DEEP REINFORCEMENT LEARNING FOR ELECTRICITY ENERGY TRADING

Miss Manassakan Sanayha

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy in Computer Engineering
Department of Computer Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2022

การเรียนรู้แบบเสริมกำลังเชิงลึกสำหรับการซื้อขายพลังงานไฟฟ้า

น.ส.มนัสกานต์ เสน่หา

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2565

Thesis Title	DEEP REINFORCEMENT LEARNING FOR ELECTRICITY ENERGY TRADING
By	Miss Manassakan Sanayha
Field of Study	Computer Engineering
Thesis Advisor	Associate Professor PEERAPON VATEEKUL, Ph.D.

Accepted by the FACULTY OF ENGINEERING, Chulalongkorn University in Partial Fulfillment of the Requirement for the Doctor of Philosophy

..... Dean of the FACULTY OF ENGINEERING

(Professor SUPOT TEACHAVORASINSKUN, D.Eng.)

DISSERTATION COMMITTEE

..... Chairman

(Associate Professor Manisa Pipattanasomporn, Ph.D.)

..... Thesis Advisor

(Associate Professor PEERAPON VATEEKUL, Ph.D.)

..... Examiner

(Assistant Professor NATTEE NIPARNAN, Ph.D.)

..... Examiner

(Ekapol Chuangsuwanich, Ph.D.)

..... Examiner

(Associate Professor NAEBBOON HOONCHAREON, Ph.D.)

..... Examiner

(Professor BOONSERM KIJSIRIKUL, Ph.D.)

..... External Examiner

(Surat Tanterdtid, D.Eng.)

มนัสกานต์ เสน่หา : การเรียนรู้แบบเสริมกำลังเชิงลึกสำหรับการซื้อขายพลังงานไฟฟ้า. (DEEP REINFORCEMENT LEARNING FOR ELECTRICITY ENERGY TRADING) อ.ที่ปรึกษาหลัก : รศ. ดร.พีรพล เวทีกูล

การยกเลิกกฎระเบียบและการเปิดเสรีของตลาดพลังงานในทศวรรษ 1990 ได้กระตุ้นให้มีการซื้อขายไฟฟ้าในระยะสั้น ทำให้ตลาดพลังงานสามารถผลิตผลผลิตสุทธิได้ในช่วงระยะเวลาหนึ่ง อันเป็นผลมาจากระบบกระจายนี้ อุตสาหกรรมพลังงานต้องการระบบที่ให้ทันสมัยอย่างเร่งด่วนเพื่อจัดการกับปัญหาต่างๆ ได้แก่ สภาพอากาศในปัจจุบัน ทรัพยากรหมุนเวียน และกรอบการทำงานด้านพลังงาน ในวิทยานิพนธ์นี้ได้เสนอวิธีการเรียนรู้การเสริมแรงเชิงลึกสำหรับการซื้อขายพลังงานทั้งแบบค้าส่งและระดับท้องถิ่น สำหรับปัญการในโลกแห่งความเป็นจริง เพื่อการแลกเปลี่ยนพลังงานให้เหมาะสมและคุ้มค่าที่สุด โดยเสนออัลกอริทึม MB-A3C สำหรับการเสนอราคาพลังงานล่วงหน้าเพื่อลดต้นทุนของ ผู้ผลิตไฟฟ้าจากพลังงานลม โดยได้แสดงให้เห็นว่าแบบจำลองสามารถสร้างกลยุทธ์ที่ทำให้ต้นทุนเฉลี่ยต่อวันลดลงมากกว่า 15% ในเดนมาร์กและสวีเดน (ชุดข้อมูล Nord Pool) และได้ขยายแบบจำลองเป็น MB-A3C3 เพื่อทดลองในชุดข้อมูลขนาดใหญ่ระหว่างปี 2555-2556 จำนวน 300 ครัวเรือนในซิดนีย์ ประเทศออสเตรเลีย เมื่อการซื้อขายกันเองระหว่างบ้านเพิ่มขึ้นและการซื้อขายภายนอก (การซื้อขายไปยังกริด) ลดลง ทำให้ MB-A3C3 ช่วยลดค่าไฟลง 17% อย่างมีนัยสำคัญ ถือเป็นการปิดช่องว่างระหว่างปัญหาในโลกแห่งความเป็นจริงและปัญหาทางทฤษฎีในด้านการช่วยลดต้นทุนการผลิตพลังงานลมและค่าไฟฟ้าได้อย่างมีประสิทธิภาพ

สาขาวิชา	วิศวกรรมคอมพิวเตอร์	ลายมือชื่อนิสิต .....
ปีการศึกษา	2565	ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

6273016521 : MAJOR COMPUTER ENGINEERING

KEYWORD: reinforcement learning, Deep learning, energy bidding, peer-to-peer energy trading

Manassakan Sanayha : DEEP REINFORCEMENT LEARNING FOR ELECTRICITY ENERGY TRADING. Advisor: Assoc. Prof. PEERAPON VATEEKUL, Ph.D.

The deregulation and liberalization of the energy market in the 1990s prompted short-term electricity trading, allowing energy markets to produce net output over a range of time periods as a result of this decentralized system, most commonly minutes to days ahead of time. The energy industry urgently requires a system that has undergone substantial modernization in place to handle a variety of issues, including the current climate, renewable resources, and the energy framework. In this dissertation, we investigate a deep reinforcement learning framework for both wholesale and local energy trading, which probes the challenge of RL to optimize the real-world problem in the energy exchange. First, we introduce the MB-A3C algorithm for day-ahead energy bidding to reduce WPP's costs. Also, we have illustrated that our model can generate a strategy that obtains a more than 15% reduction in average cost per day in Denmark and Sweden (Nord Pool dataset). Second, the MB-A3C3 approach is carried out and conducted on a large-scale, real-world, hourly 2012–2013 dataset of 300 households in Sydney, Australia. When internal trade (trading among houses) increased and external trade (trading to the grid) decreased, our multiple agent RL (MB-A3C3) significantly lowered energy bills by 17%. In closing the gap between real-world and theoretical problems, the algorithms herein aid in reducing wind power production costs and customers' electricity bills.

Field of Study: Computer Engineering Student's Signature .....

Academic Year: 2022 Advisor's Signature .....

ACKNOWLEDGEMENTS

I would like to thank my esteemed supervisor – Assoc. Prof. Dr. Peerapon Vateekul for his invaluable supervision, support and tutelage during the course of my PhD degree. My gratitude extends to the Faculty of Engineering for the funding opportunity to undertake my studies at the Department of Computer Engineering, Chulalongkorn University. Additionally, I would like to express gratitude to Dr. Surat Tanterdtid for his treasured support which was really influential in shaping my experiment methods and critiquing my results. I also thank Prof. Dr. Boonserm Kijsirikul, Asst. Prof. Dr. Nattee Niparnan, Dr. Ekapol Chuangsuwanich, Assoc. Prof. Dr. Naebboon Hoonchareon, and Assoc. Prof. Dr. Manisa Pipattanasomporn for their mentorship. I would like to thank my friends, lab mates, colleagues and DataMind team – Dr. Teerapong Panboonyuen, Miss Phattharat Songthung, Mr. Dhanaphon Supha-asawaphokhin, and Mr. Wisit Wongchaianukul for a cherished time spent together in the lab, and in social settings. My appreciation also goes out to my family, my comfort zone and close friend for their encouragement and support all through my studies.

Manassakan Sanayha

TABLE OF CONTENTS

LIST OF TABLES

Page

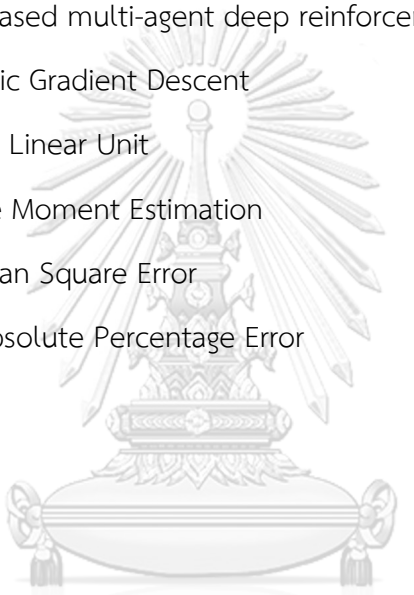
LIST OF FIGURES

Page

LIST OF ABBREVIATIONS

WPP	Wind power producers
VAR	Vector Auto-Regression
ARIMA	Auto-Regressive Integrated Moving Average
ANN	Artificial Neural Network
MLP	Multi-Layer Perceptron
RNN	Recurrent Neural Networks
RL	Reinforcement Learning
DRL	Deep Reinforcement Learning
DDPG	Deep Deterministic Policy Gradient
MDP	Markov Decision Process
A3C	Asynchronous Advantage Actor-Critic
A2C	Advantage Actor-Critic
MBRL	Model-based Reinforcement Learning
MPC	Model Predictive Control
MBPG	Model-Based Policy Gradient
LEMs	Local Energy Markets
SARL	Single-Agent Reinforcement Learning
MARL	Multi-Agent Reinforcement Learning
MA-DRL	Multi-Agent Deep Reinforcement Learning
DER	Distributed energy resources
Dec-POMDP	Decentralized partially observable Markov decision process
CNNs	Convolutional Neural Networks
LSTM	Long Short-Term Memory
DPPO	Distributed Proximal Policy Optimization
MADDPG	Multi-Agent Deep Deterministic Policy Gradient

A3C3	Asynchronous Advantage Actor-Critic with Communication
Conv-A3C	Convolution A3C
A3C3-Conv1D	Convolutional 1-dimensional A3C3
ToU	Time-of-use
FiT	Feed-in tariff
DTW	Dynamic time warping
MB-MADRL	Model-based multi-agent deep reinforcement learning
SGD	Stochastic Gradient Descent
ReLU	Rectified Linear Unit
Adam	Adaptive Moment Estimation
RMSE	Root Mean Square Error
MAPE	Mean Absolute Percentage Error



CHAPTER I

INTRODUCTION

When linked to the electric utility's lower-voltage distribution lines, distributed generation generates energy from natural resources to assist the delivery of clean, reliable power to more consumers and decreases electricity losses along transmission and distribution lines. Energy-related commodity markets trade net produced output over a range of time periods as a result of this decentralized system, most commonly minutes to days ahead of time. Furthermore, the deregulation and liberalization of the energy market in the 1990s prompted short-term electricity trading. The energy industry urgently requires a system that has undergone substantial modernization in place to handle with a variety of issues, including the current climate, renewable resources, and the energy framework.

Wind power is having significant impacts on the timing and location of electricity prices in the wholesale electricity market, competing generators offer their electricity output to retailers. The wind power producers (WPP) can be penalized if the plan deviates from the actual wind power that they can produce. Thus, it is difficult to manually determine and bid energy, which impacts WPP's profit considerably. Algorithmic forecasting and bidding techniques are more suitable and can accomplish this task automatically.

For electricity forecasting, there are some classical statistical and machine learning methods applied to the dataset. The statistical methods, such as Vector Auto-Regression (VAR) and Auto-Regressive Integrated Moving Average (ARIMA), are applied in [1-3] for demand forecasting, but such methods have the limitation when applied in a large and complex dataset. Apart from power forecasting, various statistical methods also have been applied to electricity price forecasting with varying degrees of success [4-8]. Then, an Artificial Neural Network (ANN)-based application is proposed

to forecast a day-ahead electricity price [9-13]. [14] forecast load with Multi-Layer Perceptron (MLP) compared with a linear model. Recently, Recurrent Neural Networks (RNN) with attention mechanism was applied to forecast short-term solar irradiance [15] and to predict short-term wind power [16].

Interestingly for energy bidding, it is crucial for WPP to have an automatic bidding strategy to maximize the profit. Deep Reinforcement Learning (DRL) is proposed as a bidding algorithm in [17-19] for profit optimization. Furthermore, a Deep Deterministic Policy Gradient (DDPG) following Markov Decision Process (MDP) with ANN-based model is proposed by [20]. Intending to leverage the capability of the algorithm, [21-24] proposed the DRL approach to maximize profit. The Asynchronous Advantage Actor-Critic (A3C), which agents work asynchronously [25], is introduced to increase the Wind Power Producer (WPP)'s profit [26] and economical dispatching [27]. Advantage Actor-Critic (A2C), the synchronous version of A3C, can also reduce the cost in power dispatch optimization [28].

The Model-based Reinforcement (MBRL) algorithm has shown more promising results in various domains since it includes Model Predictive Control (MPC). The Dyna architecture integrates learning, planning, and reactive execution [29]. The Model-Based Policy Gradient (MBPG) applies MDP models to compute the policy gradient in closed form [30]. The modernized MBRL algorithms conducted on the MuJoCo benchmark are proposed in to optimize reward function. Interestingly, MBRL has never been applied in the wind energy bidding task [31, 32].

Local Energy Markets (LEMs), on the other hand, enable localized energy exchange among community agents by integrating distributed generation, microgrid, and smart grid into a single electricity market at the distribution side [33, 34]. Since the millennium's turn, academic research has focused on the LEM concept to aid Europe's energy transition such as peer-to-peer energy trading in local energy communities [35]. The integration of distributed generation such as microgrids, smart grids, multi-energy systems, and virtual power plants are developing, as are new ideas called MAS (Multi-

Agent Systems), which is a collection of autonomous, interacting creatures that share a common environment. They are transforming our electrical power infrastructure into a more decentralized, highly efficient energy management system [36]. While DRL is applied for peer-to-peer Energy Trading among Microgrids [37], it also extends to multi-agents for MARL in [22, 38-41]. Interestingly, MARL is enhanced by an A3C framework with an internal communication mechanism to optimize the reward for the convergence of the algorithm [42-46]. Mean-field theory is applied in MARL to simplify the information for each agent [47-52]but hasn't been applied in the energy trading field yet [47-52].

In this dissertation, we proposed a deep reinforcement learning framework for both wholesale and retail energy trading which probes the challenge of RL to optimize the real-world problem in the energy sector. The MB-A3C, which is a Single-Agent Reinforcement Learning (SARL) is proposed for wholesale and extended to Multi-Agent Deep Reinforcement Learning (MA-DRL) for the retails.

First, we introduce the MB-A3C algorithm for day-ahead energy bidding to reduce WPP's cost when participating in the wholesale electricity market. The proposed algorithm can handle both the inaccuracy of wind energy forecasting and the dynamic activation of regulation prices. Intensively, experiments were conducted based on the Nord Pool datasets. Our model was compared to four DRL algorithms: Conv-A3C, DDPO, DDPG, and MBPG with five scenarios of wind power production. Also, we have illustrated that our model can generate a strategy that obtains an average cost per day compared to the best scenario generated with an assumption of knowing actual information.

Second, the extended version of SARL: MA-DRL is proposed to trade energy in the local energy market which composes of more than one individual prosumers. Each prosumer, an autonomous agent, determines and submits their bids whether to buy or sell energy in each period with the optimized policy model which is trained and optimized through a reinforcement learning approach. When participating in the trading

market, these agents also have to share their information with other agents to improve the system's performance as MARL concept.

1.1. Objectives

1. To propose a deep reinforcement learning architecture for energy trading paradigms that maximizes agent's revenue during participating in the energy market on two market scales: wholesale and retail energy markets.
2. To evaluate the performance of the proposed deep reinforcement learning for energy trading on two market scales: wholesale and retail energy markets.

1.2. The Scope of Work

1. Evaluate the proposed deep reinforcement learning along with the following.
    a. Experiment on public datasets, which consider as a real-world challenge in RL.
    b. Experiment with the wholesale and retail energy market.
    c. Compare proposed with baseline algorithms.
    d. Propose Single-Agent Reinforcement Learning (SARL) algorithm on the wholesale energy market and Multi-Agent Reinforcement Learning (MARL) algorithm on the retail energy market.
2. Evaluate the proposed deep reinforcement learning on three methods based on many aspects of RL and baseline methods.
    a. Forecasting accuracy evaluation on deep learning approach
    b. Convergence of deep reinforcement learning policy model
    c. Reward optimization

1.3. Research Funding

This research project is supported by Second Century Fund (C2F), Chulalongkorn University.

