,0.2,0.4,0.6,0.8,1] \qquad (5\text{-}11)$$

5.3.2.3 Reward

The heart of reinforcement learning is to maximize the value of the reward function, which represents the reinforcement values. The control goal of the system usually determines the reward function. The optimization objective studied in this paper aims to minimize the customer's electricity costs. Combining the energy balance Eq. (5-9) and the constraint condition Eq. (5-8), The reward at each step is:

$$R=-(a*\frac{1}{(T+1)}\int_{t=0}^{T}\left(P_{gird}(t)*C_{gird}(t)-P_{sell}(t)*C_{sell}\right)\mathrm{d}t)+k_{punish} \qquad (5\text{-}11)$$

The first part of the formula is the average power consumption of each time slot during the period of 0~T, Where a denotes the reward factor, which is a fixed constant that regulates orders of magnitude, $P_{gird}(t)$ denotes the electricity purchased from the public grid by the system at time t, and $C_{gird}(t)$ denotes the real-time electricity price at time t;$P_{sell}(t)$ denotes the electricity sold by the system to the public grid at time t, and C_{sell} denotes the feed-in tariff. The formula's second part is the constraint condition's punishment factor. Since the goal of RL is to maximize the reward function, we need to precede this part with a minus sign. k_{punish} indicates the battery capacity constraint punishment. It means when the constraint (the SoC value (20%~95%) shown in table2) is satisfied, $k_{punish}=0$; if the constraint is not satisfied, k_{punish} is a very large negative constant.

5.3.3 Experimental setting

5.3.3.1 Implementation Details

Since the ultimate goal of this study is to solve the operational optimization problem of ESS in practical applications, all the cases in this experiment were optimized hour by the hour using the actual measured hourly data set. The training set used in this experiment is hourly data from April 1, 2017, to September 30, 2018, and the test data includes hourly data from October 1, 2018, to September 30, 2019. To verify the optimization effects of various RL algorithms under the model-based framework, the authors designed four different types of optimization algorithms in the experiments:

- M.0 (baseline control): The actual operating state of the system, which is the scheduling model described in Section 5.3.1.
- M.1 (MB-Q-learning): M.1 used the Q-learning algorithm based on the model-based framework proposed in this paper. The Q-learning algorithm is chosen for comparison since it is the most basic value-based algorithm.
- M.2 (MB-DQN): M.2 used the DQN algorithm based on the model-based framework proposed in this paper.
- M.3 (MB-D3QN): M.3 used the D3QN algorithm based on the model-based framework proposed in this paper. To compare with the M.2 model, M.2 and M.3 use the same hyperparameter Settings.
- M.4 (MF-D3QN): To ascertain the efficacy of the model-based framework proposed in this study, we conducted a comparative verification using a model-free D3QN algorithm. M.4 was designed with identical hyperparameters to M.3. However, its training process was solely based on a model-free environment.

5.3.3.2 Training Setting

The simulation environment in this paper is based on the OpenAI Gym framework, and the neural network algorithm was implemented by Pytorch. To make a fair comparison, each algorithm adopts the same Agent Hyperparameters and neural network structure. The details on the hyperparameters in the training process are listed in Table 5-2. The setting of the learning rate (0.0002) is selected by the author through several experiments and experiences. Each experiment uses different seed generators five times to take the average value, and each episode is iterated 200 times with 365×24 steps.

Table 5-2 Hyperparameters of the DQN[20,21]

Structures	Hyperparameters	Values
Evaluation network	Learning rate	0.0002
	Discount factor	0.99
	Greedy policy	0.1
	Activation function	ReLU
	Optimizer	Adam
	Batch size	64
Target network	N	256

5.3.3.3 Performance Metrics

The main objective of the algorithms proposed in this paper is to obtain the maximum operating income under the condition of real-time electricity prices. Therefore, this paper mainly evaluates the performance of the algorithms based on their annual cost (c_a), monthly cost (c_m) and monthly PV self-consumption ratio(X), which calculation formulas are shown in Eq.(5-12) and Eq.(5-13). In addition, Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R-Square Value (R2_SCORE) are used as indicators of the evaluation model, which were calculated according to Equ. **错误!未找到引用源。**, **错误!未找到引用源。**, and **错误!未找到引用源。**. The y_i denotes the real observations, $\bar{y}_i$ denotes the average of the observed value, $\tilde{y}_i$ denotes the predicted value, N denotes the number of test samples.

$$c_a = \int_{t=0}^{8759} \left(P_{gird}(t) * C_{gird}(t) - P_{sell}(t) * C_{sell}\right) \mathrm{d}t \qquad (5\text{-}12)$$

$$c_m = \int_{t=0}^{T} \left(P_{gird}(t) * C_{gird}(t) - P_{sell}(t) * C_{sell}\right) \mathrm{d}t \qquad (5\text{-}13)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2} \qquad (5\text{-}14)$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \tilde{y}_i| \qquad (5\text{-}15)$$

$$R^2 = 1 - \frac{\Sigma_{\mathrm{i}=1}^{n}(y_i - \tilde{y}_i)^2}{\Sigma_{\mathrm{i}=1}^{n}(y_i - \bar{y}_i)^2} \qquad (5\text{-}16)$$

The 8759 in Eq.(5-12) is the number of hours in a year; The T in Eq.(5-13) denotes the number of hours in the target month.

5.4 Result and Discussion

Section 5 of this study details the simulation methodology employed, which primarily utilized the RL algorithms outlined in Section 3 to achieve cost-effective optimization of the microgrid environment described in Section 2. This section aims to provide a summary of the simulation experiment's outcomes. Firstly, we will present an overview of the performance of the five mentioned algorithms in the test set. Secondly, to evaluate the optimization capabilities of the proposed algorithms in greater detail, we will analyze energy scheduling scenarios on a typical week generated by the agents. Finally, we will discuss the implications of these findings.

5.4.1 Cost-effective Optimization Analysis

Table 5-3 presents the energy consumption of the scenarios utilizing different algorithms in the evaluation data and the actual consumption of the system. All model-based RL models achieved cost-benefit optimization; further analysis showed that the model-based Q-Learning, DQN, and D3QN reduced costs by 1.14%, 7.49%, and 11.27% compared to the baseline model, respectively, with actual consumption in these twelve months, which indicates that the model-based methods' ability to acquire experience from a limited dataset is superior to that of the model-free methods. Besides, the model-free D3QN does not achieve the optimization goal, which means the traditional model-free methods cannot learn optimal control strategies from a limited dataset in this scenario. Comparing the Root Mean Square Error (RMSE) of the monthly cost of the three RL models with the baseline model shows that the MB-D3QN model demonstrated the most significant cost fluctuation compared to the baseline model, which indicates that the MB-D3QN scheduling mode experienced significant changes when compared to the baseline model. Furthermore, the MB-D3QN algorithm achieved superior optimization effects compared to MB-DQN. The former reduced the overall cost by 4.09% compared to the latter, proving that D3QN can effectively overcome the shortcomings of DQN. The detailed comparison of the annual cost of the five models is shown in Fig. 5-2.

Table 5-3 The monthly cost results.

Month	M.1(Yen)	M.2(Yen)	M.3(Yen)	M.4(Yen)	M.0(Yen)
Jan	23105.95	21653.43	21536.78	22715.46	22675.48
Feb	19670.69	18384.27	18126.05	19424.70	19122.73
Mar	2159.84	1358.11	1799.84	2245.94	2202.35
Apr	-1275.11	-2282.75	-1996.41	-1260.73	-1352.73
May	-2258.36	-2472.63	-2593.28	-2134.72	-2081.50
Jun	-4481.23	-4458.57	-4706.14	-4162.99	-4077.52
Jul	-726.42	-933.86	-1098.56	-302.45	-478.20
Aug	1495.27	1412.51	1352.54	2013.29	1806.39
Sep	579.74	604.34	412.86	1203.74	906.54
Oct	-1150.95	-926.36	-1113.40	-318.42	-795.48
Nov	9121.12	9654.36	8883.95	9512.72	9121.54
Dec	16226.86	16465.47	15463.64	16240.30	16139.36
total	62467.41	58458.31	56067.88	65176.84	63188.98
AVG	5205.62	4871.53	4672.32	5431.40	5265.75
RMSE	300.95	600.50	643.69	10335.08	NA
MAE	250.72	537.38	593.42	188.77	NA

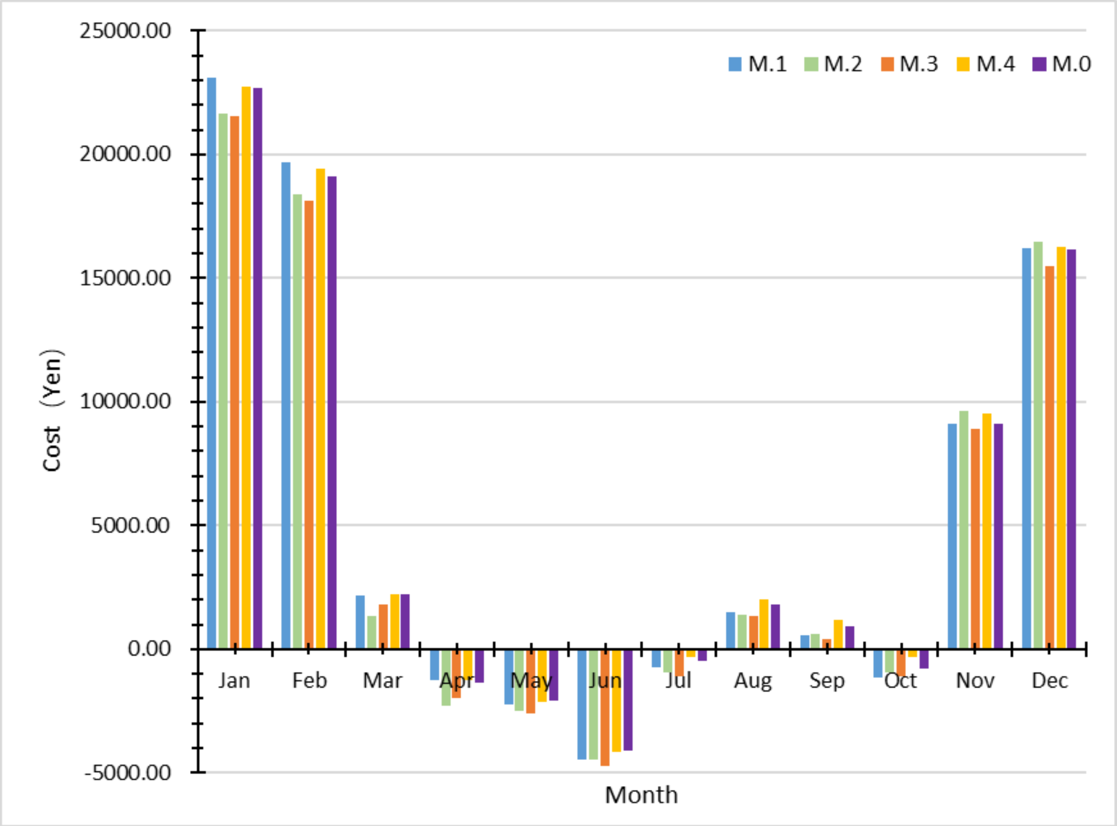


Fig. 5-2 The annual cost comparison

Further analysis shows that the cost savings of M1, M2, and M3 in the cooling season are -1065.51(Yen), 901.59(Yen), and 3048.69(Yen), respectively; In the heating season, the cost savings are 1289.85(Yen), 1532.79(Yen) and 2196.51(Yen); In the transition season, the cost savings is 497.22(Yen), 2296.27(Yen) and 1875.89(Yen), the details are shown in Fig. 5-3. We can observe that both MB-D3QN and MB-DQN can achieve cost optimization in all three seasons. However, MF-D3QN failed to achieve the optimization target in all three seasons, and MB-QL failed in the heating season. It shows that the optimization effect of MB-D3QN is the best in the cooling and heating seasons, while MB-DQN achieves the best optimization effect in the transition season. In the cooling season, the cost of M.3 was reduced by 4.54% compared to the baseline model, and M.2 was only 1.34%, with no optimization effect achieved in M.1.

It should be noted that M.1 and M.2 cannot achieve cost-effective optimization in some winter months. We suspected that this phenomenon was associated with the distribution of test samples. As seen in Fig. 3-7, the PV generation is less than the electricity load from November to March of the following year. The monthly average real-time electricity price in these five months is significantly higher than the feed-in tariff. Since the agents mainly realize cost-effective optimization by improving the photovoltaic absorption rate, it is challenging to produce optimized samples under the condition that the total amount of PV generation is less than the electricity demand. Even if some individual operations can be implemented, they are constrained and cannot be learned by the agent. This also indicates that the ratio of PV generation to electricity load and the monthly average real-time electricity price are two important factors affecting the model's efficiency.

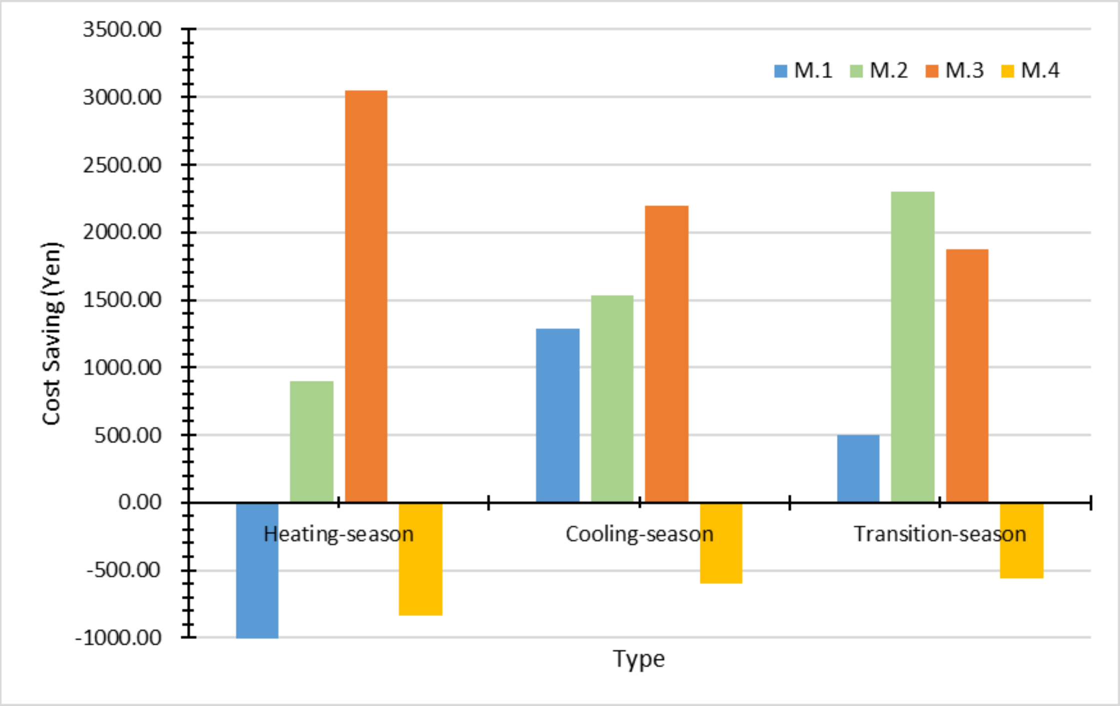


Fig. 5-3 The seasonal cost-saving comparison

5.4.2 Visualization analysis of optimization results

5.4.2.1 Comparison of model-based methods

To analyze the characteristics of the actions of these three model-based methods in typical working periods and whether the actions are optimal, we selected three random weeks in the midseason, the cooling season, and the heating season for specific research, and the results are shown in Table 5-4. It can be seen from Table 5-4 that compared with the baseline model, the cost of Q-learning in these three seasons is reduced by 16.13(Yen), 76.98(Yen), and -3.67(Yen), respectively; DQN decreased by 143.16(Yen), 137.53(Yen) and -12.42(Yen), respectively; D3QN decreased by 164.64(Yen), 163.09(Yen) and 148.88(Yen), respectively.

Table 5-4 The comparison of weekly costs

Date	M.1 (Yen)	M.2(Yen)	M.3 (Yen)	M.0 (Yen)
4.1-4.7	483.95	356.92	335.44	500.08
7.1-7.7	-39.02	-99.57	-125.13	37.96
12.1-12.7	1181.9	1190.65	1029.35	1178.23

Fig. 5-4 illustrates the regulation effects of the three model-based methods during typical weeks of the transition season. The MB-D3QN algorithm achieved the best optimization performance among the three model-based algorithms. The SOC curves of the three methods have been significantly optimized and adjusted compared to the baseline model, indicating that the three methods can effectively obtain efficient regulation strategies from the training set. While all three methods achieved their optimization goals during this season, it is evident that the optimization strategy of Q-learning and DQN went awry on April 2 and April 4, indicating that the stability of Q-learning and DQN is not robust. This is likely due to the overestimation of the Q value, leading to lower accuracy compared to D3QN when processing a large number of observation states. Furthermore, it is worth noting that despite the lack of a cyclical pattern in the fluctuation of RTP and power demand during the transition season, both DQN and D3QN demonstrate strong optimization performance, which suggests that these algorithms possess inherent strengths in handling such scenarios.

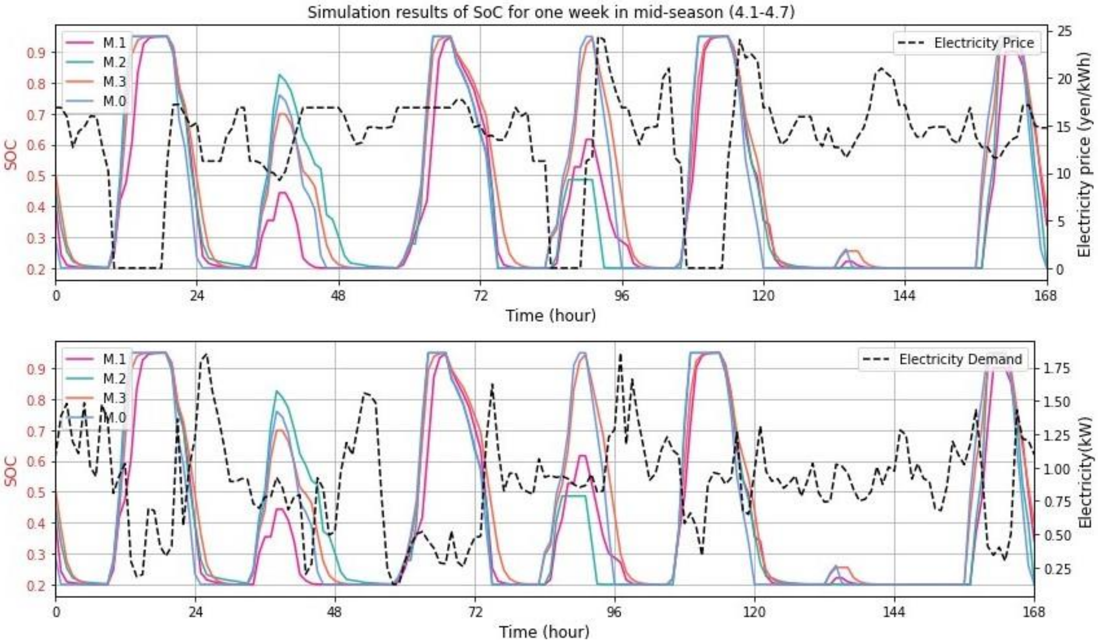


Fig. 5-4 One-week simulation results of SoC for PV-battery system: Simulation results of the transition season

Next, we analyze the detailed control results of the three model-based optimization strategies on April 3, a transition season day. As shown in Fig. 5-5, the charging rate of the three strategies decreases between 8 a.m. and 12 a.m., which can be attributed to the fact that the RTP is rising from the trough to the peak during this period, and the power demand is relatively low. Consequently, selling as much photovoltaic power as possible to the public grid under the full battery is a wise choice. Another noteworthy phenomenon is that all three optimization models reduce discharge power from 16:00 to 22:00 to achieve delayed discharge. The high RTP levels can explain this from 18:00 to 22:00, and the peak demand occurs at 23:00. Storing electricity for the upcoming high consumption period is an effective way to optimize cost in this scenario.