1.4. Publication

- Sanayha, Manassakan & Vateekul, Peerapon. (2022). Model-based deep reinforcement learning for wind energy bidding. International Journal of Electrical Power & Energy Systems. 136. 107625. 10.1016/j.ijepes.2021.107625.
    - International Journal of Electrical Power & Energy Systems, Elsevier Journal, Q1 with Tier 1, Rank 1
    - Impact Factor = 4.63
    - https://orcid.org/0000-0001-5402-839X
- M. Sanayha and P. Vateekul, "Model-Based Approach on Multi-Agent Deep Reinforcement Learning With Multiple Clusters for Peer-To-Peer Energy Trading," in IEEE Access, vol. 10, pp. 127882-127893, 2022, doi: 10.1109/ACCESS.2022.3224460.
    - IEEE Access Journal, Q1
    - Impact Factor = 3.476

CHAPTER II

BACKGROUND

In this chapter, the background knowledge related to the dissertation is presented. The algorithm, problem formulation, data description, experimental scenario, and evaluation of two scale markets is explained.

Involving in the liberalized energy market, parties must cooperate while also trying to earn a profit. The list of the most significant players in the energy market is depicted in Table 1.

Table 1. The important players in energy market**.**

Player	Function	Function in the unbundled European energy system (long)
Producer	Generates electricity	Generates electricity in a power plant, which could be a nuclear, coal-fired, or STAG plant, an offshore wind park, or a combined heat and power plant (CHP) etc. If the plant can be managed flexibly, the producer may be able to provide ancillary services to the grid operator.
Consumer	Consumes electricity	Consumes electricity to drive industrial operations, domestic appliances, lighting, and heating, among other things.
Prosumer	Consumes and produces electricity	When own output is insufficient, it uses power from the grid, and when own production exceeds own use, it puts electricity on the grid.

Player	Function	Function in the unbundled European energy system (long)
Transmission system operator (TSO)	Transmits electricity to the high-voltage grid	Long-distance transmission of energy produced at large facilities. To reduce line losses, high voltages of up to 400kV are employed. The TSO is ultimately responsible for ensuring that demand and supply are in balance at all times.
Distribution system operator (DSO)	Distributes electricity to the low-voltage grid	Distributes electricity to end customer at voltages ranging from 400V to 70kV.
Energy supplier	Supply the electricity to households and small businesses	The DSO no longer sells energy to the end-user following the unbundling. Customers have the option of selecting a preferred provider (depending on the tariffs and services offered).
Balancing responsible party (BRP)	Maintains a balance of electricity injection and intake at its access point.	The BRP is required to make balanced nominations to the grid operator based on client consumption and/or production information and forecasting. A BRP is mandatory for every party injecting or taking data from the grid.
Regulator	Ensures that the free market is a level playing field.	Because the transmission and distribution grids are operated as natural monopolies, an independent party is required to ensure that the TSO and DSO do not exploit their market position. They also monitor producers and consumers to ensure that (large) players do not try to manipulate pricing.

Player	Function	Function in the unbundled European energy system (long)
Power Exchange	Platform for exchanging energy	Energy trading takes place on power exchanges, which are anonymous and transparent. Market players submit demand or supply bids using a multilateral trading platform. Every period, the market operator will aggregate all demand and supply bids and clear the market. The products available on the power exchanges are conventional products with sufficient demand to assure liquidity and a reasonable price.
Aggregator	Provides grid operators with balancing services as well as decentralized units with access to energy markets.	In a single portfolio, an aggregator connects together multiple decentralized production and consuming units. He coordinates the operation of the pool of units and provides balancing services to the TSO or DSO, much as large plants have for decades. In addition, all relevant marketplaces exchange the energy of the decentralized units.

In a traditional energy grid, generated electricity from generators is transmitted to consumers via transmission and distribution networks in a centralised power generation system. Power is generated by a few large-scale generation units and distributed to a variety of residential, commercial, and industrial consumers under the centralised generation system. By increasing the level of DER integration on the consumer side, a distributed power generation is forming, in which a large number of small-scale generation units with capacities ranging from a few kilowatts to a few

megawatts are connected to the distribution grid, resulting in bidirectional power flows. The growing utilisation distributed energy resources (DER) has shifted the electrical system from a centralized to a decentralized paradigm. The structure of a centralised and distributed generation system is depicted in Figure 1 which the connection between two scales energy market is indicated. The retail market may consider grid or network energy threshold or energy price from the wholesale market's bidding result.

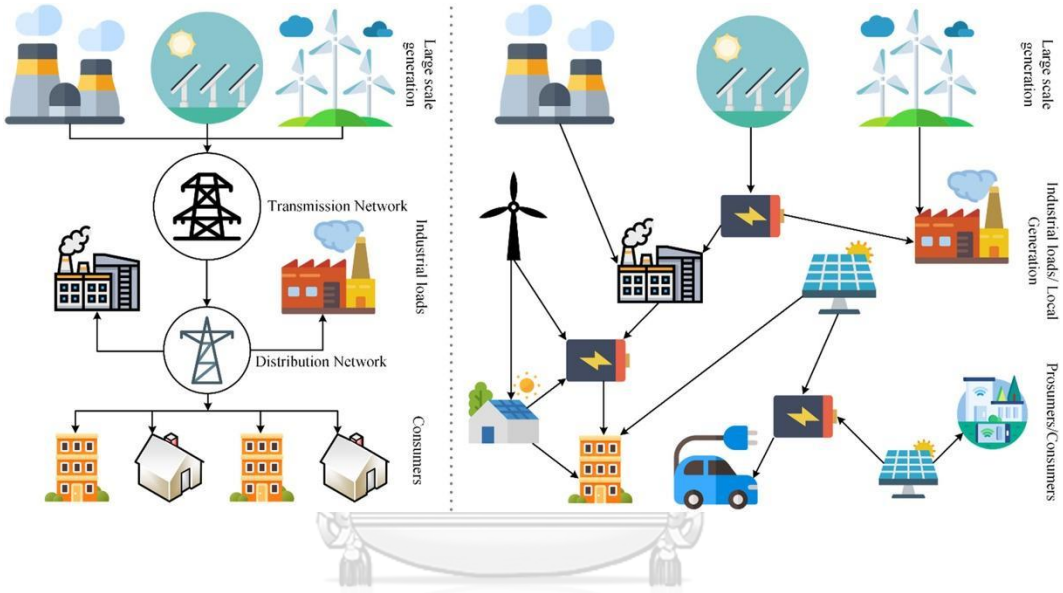


Figure 1. The centralized (left) and decentralized (right) power generation system [53].

2.1. Day-Ahead (Spot) Market

The day-ahead market (spot market) quotes market prices hourly based on purchase and sales bids. Considered as the main trading power arena, the day-ahead market is designed for the 24-hour delivery of electricity in real-time before the day of operation in the wholesale market to avoid price volatility: futures contracts, on the other hand, are exchanged for delivery at a specific time in the future. The spot market for Nord Pool (Elspot)'s process begins with the manufacturers and consumers sending

their tenders to the market 12 to 36 hours (mostly 24) before delivery, indicating the requested quantity and the corresponding amount of electricity.

The marketplaces in the Nordic power market shown in Figure 2 following with the resolution of bidding timestamp. The financial markets provide stability to the business. The day-ahead market (spot market) quotes market prices hourly on the basis of purchase and sales bids. Intraday market creates physical balance. The day-ahead market, the main trading power arena, is designed for the 24-hour delivery of electricity in real-time before the day of operation in the wholesale market to avoid price volatility. The prices and amounts in spot market are based on supply and demand.

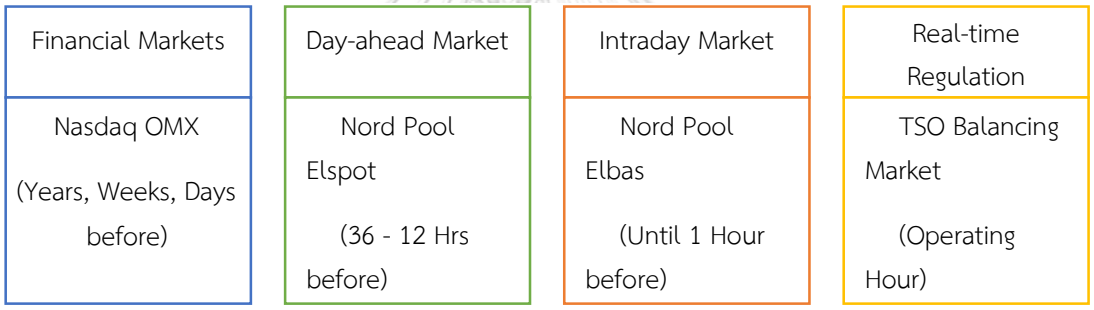
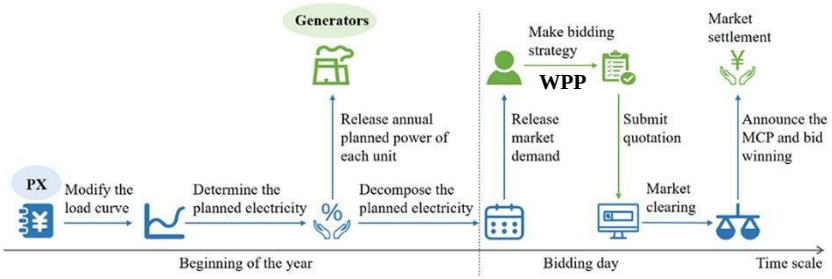
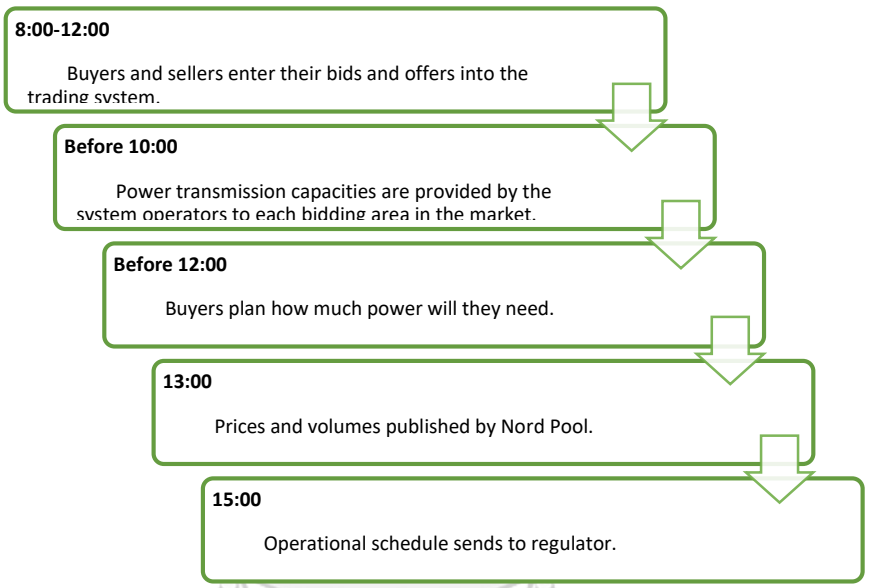


Figure 2. The marketplaces in the Nordic power market**.**

The spot market for Nord Pool (Elspot) is a day-ahead market (Figure 3) in which the power price is decided by supply and demand. Where bidding closes at noon for deliveries from midnight and 24 hours in advance, the outcome of prices and the total sums exchanged are released.



(a) The process of trading electricity on the market modified from [54].



(b) The timeline of day-ahead market.

Figure 3. The timeline of day-ahead market.

The process begins with the manufacturers and consumers sending their tenders to the market 12 to 36 hours before delivery, indicating the quantity of electricity supplied or requested and the corresponding amount. The price which clears the market (balancing supply with demand) for each hour was then calculated by the Nord Pool power exchange.

Principally, all power producers and consumers are able to trade at the exchange, but in fact, only major consumers such as distribution and trading companies

or large industries and generators perform on the market while minor companies form trading cooperatives: the case for wind turbines or engage with larger traders to perform on their behalf. The Nordic countries is traded on the spot market 45% of total electricity production approximately. The remaining share is sold by bilateral, long-term contracts, but the spot price has a significant effect on the rates agreed on in those contracts. The proportion sold on the spot market is as high as 80% in Denmark.

2.1.1. Single-Agent Reinforcement Learning (SARL)

The single agent reinforcement learning framework is based on the model shown in Figure 4, in which an agent interacts with the environment by choosing actions to take and then perceiving the effects of those actions, as well as a new state and a reward signal indicating whether it has achieved some goal (or has been penalized, if the reward is negative). The agent's goal is to maximize some metric over the rewards, such as the sum of all payouts following a series of acts. The framework of Markov decision processes, upon which the solutions for the reinforcement learning issue are formed, can be used to describe this general idea.

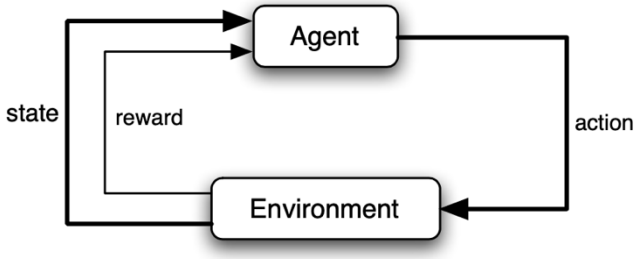


Figure 4. The traditional reinforcement learning method**.**

In the RL algorithm, the Markov Decision Process (MDP) concept is defined that the environment state focuses only on the current state and the results are partially random under the control of the decision-maker.

Markov Decision Processes [55-59] are, in fact, the foundation for much of the research on agent control. They can be defined as a tuple (S, A, T, R) where:

- A is an action set.
- S is a state space.
- $T : S \times A \times S\_ \rightarrow [0,1]$ is a transition function defined as a probability distribution over the states.
- $R : S \times A \times S\_ \rightarrow R$ is a reward function representing the expected value of the next reward, given the current state S and action A and the next state.

These four parts must be defined differently for various problems solved by RL. The notion behind MDPs is that the agent performs some action A on the environment in state s and then waits for the environment's response in the form of state S' and a real number reflecting the immediate reward the agent receives for performing an in S.

2.1.2. Problem Formulation

This phase's experiment considers the situation in which WPP participates in the reserve market. They have to plan some reserve energy to compensate for the deviation from the volume committed when the up-regulation price is activated as described in Figure 5.

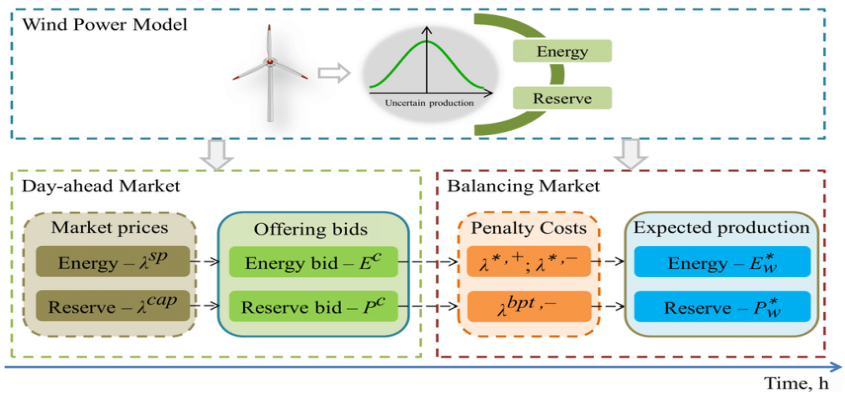


Figure 5. The energy and reserve market concept for wind power participation [60].

The revenue of WPP (π_t) consists of the income from the sale of energy with the regulation cost (π_t^R), and the expense of purchasing and deploying reserve capacity ($C_{R,t}$) as it invests in both the energy and reserve markets. In this paper, we have followed all profit and circumstance setups from [26]. Eq. (1) shows the profit formula, which is then simplified into Eq. (2) to inspect which term needs to be optimized as a reward function.

$$\pi_t = \pi_t^R + c_{R,t}$$

$$= p_{DA,t}E_{a,t} - \eta_{up,t}R_{up,t} + \begin{cases} \Delta p_{up,t}\left(E_{a,t} - E_{bid,t} + R_{up,t}\right) - \mu_{up,t}R_{up,t}, & if\ E_{a,t} \le E_{bid,t} - R_{up,t} \\ -\mu_{up,t}\left(E_{bid,t} - E_{a,t}\right), & if\ E_{bid,t} - R_{up,t} \le E_{a,t} \le E_{bid,t} \\ \Delta p_{down,t}\left(E_{a,t} - E_{bid,t}\right), & if\ E_{a,t} \ge E_{bid,t} \end{cases} \quad (1)$$

$$\pi_t = p_{DA,t}E_{a,t} + c_t^S \quad (2)$$

where $p_{DA,t}$ is the price of wind energy sold, $E_{a,t}$ is the averaged generated energy during interval t, and $E_{bid,t}$ is the committed volume of WPP at time interval t. $\eta_{up,t}R_{up,t}$ is the cost of the purchase reserve with the up-reserve volume of $R_{up,t}$ and the price of the reserve capacity of $\eta_{up,t}$ by the WPP to deal with the situation where the up-regulation price is activated and wind energy generated is less than the committed value. $\Delta p_{up,t} = p_{up,t} - p_{spot,t}$ and $\Delta p_{down,t} = \Delta p_{down,t} - p_{spot,t}$ are the up and down regulation price deviations from the spot price, respectively. $\mu_{up,t}$ is the price of the reserve energy dispatched in real-time.

This dissertation aims to increase the revenues of WPP through strategic bidding on the energy and reserve markets. Within Eq. (1), $p_{DA,t}E_{a,t}$ is the real-time revenue from wind energy produced. Maximizing the benefit thus implies maximizing the

second term c_t^S since the amount committed does not have any impact on the first term. The objective function shall be formulated in Eq. (3).

$$max(F) = max \sum_{t=1}^{T} c_t^S \quad (3)$$

We assume that the WPP's intra-day activities are not included in the spot market bidding model. Therefore, the effect of a single WPP's bidding strategy is not considered. WPP is assumed as a price-taker who has to accept prevailing market prices and lacks market share to influence market prices on its own [26].

In the RL algorithm, the Markov Decision Process (MDP) concept is defined that the environment state focuses only on the current state and the results are partially random under the control of the decision-maker. There are four parts including the state space, the action space, the transition function, and the reward function. They must be defined differently for various problems solved by RL [61]. We defined the state, action, and reward function for wind energy bidding following [26] in Eqs. (4), (5), and (6). The transition function is not defined since it does not necessitate in the time series data.

$$S_t = \left(E_{forecast,t}, E_{bid,t-1}\right) \quad (4)$$

where S_t is the state at time step t. $E_{forecast,t}$ is the wind energy forecasting value at time step t and $E_{bid,t}$ is the previous committed value.

$$A_t = \left(\Delta E_{bid,t}, R_{up,t}\right) \quad (5)$$

where A_t is the action at time step t. $\Delta E_{bid,t} = E_{bid,t} - E_{bid,t-1}$ represents the increments of the committed value at time step t related to the value at time step $(t-1)$.

$$R_t = c_t^S \tag{6}$$

where R_t is the immediate reward, the agent obtains when the action is executed according to S_t.

With the MDP at each step, the agent performs an action according to the current state, obtains an immediate $r(S_t, A_t)$ reward, and then transfers an environment to a new state. The total reward (R_t) for one episode of MDP corresponds to one bidding day with a discounted cumulative reward that can be written in Eq. (7).

$$R_t = r(s_t, a_t) + \gamma r(s_2, a_2) + \cdots + \gamma^{T-t_r}(s_T, a_T) \tag{7}$$

where $\gamma \in [0,1]$ is a discount factor that is introduced to represent environmental uncertainty and T is an episode that corresponds to 24 hours according to the hourly resolution of the datasets [62].

2.2. Peer-to-peer energy trading

The development of decentralized energy resources has revolutionized energy distribution systems in recent decades. Simultaneously, the way energy is generated and consumed is radically changing, and conventional energy consumers are increasingly becoming prosumers. A Local Energy Market, which trading's market structure is shown in Figure 6, is a term used to describe efforts to create a marketplace to coordinate the generation, supply, storage, transportation, and consumption of energy from decentralized energy resources (such as renewable energy generators, storage, and demand-side response providers) within a defined geographic area.

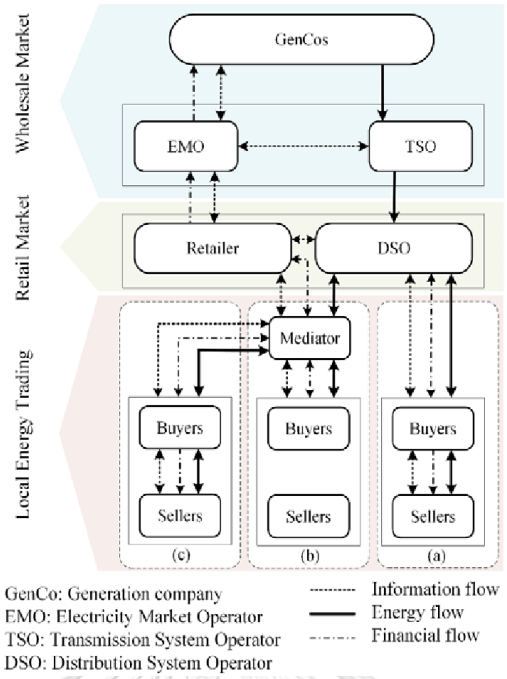


Figure 6. The market structure with local energy trading [63].

The transition to increasingly decentralized, distributed generation assets challenges existing governance, regulatory, and economic institutions to the test. Local Energy Markets (LEMs) are emerging as one solution to the issue of coordinating this increasingly complex system in the UK and elsewhere in Europe. In the UK, LEM designs are still in the early stages of development, with a wide range of design and functionality. The value, costs, and benefits of various market arrangements have yet to be thoroughly tested and analyzed, but continuing improvements to network charges, market settlement, and retail supply will have an impact on LEM implementation and success.

Prosumer power generation is inconsistent and difficult to forecast, as it is severely affected by the solar radiation and temperature (which is constantly changing). There are numerous solutions available to prosumers that have a surplus of electrical energy. The energy can be stored for later use in a storage device, exported to the

electrical grid, or sold to other energy consumers. P2P energy trading refers to direct energy transfer between consumers and prosumers as illustrated in Figure 7.

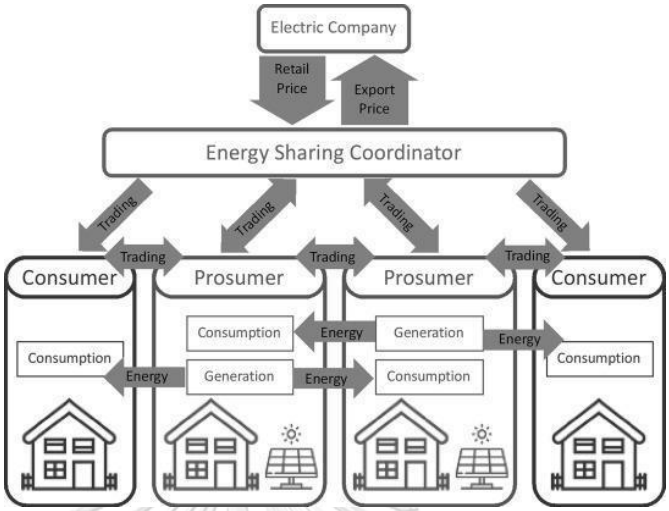


Figure 7. The model of peer-to-peer energy trading [64].

In the traditional market paradigm, producers and consumers interact with merchants according to their net consumption. However, peer-to-peer trading calls for the employment of cutting-edge technology and business structures with market rules that control the P2P paradigm [65]. Producers share their production and consumption in local exchanges prior to trading with retailers at an internal price that is often set between export and retail prices. A subgroup of producers who do not own any local power operations can be thought of as consumers. As a result of the stochastic nature of renewable energy sources like solar photovoltaic (PV) power, producers and consumers must make difficult quota decisions. Since every player's plan is updated in real time, selecting a decent trade strategy might be difficult.

2.2.1. The double auction market mechanism

Many consumers and producers involved in the energy industry are connected through the double auction (DA) market [66, 67]. The auction term in the electrical market is fixed at a predetermined period of time, i.e., an hourly resolution [68]. Following are the steps:

1. Whenever an auction session starts, traders broadcast their instructions to the market. Directives involve an energy quantity and a trade price.
2. Sale orders and purchase orders must correspond. The orders are matched by an algorithm.
3. The auctioneer applies the traditional mid-price strategy to calculate the market clearing price when two orders are matched. The minimum amount between the matched orders determines the transaction quantity.

For matching price determination, the mid-price on the financial markets is the difference between the best price offered by sellers of stocks or commodities and the best price offered by purchasers of stocks or commodities. It can be merely described as the average of the ask price and the bid price. This method was also utilized in P2P energy trading [69]. The auctioneer balances the remaining energy and unmatched orders with the utility provider at grid pricing for time-of-use (ToU) and feed-in tariff (FiT) at the conclusion of the auction. FiT and ToU impose restrictions on all merchants' pricing plans in order to ensure economic gains. The costs of offers and bids always remain within the range of the grid prices. The clearing price is centered on the buy-sell gap [70].

2.2.2. Multi-Agent Reinforcement Learning (MARL)

A framework for examining the sequential decision-making issues that agents (producers and consumers) confront is called MARL [71]. Smart grid applications, such as P2P energy trading in the DA market, can also utilize MARL [72]. The multi-agent architecture is based on the same concept as Figure 8, but there are multiple agents deciding on actions over the environment this time. The major distinction is that each actor is likely to have an impact on the environment, and so actions can have varied outcomes depending on what the other agents are doing. Generally, MARL is the learning problem in a multi-agent system, where numerous agents are learning at the same time.

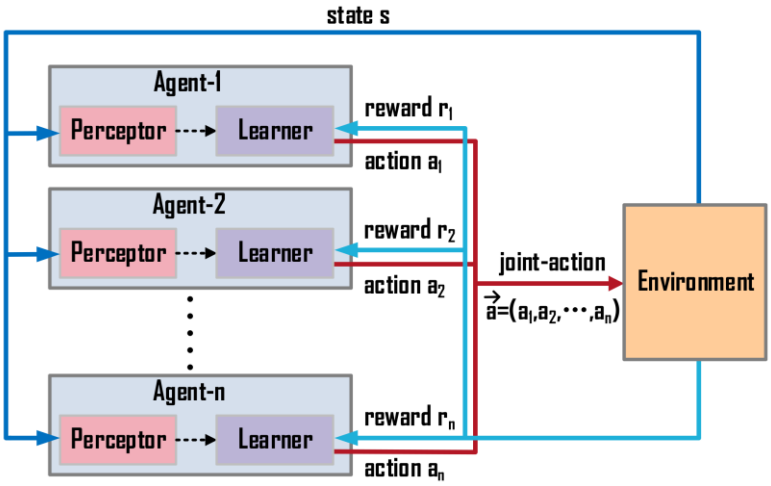


Figure 8. The MARL diagram [40].

The stochastic game is a multi-agent version of the Markov decision process. Game theory, which is meant to solve multi-agent situations and in which the solutions include compromises and collaboration, has been used to simulate such a domain. While it has a different name, it is very similar to an MDP. All actors' states merge into a single state, with distinct rewards corresponding to each potential combined action. Transition functions work similarly to single-agent transition functions, substituting states and actions as needed.

However, in MARL, we have numerous agents. In MARL, the term "state" refers to the combined state of all agents. As a result, state transitions become reliant on the combined activities of all agents, i.e. on how all of them act, rather than just one. To demonstrate the concept of non-stationary transitions, imagine the agent has some coworkers with whom he wants to collaborate. The size of the state and action spaces grows exponentially as the number of agents grows. Due to problems like the curse of dimensionality, this might make learning difficult. The difficulty can get too huge at times, causing convergence to take too long. It makes intuitive sense. The more options our policy has to make for our agents, the longer it will take them to figure out how to make good ones.

2.2.3. Problem Formulation

The MDP principle in RL implies that the state of the environment is primarily concerned with the current state, and that outputs are partially random and under the control of the decision-maker. The state-space, action space, transition function, and reward function are the four components of MDP. For the various problems treated by RL, all states must be defined differently. The above-mentioned DA market clearing procedures are a model for multi-agent decision-making, defined as a decentralized, partially observable Markov decision process (Dec-POMDP) with discrete time steps [54]. N agents include a set of global state S, a collection of private observations O, a collection of action sets A, a collection of reward functions R, and a state transition function T. One auction period (t = 1h) is the time span between two sequential stages. Each agent n selects an action $(a_{n,t})$ based on its policy and private observation $(o_{n,t})$ at time step t. In Eqs. (1), (2), and (3), the state space, action space, and reward function for the trading of solar energy are expressed [36]. The transition function is not defined since it is not needed in the time series data:

$$s_{n,t} = \left[P_{n,t}^{inf}, E_{n,t}^{es}, \lambda_t^b, \lambda_t^s, q_{actual_{n,t-1}}^{da}, q_{forecast_{n,t}}^{da}, \lambda_{actual_{n,t-1}}^{i}, \lambda_{forecast_{n,t}}^{i}\right] \quad (10)$$

where $s_{n,t}$ is the state of agent n at time step t. $P_{n,t}^{inf}$ and $E_{n,t}^{es}$ are the inflexible load information and ES battery energy content at time step t. λ_t^b and λ_t^s are the grid information of ToU and FiT at time step t. $q_{actual_{n,t-1}}^{da}$ and $\lambda_{actual_{n,t-1}}^{i}$ are the previous trading quantity and price at time step $t-1$. $q_{forecast_{n,t}}^{da}$ and $\lambda_{forecast_{n,t}}^{i}$ are the forecast trading quantity and price at time step t.

$$a_{n,t} = \left[a_{n,t}^q, a_{n,t}^p\right] \quad (11)$$

where $a_{n,t}$ is the action of agent n at time step t. $a_{n,t}^q$ and $a_{n,t}^p$ represent the energy and price decision submitted to DA market at time step t.

$$r_{n,t} = -(\lambda_{n,t} q_{n,t}^{da} \Delta t + \lambda_t^b \left[q_{n,t}^{grid}\right]^+ \Delta t + \lambda_t^s \left[q_{n,t}^{grid}\right]^- \Delta t) \quad (12)$$

where $r_{n,t}$ is the immediate reward the agent n at time step t obtains when the action is executed according to $s_{n,t}$.

At step t, agent n receives its reward $r_{n,t}$ in the form of a negative cost of energy bill, resulting from DA market clearing results. The agents who are successfully cleared in the DA market will get the local price $\lambda_{n,t}$ and the cleared quantity $q_{n,t}^{da}$, after which each agent n can calculate its n,t corresponding cost in the DA market; the remaining unmatched quantity $q_{n,t}^{grid}$ will be bought or sold through the utility company at ToU λ_t^b or FiT λ_t^s. The agents' quantity $q_{n,t}^{grid} = q_{n,t}^{da}$ will be immediately exchanged at ToU or FiT if they are unable to be cleared in the DA market.

As market's equation expressed in Eq. (13), households trade their energy to others in competitive cooperative manner to maximize the profit: minimize energy bills.

$$R_t = \sum_{i=1}^{n} -(\lambda_{i,t} q_{i,t}^{da} \Delta t + \lambda_t^b \left[q_{i,t}^{grid}\right]^+ \Delta t + \lambda_t^s \left[q_{i,t}^{grid}\right]^- \Delta t) \quad (13)$$

where R_t is the reward of every agent (community's reward) from 1 to n at time step t obtains when all agents' action is executed according to their state.

2.3 Energy trading in Thailand

Thailand's main stock exchange announced an agreement with the state-owned Electricity Generating Authority of Thailand (EGAT) to develop an energy trading platform as part of a plan to become Southeast Asia's electricity trading centre. The agreement was reached during the Association of Southeast Asian Nations' (ASEAN) energy ministers' session in Bangkok this week. ASEAN has long sought to construct a regional electricity grid to support member members where demand outstrips supply. The Thai Stock Exchange and EGAT will investigate into alternatives for ASEAN members to develop "a wholesale electricity market." As part of a power integration initiative, Thailand, Laos, and Malaysia have already committed to making around 300 megawatts of electricity capacity accessible for trading. Power trade for up to three to

five years ahead would be possible on the planned Thai platform, as well as daily energy trading. Thailand utilizes an advanced single-buyer model, with EGAT serving as both a supplier and a buyer of electricity. As a result, on April 30, 2018, Thailand's Power Development Plan (PDP) was changed to incorporate three important aspects: (1) Demand Response (DR) and Energy Management Systems (EMS); (2) Renewable Energy Forecasting; (3) Micro-Grid and Energy Storage Systems (ESS). Following the PDP 2018, the Alternative Energy Development Plan (AEDP 2018) was updated to keep RE at 30% of total final energy consumption. Governments have consistently encouraged RE from solar by launching incentive programs to entice solar producers to join the program and sell their products through the distribution network. To provide permits, the Energy Regulatory Commission (ERC) created the self-consumption program for residential PV rooftops.