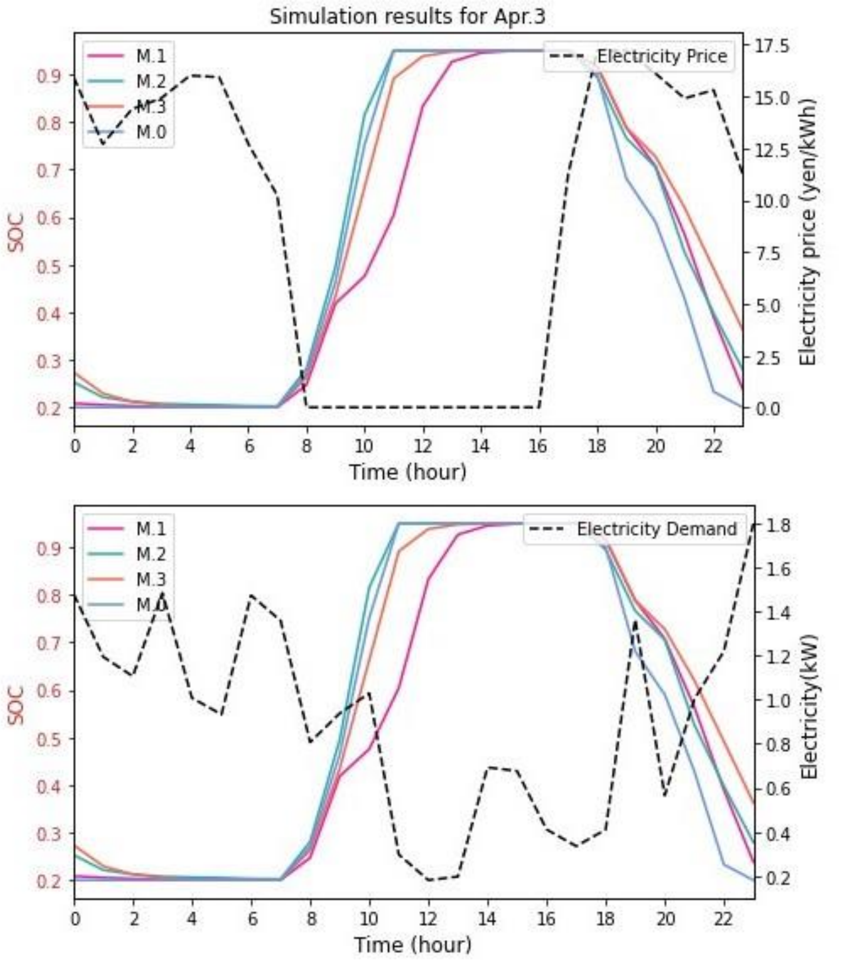


Fig. 5-5 Simulation results of the PV-battery system for one day：Simulation results for one day in transition season

Fig. 5-6 displays the control performances of the three model-based techniques during typical weeks of cooling season. According to Table 5-4, all three methods have achieved their optimization objectives, with D3QN obtaining the best optimization effect. Compared to the transition season, the RTP and power demand patterns during the cooling season exhibit greater periodicity. As a result, the efficacy of the three optimization methods has been enhanced, and the control strategies tend to be similar. However, the Q-learning method failed again on April 2, suggesting that the stability of this method is not ideal. Moreover, on April 2, the strategies employed by DQN and D3QN appeared to be significantly distinct.

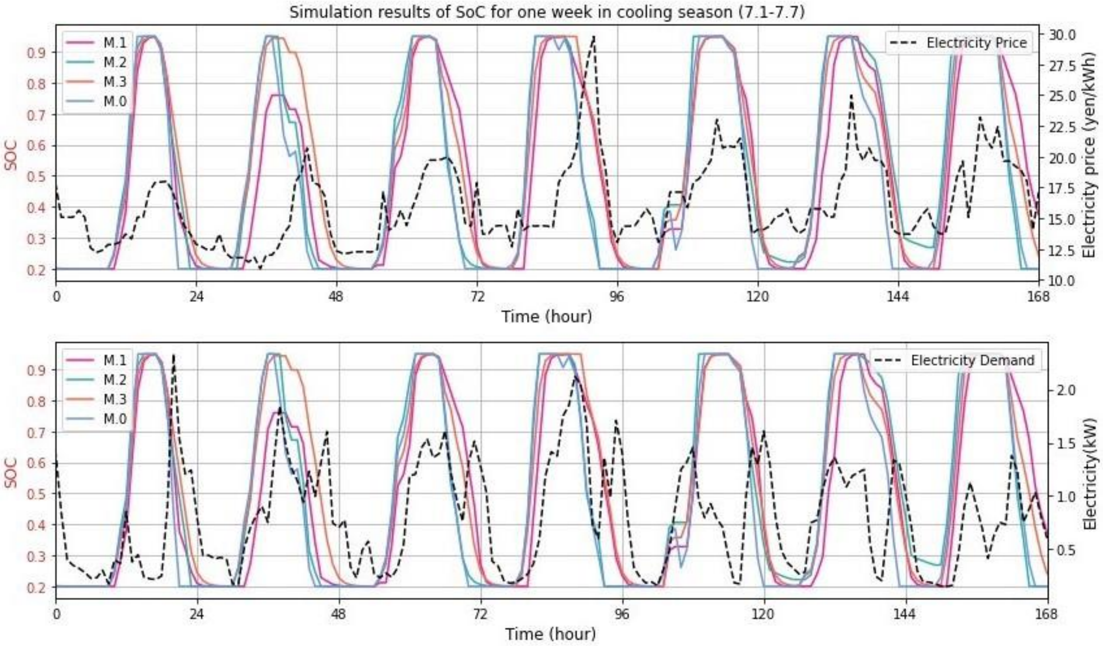


Fig. 5-6 One-week simulation results of SoC for PV-battery system: Simulation results of the cooling season

A detailed analysis of the optimization strategies of the three models on April 2 is presented in Fig. 5-7. It is observed that during the low RTP period from 6:00 to 12:00, all three models chose to decrease charging power to sell as much PV generation to the public grid as possible. However, Q-learning made an erroneous decision by excessively selling PV generation and missing the charging period, resulting in an incomplete battery charge. From 13:00 to 15:00, the baseline model began battery discharge at 13:00. In contrast, the three optimization models accurately predicted the peak demand and rising trend of RTP at 15:00 and decided to retain power. From 16:00 to 22:00, all three models reduced discharge power according to the rising trend of RTP. However, Q-learning and DQN were suboptimal as they released most of the power before the RTP peak at 19:00, resulting in inferior optimization compared to DDQN. The actions above demonstrate that D3QN has a clear advantage in the cooling season when data periodicities are noticeable, whose action selection is more scientific and intelligent.

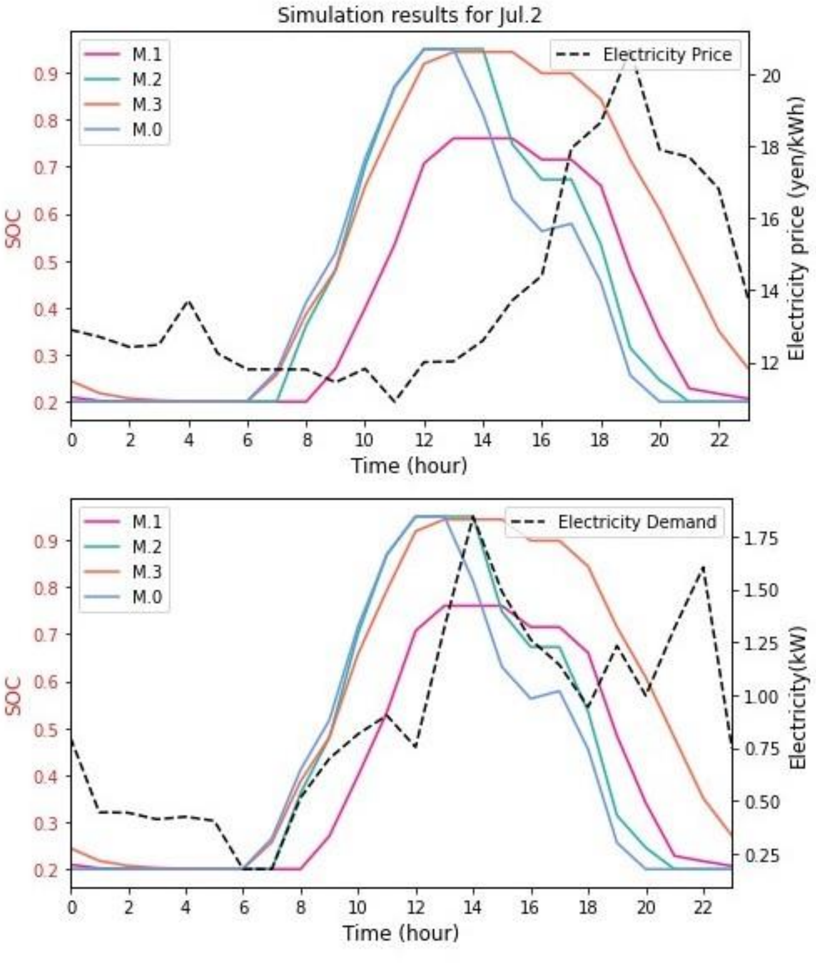


Fig. 5-7 Simulation results of the PV-battery system for one day：Simulation results for one day in the cooling season

The effects of the three model-based methods on regulation during typical weeks of the heating season are presented in Fig. 5-8. It is observed that the distribution of RTP and power demand curves during the heating season is similar to that of the transition season. However, on December 1 and 4, Q-learning and DQN exhibited errors in their regulation strategies, while D3QN exhibited relatively stable performance. It suggests that D3QN can make accurate decisions based on learned empirical knowledge even when the regularity of observed values weakens in a given scenario.

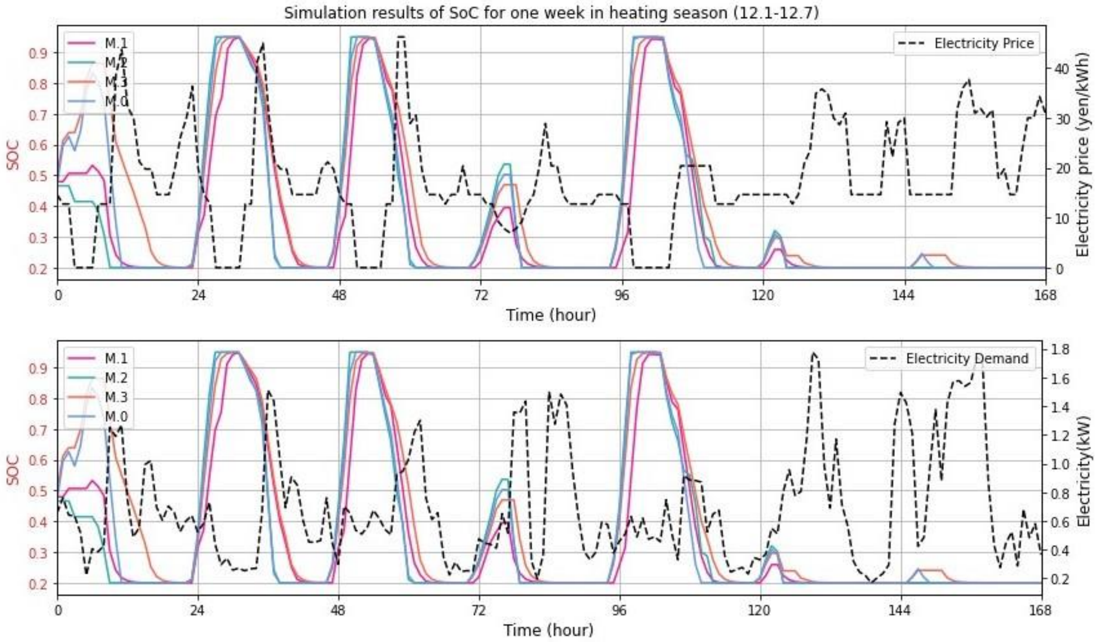


Fig. 5-8 One-week simulation results of SoC for PV-battery system: Simulation results of the heating season

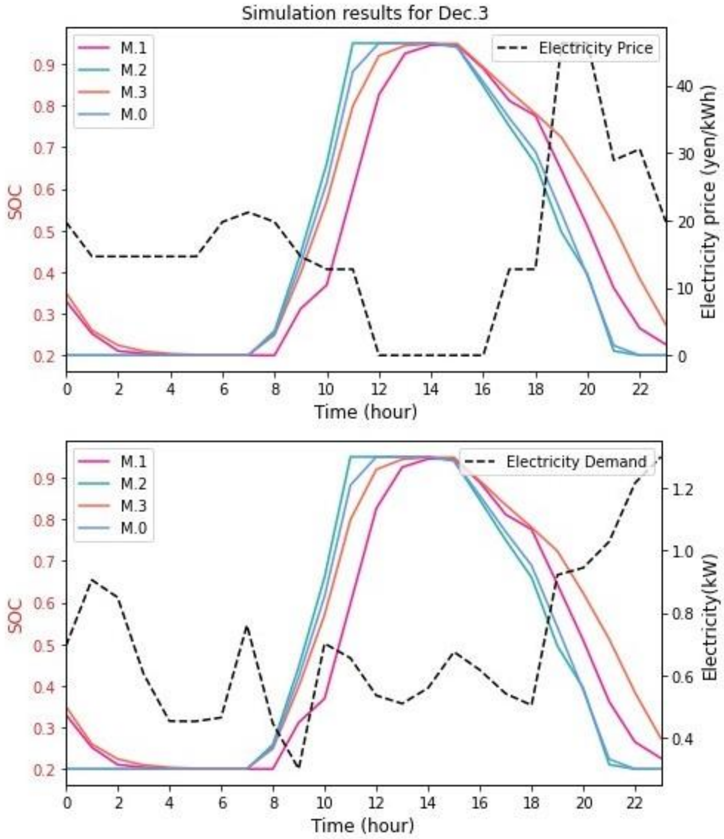


Fig. 5-9 Simulation results of the PV-battery system for one day: Simulation results for one day in the heating season

In addition, it can be seen from Fig. 5-9 that the optimization strategies of the three methods are similar to the previous analysis, which can be summarized as follows :(1) When charging if the RTP is low, the charging rate should be appropriately reduced to obtain profits through selling PV generation to the grid. (2) When discharging, it will determine the discharge power according to RTP trends. When the RTP is low, and its increasing trend is slow, it will choose to reserve the power for the time point of high electricity price that may come in the future. As above analysis, these three RL agents have all learned the above rules, except for the difference in action accuracy.

5.4.2.2 Comparison of model-based and model-free algorithms

To assess the effectiveness of the model-based RL framework proposed in this study, we compared the performance of D3QN based on this framework with that of traditional model-free D3QN on the same verification set, using identical hyperparameters and experimental methods. The results for the entire year are presented in Table 5-3. Evidently, the model-free D3QN method did not perform well and failed to achieve the optimization objective throughout the year, except in May and June. To further understand the differences in regulation strategies between the two methods, we conducted statistical and visual analyses on typical weeks in the three seasons. The detailed cost statistics are presented in Table 5-5:

Table 5-5 The comparison of weekly costs

Date	M.3 (Yen)	M.4 (Yen)	M.0 (Yen)
4.1-4.7	335.44	535.49	500.08
7.1-7.7	-125.13	116.31	37.96
12.1-12.7	1029.35	1279.45	1178.23

Fig. 5-9 compares the regulation effects of models M.3 and M.4 during typical weeks of the transition season. The figure demonstrates that the disparity between the regulation curves of M.3 and M.4 is primarily manifested in the discharge stage. A detailed analysis reveals that M.4 can anticipate the occurrence of RTP or peak power demand in the future and, therefore, reserves power for this purpose. However, M.4's judgment regarding the discharge power and time is erroneous, as it fails to identify the optimal discharge period, such as the time intervals of 20h to 28h and 72h to 80h, which is primarily attributed to the model's inaccurate peak judgment time and excessive power retention.

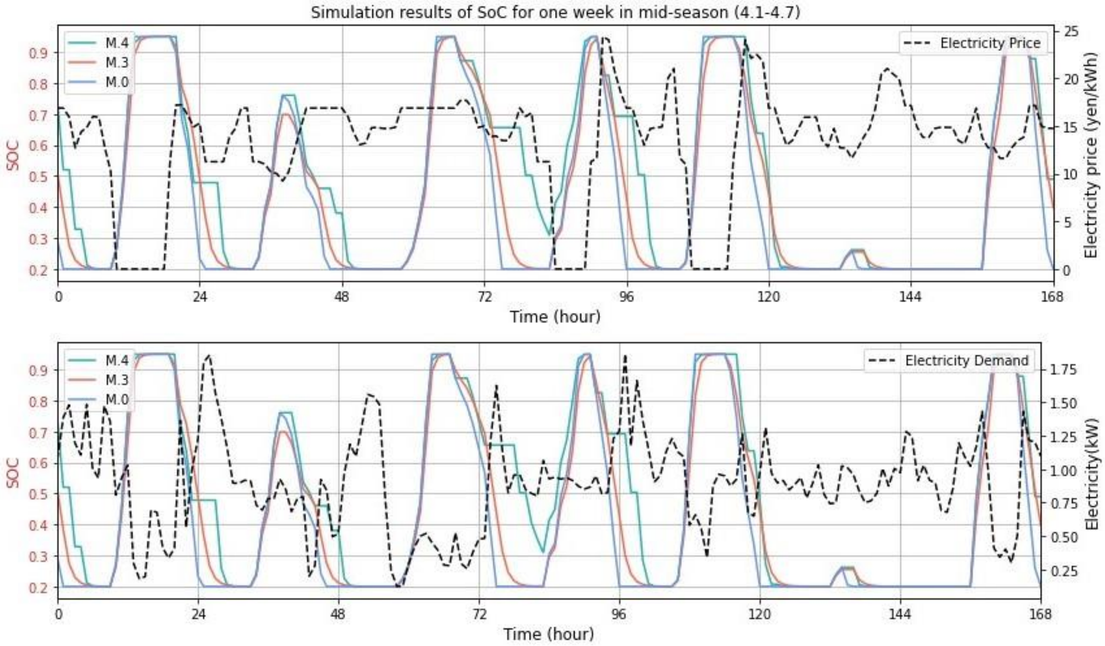


Fig. 5-9 Comparison of one-week simulation results between M.3 and M.4: Simulation results of the transition season

Fig. 5-10 compares the control effects of M.3 and M.4 in a typical week of the cooling season. As mentioned in Section 5.4.2.1, the distribution of RTP and power demand during the refrigeration season exhibits clear periodicity, making the optimization effect of M.3 in this period the best throughout the year. However, the performance of M.4 during this period is still poor due to the inaccurate time point of peak value judgment. Specifically, M.4 exhibits early discharge before the peak (e.g., 32h to 40h and 82h to 90h) and over-retention, missing the optimal discharge time (e.g., 116h to 124h and 160h to 168h). These results suggest that the improvement in data quality has a negligible impact on the performance of M.4 in the case of small sample sizes.

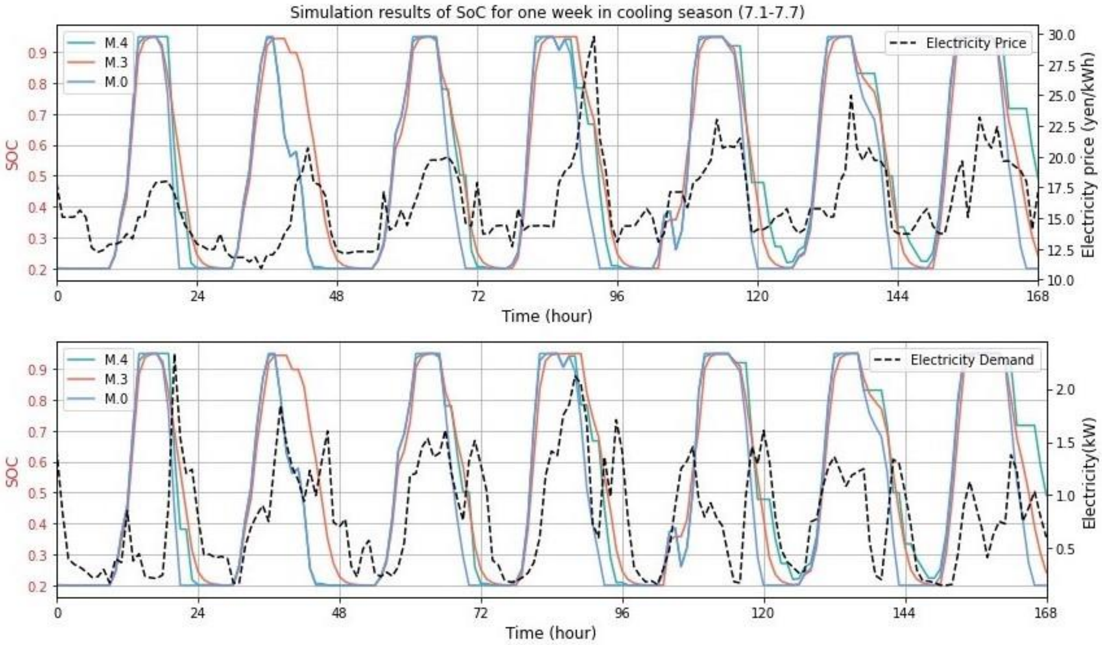


Fig. 5-10 Comparison of one-week simulation results between M.3 and M.4: Simulation results of the cooling season

Fig. 5-11 shows the comparison of the regulation effects of M.3 and M.4 in a typical week of the heating season. The distribution of RTP and demand in the heating season is similar to that in the transition season, both of which lack periodic rules. In this scenario, M.3 exhibits stable optimization of the battery's action, while M.4's performance remains similar to the transition season's. It tends to over-reserve the power and miss the optimal discharge time (for example, 32h to 40h and 62h to 70h) and even makes a mistake in the charging time selection on December 1. These observations suggest that when the sample data quality declines, M.4 tends to overestimate the future Q value, leading to excessive “over-reserve” action.