Despite several limitations, some small prosumers who would like to sell their PV production must join in the incentive scheme. Because Thailand's power market is governed by the Enhanced Single Buyer model, small producers are unable to sell their energy to anyone except the authorized utility company. Furthermore, they are unable to sell surplus power generation more than the Power Purchasing Agreement's power capacity (PPA). As a result, if their PV rooftop systems produce more power than the contract capacity, they will forfeit the extra production without compensation. As a result, many rooftop owners choose to install on-grid PV systems rather than participate in the government's incentive program.

2.3.1 Current algorithms

The National Control Center (NCC) requires supported information from a variety of sources for electricity generation planning. Short-term forecasting is done by applying linear regression to historical data. Specific application utilized Artificial Neural Network (ANN) also applied to forecast demand and price in short term for renewable energy project. Nonetheless, data and techniques are critical for leveraging system performance and supporting decision making.

2.3.2. Market rules

The retail energy market is not yet available in Thailand, only sandbox project experimented. The market rule and limitation should be further considered are the supportive policies encouraging decentralization of power systems and better utilization of existing grid infrastructure and local distributed energy generators allowance to sell their electricity at the desired price to consumers willing to pay that price. Moreover, the compatibility of the system platform with the real market: block chain, smart meter, participants' behavior, game theory and constraints, Thailand governors, policy makers e.g., taxation and legal rights are the gaps that need further actions.

2.3.3. Effect of energy trading to main grid

Peer-to-peer (P2P) trading systems-based local electricity markets have evolved as a creative mechanism to sell electricity from prosumer to consumer, to effectively and highly value local flexibility, and to assist grid management. The local market results in total cost reductions for the consumer and provides the framework for creating price plans (such as loss management strategies) that are specific to DSO operations.

Depending on the local environment, these LEMs may use solar, wind, or other sources. LEM-based renewable energy system for off-grid electrification in underdeveloped nations. They also looked at LEM usage in rural areas.

New methods to enhance the overall efficiency of energy management and distribution have been made possible by renewable energy and its trade. For instance, cutting-edge forecasting algorithms can forecast the output of renewable energy sources (solar and wind). The stability of the power distribution system can be achieved by accurately anticipating the output of solar and wind energy. Another illustration is the large-scale integration of renewable energy without causing grid congestion, which is made possible by virtual power line storage systems at both the supply and demand sides. Dynamic line ratings are also now utilized to lessen congestion on electricity lines.

CHAPTER III

RELATED WORKS

In this chapter, the related algorithms related to the dissertation are presented. Deep learning, which has dominant performance on complex data, plays two essential roles in the RL framework: to forecast stage's inputs (Attention-LSTM) and to optimize for the best policy (CNN and LSTM). The SARL algorithms: A3C, DPPO, DDPG, and MBPG, and MARL are also depicted, respectively.

3.1. Deep Learning

Deep Learning is a machine learning discipline that allows machines to learn from experience and comprehend the world in terms of a hierarchy of concepts, with each idea described in terms of its relationship to simpler notions. It has been increasingly valuable in recent years as the amount of data available has grown, and it may be used in a variety of disciplines, including image segmentation, object detection, video classification, speech recognition, reinforcement learning, robotics, and so on. As a result, they've sparked a lot of research interest in recent years, and they've achieved state-of-the-art results in a variety of domains, including sentiment classification. It can be divided current state-of-the-art deep learning algorithms for opinion mining into two groups: CNN and RNN [73].

3.1.1. Convolutional Neural Networks (CNNs)

CNNs are multilayer neural networks that consist of one or more convolutional layers, generally with a subsampling step, followed by one or more fully connected layers. Besides, CNNS, which is also applied to one-dimension data [74], is suitable for multiple time-series inputs (wind power and wind bidding) since it considers all series

with multiple time steps altogether. Generally, the 1D-CNN architecture for time-series data, for example in Figure 9, consists of convolutional, pooling, fully connected layers. Also, the CNN network permits the volume and complexity of time series data for the deep learning concept.

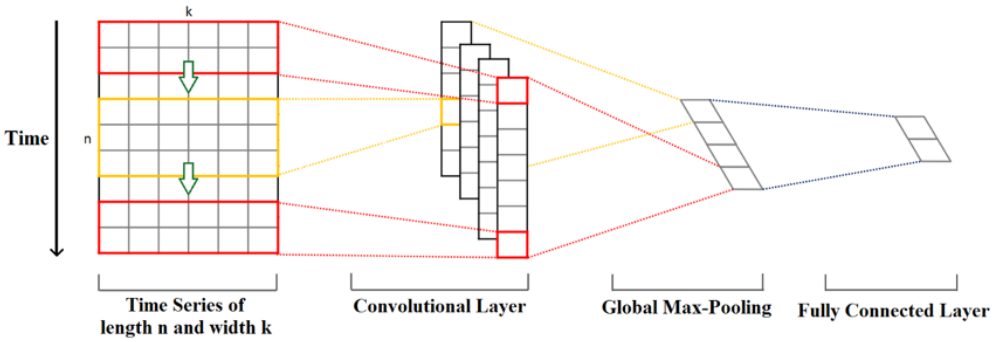


Figure 9. The example of 1D-CNN architecture for time series data **[**10**].**

3.1.2. Recurrent Neural Network (RNN)

Recurrent neural network (RNN) makes connections between nodes to form a directed graph along a sequence, which allows to exhibit dynamic temporal behavior for a time sequence. Unlike feedforward neural networks, RNN can use the internal state or memory to process sequences of inputs, which makes them applicable to tasks such as speech recognition.

A special form of RNN capable of long-term dependencies learning: LSTM. The Attention-LSTM only focuses on the previous hour of the input sequence but the 24 hours context. While the traditional LSTM networks use only the last hidden state as output, the Attention-LSTM (Figure 10) multiplies output hidden states by trainable weights [75]. As a consequence, they can capture more discriminatory task-related features by calculating all inputs together.

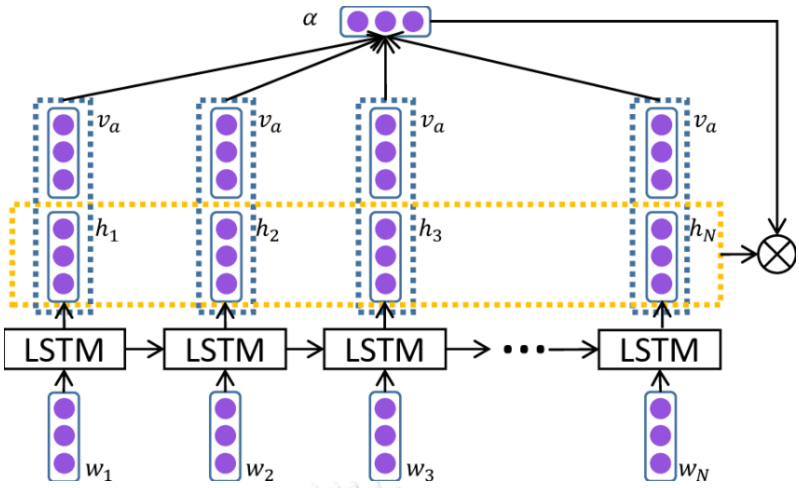


Figure 10. The Attention-LSTM diagram [76].

3.2. Single-Agent Reinforcement Learning (SARL)

As the complexity of the domain increases, it is difficult to control energy flows by using existing technologies based on physical models. Besides, data-driven models, such as RL algorithms have found widespread applications in many sectors [77]. RL algorithm is categorized into two sets: MBRL tries to understand the world and create a model to represent it, while MFRL learns policy or value function directly as shown in Figure 11.

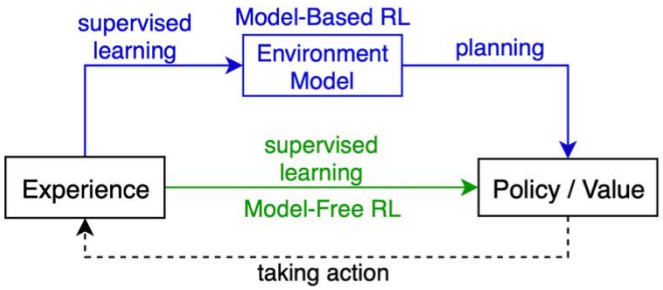


Figure 11. The difference between MBRL and MFRL.

Several methods, such as deep learning, multi-threaded, and supervised model-based, are combined with RL to increase the performance conducted in many areas of research including energy systems [77]. Both MFRL and MBRL algorithms are

introduced to solve issues in energy fields like A3C, DPPO, and DDPG which are MFRL methods, and MBPG that model the environment with policy gradient search.

One of the simplest forms of single-agent reinforcement learning problems is the multi-armed bandit problem, where players try to earn money (maximizing reward) by selecting an arm (an action) to play (interact with the environment). This is an oversimplified model, where the situation before and after each turn is the same. In most scenarios, you may face different conditions and take multiple actions in a row to complete one turn. This introduces the discrimination between states and their transitions [78]. As is common in RL, these probabilities are updated based on feedback received from the environment. While initial studies focused mainly on a single automaton in n-armed bandit settings, RL algorithms using multiple automata were developed to learn policies in MDPs [79].

3.2.1. Asynchronous Advantage Actor-Critic (A3C)

Asynchronous Advantage Actor-Critic (A3C) is essentially asynchronous parallel training, in which several workers in parallel settings update a global value function separately (Figure 12). Exploration of the state space is made more effective and efficient using asynchronous actors.

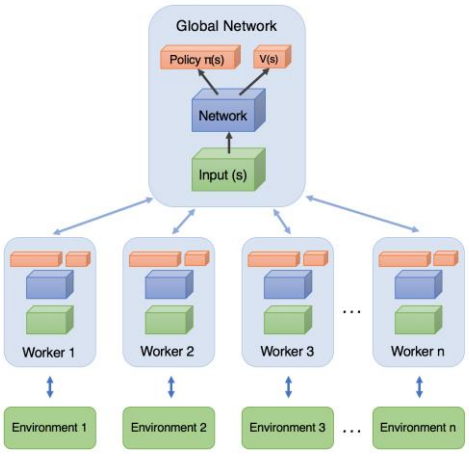


Figure 12. A3C diagram [80].

Being applied in energy strategic bidding to maximize WPP's revenue in [26], A3C maintains the policy $\pi(a_t \mid s_t; \theta)$ and the estimation of the value function $V(s_t; \theta_v)$ [25]. Both of them are updated after every T_{max} action or when a terminal state is reached. The value-based "Critic" method relies on parallel actor-learners and accumulated updates to improve training stability. The A3C algorithm breaks the data correlation by running multiple workers in parallel. The stability of the algorithm during training is significantly improved. Moreover, the memory requirements are reduced compared to the deep Q-learning method.

3.2.2. Distributed Proximal Policy Optimization (DPPO)

Distributed Proximal Policy Optimization (DPPO) [81], which is applied for intelligent economic dispatch in combined heat and power system [24], is essentially Proximal Policy Optimization (PPO) in a distributed setting by using multiple workers and a server parameter. A shared model is updated every time many workers are ready to send updates [82]. In terms of implementation, a central parameter server and a specific worker update the model, while numerous concurrent workers interface with the simulator, according to DPPO (Figure 13).

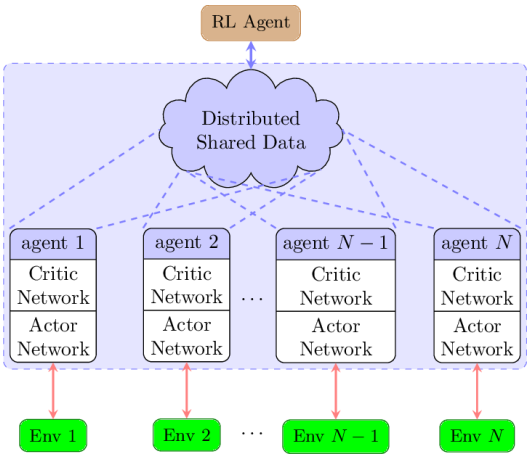


Figure 13. The schematics of DPPO algorithm [83].

The implementation of the Trust Region Policy Optimization (TRPO) is also applied to DPPO to restrict the amount by which any update is allowed to change the policy.

3.2.3. Deep Deterministic Policy Gradient (DDPG)

The model-free approach Deep Deterministic Policy Gradient (DDPG) able to learn competitive policies for most of the tasks by applying low-dimensional observations with the same hyper parameter and network structure (Figure 14). In many cases, DDPG also able to learn good policies directly from pixels, keeping hyper parameters and network structure constant again. DDPG is also proposed to solve the problem of STATCOM-ADC parameter adjusting, which considers the uncertainty of the wind power production [20].

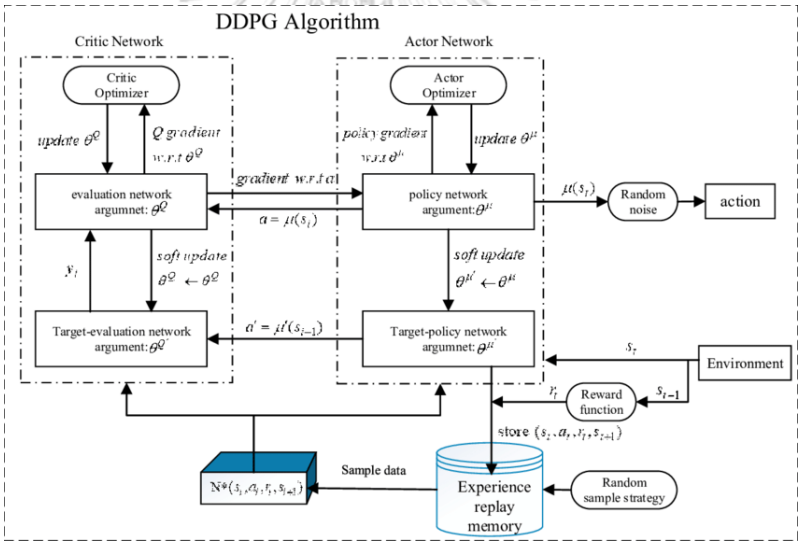


Figure 14. DDPG algorithm structure [84].

3.2.4. Model-Based Policy Gradient (MBPG)

Model-Based Policy Gradient (MBPG) is introduced as MBRL with a gradient formula, exploration, and pruning strategy (Figure 15). The idea behind Dyna-style planning is that it allows us to execute the various model-free updates you've already heard about without having to operate in the real world. For large-scale MDPs, such as resource-restricted scheduling, it is impossible to store the entire MDP in a tabular

form. The overall effectiveness of MBPG depends on whether a sufficiently insightful partial model fits into the available memory. Therefore, MBPG introduces effective exploration which gathers training data in the relevant parts of state space, and heuristic pruning which removes irrelevant parts of the incomplete MDP model.

Table 1. Algorithm MBPG

Initialize:
$\theta_i = 0$, or a random number from -0.1 to 0.1
$i = 0, 1, \ldots, K$
$N^\pi(s) = P_0(s)$,
$V^\pi(s) = 0$,
$\nabla_\theta \pi(s, a; \theta) = 0$

Repeat *until convergence*:
Exploration:
1. Remove unneeded (s, a) pairs from the model $P(s'|s, a)$ and $R(s'|s, a)$
2. Explore using the current stochastic policy $\pi(s, a; \theta)$
3. Update the model.

Gradient Ascent:
3. Compute the expected number of visits to each state $N^\pi(s)$ by, $N^\pi(s) = \sum_{s_p} N^\pi(s_p) \sum_a \pi(s_p, a; \theta) P(s|s_p, a) + P_0(s)$
4. Compute $V^\pi(s)$ by value iteration, $V^\pi(s) = \sum_a \pi(s, a; \theta) \sum_{s'} P(s'|s, a) [R(s'|s, a) + V^\pi(s')]$
5. Compute $\nabla_\theta \pi(s, a; \theta)$ by $\nabla_\theta J^\pi(\theta) = \sum_s N^\pi(s) \sum_a \nabla_\theta \pi(s, a; \theta) \sum_{s'} P(s'|s, a)[R(s'|s, a) + V^\pi(s')]$.
6. Perform a line search in the direction of $\nabla_\theta \pi(s, a; \theta)$. This requires repetition of steps 4, 5, and 6, to update the gradient.

Figure 15. MBPG algorithm [30].

In this dissertation, there are four baselines: three model-free algorithms including A3C, DPPO, DDPG, and one model-based technique, MBPG. In energy bidding, MFRL is applied in such problem more than MBRL which haven't been chosen for solve problem in this area.

3.3. Multi-Agent Reinforcement Learning (MARL)

Several real-world problems, spanning from satellite development to traffic monitoring, are naturally described as cooperative multi-agent systems. These systems

necessitate algorithms capable of learning good policies with autonomous agents based merely on local partial observations of the environment. However, due to partial observability and non-stationarity from an agent's perspective, as well as the structural credit assignment problem and the curse of dimensionality, multi-agent environments are more complex, and attaining coordination in such systems remains a difficult task. Recent research on energy trading applied MADDPG, the single-threaded actor-critic with multiple agents, approach to leverage the capability of each trading system in LEM. The multi-threaded A3C is combined in MARL for the convergence of training for best policy has success but not yet implemented in the energy trading field.

3.3.1. Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [85]

MADDPG, or Multi-agent DDPG, is a multi-agent policy gradient methodology that use decentralized agents to generate a centralized critic based on the observations and behaviors of all agents (Figure 16). It results in learned rules that only use local data (their own observations) to execute. It applies not just to cooperative interactions, but also to competitive or mixed interactions involving both physical and communicative action.

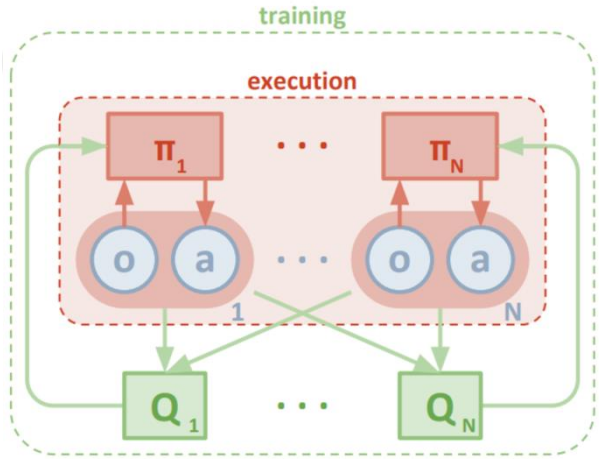


Figure 16. The schematic of the MADDPG algorithm [85].

MADDPG does not necessitate a differentiable model of the dynamics of the environment or any specific structure on the communication mechanism between agents. The critic has access to additional information about the policies of other agents, but the actor just has access to local knowledge. Following training, only local actors are used in the execution phase, acting in a distributed environment as the algorithm is described in Figure 17.

Algorithm 1: Multi-Agent Deep Deterministic Policy Gradient for N agents

for episode $= 1$ to M **do**
  Initialize a random process $\mathcal{N}$ for action exploration
  Receive initial state $\mathbf{x}$
  for $t = 1$ to max-episode-length **do**
    for each agent i, select action $a_i = \boldsymbol{\mu}_{\theta_i}(o_i) + \mathcal{N}_t$ w.r.t. the current policy and exploration
    Execute actions $a = (a_1, \ldots, a_N)$ and observe reward r and new state $\mathbf{x}'$
    Store $(\mathbf{x}, a, r, \mathbf{x}')$ in replay buffer $\mathcal{D}$
    $\mathbf{x} \leftarrow \mathbf{x}'$
    for agent $i = 1$ to N **do**
      Sample a random minibatch of S samples $(\mathbf{x}^j, a^j, r^j, \mathbf{x}'^j)$ from $\mathcal{D}$
      Set $y^j = r_i^j + \gamma\, Q_i^{\boldsymbol{\mu}'}(\mathbf{x}'^j, a'_1, \ldots, a'_N)|_{a'_k = \boldsymbol{\mu}'_k(o_k^j)}$
      Update critic by minimizing the loss $\mathcal{L}(\theta_i) = \frac{1}{S}\sum_j \left(y^j - Q_i^{\boldsymbol{\mu}}(\mathbf{x}^j, a_1^j, \ldots, a_N^j)\right)^2$
      Update actor using the sampled policy gradient:

$$\nabla_{\theta_i} J \approx \frac{1}{S}\sum_j \nabla_{\theta_i} \boldsymbol{\mu}_i(o_i^j) \nabla_{a_i} Q_i^{\boldsymbol{\mu}}(\mathbf{x}^j, a_1^j, \ldots, a_i, \ldots, a_N^j)\big|_{a_i = \boldsymbol{\mu}_i(o_i^j)}$$

    end for
    Update target network parameters for each agent i:

$$\theta_i' \leftarrow \tau\theta_i + (1-\tau)\theta_i'$$

  end for
end for

Figure 17. MADDPG algorithm [85].

3.3.2. Asynchronous Advantage Actor-Critic with Communication (A3C3) [44]

In A3C, a policy-based methodology that changes both the policy and the value-function in the forward view using a combination of n-step returns, the actor-critic architecture [80] and the advantage function [83] are utilized to establish the best policy. It also supports concurrent learning by allowing many actors-learners to adjust the neural network's parameters asynchronously. This strategy, however, restricts the number of only single agent or requires the system to be homogeneous.

Multi-agent learning has significant challenges in attempting to allocate a hidden communication channel. Recent research has frequently merged a customized neural network with reinforcement learning to facilitate communication between agents. [86] present a more scalable technique that not only deals with many agents but also allows collaboration across distinct functional agents and may be used in conjunction with any deep reinforcement learning method. Exploration is a difficult challenge in multi-agent reinforcement learning, especially when the rewards are scarce. A comprehensive technique, which is proposed for efficient exploration that incorporates agents sharing their experience, the Shared Experience Actor-Critic (SEAC) implements experience sharing in an actor-critic architecture by combining the gradients of many agents [45].

A distributed asynchronous actor-critic method, with differentiable communication and a centralized critic in a multi-agent context, is investigated to employ a centralized critic to estimate a value function [44]. The decentralized actors approximate each agent's policy function, and decentralized communication networks allow each agent to exchange important information with its team (Figure 18).

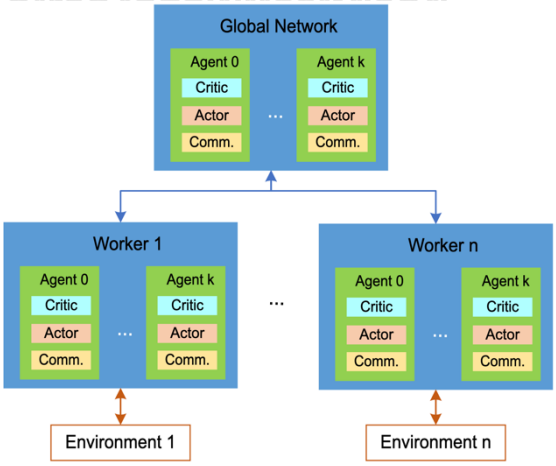


Figure 18. A3C3 architecture with n distinct workers**.** Each worker interacts with its own environment and its own collection of j agents**.** Workers asynchronously update

the global networks and transfer those weights into their local networks when samples are gathered in mini-batches [44].

When more knowledge, such as the global status of the environment, is available, the critic can include it and optimize the actor networks. The actor networks of an agent's teammates optimize its communication network, so that each agent learns to output knowledge that is valuable to the policies of others. A3C3 can handle a high number of agents, noisy communication channels, and can be horizontally scaled to reduce learning time.

A3C3, multi-agent A3C with a communication network in each multi-threaded worker, to determine reward optimization in cooperative-competitive manners of every agent in the system [44]. The actor, the centralized critic, and the communication network are the three major components of A3C3. A3C3 can learn policies that are very successful in achieving shorter distances to their goals than MADDPG through cooperative communication. Even though A3C3 has never been used in P2P energy trading, it has been chosen as our core model since it outperforms MADDPG by maximizing agent rewards.

CHAPTER IV

CONCEPT AND RESEARCH METHODOLOGY

The research's concept is divided into two phases: wind energy bidding in the wholesale electricity market and energy trading in the local electricity market. SARL is applied in the first phase and MARL in the second due to the involvement of agents in each market type. Each phase's data description, experimental settings, and evaluation procedure are detailed in this chapter.

4.1. MB-DRL for Wind Energy Bidding in the Wholesale Electricity Market

In this section, details of our model called "MB-A3C" are explained. There are three main components as illustrated in Figure 19: (i) forecasting model using the Attention-LSTM, (ii) policy model using the Convolution A3C (Conv-A3C), and (iii) MBRL framework.

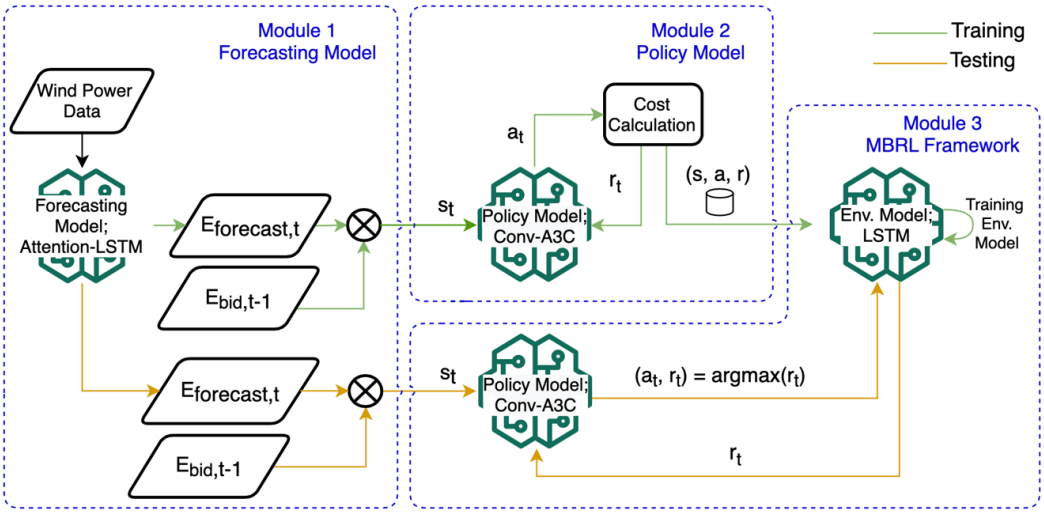


Figure 19. The MB-A3C diagram comprising with forecasting model, policy model and MBRL framework.

First, the forecasting module predicts the wind energy, and then the predicted wind energy is formulated as the current state. Second, the policy model takes the current state to propose a suitable policy resulting as an action for the day-ahead market (amount of bidding and reserve energies). Then, the cost can be projected using all factors. Third, in MBRL, another model is proposed to forecast a future cost since there are other factors (e.g., the market-clearing price: spot price) that do not know yet in the testing phase. This is the main contribution module for MBRL.

4.1.1. Forecasting Model

Here we discuss the first module in Figure 19. The Attention-LSTM for wind energy forecasting (Figure 20) not only focuses on the previous hour of the input sequence but the 24 hours' context. While the traditional LSTM networks use only the last hidden state as output, the Attention- LSTM multiplies output hidden states by trainable weights [75]. As a consequence, they can capture more discriminatory task-related features by calculating all inputs together.

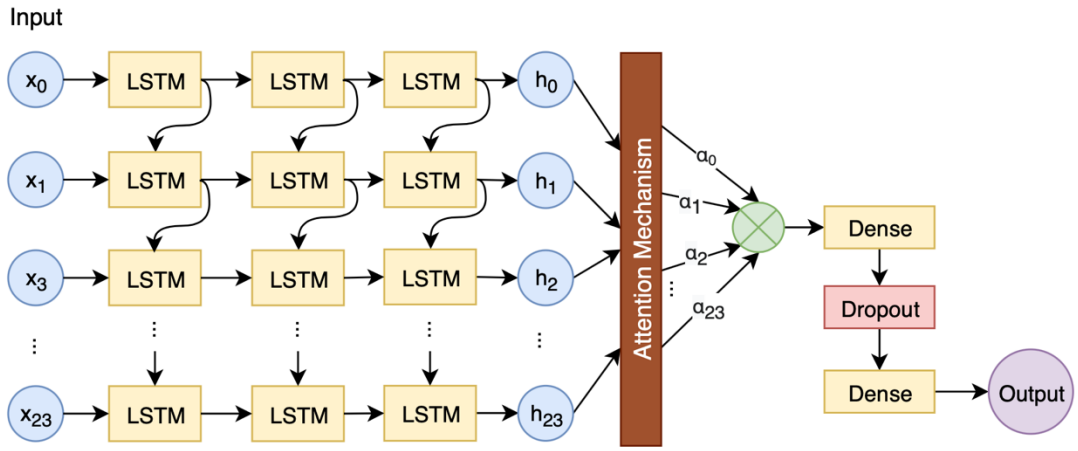


Figure 20. The Attention-LSTM architecture for wind energy forecasting. Denote x_0 to x_{23} as 24 inputs (hours) and a_0 to a_{23} as attention weights.

It is seen that the Attention-LSTM not only focuses on the previous hour of the input sequence but also on the 24-hour context. Unlike traditional LSTM networks,

which use only the most recent hidden state as output, Attention-LSTM networks multiply hidden state output by trainable weights. Consequently, they can capture more discriminatory task-related features by calculating all inputs together.

The baseline is enhanced by the leverage forecasting algorithm for wind energy. Because Attention-LSTM has made numerous contributions in such tasks, we investigated it in our algorithm and found that it outperformed the baseline paper.

For time series data, fundamental and modified attention mechanisms have been examined in several articles. It has been established that time series forecasting activities benefit greatly from attention to gains in general performance as compared to models like RNNs and LSTMs [87-89].

4.1.2. Policy Model

We proposed the convolutional A3C (Conv-A3C) which is our improvement from A3C. The model updates its parameters using experience and reward information to develop an optimal bidding strategy. We defined this optimal bidding strategy as a policy model (Module 2 in Figure 23). A3C performs with the three assumptions below.

1. The term “Asynchronous” means multiple agents work together on the same issue and share information about what they have learned. Thus, the solution is reached more rapidly.
2. An Actor-Critic is two networks delivering two outputs. One is the values for the different actions: a policy. The other calculates the value of the state the agent is currently in and reviews the action by value.
3. The Advantage informs if there is an improvement in a given action compared to the expected average value of that state based on it.

The reward obtained in this method only has a direct impact on the state action pairs at the current time step, which is why the algorithm slowly converges. The A3C algorithm adopts a multi-agent approach with the advantage function in Eq. (7) to speed up the learning process.

$$\sum_{i=0}^{k-1} \gamma^i r_{t+i} + \gamma^k V(s_{t+k} \mid \theta^v) - V(s_t \mid \theta^v) \quad (7)$$

where $V(s_t \mid \theta^v)$ is the state value function of state s_t with parameter θ^v at interval t.

In addition, there is a constraint (a maximum bound) for the policy to make it more realistic. If the total energy exceeds this bound, the process of bidding should be stopped. In the WPPs power plant, this constraint is “a network-determined power consumption”, which is usually planned one day in advance.

For the cost (reward) calculation, during the training phase, the actual data needed for cost calculation is duly supplied in accordance with the profit formula, as given in Eq. (1). As for the testing phase, WPP cannot supply all the inputs needed for the reward function in advance. Therefore, the predicted cost is obtained via model prediction.

4.1.2.1. The Critic Networks

The critic network is optimized by minimizing the loss function defined for the value function $V(s_t)$ approximation, which maps the current state s_t to the scalar. A3C uses multi-step rewards to optimize the critic network parameter. The function of loss is defined as Eq. (8) to accelerate the learning process [25].

$$L(\theta^v) = E_{\theta^v}\left(\sum_{i=0}^{k-1} \gamma^i r + \gamma^k V(s \mid \theta^v) - V(s \mid \theta^v)\right)^2 \quad (8)$$

when θ^v is set to the critic network parameter, r_t is the immediate reward the agent has acquired when the action is executed at state s_t, $V(s_t)$ is the value function.

4.1.2.2. The Actor Networks

The actor network's parameter is optimized by a policy-based method to maximize the cumulative discounted reward obtained after executing an action in state s_t due to the A3C policy-based algorithm variant. According to the RL process, samples are collected with the trajectory $\tau = \{s_t, a_t, s_t, \dots, s_T, a_T\}$ which the cumulative discounted reward from time step t is shown in Eq. (9). The trajectory corresponding to the bidding day is divided into 24 hours for wind energy bidding. There is a probability of a trajectory shown in Eq. (10).

$$R_t = \sum_{\tau}^{T} R(\tau) p_\pi(\tau) = E_{\tau \sim p_\pi(\tau)}[R(\tau)] \tag{9}$$

where $p_\pi(\tau)$ is the probability of the trajectory τ.

$$p_\pi(\tau) = p(s_t) \prod_t^T p_\pi\ (a_t \mid s_t) p(s_{t+1} \mid s_t, a_t) \tag{10}$$

where $p_\pi(a_t \mid s_t)$ is determined by the policy π, which is parameterized by θ^μ.