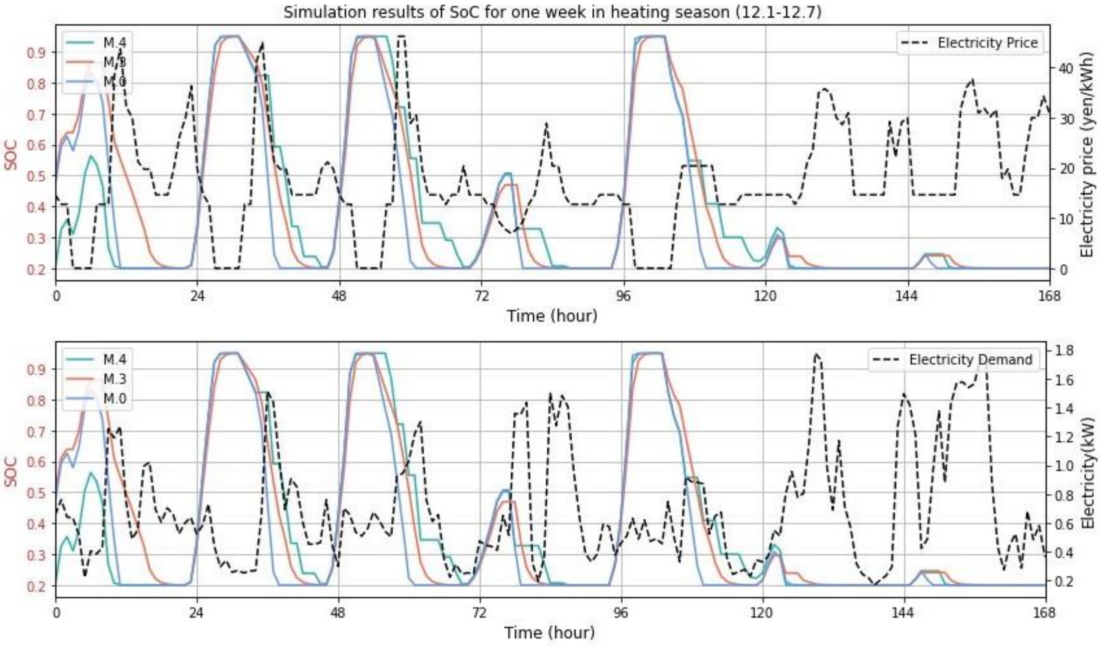


Fig. 5-11 Comparison of one-week simulation results between M.3 and M.4: Simulation results of the heating season

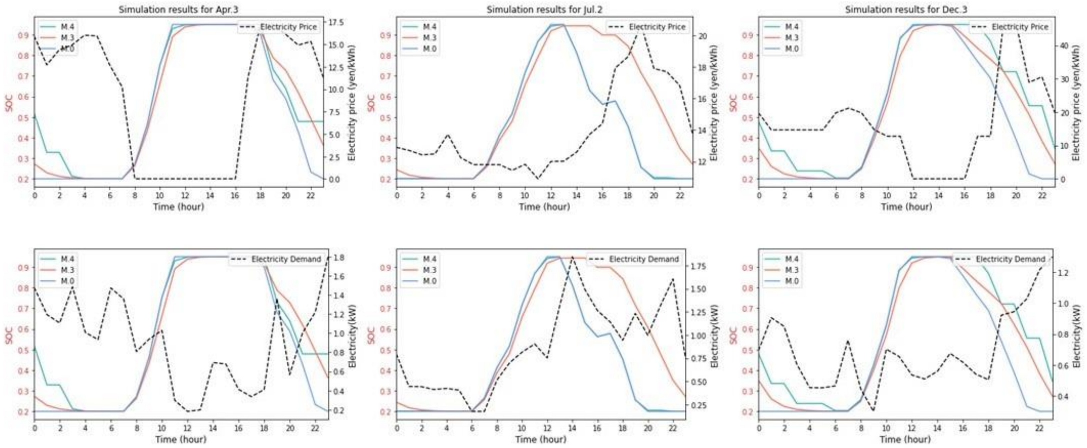


Fig. 5-9 Comparison of one-day simulation results between M.3 and M.4

Fig. 5-12 provides a detailed comparison of daily control effects. In conclusion, the proposed hybrid model-based RL method exhibits significant advantages over traditional model-free RL methods when trained on half of the data from one year. Due to the limited number of data samples, the model-free RL method struggles to acquire sufficient empirical knowledge. Conversely, the model-based hybrid RL method can leverage inherited knowledge and experience from the battery model, thereby reducing the need for wasteful exploration and improving the utilization of data samples. As such, it is better suited for applications in such scenarios.

5.4.3 Improvements

Potential improvement could be made by adopting a more scientific real-time electricity price simulation method. In this experiment, the simulated RTP fluctuates wildly while the actual application RTP fluctuates less violently. However, since we focused on the performance of the RL agent under real-time electricity price in this case study, the real-time electricity price simulated by spot price can also provide a reasonable electricity price curve, and its correlation is sufficient.

5.5 Conclusion

This paper presents a model-based RL controller using the D3QN algorithm for cost-effective energy management of the grid-connected residential PV-battery system. In addition, we designed a new reward function for the optimization goal of reducing energy costs and ensuring the local absorption rate of PV generation. Results compared and analyzed the performance of Q-learning, DQN, and D3QN agents in optimizing the scheduling strategy of the residential PV-battery system based on real-world monitored data and real-time electricity price. The experimental results proved the effectiveness of the reward function design, and both DQN and D3QN algorithms can reduce the energy cost when the PV self-consumption ratio is higher than the baseline model. The case analysis based on the measured data also proves that the MB-D3QN algorithm provides a more efficient scheduling strategy. Compared to the baseline model, it reduces the annual electricity cost by 11.27%. According to the analysis of cost-effectiveness and influencing factors, it could be concluded that the optimization effect of the MB-D3QN method was mainly affected by the difference between the average PV generation and average load and then by the average RTP. The analysis of the Soc control effect proves that MB-D3QN can intelligently judge the future load and electricity price peak and take reasonable charge and discharge action. The comparison between the model-based D3QN method and the model-free D3QN method shows that the model-based approach proposed in this study can significantly improve sample utilization and effectively learn empirical knowledge from limited small sample data. Additionally, it can develop more scientific strategies compared to the model-free method. This further supports the superiority of the model-based approach in the design and control of energy storage systems, which presents a promising practical application potential.

Future research will focus on optimizing the reward function's design and try to add other control objectives(such as heat pump or air conditioner) to achieve multi-objective optimization of the energy system[22]. Secondly, we should continue improving the model's robustness to be applied to other zero-energy houses[23]. In addition, we will predict PV generation, electricity load, and electricity price using the LSTM network and try to add these predicted values into the agent as observations to see how it affects the accuracy of the actions.

Reference

[1] Komiyama R, Fujii Y. Assessment of post-Fukushima renewable energy policy in Japan's nation-wide power grid[J]. Energy Policy, 2017, 101: 594–611.

[2] Bogdanov D, Ram M, Aghahosseini A, Gulagi A, Oyewo A S, Child M, Caldera U, Sadovskaia K, Farfan J, De Souza Noel Simas Barbosa L, Fasihi M, Khalili S, Traber T, Breyer C. Low-cost renewable electricity as the key driver of the global energy transition towards sustainability[J]. Energy, 2021, 227: 120467.

[3] Lepszy S. Analysis of the storage capacity and charging and discharging power in energy storage systems based on historical data on the day-ahead energy market in Poland[J]. Energy, 2020, 213: 118815.

[4] Li Y, Gao W, Zhang X, Ruan Y, Ushifusa Y, Hiroatsu F. Techno-economic performance analysis of zero energy house applications with home energy management system in Japan[J]. Energy and Buildings, 2020, 214: 109862.

[5] Pallonetto F, De Rosa M, Finn D P. Impact of intelligent control algorithms on demand response flexibility and thermal comfort in a smart grid ready residential building[J]. Smart Energy, 2021, 2: 100017.

[6] Al-Hinai A, Alyammahi H, Haes Alhelou H. Coordinated intelligent frequency control incorporating battery energy storage system, minimum variable contribution of demand response, and variable load damping coefficient in isolated power systems[J]. Energy Reports, 2021, 7: 8030–8041.

[7] L. Ren, L. Zhao, S. Hong, S. Zhao, H. Wang, L. Zhang. Remaining Useful Life Prediction for Lithium-Ion Battery: A Deep Learning Approach[J]. IEEE Access, 2018, 6: 50587–50598.

[8] Zhao X, Gao W, Qian F, Ge J. Electricity cost comparison of dynamic pricing model based on load forecasting in home energy management system[J]. Energy, 2021, 229: 120538.

[9] Elma O, Taşcıkaraoğlu A, Tahir İnce A, Selamoğulları U S. Implementation of a dynamic energy management system using real time pricing and local renewable energy generation forecasts[J]. Energy, 2017, 134: 206–220.

[10] Doostizadeh M, Ghasemi H. A day-ahead electricity pricing model based on smart metering and demand-side management[J]. Energy and Exergy Modelling of Advance Energy Systems, 2012, 46(1): 221–230.

[11] Andrew A M. REINFORCEMENT LEARNING: AN INTRODUCTION by Richard S. Sutton and Andrew G. Barto, Adaptive Computation and Machine Learning series, MIT Press (Bradford Book), Cambridge, Mass., 1998, xviii + 322 pp, ISBN 0-262-19398-1, (hardback, £31.95).[J]. Robotica, 1999, 17(2): 229–235.

[12] Teoh K, Ismail R, Naziri S, Hussin R, Isa M, Basir M. Face Recognition and Identification using Deep Learning Approach[J]. Journal of Physics: Conference Series, 2021, 1755(1):

012006.

[13] Ramana K, Kumar M R, Sreenivasulu K, Gadekallu T R, Bhatia S, Agarwal P, Idrees S M. Early Prediction of Lung Cancers Using Deep Saliency Capsule and Pre-Trained Deep Learning Frameworks[J]. Frontiers in Oncology, 2022, 12.

[14] Mnih V, Kavukcuoglu K, Silver D, Rusu A A, Veness J, Bellemare M G, Graves A, Riedmiller M, Fidjeland A K, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S, Hassabis D. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529–533.

[15] Tangkaratt V, Mori S, Zhao T, Morimoto J, Sugiyama M. Model-based policy gradients with parameter-based exploration by least-squares conditional density estimation[J]. Neural Networks, 2014, 57: 128–140.

[16] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing Atari with Deep Reinforcement Learning. arXiv. https://doi.org/https://arxiv.org/abs/1312.5602v1[J].

[17] Duryea E, Ganger M, Wei H. Deep Reinforcement Learning with Double Q-learning[J]. 2016. ,2016.

[18] D. Lopez-Martinez, P. Eschenfeldt, S. Ostvar, M. Ingram, C. Hur, R. Picard. Deep Reinforcement Learning for Optimal Critical Care Pain Management with Morphine using Dueling Double-Deep Q Networks[C]//2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC).

[19] Yoon Y R, Moon H J. Performance based thermal comfort control (PTCC) using deep reinforcement learning for space cooling[J]. Energy and Buildings, 2019, 203: 109420.

[20] Gao Y, Matsunami Y, Miyata S, Akashi Y. Operational optimization for off-grid renewable building energy system using deep reinforcement learning[J]. Applied Energy, 2022, 325: 119783.

[21] Harrold D J B, Cao J, Fan Z. Data-driven battery operation for energy arbitrage using rainbow deep reinforcement learning[J]. Energy, 2022, 238: 121958.

[22] Zhao L, Yang T, Li W, Zomaya A Y. Deep reinforcement learning-based joint load scheduling for household multi-energy system[J]. Applied Energy, 2022, 324: 119346.

[23] Biemann M, Scheller F, Liu X, Huang L. Experimental evaluation of model-free reinforcement learning algorithms for continuous HVAC control[J]. Applied Energy, 2021, 298: 117164.

Chapter 6

OPERATIONAL OPTIMIZATION FOR BUILDING ENERGY SYSTEMS USING REINFORCEMENT LEARNING CONSIDERING REAL-TIME ENERGY PREDICTION

CHAPTER SIX: OPERATIONAL OPTIMIZATION FOR BUILDING ENERGY SYSTEMS USING REINFORCEMENT LEARNING CONSIDERING REAL-TIME ENERGY PREDICTION

6.1 Introduction

In recent years, owing to the rapid development of industrialization and urbanization, the global energy demand has risen sharply, which has brought severe challenges to mitigate climate change. Since building energy consumption accounts for about 40% of global energy consumption[1], increasing the proportion of renewable energy sources(RES) to reduce building energy consumption has become a research hotspot[2]. Since photovoltaic(PV) technology has the advantages of excellent cost and convenient deployment, making it one of the most widely used RES[3]. For example, the Japanese government has introduced a series of incentive policies for applying RES[4]. Consequently, more and more households in Japan are opting for the household multi-energy system (HMES)[5], which integrates electricity, natural gas, and renewable energy sources (such as photovoltaic and wind power) as energy sources. As a bidirectional grid-connected energy system, HMES can meet multiple load demands of users and sell excess renewable energy to the grid, reducing household energy payment costs[6]. Therefore, the HMES integrating RES undoubtedly has significant research value and application potential.

However, due to the multiple uncertainties in the application of the HMES, the energy scheduling of the system faces significant challenges. Firstly, renewable energy production is greatly affected by environmental factors (such as weather conditions)and has strong intermittency and uncertainty. Secondly, with the development of the electricity market, many countries have adopted the real-time electricity price (RTP)[7], which is also highly uncertain due to the fluctuations of the electricity futures trading price. Third, for residential customers, the differences in living habits and rapid electrification will also lead to the uncertainty of electricity demand. The energy storage system (ESS) is an effective approach to deal with these uncertainties[8]. The ESS can not only effectively alleviate the instability caused by the fluctuation of renewable energy but also optimize the economy of the energy system according to the dynamic information of energy prices, and the grid-connected residential photovoltaic-battery system based on HMES has become Japan's fastest-growing renewable energy technology[6]. It should be noted that although the ESS has the above advantages, it also increases the system cost and the complexity of system optimization[9].

The primary objective of ESS management is to ensure the economical and efficient operation of microgrids and to optimize energy scheduling. Numerous studies have been conducted on controlling energy storage systems, including classical optimization methods, heuristic optimization methods, reinforcement learning methods, and others. While classical optimization methods, such as mixed integer linear programming[10], dynamic programming[11], and stochastic linear programming[12], are well-suited for solving sequential optimization problems, they have certain limitations. One of the major drawbacks of these methods is that each iteration requires restarting, which involves significant computational resources and cannot facilitate real-time decision-making[13]. In addition, these methods cannot predict the many uncertainties in the system because they do not include domain-specific knowledge and cannot use historical data or model predictions.

Prediction can be an important component of DRL in many cases. Currently, two approaches exist for integrating predictions into DRL. The first approach involves training a prediction model using supervised or unsupervised learning techniques to predict future states, which can then inform the agent's decision-making process. The second approach involves incorporating future predictions into a value function, which estimates the expected cumulative return of taking a particular action while in a given state to inform the agent's decision-making process further. While both approaches have their merits, the first is the most commonly used. The authors in[14] proposed a novel approach for energy management in HVAC systems using deep reinforcement learning based on multi-step prediction. Specifically, the proposed algorithm integrates LSTM prediction of outdoor temperature with DDPG to dynamically adjust the output power of HVAC systems based on RTP. The simulation results show that the proposed method can achieve significant cost savings of more than 12% while maintaining user comfort levels. Duo Yang et al.[15] proposed a novel approach to energy management using reinforcement learning, which includes a real-time driving speed prediction method and a power allocation method based on RL. The experimental results demonstrated that the proposed EMS could significantly reduce the life decay rate of fuel cells while increasing fuel economy by 6%. Dian Zhuang et al.[16] proposed a novel data-driven predictive control method for optimizing the energy consumption and thermal comfort of HVAC systems. The proposed method integrates a reinforcement learning (RL) agent with 16-time series prediction models based on CNNs and LSTM. The authors evaluated the 16 prediction models under various scenarios and selected the optimal model for integration with the RL agent. Fang Liu et al.[17] proposed a novel data-driven strategy for wind ESS management, which involves leveraging LSTM to develop a power prediction model that quantifies wind power uncertainties and uses RL to solve energy storage management problems. Experimental results demonstrate that this method can significantly reduce daily transaction and wear costs, making it an effective and practical solution for wind energy storage management. In the above examples, prediction is used to augment the agent's decision-making process by providing additional information about the future state of the environment. This information can help the agent make more informed decisions and improve performance.

Based on the above-reviewed work, the contributions of this study can be summarized as follows:

- We adopted an existing grid-connected residential photovoltaic-battery system’s data as the research object. We used an RL-based approach to optimize the system's operation, including reducing energy costs while maintaining the renewable energy self-consumption ratio within a predetermined range. Therefore, we designed a new reward function to achieve these goals and proved its effectiveness through experiments.
- A model-based Twin Delayed Deep Deterministic Policy Gradients(MB-TD3) algorithm considering real-time prediction Values is developed to optimize residential ESS. We developed nine prediction models for the building's electricity demand, PV generation, and RTP. After evaluating the performance of each model, we selected the optimal one

and combined it with our RL agents, which could improve the RL algorithms' efficiency and effectiveness significantly. Detailed case analysis and comparison of four advanced RL algorithms are presented, where the baseline model uses the strategy used in actual buildings. We evaluated the efficiency of these algorithms in terms of energy cost and renewable energy self-consumption ratio and proved that the efficiency of MB-TD3 is optimal.

- We showed that all the MB-RL algorithms could reliably ensure that the renewable energy self-consumption ratio is higher than the baseline model while reducing the electricity purchase cost. This emphasizes these model-based RL algorithms' ability to learn optimal control policies with fewer datasets, providing valuable insights for real-world implementations.

The organization of work is as follows. Section 2 describes the algorithm details of this paper. Section 3 provides a detailed explanation of the implementation and evaluation of our prediction model. Section 4 outlines the RL agent implementation. Section 5 discusses the results of the case study. Finally, section 6 provides this paper's conclusions and future outlook.

6.2 Methodology

6.2.1 The selection of the algorithm

RL, as a branch of machine learning, is a computational method that can solve sequential decision problems[18,19]. All the RL problems can be defined as MDP, which represents the process by which an agent guides its behavior by obtaining rewards from interaction with the environment. Formally, an MDP is a five-tuple (S, A, P, R, R), where:

- S is the state space, which represents the available information that the RL agents use to make decisions.
- A is the action space, which means the RL agents make different decisions when interacting with the environment.

- P is the state-transition probability, which describes the probability distribution of going from state s to state s' when action a is taken.
- R is the reward (or cost) function, usually the objective function in a control problem.
- r is the discount factor. The discount factor is used to overcome the feedback delay in the interaction between the agent and the environment. By discounting the rewards for multiple steps, the sum of the accumulated rewards for numerous steps in the future can be obtained. Then the short-term optimization objective and long-term optimization objective can be balanced.