The partial derivation of the expected reward can be obtained concerning the neural network parameter θ^μ. The variance can be decreased by removing the baseline term $b_t(s_t)$, which in practice is uncommonly replaced by the value function $V(s_t)$. The reward's partial derivatives related to θ^μ is written in Eq. (11).

$$\nabla R_{\theta^\mu} \approx \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T} \big(R_t(\tau^n) - V(s_t)\big) \nabla log\, p_{\theta^\mu}(a_t^n \mid s_t^n) \tag{11}$$

where $R_t(\tau^n)$ is the accumulated discounted reward obtained after executing action at in state s_t of the n-th trajectory, which is an evaluation of action value function $Q(a_t, s_t)$. The term $R_t(\tau^n) - V(s_t)$ is an estimate of the advantage function

$A(a_t, s_t) = Q(a_t, s_t) - V(s_t)$, which represents how good it is to select action a_t in state s_t.

Following the encouragement of an agent to explore more in the environment, the A3C algorithm adds to the objective function the term $H\left(\pi(s_t; \theta_\mu)\right)$, which is the entropy of the policy π. The policy gradient [27] is therefore formulated to Eq. (12).

$$\nabla R_{\theta^\mu} \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T}\nabla_{\theta_\mu} log\, \pi(a_t \mid s_t; \theta_\mu)\left(\sum_{i=0}^{k-1}\gamma^i r_{t+i} + \gamma^k V(s_{t+k} \mid \theta_v) - V(s_t \mid \theta_v)\right) + \beta\nabla_{\theta_\mu} H\left(\pi(s_t; \theta_\mu)\right) \quad (12)$$

where β is the weight factor for the period of regularization. ANN is applied to learn a robust strategy to address environmental uncertainties. The parameters of the actor network are updated $by\ \theta_\mu \leftarrow \theta_\mu - \eta_\mu \nabla_{\theta^\mu} R_{\theta^\mu}$ where η is the learning rate of the actor network [27].

4.1.2.3. The Convolution A3C (Conv-A3C)

The Convolutional Neural Network (CNN) is applied to both actor and critic networks. Figure 21 illustrates our proposed Conv-A3C network architecture.

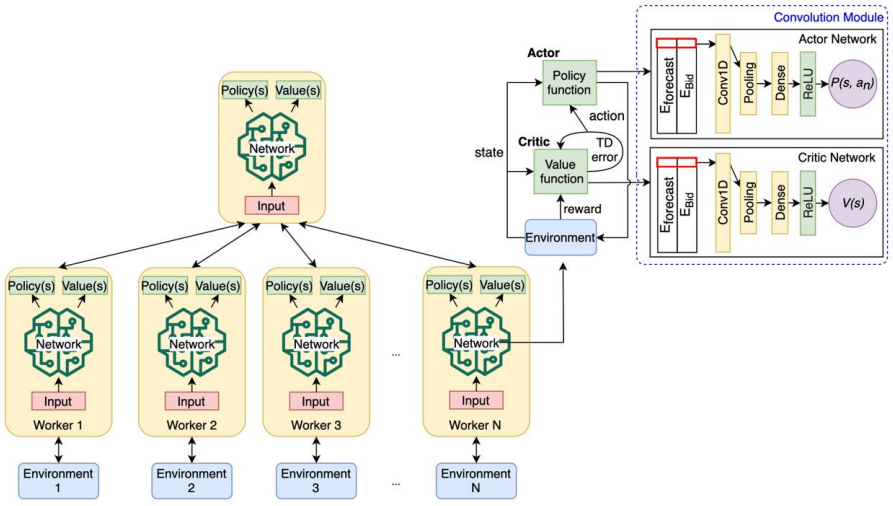


Figure 21. The Conv-A3C diagram with the CNN architecture in actor and critic network.

It is suitable for multiple time-series inputs (wind power and wind bidding) since it considers all series with multiple time steps altogether (filter size = 2x1). Also, the CNN network permits the volume and complexity of time series data for the deep learning concept.

4.1.3. Model-Based Reinforcement Learning (MBRL) Framework

Model-Based Reinforcement Learning (MBRL) learns optimal behavior indirectly by learning the environment when taking action and observing outcomes that include the next state and immediate reward. Dyna architecture is a variation of MBRL that proceed to update the value functions instead of only building a model with real experience. From module 3 in Figure 19, we applied MBRL with LSTM network architecture to model the environment, which characterizes time series as shown in Figure 22.

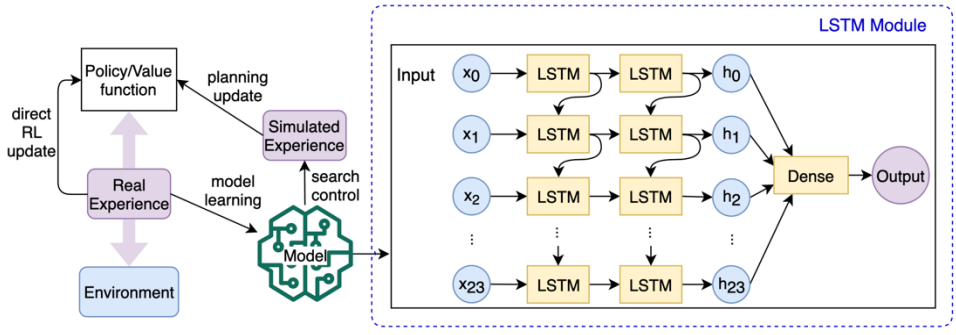


Figure 22. The MBRL diagram with LSTM architecture for environment model.

The workflow begins with a standard reinforcement learning agent, learning of domain knowledge, model-generated experiences, and planning. The LSTMs assure the ability to learn the context involved to create predictions in time-series forecasting problems, rather than having this context pre-specified and fixed, which is applied to predict future value need to be optimized in specific bidding.

4.1.4. The overall process

We train our MB-A3C with the procedure demonstrated in Figure 23. The process begins with wind energy forecasting. Then, 24 forecast values are passed to the policy model viz. Conv-A3C as a state to collect samples during training. The training process continues until the average cost per day calculated in accordance with Eq. (1), using actual data, stabilizes. Next, the collected samples utilize the LSTM algorithm for the 24 cost prediction. After that, the MPC process is applied for the testing phase.

Algorithm 1: MB-A3C algorithm

1. Initial inputs r_t, η_v, η_u, T_{max}, t_{max}, γ, β, and output π
2. Initial environments; state, action, and reward space, of each dataset with [44].
3. Assume global shared parameter vectors (θ_μ, θ_v), and global shared counter T = 0, thread-specific parameter vectors (θ'_μ, θ'_v)
4. Run A3C with D_{train} to collect trajectories $D_{env} = (s, a, r)$
5. Train env. model on $D_{env.}$
6. Run A3C with env. model from 4. for D_{test}

for *<episode = 1: T_{max}>* **do**

1. Reset gradients of actor and critic network $d\theta_\mu \leftarrow 0$, $d\theta_v \leftarrow 0$
2. Reset $\theta'_\mu = \theta_\mu$, $\theta'_v = \theta_v$

for *<t = 1: 24>* **do**

1. Get state s_t
2. Perform a_t according to policy $\pi(a_t \mid s_t; \theta'_\mu)$ and the network-determined consumption constraint.

if *($a_t \leq$ consumption)* **then**

Perform a_t according to policy $\pi(a_t \mid s_t; \theta'_\mu)$

else

Perform a_t according to consumption

3. Every actor send its (s_t, a_t) to env. model.
4. Env. model predicts r_t and send to A3C model.
5. A3C action with $r_t = \text{argmax}\, r_t \{a_{1_t}, a_{2_t}, \ldots, a_{n_t}\}$, receive r_t, then transfer to new state

if *<T/T_{max} == 0:>* **then**

6. $R = \begin{cases} 0, & \text{to terminal } s_t \\ V(s_t, \theta'_v), & \text{to not terminate } s_t \end{cases}$

for *<$i \in \{t, t-1, \ldots, t_{\max}\}$>* **do**

7. $R \leftarrow r_i + \gamma R$
8. Accumulate gradients with θ'_v and θ'_μ according to Eqs. (9) and (13)

.1

9. Update the gradient of networks: θ_μ by $d\theta_\mu$ and θ_v by $d\theta_v$
10. Reset gradients of actor and critic network $d\theta_\mu \leftarrow 0$, $d\theta_v \leftarrow 0$
11. Reset $\theta'_\mu = \theta_\mu$ and $\theta'_v = \theta_v$

Figure 23. MB-A3C algorithm**.** where r_t is the reward function. η_v and η_u are the actor's and critic network's learning rate. T_{max} is the maximum training episode which is the updated time-step. γ is the discount factor. β is the entropy regularization term. π is the policy. D_{train}, $D_{env.}$, and D_{test} are training, environment, and testing dataset, respectively.

Table 2 depicted the example of information from MB-A3C. Begin with training phrase, reward is calculated from cost calculation in policy model (Module 2). Then, it is predicted via environment model in MBRL framework (Module 3).

Table 2. Bidding information example from MB-A3C from training to testing process.

EP	Hour	State		Action		Reward	Remark
		Wind forecast (t)	Wind bid (t-1)	Wind bid (t)	Wind reserve (t)	WPP's cost	
1	1:00	7.2	2.61	5	4.59	-8.43	Training
	2:00	8.19	4.42	7	3.77	-2.12	
	3:00	9.59	1.45	4	8.14	-9.86	
	4:00	6.16	0.08	3	6.08	-1.78	
	5:00	1.36	12.01	2	10.65	3.27	
	6:00	1.7	10.78	4	9.08	-8.93	
	7:00	5.26	3.65	5	1.6	-4.37	
	...	...	...	...	...	...	
	23:00	11.14	11.5	12	0.35	-6.53	
	0:00	2.41	0.28	2	2.13	-4.76	
	WPP's Cost					-40.11	
...	...	...	...	...	...	...	...
50k	1:00	6.51	12.55	10	6.03	-10.11	Testing
	2:00	7.9	0.22	5	7.69	-3.41	
	3:00	7.24	6.24	7	1	0.82	
	4:00	7.25	11.27	8	4.02	-1.24	
	5:00	11.02	4.06	5	6.96	-9.34	
	6:00	9.23	4.81	8	4.42	-6.2	
	7:00	3.86	1.84	3	2.02	0.08	
	...	...	...	...	...	...	
	23:00	4.76	9.16	8	4.39	-6.01	
	0:00	5.49	10	8	4.51	-4.87	
	WPP's Cost					-63.79	

4.2. MB-MA-DRL for Energy Trading in the Solar-installed households

The MB-A3C3 structure, which consists of three modules, is shown in Figure 24. During the training phase, agents are categorized based on their daily trading activity (Module 2) after executing A3C3 to collect environmental data (Module 1). After that, a forecasting module (Module 3) is used to estimate the trading quantity and price of agents using centralized data from Module 2. The present condition is then used to generate the predicted trading quantity and price for the testing phase. The policy model then analyses the current situation and proposes a policy that will result in DA market behavior (amount of trading quantity and price).

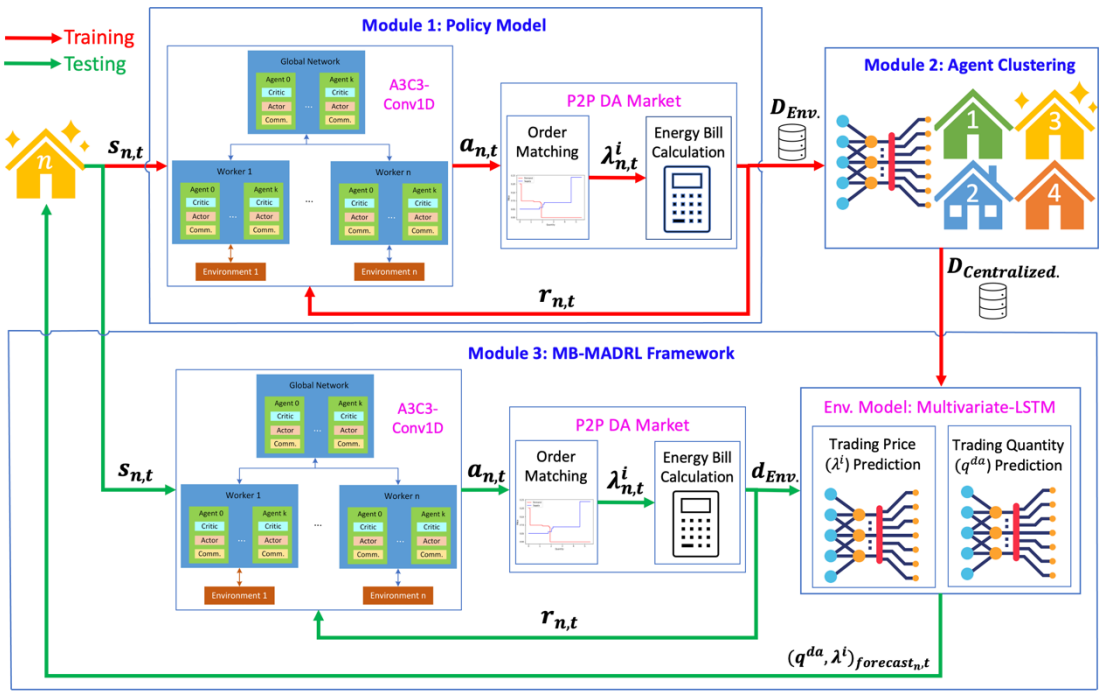


Figure 24. Schema of the three modules: (1) Policy model, (2) Agent clustering, and (3) MB-MADRL framework.

4.2.1. Policy model: A3C3-Conv1D with DA mechanism

The convolutional 1-dimensional A3C3 (A3C3-Conv1D) neural network was enhanced via application of A3C3. To build an optimal trading strategy for each agent,

model parameters were used and updated based on experience and reward information. A customized P2P energy trading environment was used to implement this optimal trading strategy as a policy model. The assumptions listed below are attributed to the A3C3-Conv1D, and A3C3 alike:

1. The concept of “asynchronous” refers to when numerous threads collaborate on the same task and communicate what they have learned. Then, a solution is achieved more efficiently.
2. “Multi-agent actor” and “centralized-critic”: The “multi-agent actor” provides values based on their current policy for various actions. A “centralized critic” combines agent observations and environmental state information to provide an estimate of the current state and evaluates actions.
3. The term “advantage” describes how much better a certain action is relative to the predicted average value of the situation on which it is based.
4. “Communication” allows agents to share important information explicitly via a communication network based on the performance of other agents.

Furthermore, policies have constraints (a maximum bound) to make them more realistic. The system constraints: a households’ load and energy storage. The trading constraints: price threshold, network capacity, and system status. If trading system fail, buy/sell with main grid. If main grid fails, each household buy/sell with community especially those who owns energy storage.

A household’s energy storage, which is determined, is placed to offer trading quantity with minimum and maximum energy levels between 2 and 10 kWh.

1. For trading prices, we followed the grid pricing provided by the data owner, as in Table 3, which includes the time-of-use (ToU) tariff, a flexible grid purchase price for the period, and the feed-in tariff (FiT), a set grid sale price of 0.04 $/kWh for the entire day [69]. The agent's trading price output, whether buy or sell, is limited to grid prices.

Table 3. Grid Pricing by period.

Time	ToU ($/kWh)		FiT ($/kWh)
	Time	Value	
Shoulder	09:00-16:00	0.13	0.04
Peak	17:00-20:00	0.18	
Off-Peak	21:00-08:00	0.08	

2. The network capacity threshold is considered peak demand when trading in the DA market. The algorithm employs a daily peak demand of 600 kW, which is sufficient to satisfy the network's capacity [90].

3. The condition in the trading algorithm will suggest trading directly from the grid if forecasted values deviate from historical weekly data.

4.2.1.1. The Actor Networks

As depicted in Figure 25 local policy is learned by the actor network. In this example, the actor receives all of the other agents' observations and broadcast messages as input.

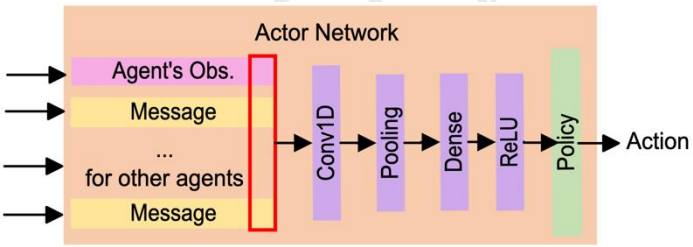


Figure 25. Agent's actor network.