RL aims to find the optimal policy π through the Markov decision process, which refers to the mapping of states to actions[20]. Specifically, the mapping is constructed by the state-value function $v_\pi(s)$ and the action-value functions $q_\pi(s,a)$, which formula is as follows:

$$v_\pi(s) = E_\pi\left[\sum_{k=v}^{\infty} r^k R_{t+k+1} | S_t = s\right] \qquad (6\text{-}1)$$

$$q_\pi(s,a) = E_\pi\left[\sum_{k=v}^{\infty} r^k R_{t+k+1} | S_t = s, A_t = a\right](6\text{-}2)$$

Through Eq.1 and Eq.2, we can obtain the Behrman equation of the state-action value function:

$$q_\pi(s,a) = E_\pi[R_{t+1} + rq(S_t, A_t)|S_t = s, A_t = a](6\text{-}3)$$

If the optimal state-action value function $q^*(s,a)$ is known, the optimal policy can be determined by directly maximizing it:

$$\pi_*(a) = \begin{cases} 1 & if\ a = \underset{a\in A}{argmax} q^*(s,a) \\ 0 & otherwise \end{cases} \qquad (6\text{-}4)$$

The above summarized the basic principles of RL. Next, we will discuss the classification of RL algorithms. The overview of the most popular algorithms is summarized in Table 6-1. According to the different action selection strategies, RL algorithms can be divided into two branches: value-based algorithms and policy-based algorithms. The value-based algorithm calculates the expectation of reward through the potential reward as the basis for selecting actions. And the policy-based algorithm trains a probability distribution through policy sampling and enhances the probability of the desired action with a high return value[18]. Therefore, value-based algorithms can only be used for discrete action space, while policy-based algorithms have more advantages in continuous action space control. Currently, the most popular actor-critic method combines the benefits of these two branches. Specifically, the actor-network will take actions based on the probability distribution of policies. The critic-network will give the value of actions to the actions, making it more convenient for the latter to deal with continuous control. Therefore, this paper focuses on them.

Table 6-1 Common properties of the popular RL algorithms.

Algorithm	Type	Data usage	Action space
DQN	value-based	Off-policy	Discrete
DPG	policy-based	Off-policy	Continuous
DDPG	actor-critic	Off-policy	Continuous
TD3	actor-critic	Off-policy	Continuous
TRPO	policy-based	On-policy	Discrete/Continuous
PPO	actor-critic	On-policy	Discrete/Continuous

The RL algorithm can be off-policy or on-policy according to the interaction between the RL agent and the environment. For the off-policy method, the agent can learn by interacting with the environment in person or through accumulated experience (such as experience replay or replay buffer mechanism)[21]. In contrast, for the on-policy method, the agent can only interact with the environment to update the network. As the research object of this study is the measurement data collected by the actual HMES, the amount of data is limited, and the data collection is slow, so we would prefer to choose the off-policy method because they are more sample-efficient. In contrast, the on-policy method is more suitable for scenarios where data is generated using simulators.

6.2.2 Deep deterministic policy gradient (DDPG)

DQN is a value-based RL algorithm that uses a deep neural network (Q-network) to fit Q values. It overcomes the dimension disaster in the traditional Q-learning algorithm. The Q-network estimates the Q value of each discrete action, and $\varepsilon - greedy$ strategy is used to select the action with the highest Q value. In addition, The DQN adds the experience replay mechanism in the training process, which allows the DQN to randomly extract a batch of historical training data from the buffer for gradient descent of the network. This mechanism prevents the training data from being highly temporal correlated, improving the model's efficiency. The update rule of the DQN algorithm is as follows:

$$Q(s_t, a_t; \theta_t) \ = \ Q(s_t, a_t; \theta_t) \ + \alpha\left[r_i + \gamma max_{\alpha_{t+1}} Q(s_{t+1}, a_{t+1}; \theta') - Q(s_t, a_t; \theta_t) \ \right] \qquad (6\text{-}5)$$

Where θ_t means the parameters of the evaluation network, and θ'means the parameters of the target network. DQN estimates the new Q value more accurately by establishing these two independent neural networks.

It can be seen from Eq.6-5 that there is a calculation to find the maximum value when the DQN network is updated. For continuous action space, this maximization operation is impossible. Therefore, the DQN can only handle a finite discrete action space. DDPG algorithm is proposed to solve this problem[22]. Based on the DPG (deterministic policy gradient) algorithm, DDPG integrates the advantages of the DQN (such as experience replay mechanism and independent target network) and enables it to deal with the continuous action space by introducing the actor-critic framework.

The learning process of DDPG is shown in Fig.6-1. As we can see, the DDPG consists of two DNNs: online actor network $\mu(s|\theta^{\mu})$ and online critic network $Q(s, a|\theta^{Q})$, target actor $\mu'(s|\theta^{\mu})$ and target critic networks $Q'(s, a|\theta^{Q})$ are also used to stabilize learning, which has the same structure and initial parameters as the online network. It should be noted that the weights of all the above networks are fixed in training and updated at the end of each step. DDPG also borrows from DQN's experience replay mechanism, which is one of the common strategies used in on-policy methods. It can learn by sampling previous transitions from limited data, thus effectively improving the sample efficiency. It is beneficial for the training environment with relatively small data samples.

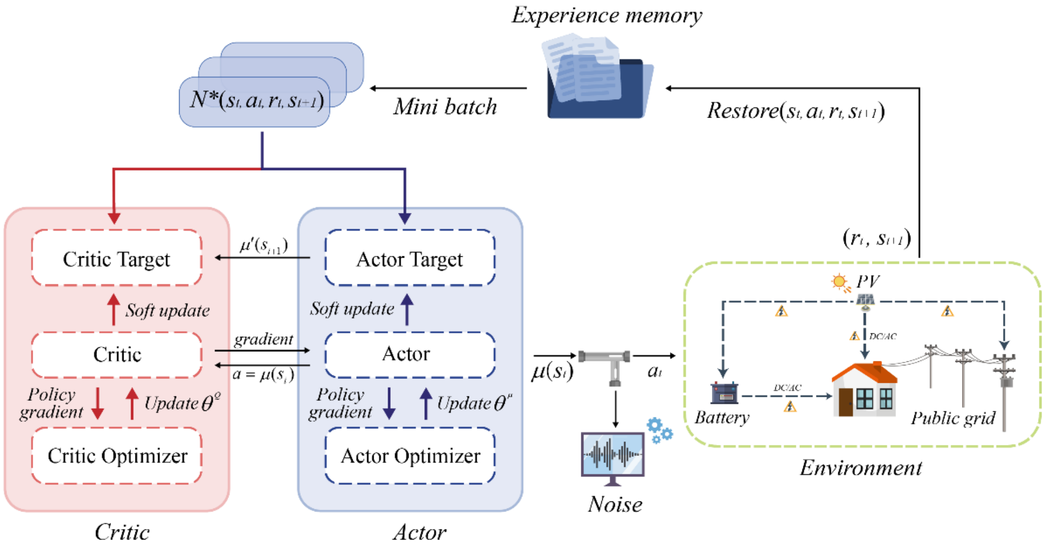


Fig. 6-1 Basic schematic of the DDPG.

When the agent begins to explore, the action selection is performed by passing the current state and random noise x_t through the actor network:

$$a_t = \mu(s_t|\theta^\mu) + x_t \tag{6-6}$$

Once the environment has executed the a_t, the agent will observe the reward r_t and the new state s_{t+1}, and then store the transition data tuples (s_t, a_t, r_t, s_{t+1}) in an experience replay buffer. Mini-batch sampling is performed using a replay buffer for training. The critic and target critic networks will evaluate the target value y_i by observing s_t and a_t, then update the critic network by minimizing the loss function L[22], as shown in Eq.7 and Eq.8:

$$y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'}) \tag{6-7}$$

$$L = \frac{1}{N}\Sigma_i(y_i - Q(s_i, a_i|\theta^Q))^2 \tag{6-8}$$

Next, the agent will calculate the policy gradient of the local actor network and update parameters through gradient ascent using the deterministic policy gradient[22]:

$$\nabla_{\theta^\mu}\mu|_{s_t} \approx \frac{1}{N}\sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_t, a=\mu(s_t)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_t} \tag{6-9}$$

At the end of each step, the agent updates the parameters of the target actor and critic networks by the running average method:

$$\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)|\theta^{Q'} \qquad (6\text{-}10)$$

$$\theta^{\mu} \leftarrow \tau\theta^{\mu} + (1-\tau)|\theta^{\mu'} \qquad (6\text{-}11)$$

Where τ is a smoothing parameter by which to maintain the stability of training.

6.2.3 Twin delayed deep deterministic policy gradient (TD3)

Based on the above introduction, we can conclude that DDPG is a variant of DQN to solve the continuous control problem. Therefore, it inherits a series of advantages of DQN but also disadvantages, such as overestimation. Overestimation means the estimated value function is larger than the actual. As seen from Eq.6-5, when the DQN agent finishes the explore, instead of the actual action of the next interaction, it updates the value function with the action currently considered to have the largest value so that it will overestimate the value of Q.

To solve the overestimation problem, Hasselt first proposed the double Q-Learning method, whose application in DQN is called Double DQN (DDQN)[23]. DDQN uses two value function estimates to perform the best action of the next interaction and target estimate using different value estimates, which effectively optimizes the Q-Value overestimation problem. Fig. 6-2 visually represents the structural differences between TD3 and DDPG. TD3 aims to solve the overestimation problem for DDPG by using a similar approach with a second critic and target critic pair[24]. Instead of using a specific target network in Eq.6-7, TD3 uses the minimum of the two critic networks to calculate the target values y_i, as shown in Eq.6-12:

$$y_i = r_i + \gamma \underset{i=1,2}{min} Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'}) \qquad (6\text{-}12)$$

In addition, TD3 reduces the update frequency of the Actor-network. The authors of TD3 found that if the policy is learned after the Q value is stabilized, there will be fewer wrong updates in the Actor-network, which can help stabilize the training[24]. Since TD3 is a minor improvement based on DDPG, its algorithm flow is basically the same as DDPG. Due to space limitations, the algorithm flow of TD3 will not be introduced in this section. The selection of experimental algorithms will be detail discussed in the following sections.

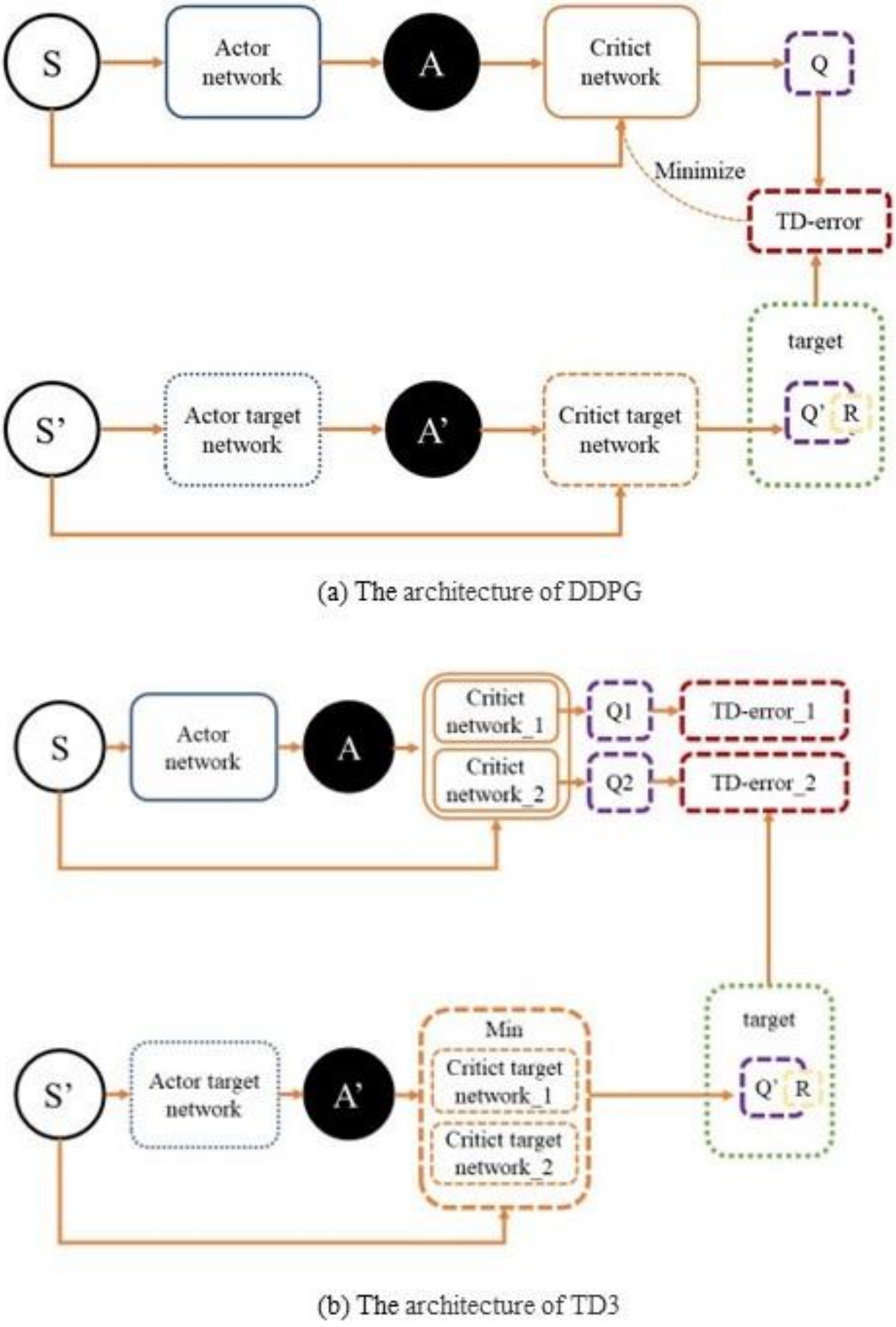


Fig. 6-2 Comparison of TD3 and DDPG.

6.2.4 Soft Actor-Critic (SAC)

The Soft Actor-Critic (SAC) algorithm is a deep reinforcement learning technique within the Actor-Critic framework. First proposed by Tuomas Haarnoja et al.[25], SAC has demonstrated its effectiveness in solving complex, continuous motion control problems. The algorithm's primary distinguishing feature is its use of entropy regularization. Entropy can be thought of as a measure of chaos and randomness. The SAC algorithm discourages overly deterministic strategies by encouraging agents to explore more widely through entropy regularization, which will help prevent the agent from getting stuck in local optima. This leads to a more diverse and effective set of learned behaviors and better overall performance.

SAC also uses the Actor-Critic framework, so its algorithm update process is similar to DDPG. The Bellman targets for the loss function in Eq. 6-8, which are used to update the critic, were briefly described. However, a different loss function for the SAC algorithm minimizes the Bellman error of the maximum entropy RL objective. Specifically, the following loss function is employed:

$$y_i = r_i + \gamma(\min_{i=1,2} Q'(s_{i+1}, a_{i+1}) - \alpha \log_{\pi_\theta}(a_{i+1}|s_{i+1}) \qquad (6\text{-}13)$$

In the SAC algorithm, the policy π_θ is commonly parameterized using a Gaussian distribution. The choice is theoretically justified in [25], where it is argued that the goal is to minimize the Kullback-Leibler divergence between π_θ and the softmax policy. Specifically, the function takes in a state-action pair and produces a corresponding value output. The policy takes in the current state and produces a distribution of possible actions. When an action is required, a Gaussian distribution with mean and the standard deviation is used for sampling, and the sample obtained is treated as the decision-making action for the policy.

From the above discussion, we can find that although there are many similarities between SAC and TD3, there are also some key differences between them, which can be summarized as follows:

- Similar to SAC, TD3 utilizes an exploration strategy that involves adding noise to the actions taken by the agent. However, the noise in TD3 is generated by the target policy smoothing function, which samples the action from a shifted Gaussian distribution centered on the action generated by the actor-network, which can improve the stability of the algorithm by preventing overestimation of the Q value and reducing the variance of the target Q value used in critic updates.

- TD3 uses three Q-value networks. The two Q-value networks estimate the target Q-values for each state-action pair, and the third Q-value network is used to compute the minimum Q-value between the two target Q-values. This approach of using multiple Q-value networks in TD3 helps to reduce the overestimation bias that can occur when using a single Q-value network. In contrast, SAC uses a single soft Q-value network that is updated at every time step, which can be more computationally efficient but may still suffer from overestimation bias.

- In TD3, the policy selects an action based on the current state, and then a small amount of noise is added to the action to encourage exploration. On the other hand, SAC uses a stochastic policy, meaning that it outputs a probability distribution over actions instead of a deterministic action.

- Compared to SAC, TD3 introduced several new hyperparameters, including a discount factor for target values, a clip range for critic updates, and the frequency of target policy smoothing. These hyperparameters can significantly impact the performance and stability of an algorithm, and tuning them can be a challenge. In contrast, SAC is relatively insensitive to hyperparameters and requires fewer hyperparameters to be tuned.

6.3 DL-based solution for energy prediction

The prediction models proposed in this study were verified using the dataset of an actual Japanese house in the “Jono Zero Carbon Smart Community” in Kitakyushu, described in detail in

Chapter 3. All the cases in this experiment were optimized hour by the hour using the actual measured hourly data set. The training set used in this experiment is hourly data from April 1, 2017, to September 30, 2018, and the test data includes hourly data from October 1, 2018, to September 30, 2019. Examples of data sets are shown in Section 3.4.2. According to the evaluation results of the prediction models in Chapter 4, we focused on evaluations of the following four deep learning algorithms: RNN, LSTM, and LSTM with attention mechanisms (A-LSTM).

This study proposes nine DL-based model architectures containing RNN, LSTM, and A-LSTM. These three algorithms were employed to predict the target building's electricity demand, PV capacity, and RTP. These prediction models were then used to determine the optimal one through an evaluation process, which would be integrated into the RL-based controller.

6.3.1 Implementation Details

The RNN model was used as the baseline model to outline how the model was built. After configuring the baseline model parameters, the same configuration was used to build the LSTM and A-LSTM models. To optimize the hyperparameters of the models, the Hyperopt framework was used, which implemented the tree-structured Parzen estimator (TPE) algorithm. Hyperopt is a Python library for hyperparameter optimization based on Bayesian optimization, which supports optimizing continuous, discrete, and conditional variables. To use the Hyperopt framework, four parameters had to be specified: the target function to optimize, the search space with hyperparameters, the Trials database, and the search algorithm. Therefore, the implementation details were explained based on these four aspects.

The RNN baseline model requires optimization of four parameters: the time step L of each RNN layer (determined by the length of previous data), the size of the hidden unit m of each layer, the size of the batch processing b during training (with default same hidden unit for each layer in the two-layer RNN structure), and the drop rate of the Dropout layer. To establish the range of L, we first conducted autocorrelation analysis on load data to detect data cycle patterns, as depicted in Fig.6-3. The X-axis and Y-axis of Fig.6-3 represent "hours" and "autocorrelation coefficient," respectively. It was observed that PV and RTP had obvious periodic characteristics, whereas the electricity demand did not exhibit a clear periodicity due to the randomness of user behavior. As such, we defined the conditional parameters of L as {12,24,36,48}.

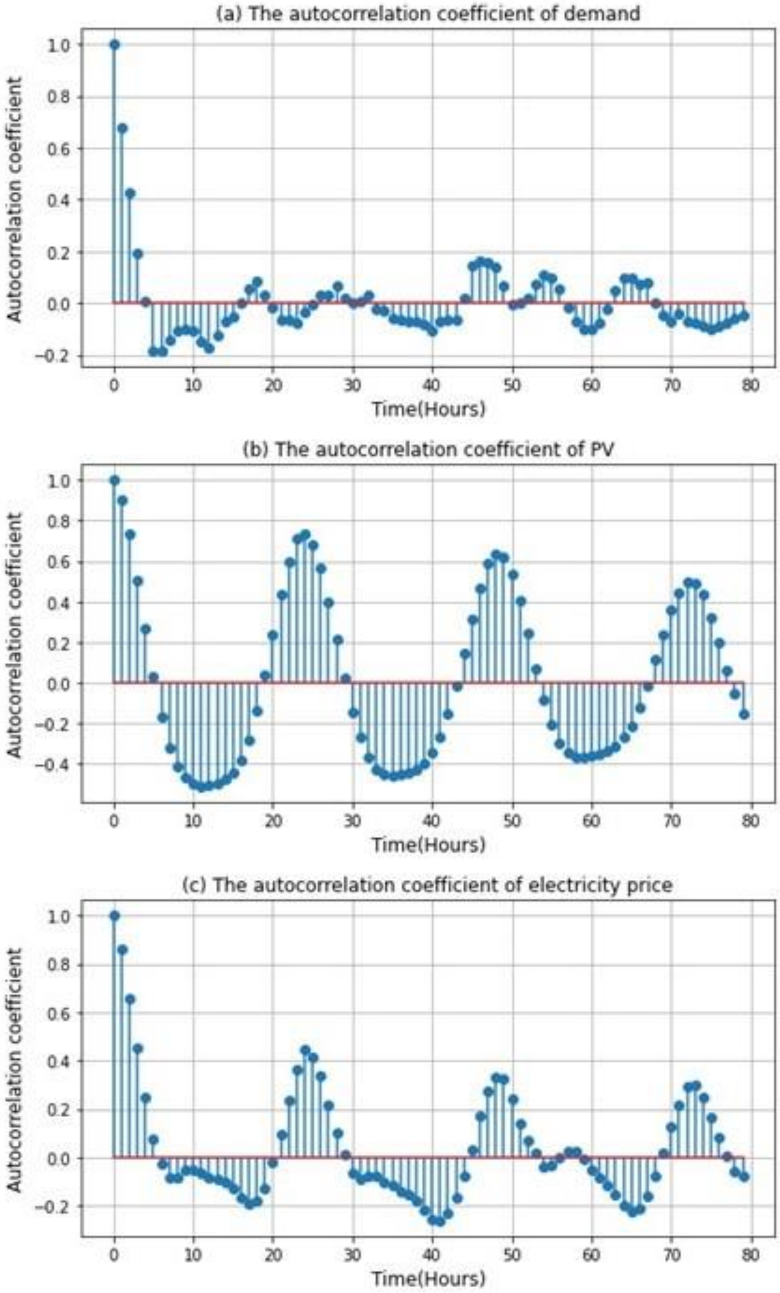


Fig. 6-3 The autocorrelation analysis results.

To avoid overfitting, we added a dropout layer after each RNN layer, with the conditional parameters of drop rate set as {0.2, 0.3, 0.4, 0.5}. Due to limited computational resources, we set the conditional parameter sets of m and b based on empirical methods to ensure prediction accuracy: m ∈ {32,64,128,256} and b ∈ {32,64,128,256}[26]. We input the aforementioned conditional parameters into the Hyperopt framework and utilized the TPE algorithm to optimize the model's super parameters. Table 6-2 shows the optimized RNN, LSTM, and A-LSTM hyperparameters, which the TPE algorithm determines.

Table 6-1 Hyperparameters for the prediction models

Prediction Object	Model	Layer1 Units	Layer2 Units	Time Step	Batch Size	Drop Rate
Electricity Demand	RNN	128	64	48	64	0.2
	LSTM	128	64	48	64	0.2
	A-LSTM	128	64	48	64	0.2
PV Generation	RNN	128	64	24	64	0.2
	LSTM	128	64	24	64	0.2
	A-LSTM	128	64	24	64	0.2
Real-time Prices	RNN	128	64	24	64	0.2
	LSTM	128	64	24	64	0.2
	A-LSTM	128	64	24	64	0.2

6.3.2 Training Setting and Performance Metrics

We use the TensorFlow2 framework in a Python environment to perform the training of the prediction models. To evaluate the time series prediction effect of these three models on the test data set, we used the same hyperparameter configuration and experimental methods in all experiments. All models have been trained and tested five times, and each training is 200 episodes. The final data used for comparison is the average of the 5 test results to reduce the errors caused by random numbers. Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R-Square Value (R2_SCORE) were used as indicators of the evaluation model, which were calculated according to Eq. (6-14), (6-15), and (6-16). The y_i denotes the real observations, $\bar{y}_i$ denotes the average of the observed value, $\tilde{y}_i$ denotes the predicted value, N denotes the number of test samples.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2} \tag{6-14}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \tilde{y}_i| \tag{6-15}$$

$$R^2 = 1 - \frac{\Sigma_{i=1}^{n}(y_i - \tilde{y}_i)^2}{\Sigma_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{6-16}$$

6.4 RL-based solution for energy storage management

This section presents the solution for energy storage management optimization of a grid-connected residential PV-battery system based on an actor-critic RL algorithm considering the predicted values. It includes a description of the complete method and parameter design adopted in the simulation experiments. Fig. 6-4 illustrates the schematic diagram of the system. The objective of the RL agent optimization in this experiment is to minimize the energy cost while ensuring the local absorption rate of PV. To achieve this goal, the agent continuously interacts with the environment and prediction model to learn, adjust, and improve the agent's behavior.

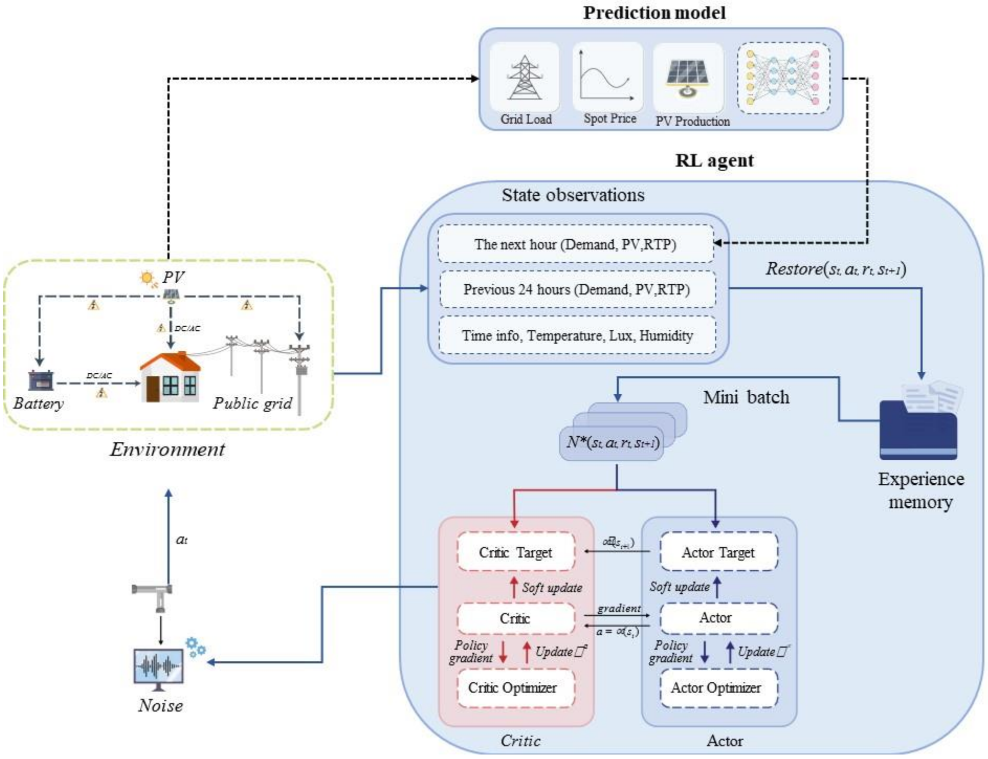


Fig. 6-4 The diagram of energy storage management decision control process.

6.4.1 Baseline control model

The baseline control model of the ESS is mainly based on the power balance formula and the battery capacity constraints, which can be expressed as:

$$P_t^{gird} = P_t^{demand} - P_t^{PV} + P_t^{battery} \qquad (6\text{-}17)$$

$$E_{min}^{battery} \leq E_t^{battery} \leq E_{max}^{battery} \qquad (6\text{-}18)$$

Where P_t^{gird} is the amount of electricity purchased or sold to the public grid at the time t; P_t^{demand} is the electricity demand at time t; P_t^{PV} is the PV generation at time T; $P_t^{battery}$ is the charge or discharge amount of the battery at time t. $P_t^{battery}$ >0 refers to the charge capacity, $P_t^{battery}$<0 refers to the discharge capacity. $E_{max}^{battery}$ denotes the maximum battery capacity and $E_{min}^{battery}$ denotes the minimum battery capacity.

In ESS's existing control logic, the battery works only when some fixed conditions are met. Algorithm 1 is the pseudocode of the specific control logic. This control logic avoids the arbitrage of renewable energy using batteries and reduces the energy loss due to battery efficiency, which is as follows:

- When the PV generation is greater than the demand and the battery capacity is less than the $E_{max}^{battery}$, the battery charges and the remaining PV generation is sold to the public grid for revenue. Where η_{cha} denotes the charge efficiency.
- When the PV generation is less than the demand and the battery capacity is greater than the $E_{min}^{battery}$, the battery discharges, and the insufficient power will be purchased from the public grid. Where $\eta_{disscha}$ denotes the discharge efficiency.
- If the above two conditions are not met, the battery will not perform any action.

Algorithm 1 Baseline Control Step

```
if $P_t^{PV} > P_t^{demand}$ and $E_t^{battery} < E_{max}^{battery}$ do
    $E_{t+1}^{battery} = E_t^{battery} + \eta_{cha} * P_t^{battery} * \Delta t$
end if
elif $P_t^{PV} < P_t^{demand}$ and $E_t^{battery} > E_{min}^{battery}$ do
    $E_{t+1}^{battery} = E_t^{battery} + \frac{P_t^{battery} * \Delta t}{\eta_{disscha}}$
end elif
else $E_{t+1}^{battery} = E_t^{battery}$
end else
```

The model-based RL method adopted in this study aims to optimize the baseline model using the strategies learned from the data rather than learning a new set of scheduling rules. It means all the proposed RL models will also interact with the environment under the rule of the Baseline control model. That is, the battery performs the action selected by the agent only after the agent determines whether the above conditions for charging and discharging are met. In this model-based RL approach, the agent can use the known rules of the baseline control model for fast and efficient learning, avoiding many unnecessary exploration actions, such as exceeding the battery capacity constraints, frequent selling PV generation to the public grid in pursuit of arbitrage, or other idle behaviors. It means that we can limit and narrow the exploration scope of agents according to the baseline model, thus reducing the number of trials and errors of agents and improving the utilization of training samples.

6.4.2 Model-based RL application

To solve the sequential decision-making problem with RL, we need to ensure that the decision process meets the Markov feature, and the decision process must be modeled as MDP. The MDP can be represented by a tuple (S, A, p, R, γ), where S denotes a state-space, A denotes an action-space, p denotes the state transition probability, R denotes the reward function, and γ denotes the discount factor which is used to calculate the cumulative reward. The rest of this section will expound on How the states, actions, and rewards are set up for this study.

6.4.2.1 States

The state observations S are the values the agents obtain when selecting actions. The state space in this study mainly consists of four parts:

- Energy features: As described in Chapter 3, PV generation, demand, and RTP are periodic in the time series. Therefore, we designed a 24-step (24h) sliding time window (when less than 24 hours, the list filling is 0) to enable the agent to learn their potential rules. Furthermore, the proposed optimal prediction models were used to predict these energy features, and the predicted value for the next hour was also considered observed values. By doing so, the RL agent could consider the future state more when making decisions.
- Time series features: The current hour in the day (X_t^{hour}) and the month (X_t^{month}).
- Environmental features: Outdoor temperature (X_t^{temp}), illumination (X_t^{lux}), and humidity (X_t^{hum}).
- Episode step: The position of the current time step in the entire optimization window (T).

In summary, the state space of proxy observation can be expressed as:

$$S_t\text{=}[\ T,\ s_{t-23}^{pv},\dots s_t^{pv},\ s_{t+1}^{pv},\ s_{t-23}^{demand},\dots s_t^{demand},\ s_{t+1}^{demand},\ s_{t-23}^{price},\dots s_t^{price},\ s_{t+1}^{price},\ X_t^{hour},\ X_t^{month},\ X_t^{temp},\ X_t^{lux},\ X_t^{hum}] \qquad (6\text{-}19)$$

To improve the training stability of the RL agent, all the observation values should be normalized in the pre-processing stage, which means that each variable's values should be scaled down to the range of [0,1].

6.4.2.2 Actions

Since this study's objective is to control the battery continuously, we used the battery control factor to achieve this operation. The battery's actual power was calculated based on the maximum charge and discharge per hour and the battery control factor. The battery control factor ranges from

−1 to 1 (the negative sign indicates that the battery is discharging, and the positive sign indicates that the battery is charging), which is also the action space used in this study.

6.4.2.3 Reward

Currently, there are two standard reward function design approaches discrete reward function and continuous reward function. As for the discrete reward function, it is easy to converge but contains less information. Conversely, the continuous reward function contains more information, but it is easy to have the problem of sparse rewards, making the training difficult to converge[27]. The optimization objective always determines the design of the reward function. In this study, the aim of the agents is to reduce the energy cost of the microgrid and ensure that the PV self-consumption ratio is not lower than the baseline model, which can be defined as a multi-objective optimization. The reward function of multi-objective optimization is often designed to consist of multiple parts and constraints. Therefore, we designed the reward function into two parts: economic reward and PV generation consumption reward.

First, the economic reward is calculated by the average cost of electricity imported to or exported from the microgrid. The average cost of electricity during timeslot 0~T is considered:

$$R_{eco} = -(a * \frac{1}{T}\int_{t=0}^{T}\left(P_{gird}(t) * C_{gird}(t) - P_{sell}(t) * C_{sell}\right)\mathrm{d}t) \qquad (6\text{-}20)$$

The minus sign indicates that if the average electricity cost is lower, the R_{eco} will be larger. Moreover, a denotes the reward factor, which is a fixed constant that regulates orders of magnitude, by which we can control the order of magnitude of R_{eco} within the range of -10 to 10; $P_{gird}(t)$ denotes the electricity purchased from the public grid by the system at time t, and $C_{gird}(t)$ denotes the real-time electricity price at time t; $P_{sell}(t)$ denotes the electricity sold by the system to the public grid at time t, and C_{sell} denotes the feed-in tariff.

Second, we used a discrete reward function to define the PV generation consumption reward. The calculation of the PV self-consumption ratio is shown in Eq. 6-21, by which we can get the agent's PV self-consumption ratio (r_{RL}) and the baseline model's PV self-consumption ratio ($r_{baseline}$).

$$r = \frac{\int_{t=0}^{T} \left(P_{pv}(t) - P_{sell}(t)\right) \mathrm{d}t}{\int_{t=0}^{T} P_{pv}(t)\, \mathrm{d}t} *100\% \qquad (6\text{-}21)$$

Where $P_{pv}(t)$ denotes the PV generation at time t, and $P_{sell}(t)$ denotes the electricity the microgrid sells to the public grid at time t. If r_{RL} is greater than $r_{baseline}$, then R_{pv} is equal to 1. In contrast, when r_{RL} is less than $r_{baseline}$, R_{pv} is assigned the value -10.

Finally, the sum of the two rewards is the primary reward function, which is as follows:

$$R = R_{eco} + R_{pv} \qquad (6\text{-}22)$$

6.4.3 Experimental setting

6.4.3.1 Implementation Details

The RL-based control models proposed in this study were verified using the dataset of an actual Japanese house in the “Jono Zero Carbon Smart Community” in Kitakyushu, described in detail in Chapter 3. Since the ultimate goal of this study is to solve the operational optimization problem of ESS in practical applications, all the cases in this experiment were optimized hour by the hour using the actual measured hourly data set. The training set used in this experiment is hourly data from April 1, 2017, to September 30, 2018, and the test data includes hourly data from October 1, 2018, to September 30, 2019. To verify the optimization effects of various RL algorithms under the model-based framework, we focused on evaluations of the following four algorithms: PPO, SAC, DDPG, and TD3. These four selected algorithms cover all actor-critic algorithm branches, shown in Section 6.2.1. In addition, we also added the baseline model for comparison, as shown below:

- M.0: The baseline model adopted in this experiment and its control flow is shown in Section 6.4.1.
- M.1: M.1 used the PPO algorithm based on the model-based framework proposed in this paper. The PPO algorithm is chosen for comparison since it is a typical on-policy RL method.
- M.2: M.2 used the SAC algorithm based on the model-based framework proposed in this paper.
- M.3: M.3 used the DDPG algorithm based on the model-based framework proposed in this paper.
- M.4: M.4 used the TD3 algorithm based on the model-based framework proposed in this paper. M.3 and M.4 used the same hyperparameters for comparison.

6.4.3.2 Training Setting

The deep learning framework adopted in this experiment is PyTorch, in which the environment code was written using the gym framework of OpenAI[28], and the RL algorithms adopted were implemented by the Pytorch version of the Stable Baselines framework[29]. The primary hyperparameters for different algorithm designs are shown in Table 6-3, and other hyperparameters follow default settings in the stable baseline. Although some algorithms may have a greater reward by fine-tuning the hyperparameters, we prefer to use more default parameters provided by Stable Baselines. Since fine-tuning the hyperparameters is impossible when deployed in a physical residential, we should pay more attention to the generalization ability of the models in practical application.

Table 6-3 Parameters for different algorithms.

Parameter	PPO	SAC	DDPG	TD3
Activation function	Tanh	Relu	Relu	Relu
Optimiser	Adam	Adam	Adam	Adam
Learning rate	0.0002	0.0002	0.0002	0.0002
Batch size	128	128	128	128
Replay memory capacity	None	1000000	1000000	1000000
Discount factor	0.99	0.99	0.99	0.99
Delay steps in TD3	None	None	None	2

6.4.3.3 Performance Metrics

We will use two metrics to evaluate the algorithm's performance: energy cost and PV self-consumption ratio. The energy cost is calculated by Eq.6-23. We will evaluate the energy cost of the above five models from annual and monthly dimensions.

$$\mathrm{c} = \int_{t=0}^{T} \left(P_{gird}(t) * C_{gird}(t) - P_{sell}(t) * C_{sell}\right) \mathrm{d}t \tag{6-23}$$

As for the PV self-consumption ratio, which calculation was shown in Eq.17. We will also evaluate it from annual and monthly dimensions. The evaluation standard is as long as the baseline model is exceeded as qualified.

6.5 Result and Discussion

6.5.1 Prediction model evaluation

6.5.1.1 Annual prediction evaluation

We utilized 18 months of data (from April 1, 2017, to September 30, 2018) as a training set to develop our models and evaluated their performance on the test set spanning from October 1, 2018, to September 30, 2019. The performance of the nine prediction models on the test set is presented in Table 6-4. The results indicate that, although the magnitudes of the three prediction targets differ, the PV generation prediction models have the best curve-fitting effect, as demonstrated by their higher R^2 values. This finding corresponds to the visible periodic pattern of PV power generation illustrated in Fig. 6-4. However, due to the fewer data quality, the prediction accuracy of electricity demand and RTP is slightly lower than that of PV. Nonetheless, the R^2 value of the optimal prediction model for these two targets still exceeds 0.75, indicating a reasonably good prediction accuracy. Therefore, we are confident that the accuracy of the prediction model will positively impact the performance of the control model.

Table 6-4 Evaluation metrics of candidate models.

Performance	Architecture	RMSE(kW)	MAE(kW)	R^2
Electricity	RNN	0.284	0.211	0.762
Demand	LSTM	0.280	0.205	0.769
	A-LSTM	0.291	0.207	0.751
Real-time	RNN	6.708	3.778	0.731
Prices	LSTM	6.148	3.309	0.774
	A-LSTM	6.564	3.789	0.742
PV	RNN	0.199	0.088	0.939
Generation	LSTM	0.188	0.093	0.946
	A-LSTM	0.185	0.090	0.947

The results demonstrate that among the three models for predicting power demand, LSTM exhibits the best performance. Compared to the second-best A-LSTM, LSTM shows a decrease of 0.011kW in RMSE, a decrease of 0.002kW in MAE, and an increase of 0.018 in R2. Similarly, among the three models predicting RTP, LSTM also outperforms the others. Compared to the second-best A-LSTM, LSTM shows a decrease of 0.416kW in RMSE, a decrease of 0.002kW in MAE, and an increase of 0.018 in R2. Notably, the performance of the three models for predicting RTP is very similar, and the performance of A-LSTM is virtually indistinguishable from that of LSTM. This experimental result confirms the conclusion drawn in Chapter 4, namely that when the training sample is less than 2 years (with a data amount of less than 16,320), A-LSTM's prediction accuracy is inferior to that of LSTM, which can be interpreted as A-LSTM having a lower sample utilization ratio than LSTM. It should be noted, however, that for high-quality training data such as the training set of photovoltaic power generation, the prediction results of the two models tend to be similar.

6.5.1.2 Seasonal prediction evaluation

It should be noted that the selection of the prediction model should consider not only the accuracy of the prediction but also the stability of the prediction performance, the sample utilization rate, and the ease of parameter tuning. Table 6-5 shows the forecast performance of the forecast model in different seasons. We can find that the prediction accuracy of photovoltaic power generation can maintain stability throughout the year. However, the prediction accuracy of power demand and RTP decreases in the transition season because the transition season is often accompanied by irregular changes in environmental factors and user behavior, which undoubtedly challenges the prediction model. The performance of the three prediction models in each season's typical week is shown in Fig. 6-5, Fig. 6-6, and Fig. 6-7.