Regarding that, the network's output layer provides a probability distribution for agent j's actions. The output layer is based directly on the environment's action space.

4.2.1.2. The Centralized Critic Networks

In Figure 26, the agent's centralized network is depicted, combining all agent observations with some additional information from the environment. If the environment allows access to its underlying state, the centralized observations become the entire environmental state s_t. Thus, policy is evaluated by the centralized critic. Sampled as a fully observable environment state shared by all agents. In some cases, this may not be feasible.

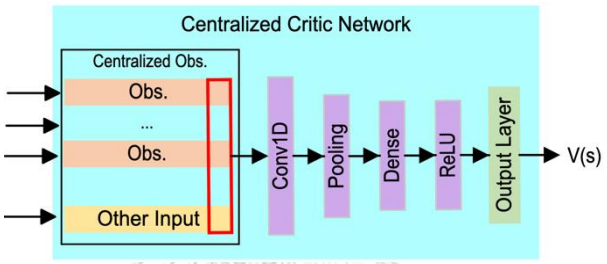


Figure 26. Agent's centralized critic network.

4.2.1.3. The Communication Network

In Figure 27, the communication network of the agent is depicted. To produce messages, the output layer has a rectifier with a ReLU activation function. Other output topologies are supported, such as continuous valued messages. The communicator network learns a communication protocol between agents.

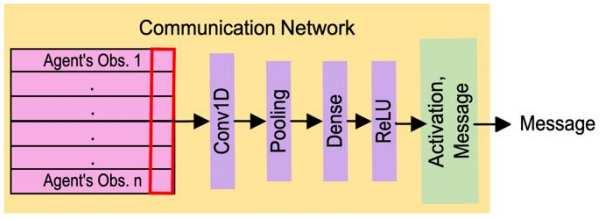


Figure 27. Agent's communication network.

4.2.3. Agent's daily trading behavior clustering

Due to their daily routines and activities, a household's daily behavior is currently different. Agents are clustered together for day-to-day trading to aggregate

similar trading activity as a single dataset for environmental modeling. Dynamic time warping (DTW) is a more precise method of calculating distance than Euclidean distance; data points are shifted between each other and the shape is prioritized above the geometry [91-93]. The assumption of Euclidean distance does not need two-time series to be of the equal length. The Euclidean distance is used to compare two data points [94]. The elbow [95] and silhouette [96] method is employed to determine the optimal k-means.

Due to the large number of agents and their diverse behavior, it is assumed that an agent's daily behavior differs hour by hour. Accordingly, 300 agents are organized into four clusters based on their daily trading behavior. In the literature, DTW is frequently used in conjunction with k-medoids and hierarchical approaches. Occasionally, DTW is used in conjunction with k-means in some articles, but this is debatable [97]. DTW has also been coupled with random-swap and hybrid among non-traditional approaches [98]. Because of its one-to-many determination, DTW is used to assess the similarity of an agent's daily trade volume. DTW calculates the shortest distance between all points, allowing for a one-to-many match.

4.2.4. Model-based multi-agent deep reinforcement learning (MB-MADRL) framework

The multivariate-LSTM, depicted in Figure 28, consists of six time-dependent variables $(x_1, \dots, x_6) = \left[P_{n,t}^{inf}, E_{n,t}^{es}, \lambda_t^b, \lambda_t^s, q_{n,t-1}^{da}, \lambda_{n,t-1}^i, r_{n,t-1}^i\right]$, is dependent on others in addition to its own previous values. The hidden layer output $(h_1, \dots, h_6)$ from one step of the network is passed to the next. The algorithm captures not only the previous hour, but also the previous 24 hours of the input sequence. The technique is used to calculate the state's predicted trading quantity $\left(q_{n,t}^{da}\right)$ and price $\left(\lambda_{n,t}^i\right)$. The multivariate-LSTM is used to forecast an agent's trading quantity and price since it can multiply the output of hidden states by trainable weights, whereas traditional LSTM networks only use the latest hidden state as output [62].

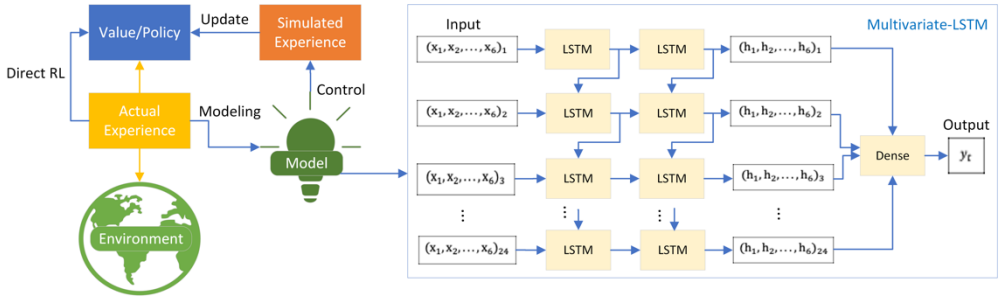


Figure 28. Schema of multivariate-LSTM having six features and a single output.

4.2.5. The overall process of MB-A3C3

As outlined below in Figure 29, Algorithm 1, the MB-A3C3 algorithm is demonstrated. The first step in the procedure is for A3C3-Conv1D to gather environmental data. Energy bills from the DA market mechanism, using actual data, stabilize after ten random runs of the training procedure. Then, the DTW algorithm is applied for time-series clustering to categorize the agents. The data from each cluster is gathered in order to combine the environmental data with the multivariate-LSTM for trading quantity and price forecasts for the following 24 hours. The MBRL technique is then used throughout the testing phase.

Table 4, depicted the example of information from MB-A3C3. Begin with training phrase, forecast trading quantity and price are formulated from actual data in policy model (Module 1). Then, it is predicted via environment model in MBRL framework (Module 3).

Algorithm 1: MB-A3C3 algorithm

1. Initial inputs r_t, η_v, η_u, η_w, T_{max}, t_{max}, γ, β, and output π
2. Initial environments; state, action, and reward space, of dataset.
3. Assume global shared parameter vectors (θ_μ, θ_v), (θ, and global shared counter T = 0, thread-specific parameter vectors (θ'_μ, θ'_v)
4. Run A3C3 with D_{train} to collect agent's trajectories $D_{env} = (s, a, r)$
5. Time-series clustering with DTW on $D_{env.}$
6. Aggregate each cluster's dataset to $D_{centralized}$
7. Train env. model on $D_{centralized}$ for each cluster.
8. Run A3C3 with env. model from 7. for D_{test}

for *<episode = 1: T_{max}>* **do**

1. Reset gradients of actor, centralized critic, and communication network $d\theta_\mu \leftarrow 0$, $d\theta_v \leftarrow 0$ $d\theta_w \leftarrow 0$
2. Reset $\theta'_\mu = \theta_\mu$, $\theta'_v = \theta_v$, $\theta'_w = \theta_w$

for *<agent = 1: N)>* **do**

for *<t = 1: 24>* **do**

1. Get state s_t
2. Perform a_t according to policy $\pi(a_t \mid s_t; \theta'_\mu)$ and constraints.
3. Every actor send its (s_t, a_t, r_t) to env. model according to agent's cluster.
4. Env. model predicts a_t and send to A3C3 model.
5. A3C3 action with $r_t = \text{argmax}\, r_t \{a_{1_t}, a_{2_t}, \ldots, a_{n_t}\}$, receive r_t, then transfer to new state

if *<T/T_{max} == 0:>* **then**

6. $R = \begin{cases} 0, & \text{to terminal } s_t \\ V(s_t, \theta'_v), & \text{to not terminate } s_t \end{cases}$

for *<$i \in \{t, t-1, \ldots, t_{\max}\}$>* **do**

7. $R \leftarrow r_i + \gamma R$
8. Accumulate gradients with θ'_v, θ'_μ, and θ'_w.

9. Update the gradient of networks: θ_μ by $d\theta_\mu$, θ_v by $d\theta_v$, and θ_w by $d\theta_w$
10. Reset gradients of actor, centralized critic, and communication network $d\theta_\mu \leftarrow 0$, $d\theta_v \leftarrow 0$, and $d\theta_w \leftarrow 0$
11. Reset $\theta'_\mu = \theta_\mu$, $\theta'_v = \theta_v$, and $\theta'_w = \theta_w$

Figure 29. MB-A3C3 algorithm. where r_t is the reward function. η_v, η_u, and η_w are the actor's, centralized critic, and communication network's learning rates. T_{max} is the maximum training episode and tmax is the updated time-step. γ is the discount factor. β is the entropy regularization term. π is the policy. D_{train}, $D_{env.}$, $D_{centralized}$, and D_{test} are training, environment, centralized environment, and testing datasets, respectively.

Table 4. Trading information example from MB-A3C3 from training to testing process.

Ep	Hour	Agent 1											300 Agents	Remark
		State									Action	Reward	Reward	
		Load	ES	Gird	Grid	Trade Quantity	Trade Sell	Trade Buy	Trading Price	Grid Price	Trade Quantity	Trade	Trade	
1	1:00	2.036	0	0.04	0.08	-2.091	0.041	0.069	0.069	0.080	-2.071	-0.156	-13.34	Training
	2:00	1.807	0	0.04	0.08	-1.879	0.046	0.060	0.060	0.080	-1.831	-0.138	-10.92	
	3:00	1.181	0	0.04	0.08	-1.324	0.050	0.078	0.078	0.080	-0.955	-0.072	-7.77	
	4:00	0.665	0	0.04	0.08	-0.772	0.044	0.077	0.077	0.080	-1.233	-0.093	-5.83	
	5:00	1.09	0	0.04	0.08	-1.186	0.042	0.066	0.066	0.080	-0.583	-0.044	-5.78	
	6:00	0.337	0	0.04	0.08	-0.463	0.048	0.075	0.075	0.080	-0.497	-0.037	-5.84	
	7:00	0.101	0	0.04	0.08	-0.170	0.042	0.080	0.080	0.080	-0.075	-0.006	-7.61	
	...	...	...	...	...	...	...	...	...	...	...	...	...	
	23:00	0.219	0.038	0.04	0.08	-0.320	0.049	0.078	0.078	0.080	-0.113	-0.008	-9.40	
	0:00	0.164	0.175	0.04	0.13	-0.136	0.048	0.111	0.111	0.130	-0.118	-0.009	-16.09	
		Community's Energy Bills											-437.977	
1.3k	...	...	...	...	...	...	...	...	...	...	...	...	...	Testing
	1:00	0.201	0.419	0.04	0.13	0.052	0.046	0.124	0.046	0.040	0.063	0.001	-20.96	
	2:00	0.38	0.619	0.04	0.13	0.239	0.044	0.127	0.044	0.040	0.372	0.014	-19.57	
	3:00	0.11	0.488	0.04	0.13	0.193	0.040	0.128	0.040	0.040	0.504	0.019	-18.57	
	4:00	0.109	0.125	0.04	0.13	-0.002	0.042	0.111	0.111	0.130	0.243	0.009	-18.04	
	5:00	0.109	1.156	0.04	0.13	0.862	0.047	0.125	0.047	0.040	0.853	0.031	-15.76	
	6:00	0.107	0.7	0.04	0.13	0.479	0.041	0.111	0.041	0.040	0.804	0.030	-14.25	
	7:00	0.409	0.2	0.04	0.13	-0.363	0.047	0.128	0.128	0.130	0.174	0.002	-15.56	
	...	...	...	...	...	...	...	...	...	...	...	...	...	
	23:00	0.651	0	0.04	0.08	-0.673	0.042	0.068	0.068	0.080	-0.416	-0.031	-15.55	
	0:00	1.149	0	0.04	0.08	-1.244	0.047	0.064	0.064	0.080	-0.620	-0.047	-14.77	
		Community's Energy Bills											-528.320	

CHAPTER V

EXPERIMENTS AND RESULTS

The experiment of two phases, wind energy and P2P energy trading are conducted due to the details.

5.1. Experiment on SARL for Wind Energy Bidding in the Wholesale Electricity Market

We conduct the experiment on Nord Pool dataset with scenarios and evaluation which detail is depicted below.

5.1.1. Data Description

The experiment was carried out analyzing six datasets from wind farms viz. two from Denmark (DK1-2) and four from Sweden (SE1-4) as obtained from Nord Pool, a European power exchange [99]. Each dataset was tested on five different scenarios in the reserve market by varying the price ratios that affect reserve capacity and dispatch prices.

The hourly resolution day-ahead market datasets are divided into two sets i.e. the training set (01/01/2016 - 31/05/2018) and the testing set (01/06/2018 - 27/10/2018). The dataset contains seven variables: wind production, wind production prognosis, consumption prognosis, and up-regulating volume in megawatt-hour (MWh), up and down regulating price, and spot price, as described in Table 5. The price data is in the currency of Danish Krone (DKK) for Denmark (similar to the baseline [3]) and Euro (EUR) for Sweden. We considered the same reward function for both countries since Sweden is in the same region of Northern Europe as Denmark.

Table 5. The description and range of day-ahead market data. The unit of wind energy and regulating volume is megawatt-hour (MWh). The price data is in currency of Danish Krone (DKK) for Denmark and Euro (EUR) for Sweden**.**

Data	Range		Unit	Description
	Denmark	Sweden		
Wind production	-2 – 3,771	0 – 2,071	MWh	Actual wind produced by WPP.
Wind production prognosis	0 – 3,973	0 – 5,694		Forecasted wind produced by WPP.
Up regulating volume	0 - 666	0 – 1,442		Reserve energy.
Consumption prognosis	750-3,463	650-18,215		The demand for next coming day.
Up regulating price	-372.51 – 5,001	0 - 670.16	DKK for Denmark, EUR for Sweden	The regulating price when up situation is activated.
Down regulating price	-837.44 – 1,898.90	-1,000.0 - 255.02		The regulating price when down situation is activated.
Spot price	-398.61 – 1,898.9	0 - 255.02		The day-ahead price announced at the end of the bidding period by regulator.

5.1.2. Experimental Scenarios

According to Eq. (1), $\eta_{up,t}$ corresponds to the cost of the opportunity when WPP participates in the reserve market and $\mu_{up,t}$ is the real-time dispatch price. The

price ratio of the reserve capacity and the reserve dispatch price is introduced in [26] to explore the impact of the reserve energy cost on the revenue of the WPP according to Eqs. (8) and (9).

$$\eta_{up,t} = \psi p_{DA,t} \tag{8}$$

$$\mu_{up,t} = (1+\omega) p_{DA,t} \tag{9}$$

The reserve cost refers to the cost of purchasing and dispatching operation when WPP is combined with other energy sources. It corresponds to different costs in specific scenarios. The reserve cost corresponds to the cost of the bilateral reserve market when WPP participates in both the energy market and the bilateral reserve market. Five scenarios interpreted in Table 6 are conducted in our experiment to investigate the effect of different circumstances on the cost.

Table 6. The description of each experimental case**.**

Case no.	φ	ω	Description
1	0	0	Only trade in energy market
2	0	0.1	No cost of purchasing from others but having operation for reserve energy dispatching from WPP's own storage devices.
3	0	0.2	
4	0.2	0.1	Purchasing and having operation for reserve energy dispatching from others
5	0.2	0.2	

5.1.3. Hyperparameters Setting and Details

The efficiency of RL is enhanced by the potential of deep learning internally. The hyperparameters of MB-A3C in Table 7 are categorized by three main modules as in

Figure 24. We used the TensorFlow software library for machine learning procedure and the OpenAI Gym for developing reinforcement learning algorithms with the customized environment of the day-ahead market data.

Table 7. The MB-A3C's hyperparameter setting**.**

Modules	Hyperparameter Setting and Details
Forecasting Model; Attention-LSTM	- Data is preprocessed by Min-max normalization - Three Keras layers with 128 LSTM hidden nodes of each. - The Stochastic Gradient Descent (SGD) optimizer - The attention weights scoring with tanh and softmax. - The Xavier normal initializer for the initial random weights of Keras layers setting. - The MSE is evaluated for early stopping during training.
Policy Model; Conv-A3C	- Data is divided by 24 for normalization. - Ten actors with 24 step size each episode due to hourly resolution of datasets. - Learning rate 0.00001 for actor and 0.0001 for critic network. - Discount factor (γ) value is 0.1. - The entropy term controlling hyper parameter (β) value is 0.01. - Tensorflow layer with 10 units, Rectified Linear Unit (ReLU) as activation function for both actor and critic networks.