Table 6-5 Evaluation metrics of candidate models in different seasons.

Performance	Period	Architecture	RMSE(kW)	MAE(kW)	R^2
Electricity Demand	Cooling Season	RNN	0.284	0.211	0.762
		LSTM	0.280	0.205	0.769
		A-LSTM	0.291	0.207	0.751
	Transition Season	RNN	0.210	0.156	0.602
		LSTM	0.203	0.149	0.626
		A-LSTM	0.214	0.156	0.584
	Heating Season	RNN	0.321	0.240	0.709
		LSTM	0.319	0.238	0.713
		A-LSTM	0.323	0.229	0.704
Real-time Prices	Cooling Season	RNN	3.047	2.151	0.715
		LSTM	2.809	1.678	0.758
		A-LSTM	2.972	1.915	0.729
	Transition Season	RNN	5.785	3.225	0.723
		LSTM	5.672	3.242	0.734
		A-LSTM	5.684	3.272	0.732
	Heating Season	RNN	7.680	4.312	0.711
		LSTM	6.702	3.667	0.780
		A-LSTM	7.396	4.414	0.732
PV Generation	Cooling Season	RNN	0.191	0.089	0.952
		LSTM	0.183	0.089	0.956
		A-LSTM	0.181	0.085	0.958
	Transition Season	RNN	0.191	0.089	0.952
		LSTM	0.181	0.085	0.957
		A-LSTM	0.183	0.089	0.956
	Heating Season	RNN	0.171	0.084	0.955
		LSTM	0.172	0.082	0.955
		A-LSTM	0.170	0.075	0.956

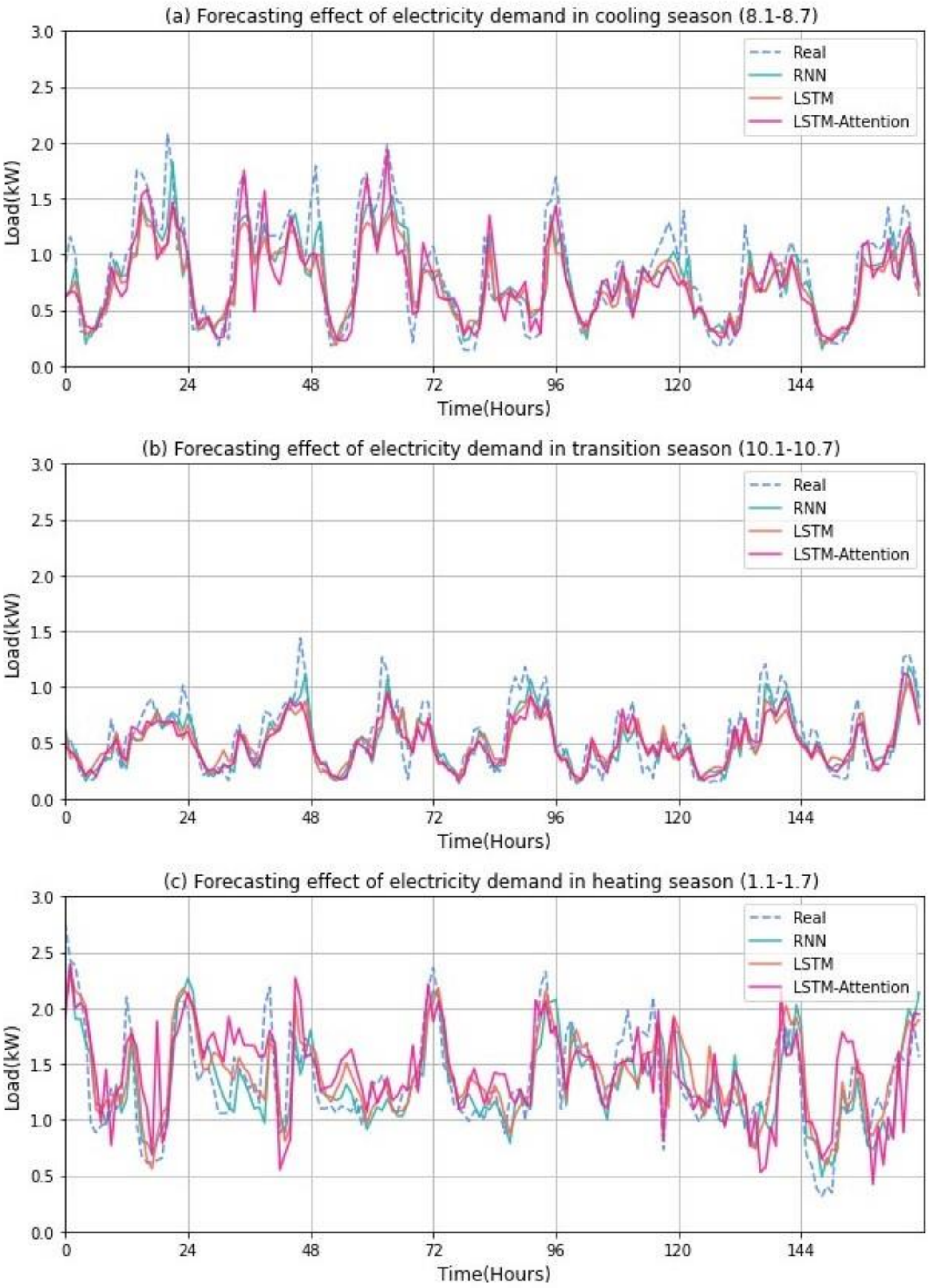


Fig. 6-5 The electricity demand prediction results of random one-week candidate models in different seasons.

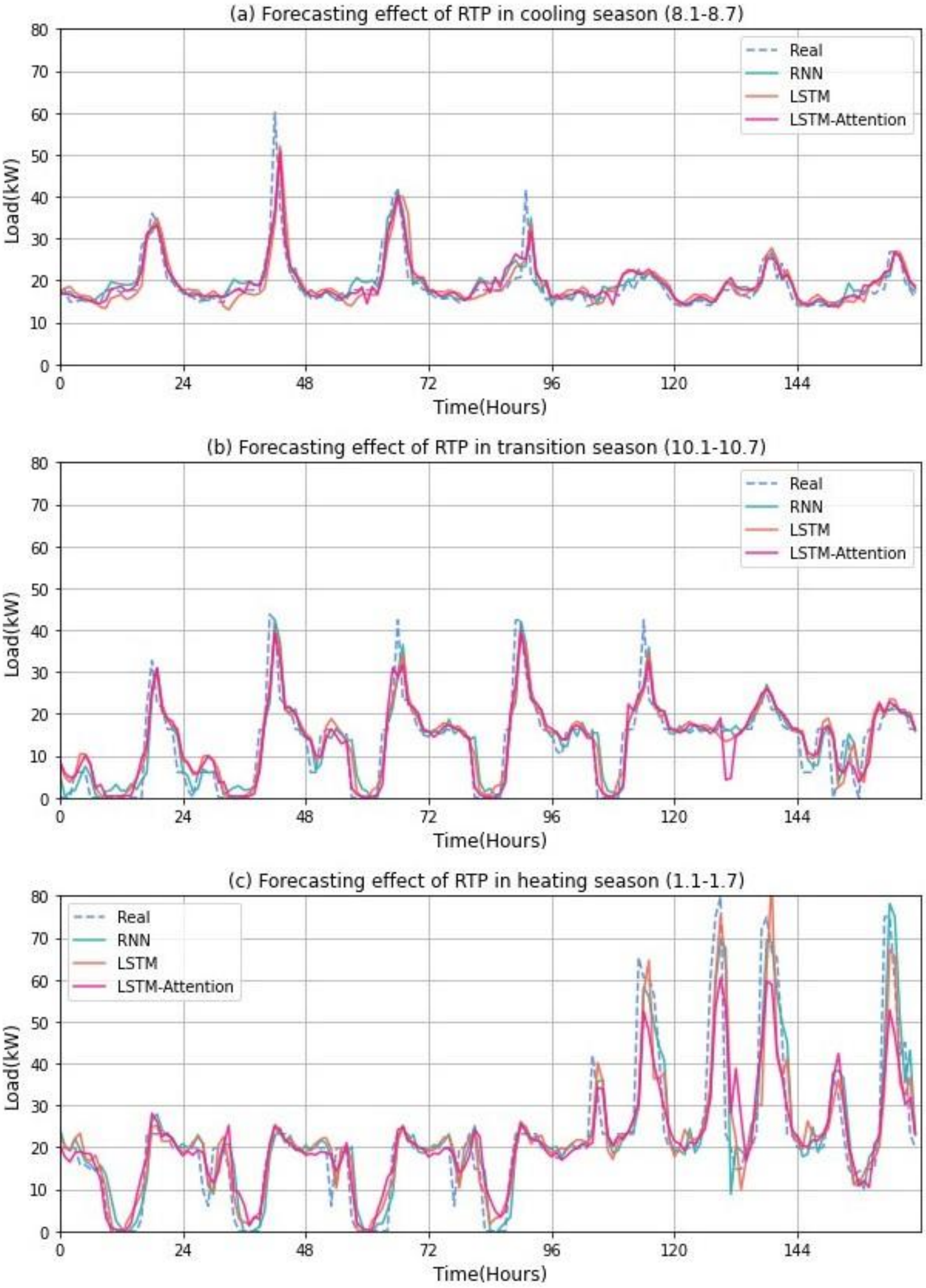


Fig. 6-6 The RTP prediction results of random one-week candidate models in different seasons.

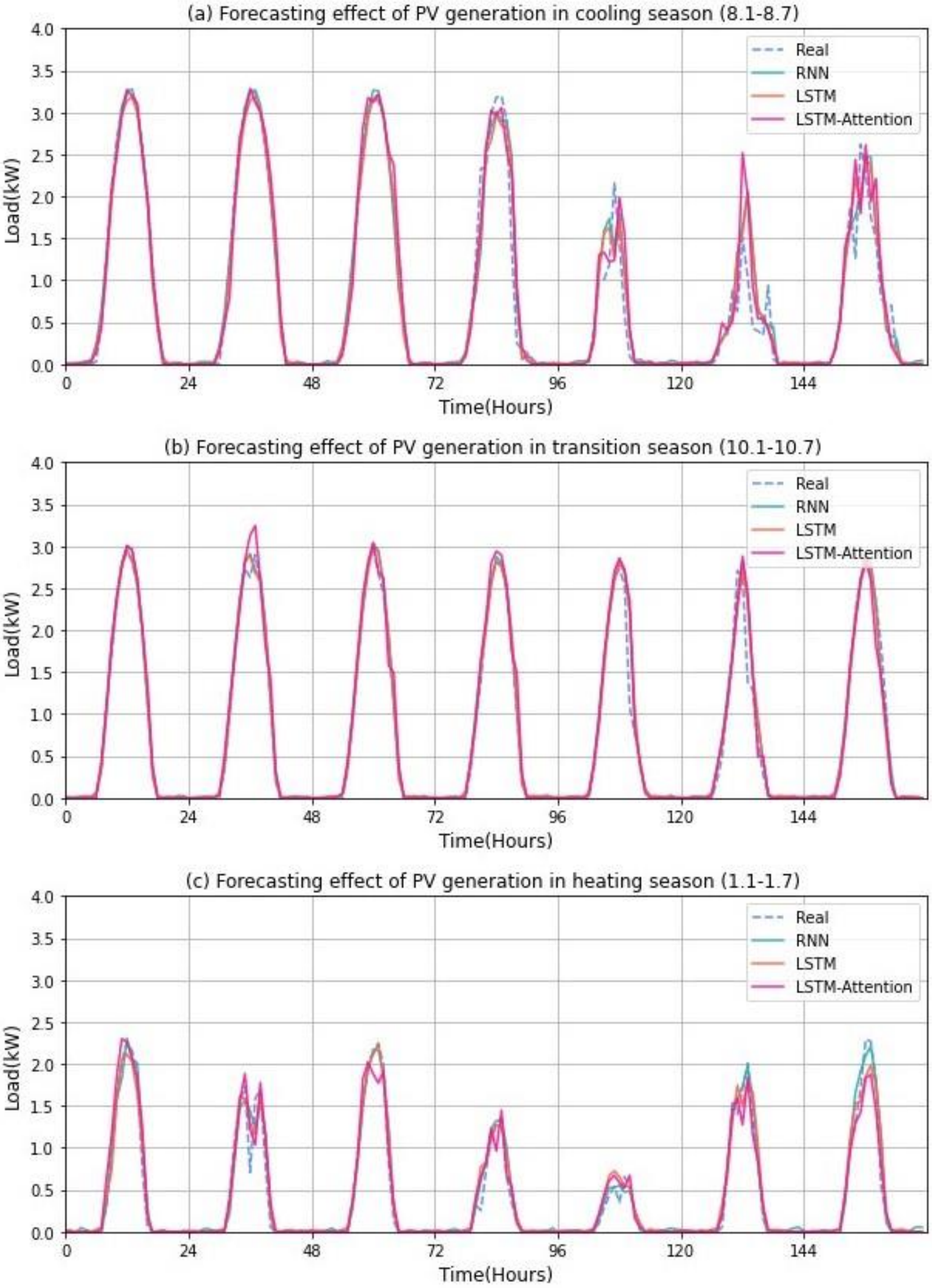


Fig. 6-7 The PV generation prediction results of random one-week candidate models in different seasons.

Fig. 6-5 illustrates that all three electricity demand prediction models fit the demand curve well.

However, it is noticeable that power demand in the transition season is at its lowest, and the prediction performance of the three models in this season is relatively poor (with the R^2 value being the lowest of the entire year). Nevertheless, the three models can predict the moment when the peak demand occurs accurately, even though there may be errors in the specific values, which can provide valuable information for the control model. Comparing all the prediction models, it can be observed that LSTM has the most stable forecasting performance, while RNN and A-LSTM exhibit considerable volatility. Therefore, we have comprehensively judged and selected LSTM as the forecasting model for power demand.

Fig. 6-6 illustrates the predictive performance of RTP over one week. It can be observed that for datasets with relatively high volatility, the actual predictive performance of RTP is similar to that of power demand (with similar R^2 values). Although there are deviations in the actual predicted values, the three prediction models can accurately predict the period of peak electricity prices. It is worth noting that all three models display a lag of 1 hour when predicting the time point of peak electricity prices. After observation, we found that the predictive model is accurate at the time point of the electricity price trend. Still, the specific value is smaller than the actual value, leading to a smaller curve growth trend. By comparing the parameters in Table 6-5, it was found that LSTM outperforms the other two models, and thus, we chose LSTM as the RTP prediction model.

Based on the high autocorrelation of PV generation data, the three prediction models demonstrated excellent performance, fitting the actual curve well, with only slight errors observed in the peak period. By examining the specific performance of each model in Fig. 6-7, it was found that A-LSTM outperforms the other two models in predicting peak time, which can provide valuable information for the control model. Although the evaluation index of LSTM is comparable to that of A-LSTM, it was observed that LSTM's performance is more stable, but its ability to predict peak time is not as accurate as A-LSTM. Therefore, this experiment selected A-LSTM as the PV generation prediction model.

In summary, based on the experimental results and comprehensive judgment, we selected LSTM as the prediction model for power demand and RTP. At the same time, A-LSTM was chosen as the PV power generation prediction model. Although the prediction models may have some errors in specific values, they can still provide valuable information for the control model. It is worth noting that the sample utilization ratio of A-LSTM is less than that of LSTM when the training sample is less than two years, which may affect its prediction effect. However, for high-quality training data such as photovoltaic power generation, the prediction results of A-LSTM and LSTM tend to be similar. The experimental results also revealed the seasonal and periodic rules of power demand and PV generation, which can be utilized in developing effective control strategies.

6.5.2 RL agent for Data-Driven control

6.5.2.1 Training process analysis

Through the callback function provided by Stable Baseline[29], we found that most models generally reach the highest cumulative reward during 50 to 60 episodes of training, and each episode simulates 13199 hours of run optimization. The average episodic rewards of the different algorithms across all 60 episodic can be found in Fig. 6-8. After taking random actions for the first ten episodes of exploration, all agents show similar initial behaviors and begin to gain benefits progressively. After around forty episodes, The average reward growth of all agents starts to slow down and gradually converges. We can see that M.1 fluctuates significantly in the initial stage, which is determined by the nature of its on-policy. Its performance tends to be stable with the increased number of training episodic. It indicates that the training performance of the PPO algorithm based on the on-policy method is weaker than that of other off-policy algorithms on small data samples. Conversely, Although TD3's average reward is close to that of other algorithms in the first ten episodes, it keeps the highest average reward after that. It proves that TD3 performs significantly better than the other algorithms, with a best average reward of more than 0, suggesting that it can extract valuable knowledge from the data more efficiently.

This section will summarize the results of the simulation experiment. All the RL models are trained for 50 episodes for fairness and efficiency. We first evaluated all the proposed RL algorithms against the two optimization objectives presented in Section 6.3.2.3. Then we analyzed the performance of these algorithms in detail through a visual analysis of three typical cases.

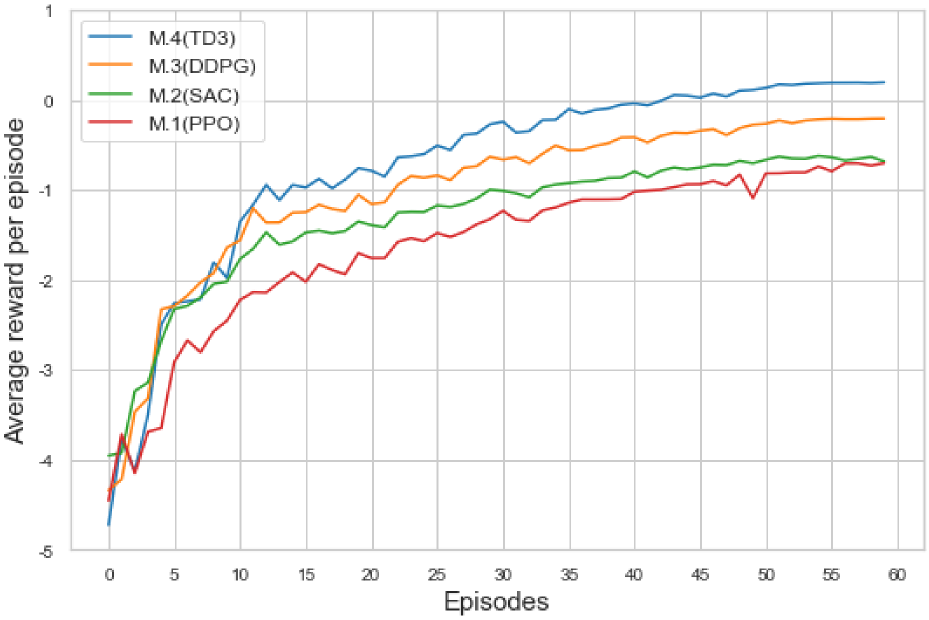


Fig. 6-8 Average rewards per episodic in the training process.

6.5.2.2 Energy cost optimization analysis

First, we compare the annual energy costs of the four algorithms and the baseline model, which is summarized in Table 6-6. Note that a positive cost represents purchasing power from the public grid, while a negative cost represents the profit from selling PV generation to the public grid. We can see all RL algorithms achieve the optimization of energy cost for the total annual cost. PPO has the worst optimization effect among the four algorithms, with its energy cost reduced by only 4.02% compared to the baseline model. In addition, it did not achieve cost optimization targets in June. The other three off-policy algorithms have achieved good results, which proves that the off-policy algorithm is more suitable to deal with the practical applications of this scenario with better sample efficiency. Among them, the annual optimization effect of TD3 is the best, which reduces the cost by 17.82% compared with the baseline model. Next came DDPG, close to TD3, with a 15.45% cost reduction relative to the baseline model. It also proves that the actor-critic framework algorithms have advantages in dealing with this scenario.

It is important to note that SAC is not optimized as well as DDPG and TD3 in this study. This is because SAC is better suited for tasks requiring more exploration and diversity, while TD3 is better suited for tasks prioritizing accuracy and stability. Additionally, this study used a pre-tuned model from the Stable Baselines framework[29], compensating for the hyperparameter tuning disadvantage of TD3. At the same time, the model-based optimization framework further limits SAC's exploration capabilities. Therefore, TD3 and DDPG have an advantage over SAC in this scenario.

Additionally, Table 6-7 and Fig. 6-9 compare energy savings costs between the models in Chapters 5 and 6. Both models use the same training and test sets and adopt a model-based framework. However, there are three main differences between the two models. Firstly, the model in Chapter 6 uses multi-objective optimization, while the model in Chapter 5 uses single-objective optimization. Secondly, the observed value of the model in Chapter 6 includes the predicted value for the next moment, whereas the model in Chapter 5 only considers the current moment. Thirdly, the RL model of the actor-critic is adopted in Chapter 6, while the value-based model is used in Chapter 5. Through comparison, we find that the TD3 improvement method proposed in Chapter 6 has the best optimization effect, resulting in an annual cost savings increase of 3155.82Yen compared to the D3QN method used in Chapter 5, which indicates that the proposed improvement method in Chapter 6 is successful and contributes to significant cost savings.

Table 6-6 Annual energy cost result and percentage difference against the Baseline model.

Month	M.0(Yen)	M.1(Yen)	M.2(Yen)	M.3(Yen)	M.4(Yen)
Jan	22575.21	21638.04	21049.83	21536.78	21375.14
Feb	19257.42	18631.07	17990.31	18126.05	18151.57
Mar	2424.09	2161.89	2079.55	1999.84	1968.18
Apr	-1276.51	-1329.10	-2021.08	-1996.41	-2013.67
May	-1990.03	-2077.12	-2552.06	-2593.28	-2500.74
Jun	-4016.58	-3949.97	-4806.07	-4706.14	-4831.08
Jul	-348.45	-372.59	-1091.39	-1098.56	-1085.25
Aug	1894.51	1846.35	1449.84	1152.54	1161.03
Sep	1046.97	1011.18	670.38	312.86	280.71
Oct	-610.77	-765.32	-781.51	-1613.40	-1689.44
Nov	9312.58	8941.64	8759.79	7883.95	6980.48
Dec	16151.86	16091.94	15592.44	15463.64	15142.14
Total cost	64420.30	61828.01	56340.03	54467.88	52939.06
VS Baseline		4.02%	12.54%	15.45%	17.82%

Table 6-7 Compares cost savings on validation sets of the models proposed in this study.

Algorithm	Framework	Num of optimization objective	Considering real-time prediction values	Cost saving (Yen)
PPO	Model-based	Two	Yes	2592.29
SAC	Model-based	Two	Yes	8080.27
DDPG	Model-based	Two	Yes	9952.42
TD3	Model-based	Two	Yes	11481.24
Q-learning	Model-based	One	NO	1952.89
DQN	Model-based	One	NO	5961.99
D3QN	Model-based	One	NO	8352.42

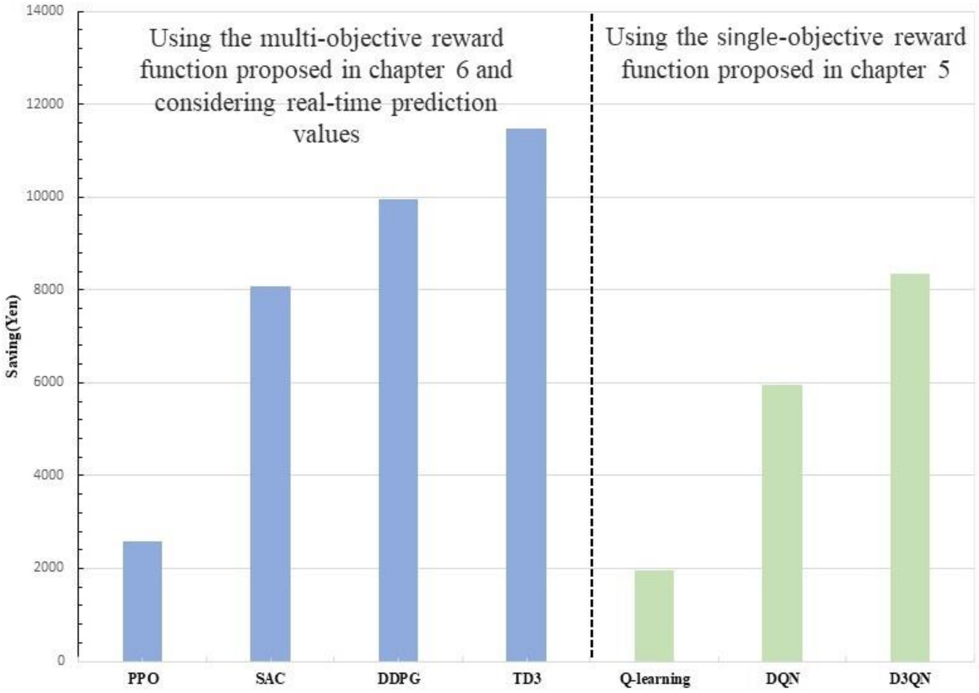


Fig. 6-9 Compares cost savings on validation sets of the models proposed in this study.

Since the test dataset's features fluctuate greatly in different months, the annual statistics cannot fully reflect the optimization effect of these algorithms. To evaluate the experimental results in more detail, we divided the year into three periods according to the use of HVAC: heating season, cooling season, and transition season. We have calculated the energy cost savings of the four algorithms in these three periods, respectively, and the results are summarized in Table 6-8.

Table 6-8 Quarterly cost savings against the Baseline model

	M.1(Yen)	M.2(Yen)	M.3(Yen)	M.4(Yen)
Heating season	1994.37	3904.70	4286.64	5647.73
Cooling season	41.48	2353.69	2915.74	3051.04
Transition season	556.43	1821.88	2750.03	2782.46

We can see that the cost optimization effect of the four algorithms is the best in the heating season. It can be seen from Fig.3-7a that the energy demand and RTP fluctuation in the heating season are the strongest, while the PV generation is the lowest in the whole year. In this case, the

RL agents can pay more attention to the fluctuations of demand and RTP, according to which agents will intelligently choose the time point of charge and discharge to realize the optimization of energy cost. For the cooling season, we can also find in Fig.3-7a that its average energy demand, RTP, and PV generation are both high, which puts forward higher requirements on the learning ability of agents. The optimization results show that M2, M3, and M4 can learn the rule of feature change well, while M1 fails to achieve this goal. We can also find that the energy demand and PV generation in the transition season are close to that in the cooling season. The only difference is that the RTP in the transition season is relatively low, so the agent only needs to focus on PV and demand schedule during this period. The performance of M.2, M.3, and M.4 remained stable in this period, while that of M1 also picked up. The reasons for these phenomena are described in detail in section 6.5.2.4.

6.5.2.3 PV self-consumption ratio optimization analysis

To ensure the self-consumption ratio of renewable energy and avoid the system using RTP fluctuations for arbitrage, we set the PV self-consumption ratio of the ESS during the optimization period should be higher than the baseline model. We calculated the annual PV self-consumption ratio of the test dataset, and the results are shown in Table 6-9. It can be seen that all the algorithms proposed in Chapter 6 have reached the optimization goal. Surprisingly, the PV self-consumption ratio of PPO was the highest, while TD3 was the least. Since the reward function used in this experiment was composed of energy cost and self-consumption ratio, it showed that different algorithms had different sensitivities to these two parts. In future research, we should try to fine-tune the reward function's weight coefficient a according to different algorithms to obtain a better optimization effect.

According to the comparison with the model proposed in Chapter 5, it can be concluded that the reward function proposed in this chapter is effective. The comparison reveals that the PV self-consumption ratio of the three models proposed in Chapter 5 does not reach the level of the baseline model because the reward function in Chapter 5 does not include rewards and punishments for the PV self-consumption ratio. This comparison proves that the reward function proposed in this chapter, which includes rewards and punishments for the PV self-consumption ratio, is important in achieving higher photovoltaic absorption rates.

Table 6-9 Annual PV self-consumption ratio results

Algorithm	PV self-consumption ratio	Num of optimization objective
Baseline	66.80%	NA
PPO	68.91%	Two
SAC	69.78%	Two
DDPG	67.98%	Two
TD3	67.66%	Two
Q-learning	65.98%	One
DQN	65.81%	One
D3QN	65.73%	One

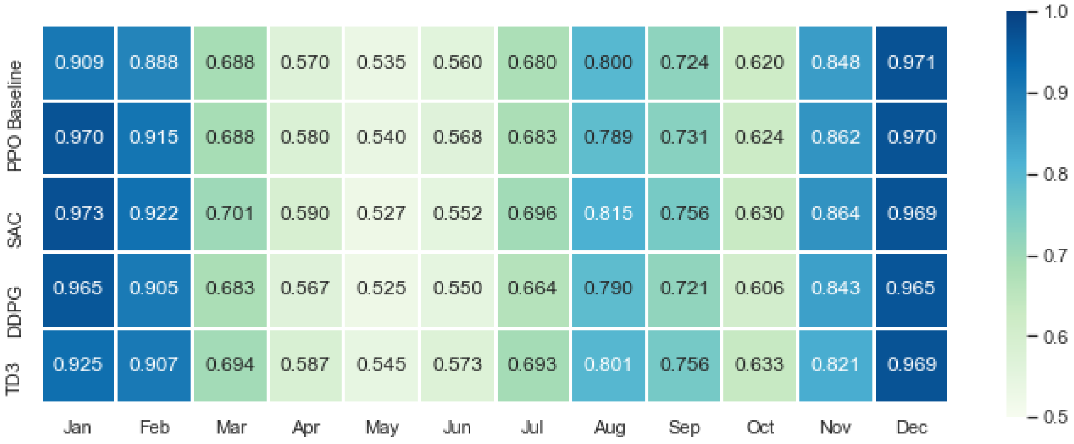


Fig. 6-10 Thermal map of PV self-consumption ratio

The comparison of the PV self-consumption ratio in different months is shown in Fig.6-10. It can be seen that the PV self-consumption ratio of M.1 and M.2 is significantly higher than that of M.3 and M.4 in winter. However, their performance is similar in other seasons. It indicates that M.1 and M.2 are more sensitive to the PV self-consumption ratio in winter, thus neglecting the energy cost optimization. The analysis in the previous section showed that the cost optimization effect of M.1 and M.2 was weaker than that of M.3 and M.4, which also confirmed this conclusion.

6.5.2.4 Visualization analysis of optimization results

This section describes the optimization performance of each algorithm in one week to demonstrate specific optimization strategies. We will discuss three typical cases of the heating, cooling, and transition seasons according to the classification standards in Section 4.2.1.

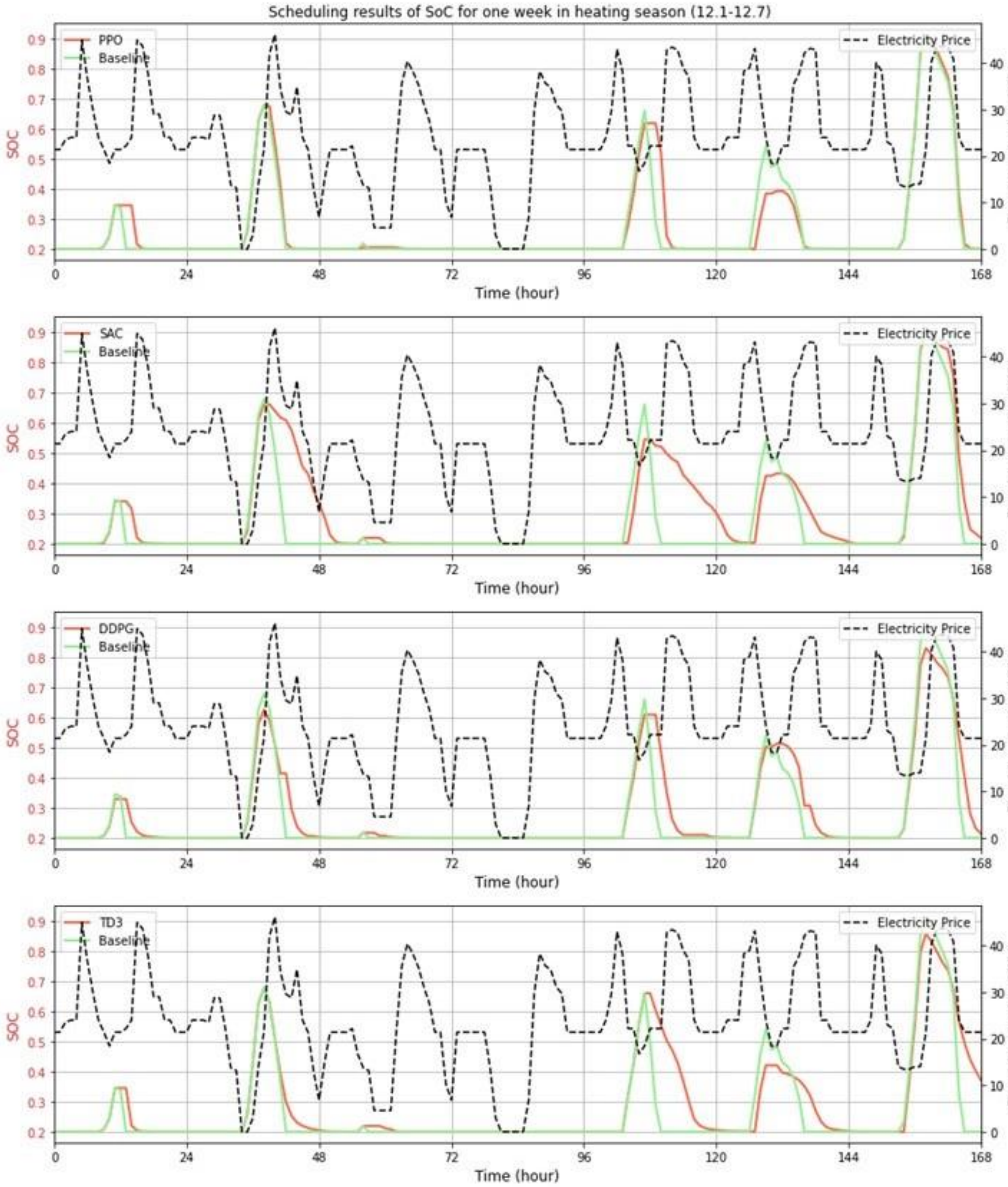


Fig. 6-11 One-week optimization results of the heating season

Fig.6-11 shows the optimization results of different optimization algorithms during one week in the heating season. The left ordinate indicates the state of charge(SOC), the right ordinate indicates the RTP and the abscissa indicates the hour. For this week's optimization task, we can see that the four algorithms have adjusted based on the baseline model. As seen from Fig.6-11, all algorithms can predict future demand and RTP trends and adopt the strategy of storing power for the possible price peak in the evening to achieve cost optimization. Both DDPG and TD3 performed well, with optimized results of 285.62(Yen) and 321.78 (Yen), respectively. However, PPO failed to execute the strategy on the second, sixth, and seventh days, thus achieving an optimization result of only 60.2 (Yen). On the contrary, SAC over-implemented this strategy. It chose too low discharge efficiency on the second and fifth days, which resulted in the discharge action missing the peak price, thus achieving a cost optimization result of 190.21 (Yen). This proves that DDPG and TD3 based on the actor-critic framework have a good learning effect on cost optimization.

Fig.6-12 shows the optimization results of different optimization algorithms during one week in the cooling season. We found that the optimization strategy of each algorithm at this period includes the following two points: (1) When the battery has stored enough power, it will hold it until the evening price peaks occur. (2) When the RTP is low in the morning, the battery will choose to delay charging, and the PV generated during this period will be sold to the public grid for profit. The reason for the appearance of strategy 2 is that PV generation is abundant in summer, and the PV self-consumption ratio can be ensured not to be lower than the baseline model even if the PV sales volume is increased. It can be seen that DDPG and TD3 are more inclined to strategy 1 in the selection of optimization strategy, and they perform well in the selection of discharge time point and slightly increased the sale of PV in the morning, with optimization results of 101.56(Yen) and 167.06(Yen), respectively. In contrast, PPO paid more attention to strategy 2. Although the sale of PV increased, the cost optimization goal was not achieved due to poor selection of discharge action. It can be seen that constrained by the model-based framework, the space for each algorithm to realize cost optimization by increasing the sale of PV is very limited, and proper discharge action selection is the key to ensuring optimization efficiency.

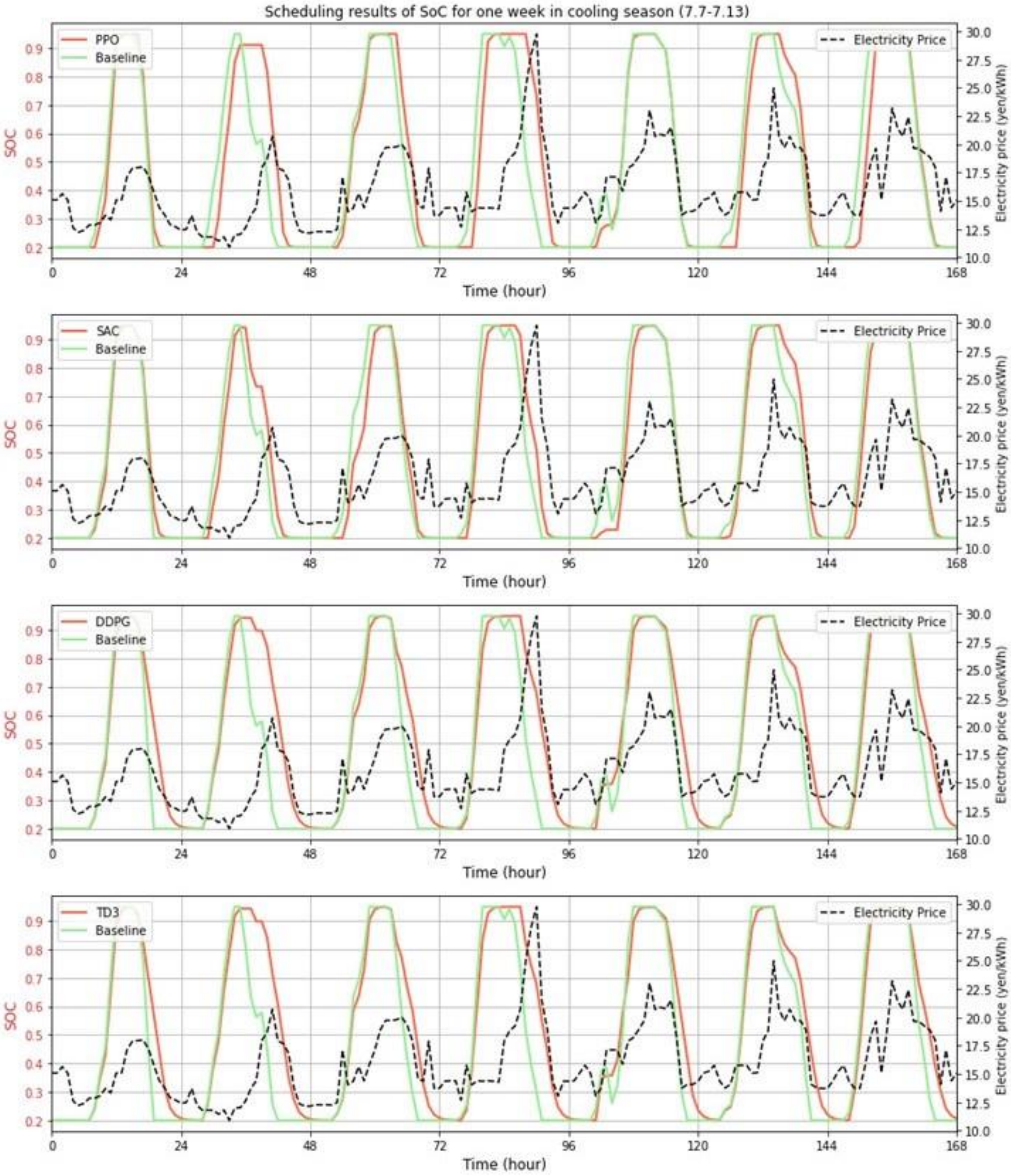


Fig. 6-12 One-week optimization results of the cooling season

Fig.6-13 shows the optimization results of different optimization algorithms during one week in the transition season. The strategies adopted by the algorithms during the transition season are similar to the cooling season because these two seasons' data distribution is very similar, except for the difference in energy demand. By calculating the optimization result, we found that the optimization effect of TD3 is still the best (805.14 Yen), followed by DDPG (168.33 Yen). It can be seen that the optimization effect of SAC is only a fine-tuning of the baseline model, so it only achieved an optimization structure of 34.57 (Yen). However, PPO failed to optimize again because it incorrectly adjusted the charging rate on the first and fourth days, resulting in insufficient power, thus reducing the utilization of renewable energy.

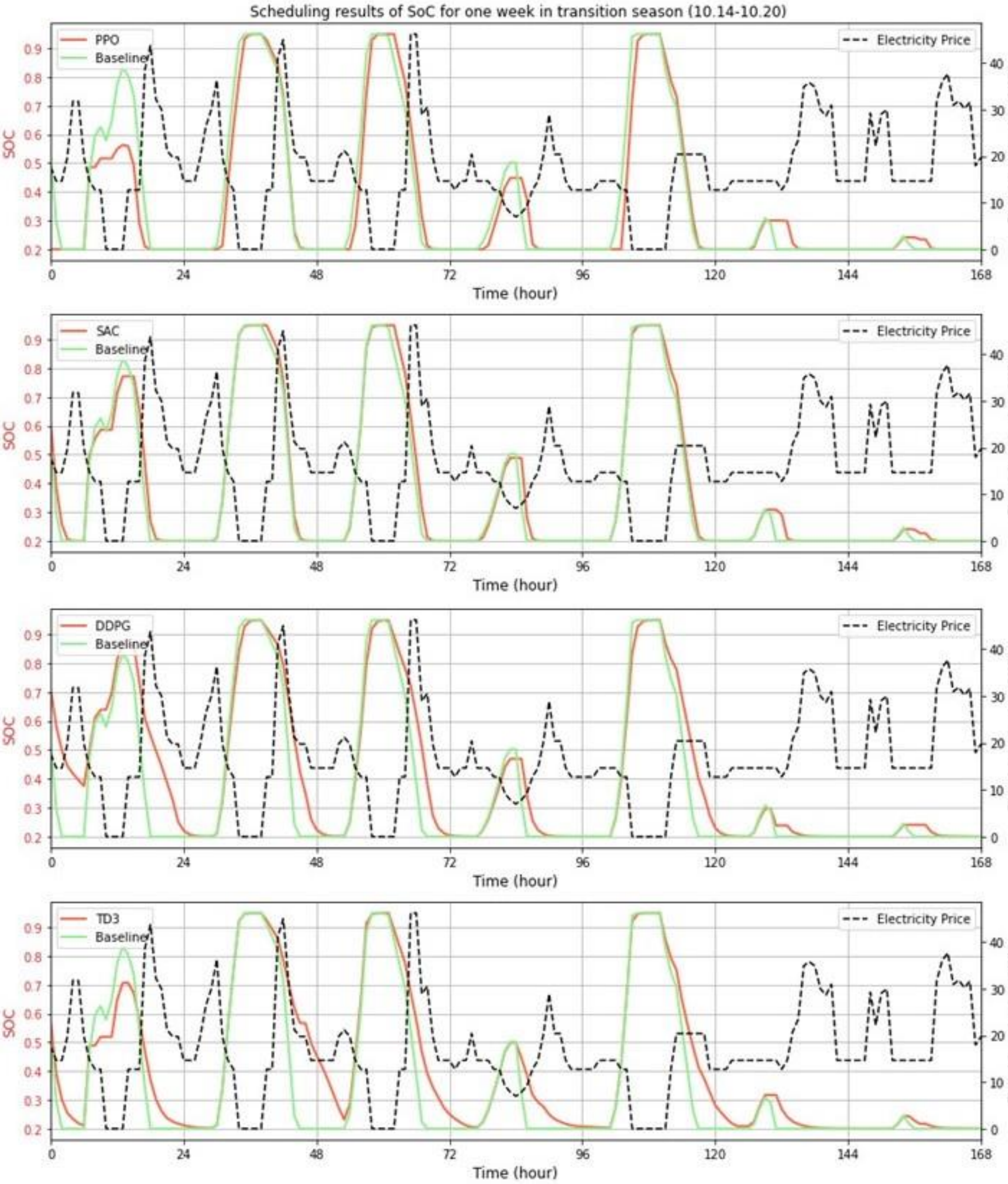


Fig. 6-13 One-week optimization results of the transition season

6.6 Conclusion

The RL-based energy control approach presents a promising potential for improving building energy efficiency due to their ability to learn strategies from environmental data and their scalability. This study proposed a model-based RL control method considering real-time prediction values for operation optimization of the residential PV-battery system. The optimization goals aim at reducing the energy cost of the microgrid and ensuring that the PV self-consumption ratio is not lower than the baseline model. To achieve this goal, this study designed a new multi-objective optimization reward function, and experimental results proved the effectiveness of the designed reward function. One of the key steps in this study was to develop and evaluate nine different prediction models with varying structures to predict power demand, real-time electricity price, and photovoltaic power generation. The optimal prediction model was selected for each variable through a comparative evaluation process. The experimental results showed that LSTM is more suitable for energy prediction of small sample data of household energy systems. Subsequently, the predicted value from the selected models was incorporated into the observed state variable of the RL models for the next time step.

In the case study, we analyzed the differences in optimization strategies between these four algorithms and evaluated their optimization efficiency during different periods. Although many RL-based energy management applications have been proposed, only a few studies have compared the optimization strategies between different RL algorithms in practical applications. This paper fills this gap, helping users better understand the performances of different algorithms in this scenario to facilitate the selection of RL algorithms for specific applications.

In implementing the control model, we first set the benchmark strategy used by the target building as the baseline model, which has been validated in Chapter 5. Then we adopted four advanced RL algorithms (PPO, SAC, DDPG, and TD3) to optimize the operation of the baseline model. The experimental results showed that the above four algorithms could achieve the optimization objective by using the designed reward function in this study. Furthermore, the TD3 algorithm had the best performance in each season of the year. It could reduce the annual energy costs by 17.82% and increase the PV self-consumption ratio by 0.86% compared with the baseline model. In addition, the improved method proposed in this chapter is superior to the models proposed in Chapter 5 in terms of cost optimization and PV self-consumption ratio, which indicates that the solution proposed in this chapter is a better approach for this scenario.

Future research will first focus on designing and optimizing reward functions in scenarios where additional energy sources (such as wind or fuel cells) and control objectives (such as heat pumps or air conditioners) are added[30–32]. Second, we will continue tuning these algorithms' hyperparameters to improve their generalization ability to deploy them in other buildings easily[33].

Reference

[1] Niu Z, Wu J, Liu X, Huang L, Nielsen P S. Understanding energy demand behaviors through spatio-temporal smart meter data analysis[J]. Energy, 2021, 226: 120493.

[2] Bogdanov D, Ram M, Aghahosseini A, Gulagi A, Oyewo A S, Child M, Caldera U, Sadovskaia K, Farfan J, De Souza Noel Simas Barbosa L, Fasihi M, Khalili S, Traber T, Breyer C. Low-cost renewable electricity as the key driver of the global energy transition towards sustainability[J]. Energy, 2021, 227: 120467.

[3] Li Y, Gao W, Ruan Y. Performance investigation of grid-connected residential PV-battery system focusing on enhancing self-consumption and peak shaving in Kyushu, Japan[J]. Renewable Energy, 2018, 127: 514–523.

[4] Komiyama R, Fujii Y. Assessment of post-Fukushima renewable energy policy in Japan's nation-wide power grid[J]. Energy Policy, 2017, 101: 594–611.

[5] Su Y, Zhou Y, Tan M. An interval optimization strategy of household multi-energy system considering tolerance degree and integrated demand response[J]. Applied Energy, 2020, 260: 114144.

[6] Li Y, Gao W, Zhang X, Ruan Y, Ushifusa Y, Hiroatsu F. Techno-economic performance analysis of zero energy house applications with home energy management system in Japan[J]. Energy and Buildings, 2020, 214: 109862.

[7] Zhao X, Gao W, Qian F, Ge J. Electricity cost comparison of dynamic pricing model based on load forecasting in home energy management system[J]. Energy, 2021, 229: 120538.

[8] Ullah Z, Elkadeem M R, Kotb K M, Taha I B M, Wang S. Multi-criteria decision-making model for optimal planning of on/off grid hybrid solar, wind, hydro, biomass clean electricity supply[J]. Renewable Energy, 2021, 179: 885–910.

[9] McIlwaine N, Foley A M, Morrow D J, Al Kez D, Zhang C, Lu X, Best R J. A state-of-the-art techno-economic review of distributed and embedded energy storage for energy systems[J]. Energy, 2021, 229: 120461.

[10] Comodi G, Giantomassi A, Severini M, Squartini S, Ferracuti F, Fonti A, Nardi Cesarini D, Morodo M, Polonara F. Multi-apartment residential microgrid with electrical and thermal storage devices: Experimental analysis and simulation of energy management strategies[J]. Applied Energy, 2015, 137: 854–866.

[11] Zhang H, Wu Q, Chen J, Lu L, Zhang J, Zhang S. Multiple stage stochastic planning of integrated electricity and gas system based on distributed approximate dynamic programming[J]. Energy, 2023, 270: 126892.

[12] Cardoso G, Stadler M, Siddiqui A, C. Marnay, DeForest N, Barbosa-Póvoa A, Ferrão P. Microgrid reliability modeling and battery scheduling using stochastic linear programming[J]. Electric Power Systems Research, 2013, 103: 61–69.

[13] Venkatasatish R, Dhanamjayulu C. Reinforcement learning based energy management

systems and hydrogen refuelling stations for fuel cell electric vehicles: An overview[J]. International Journal of Hydrogen Energy, 2022, 47(64): 27646–27670.

[14] Liu X, Ren M, Yang Z, Yan G, Guo Y, Cheng L, Wu C. A multi-step predictive deep reinforcement learning algorithm for HVAC control systems in smart buildings[J]. Energy, 2022, 259: 124857.

[15] Yang D, Wang L, Yu K, Liang J. A reinforcement learning-based energy management strategy for fuel cell hybrid vehicle considering real-time velocity prediction[J]. Energy Conversion and Management, 2022, 274: 116453.

[16] Zhuang D, Gan V J L, Duygu Tekler Z, Chong A, Tian S, Shi X. Data-driven predictive control for smart HVAC system in IoT-integrated buildings with time-series forecasting and reinforcement learning[J]. Applied Energy, 2023, 338: 120936.

[17] Liu F, Liu Q, Tao Q, Huang Y, Li D, Sidorov D. Deep reinforcement learning based energy storage management strategy considering prediction intervals of wind power[J]. International Journal of Electrical Power & Energy Systems, 2023, 145: 108608.

[18] Wang R. Reinforcement Learning: An Introduction[C]//2006 International Conference on Artificial Intelligence: 50 Years' Achievements, Future Directions and Social Impacts.

[19] Sutton R S, Precup D, Singh S. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning[J]. Artificial Intelligence, 1999, 112(1): 181–211.

[20] Sutton R, Barto A. Reinforcement Learning: An Introduction, Adaptive Computation and Machine Learning Series[J]. 1998. ,1998.

[21] Lehna M, Hoppmann B, Heinrich R, Scholz C. A Reinforcement Learning Approach for the Continuous Electricity Market of Germany: Trading from the Perspective of a Wind Park Operator[J]. 2021. ,2021.

[22] Lillicrap T P, Hunt J J, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D P. Continuous control with deep reinforcement learning[P]. 2020.

[23] Duryea E, Ganger M, Wei H. Deep Reinforcement Learning with Double Q-learning[J]. 2016. ,2016.

[24] Fujimoto S, Van Hoof H, Meger D. Addressing function approximation error in actor-critic methods[J]. Proceedings of the 35th International Conference on Machine Learning, Ser. Proceedings of Machine Learning Research, 2018, 80: 1587–1596.

[25] Haarnoja T, Zhou A, Abbeel P, Levine S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor[C]. DY J, KRAUSE A. //Proceedings of the 35th International Conference on Machine Learning. PMLR,2018: 1861–1870.

[26] Wang Z, Hong T, Piette M A. Predicting plug loads with occupant count data through a deep learning approach[J]. Energy, 2019, 181: 29–42.

[27] Grzes ´ M., Kudenko D. Plan-based reward shaping for reinforcement learning[Z]

[28] Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, Tang J, Zaremba W. OpenAI Gym[A]. arXiv,2016[2022-12-07].

[29] Raffin A, Hill A, Gleave A, Kanervisto A, Ernestus M, Dormann N. Stable-Baselines3: Reliable Reinforcement Learning Implementations[J]. Journal of Machine Learning Research, 2021, 22(268): 1–8.

[30] Gao Y, Matsunami Y, Miyata S, Akashi Y. Multi-agent reinforcement learning dealing with hybrid action spaces: A case study for off-grid oriented renewable building energy system[J]. Applied Energy, 2022, 326: 120021.

[31] Zhao L, Yang T, Li W, Zomaya A Y. Deep reinforcement learning-based joint load scheduling for household multi-energy system[J]. Applied Energy, 2022, 324: 119346.

[32] Harrold D J B, Cao J, Fan Z. Renewable energy integration and microgrid energy trading using multi-agent deep reinforcement learning[J]. Applied Energy, 2022, 318: 119151.

[33] Biemann M, Scheller F, Liu X, Huang L. Experimental evaluation of model-free reinforcement learning algorithms for continuous HVAC control[J]. Applied Energy, 2021, 298: 117164.

Chapter 7

CONCLUSION AND OUTLOOK

CHAPTER SEVEN: CONCLUSION AND OUTLOOK

7.1 Conclusion

Renewable energy has developed steadily in recent years in the context of energy shortages and safe supply requirements. The power sector, in particular, plays a crucial role in energy conservation and emission reduction. Renewable energy development can reduce dependence on fossil fuels and improve energy self-sufficiency rates. Since over 40% of total energy consumption comes from buildings, increasing the self-sufficiency rate of renewable energy in buildings is critical. While Japan's implementation of the feed-in tariff in 2011 led to explosive growth in household renewable energy equipment, the trend slowed as the feed-in tariff price decreased. Therefore, it is urgent to reduce further the cost of running household renewable energy equipment. This research focuses on applying machine learning in optimizing building energy system operations further to reduce the operation cost of building energy systems and increase the self-sufficiency rate of renewable energy.

The main works and results can be summarized as follows:

In Chapter 1, INTRODUCTION AND PURPOSE OF THE RESEARCH. Chapter 1 introduces the background of energy research, including the current situation and bottlenecks of comprehensive energy development, as well as the importance of developing variable renewable energy sources. Additionally, it presents renewable energy's development and current state globally and in Japan. The chapter also highlights recent advances in energy prediction, reinforcement learning control, and related research demonstrating how machine learning technology can address energy security and renewable energy deployment issues in building energy systems. Finally, this chapter outlines the paper's research purpose and logical framework to help reviewers better understand its content.

In Chapter 2, METHODOLOGY. Chapter 2 focuses on the key concepts and methods used in the study, which include machine learning, deep learning, deep reinforcement learning, and energy storage systems. Specifically, the chapter summarizes the fundamental theories and methodologies of deep learning and deep reinforcement learning, which form the foundation of the algorithms utilized in the subsequent research.

In Chapter 3, MATERIALS AND DATA PREPROCESSING. Chapter 3 provides an in-depth analysis of the data resources and this study's preprocessing steps. The measured energy system data from Kitakyushu Science Research Park and Jono Zero Carbon Smart Community were utilized. This section details the system under consideration, the methodology employed for data preprocessing, potential data patterns, and the creation of the training and test sets utilized in the subsequent experiments.

In Chapter 4, POTENTIAL ANALYSIS OF THE ATTENTION-BASED LSTM MODEL IN BUILDING ENERGY SYSTEM. Chapter 4 aimed to evaluate the potential of using an attentional-based LSTM network (A-LSTM) to predict HVAC energy consumption in practical applications. To assess the potential applicability of the A-LSTM model in practical scenarios, the training and testing datasets used in the experiments consist of actual energy consumption data collected from Kitakyushu Science Research Park in Japan. Five baseline models (A-LSTM, LSTM, RNN, DNN, and SVR) were developed, and the Tree-structured Parzen Estimators (TPE) algorithm was introduced to optimize the model's super parameters. The subsequent application of the models on the target database resulted in a comprehensive analysis of the results from multiple perspectives. The results indicate that the A-LSTM model achieved the highest prediction accuracy, surpassing the LSTM model with a 3.06% reduction in overall RMSE, a 6.54% decrease in MSE, and a 0.43% increase in R^2 value. Furthermore, the A-LSTM model performed exceptionally well when the length of the training set was between 4 and 6 years. However, the model's prediction accuracy sharply decreased when the size of the training set was reduced to 2 years, indicating its limitations in predicting small sample data.

In Chapter 5, OPERATIONAL OPTIMIZATION FOR BUILDING ENERGY SYSTEMS USING VALUE-BASED REINFORCEMENT LEARNING. Chapter 5 presented the proposed model-based deep reinforcement learning algorithm called Model-based Double-Dueling Deep Q-Networks (MB-D3QN). This algorithm was used to optimize the cost-effective operation of a residential house equipped with a grid-connected PV-battery system in Japan. Results compared and analyzed the performance of Q-learning, DQN, and D3QN agents in optimizing the scheduling strategy of the residential PV-battery system based on real-world monitored data and real-time electricity price. The experimental results proved the effectiveness of the reward function design, and both DQN and D3QN algorithms can reduce energy costs. The case analysis based on the measured data also proves that the MB-D3QN algorithm provides a more efficient scheduling strategy. Compared to the baseline model, it reduces the annual electricity cost by 11.27%. According to the analysis of cost-effectiveness and influencing factors, it could be concluded that the optimization effect of the MB-D3QN method was mainly affected by the difference between the average PV generation and average load and then by the average RTP. The analysis of the Soc control effect proves that MB-D3QN can intelligently judge the future load and electricity price peak and take reasonable charge and discharge action. The comparison between the model-based D3QN method and the model-free D3QN method shows that the model-based approach proposed in this study can significantly improve sample utilization and effectively learn empirical knowledge from limited small sample data.

In Chapter 6, OPERATIONAL OPTIMIZATION FOR BUILDING ENERGY SYSTEMS USING ACTOR-CRITIC BASED REINFORCEMENT LEARNING CONSIDERING REAL-TIME ENERGY PREDICTION. Chapter 6 proposed a model-based RL control method considering real-time prediction values for operation optimization of the residential PV-battery system. The optimization goals aim at reducing the energy cost of the microgrid and ensuring that the PV self-consumption ratio is not lower than the baseline model. To achieve this goal, this study designed a new multi-objective optimization reward function, and experimental results proved the effectiveness of the designed reward function. One of the key steps in this study was to develop and evaluate nine different prediction models with varying structures to predict power demand, real-time electricity price, and photovoltaic power generation. The optimal prediction model was selected for each variable through a comparative evaluation process. Subsequently, the predicted value from the selected models was incorporated into the observed state variable of the RL models for the next time step. The experimental results showed that the above four algorithms could achieve the optimization objective by using the designed reward function in this study. The TD3 algorithm had the best performance in each season. It could reduce the annual energy costs by 17.82% and increase the PV self-consumption ratio by 0.86% compared with the baseline model. In addition, the improved method proposed in this chapter is superior to the models proposed in Chapter 5 in terms of cost optimization and PV self-consumption ratio, which indicates that the solution proposed in this chapter is a better approach for this scenario.

In Chapter 7, CONCLUSION AND OUTLOOK. A summary of each Chapter is concluded.

7.2 Outlook

The main goal of this paper is to optimize building energy systems using the latest machine learning technology. However, the study's limitations are mainly evident in the application scenarios. Specifically, the experiment only focused on optimizing the operation of energy storage equipment. In contrast, buildings typically have other energy equipment, such as wind power generation, heat pumps, and fuel cells. This limitation indicates that there is still a long way to go before our research can be practically applied. Therefore, future research will first focus on designing and optimizing reward functions in scenarios where additional energy sources (such as wind or fuel cells) and control objectives (such as heat pumps or air conditioners) will be added[42–44]. Second, we will continue tuning these algorithms' hyperparameters to improve their generalization ability to deploy them in other buildings [45] easily. Third, We will deploy the solution proposed in this research in real-world buildings, where the reinforcement learning agent will be deployed on cloud computing servers. We aim to further refine and improve this research through practical applications.

Reference

[1] Gao Y, Matsunami Y, Miyata S, Akashi Y. Multi-agent reinforcement learning dealing with hybrid action spaces: A case study for off-grid oriented renewable building energy system[J]. Applied Energy, 2022, 326: 120021.

[2] Zhao L, Yang T, Li W, Zomaya A Y. Deep reinforcement learning-based joint load scheduling for household multi-energy system[J]. Applied Energy, 2022, 324: 119346.

[3] Harrold D J B, Cao J, Fan Z. Renewable energy integration and microgrid energy trading using multi-agent deep reinforcement learning[J]. Applied Energy, 2022, 318: 119151.

[4] Biemann M, Scheller F, Liu X, Huang L. Experimental evaluation of model-free reinforcement learning algorithms for continuous HVAC control[J]. Applied Energy, 2021, 298: 117164.