Modules	Hyperparameter Setting and Details
MBRL Framework; LSTM	- Data is preprocessed by Min-max normalization. - Two Keras layers with 256 LSTM hidden nodes of each. - The Adaptive Moment Estimation (Adam) optimizer. - The Xavier normal initializer for the initial random weights of Keras layers setting. - The MSE is evaluated for early stopping during training.

5.1.4. Evaluation

The purpose of our MB-A3C is to minimize the cost of purchasing and dispatching reserve energy in WPP's revenue by defined reward function: the second term of Eq. (1). The average cost per day, the cumulative reward of each episode consisting of 24 steps, is determined by the action performed by the MB-A3C algorithm during the training to optimize all parameters in Conv-A3C. Then, the optimized algorithm is tested to evaluate how well the MB-A3C completes the task according to the forecasting model, policy model, and experience with the predicted cost from the environment model in the MBRL framework.

5.1.5. The Experimental Result

The result on SARL depicted in this section is divided as overall result and effect on the forecasting model.

5.1.5.1. Overall Results

The experimental results are depicted in Table 8 with five scenarios of each dataset to assess the impact of the purchase and dispatch price on WPP's revenue with the discrepancy between ψ and ω of each scenario. The result suggests that MB-A3C learns and strategically bids to reduce costs over other RL algorithms with the penalty for the imprecision of the forecasting of wind energy and the uncertainty of price activation.

Table 8. The average cost per day of each dataset compared to RL algorithms.

Dataset	Algorithm					Upper Bound	%Diff.
	A3C	DPPO	DDPG	MBPG	MB-A3C (Ours)		
In the currency of Danish Krone (DKK) per day							
DK1	2,025.63	2,269.39	2,552.37	2,564.94	**1,960.82**	1,957.03	0.19
	1,747.30	2,037.90	1,660.17	1,691.08	**1,613.84**	1,606.08	0.48
	2,310.88	2,111.42	2,218.95	2,579.30	**2,054.10**	2,025.13	1.43
	2,999.29	2,880.65	2,972.43	2,847.05	**2,805.73**	2,729.93	2.78
	3,017.01	3,035.86	3,084.01	2,918.68	**2,891.49**	2,798.99	3.30
DK2	791.39	1,042.31	849.13	813.77	**743.82**	665.32	11.80
	1,271.04	1,161.34	1,220.46	1,418.68	**817.84**	730.94	11.89
	1,112.29	1,068.31	1,102.34	1,055.85	**891.86**	796.55	11.97
	1,787.61	1,716.92	1,771.62	1,696.90	**1,686.07**	1,577.28	6.90
	1,797.20	1,777.18	1,801.79	1,792.26	**1,760.09**	1,642.90	7.13
In the currency of Euro (EUR) per day							
SE1	53.73	58.74	68.99	59.84	**52.28**	52.18	0.19
	58.05	60.06	68.11	59.07	**57.50**	57.38	0.21
	55.99	59.00	71.89	62.35	62.72	62.59	0.21
	380.64	380.69	445.59	406.63	**369.90**	324.47	14.00
	401.51	401.56	472.38	429.87	**375.12**	329.68	13.78
SE2	**265.05**	298.12	382.72	276.60	275.38	239.30	15.08
	328.09	328.17	435.49	371.07	**291.47**	263.10	10.78
	386.18	386.28	510.08	435.76	**317.89**	286.90	10.80

Dataset	Algorithm					Upper Bound	%Diff.
	A3C	DPPO	DDPG	MBPG	MB-A3C (Ours)		
	543.07	601.56	759.43	554.87	563.26	507.81	10.92
	586.69	619.55	795.36	689.82	**569.49**	531.61	7.13
SE3	865.11	496.98	717.47	496.81	**419.11**	397.60	5.41
	854.06	487.06	704.73	**479.02**	482.48	436.86	10.44
	718.93	529.60	759.34	621.42	**524.77**	476.12	10.22
	948.86	669.04	906.57	763.98	**658.45**	575.05	14.50
	1,011.51	731.70	987.01	833.75	**718.36**	614.31	16.94
SE4	395.35	395.45	532.36	450.17	**331.03**	311.01	6.44
	389.18	389.29	524.45	443.31	**342.13**	342.11	0.01
	415.59	415.70	558.36	472.72	**373.22**	373.20	0.01
	1,151.39	1,151.64	1,478.45	1,282.26	**1,100.79**	1,097.87	0.27
	1,219.76	1,220.03	1,566.24	1,358.41	**1,131.88**	1,128.97	0.26

*****The upper bound refers to the model knowing the actual reserve and bidding energies**.** The bold number is the lowest cost according to such an algorithm**.** The **%**Diff**.** represents the difference of an average cost per day between MB-A3C **(**ours**)** and the upper bound**.**

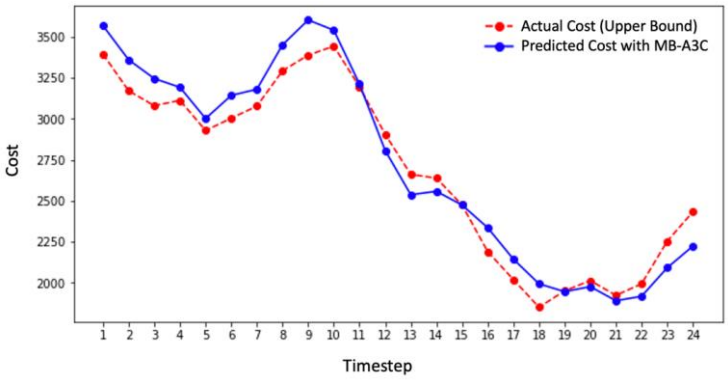
As shown in Table 8, MB-A3C is the winner in almost all cases (26 out of 30 cases). Also, the %difference is introduced in the last column to investigate the deviation between the cost from MB-A3C and the upper bound, which is derived from the actual amount and considered as the best value. In most cases, MB-A3C's results are quite close to the upper bound's results, which are derived from actual amounts.

Table 9 is a summary of the comparison results from Table 8. Based on paired t-test, MB-A3C significantly outperforms all baseline with a p-value less than 0.0001. It has a lower cost than DPPO and DDPG on the whole 30 cases. Also, MB-A3C provides the lowest average costs 1,722.57 DKK and 450.86 EUR in Denmark and Sweden which is closest to baselines (1,653.02 DKK and 420.41 EUR) with %difference at 4.21% and 7.24%.

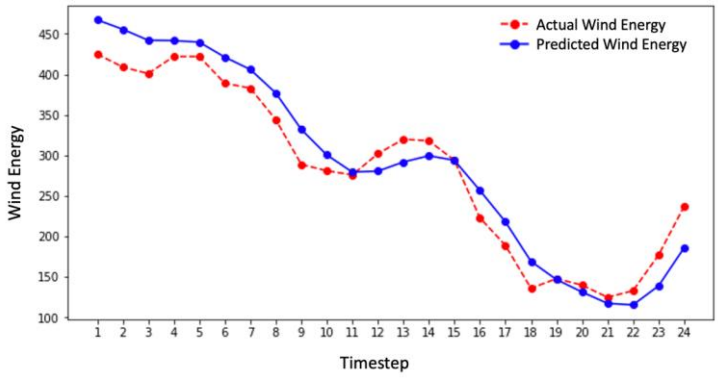
Table 9. The performance of MB-A3C is compared with the four RL algorithms.

MB-A3C comparing topics	Comparing Algorithms			
	Conv-A3C	DPPO	DDPG	MBPG
Denmark (DKK)				
#Winner cases	10	10	10	10
Average cost per day	1,885.96	1,910.13	1,923.33	1,937.85
Different of average cost per day	163.40	187.56	200.76	215.29
%Different of average cost per day	9.49	10.89	11.65	12.50
p-value	0.0026	0.0029	0.0051	0.0251
Sweden (EUR)				
#Winner cases	17	20	20	19
Average cost per day	551.44	484.01	637.25	527.39
Different of average cost per day	100.58	33.15	186.39	76.53
%Different of average cost per day	22.31	7.35	41.34	16.97
p-value	0.0044	< 0.05	< 0.05	< 0.05

As depicted below in Figure 30a, results show that MB-A3C's trend is very close to that of the upper bound. However, there are some data points where MB-A3C's predicted cost is lower than the upper bound. The reason being is such that when wind energy is bid through the MB-A3C framework, the cost is correlated to the amount of energy sold, as determined via Eq. (1). The consistency of predicted cost and wind energy in Figure 30 shows that the algorithm learns to bid less energy and incurs less cost than the upper bound.



(a) The actual and predicted cost.



(b) The actual and predicted wind energy.

Figure 30. Predicted cost of MB-A3C is less than the actual amount (upper bound) at some data points on 30th August 2018: case study no. 5, SE1 test dataset.

5.1.5.2. Effect of The Forecasting Model

In this part, we are going to investigate the reasons behind our success. There are three main modules in our algorithm: (1) forecasting model, (2) policy model, and (3) MBRL framework as described in Chapter IV. Hence, if each module can perform effectively, this should drive an impact on our algorithm.

The attention-LSTM can forecast the wind power accurately with the Root Mean Square Error (RMSE) 2.53 MWh and the Mean Absolute Percentage Error (MAPE) 1.01% on the testing dataset. Figure 31 illustrates the average RMSE prediction for each hour where each line refers to the result of each test dataset. It is noted that there is

not much difference between RMSE as regards the short-term and long-term forecasting, except for DK2 (the blue line). Such an outcome infers that this trend will not change much for the next 24 hours.

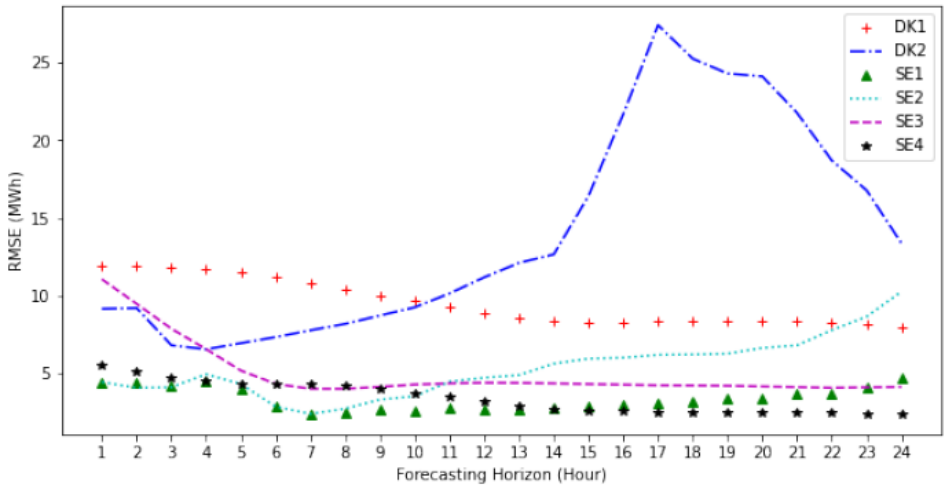


Figure 31. RMSE of each dataset for wind energy forecasting horizon: 1 to 24 hours.

The wind energy forecasting result from the attention-LSTM model is visualized in Figure 32. According to the accurate forecast of wind energy, the policy model can learn to take action and minimize the cost more efficiently.

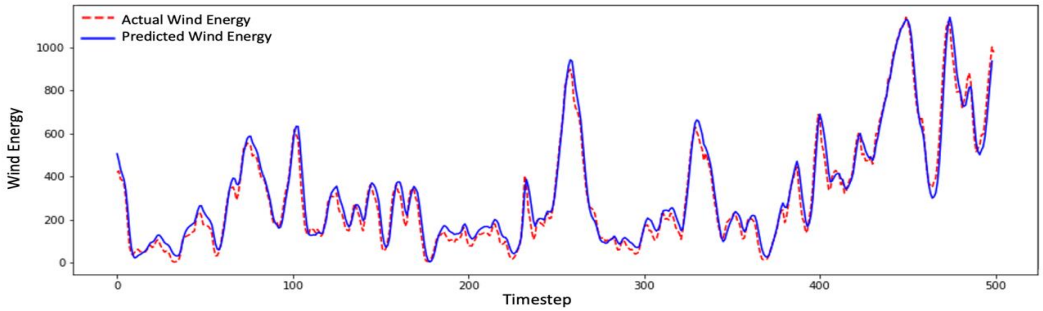


Figure 32. The results between actual and predicted wind power from Attention-LSTM model on first 500 time steps of SE4 testing set.

For the policy model, the original A3C uses neural networks, while ours uses 1D-CNN as network architecture which considers both series of forecast and previous bidding wind energy during the previous 24 hours (kernel size) altogether. The results in Table 10 indicate that Conv-A3C provides superior performance over A3C. Please note that the result is only conducted on the DK1 dataset as in [26].

Table 10. The average cost per day **(**DKK**)** of Conv**-**A3C is compared to A3C**.** The five experimental cases are conducted on the DK1 testing dataset**.**

Case no.	Baseline paper	Conv-A3C	%Reduction of Conv-A3C from baseline
1	2,449.00	2,025.63	17.29
2	1,808.00	1,747.30	3.36
3	2,449.00	2,310.88	5.64
4	3,052.00	2,999.29	1.73
5	3,067.00	3,017.01	1.63

The average RMSE prediction for each hour is illustrated below in Figure 33; each line refers to the result for each test dataset. Results show that there is no difference in RMSE for each hour of cost forecasting. The overall RMSE for the test datasets proved to be: 43.57 DKK (Denmark) and 8.58 EUR (Sweden).

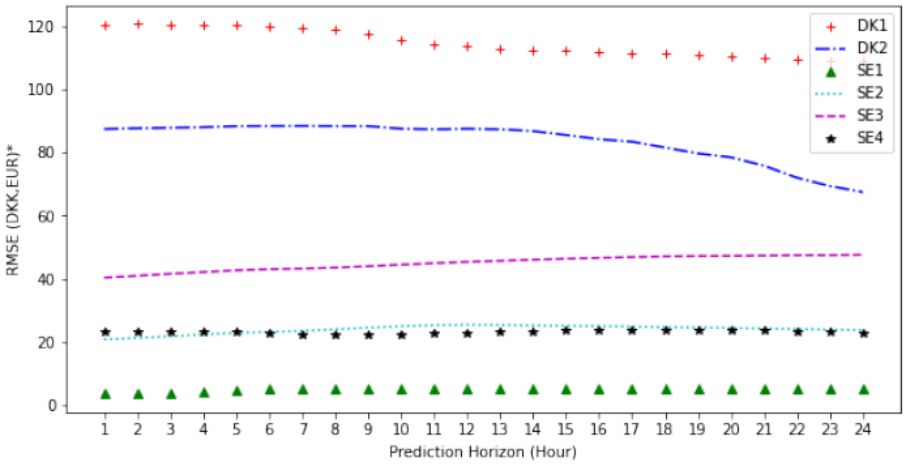


Figure 33. RMSE of each dataset for cost prediction horizon from 1 to 24 hours. *The units of RMSE are Danish Krone (DKK) for Denmark and Euro (EUR) for Sweden.

As illustrated below in Figure 34, comparison between actual and predicted cost for the environmental model is shown.

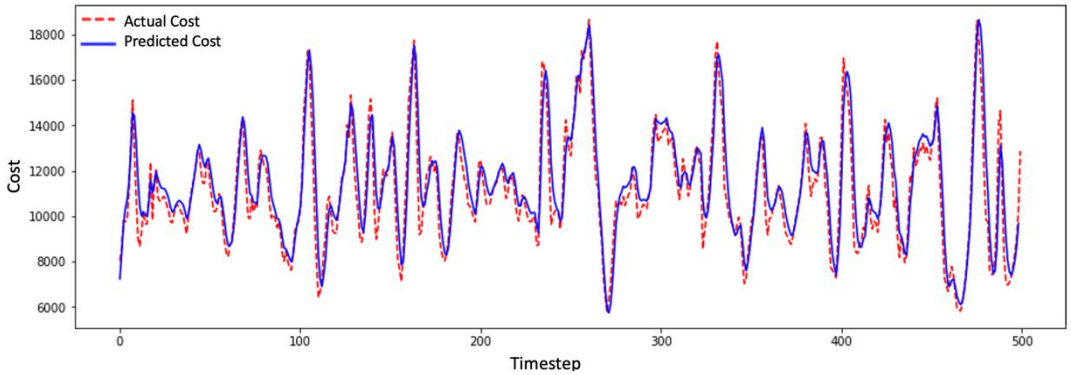


Figure 34. The cost prediction on the first 500 time steps comparison between the actual and predicted result**.** The result is derived from LSTM model on the SE4 testing set with case study no**.** 5**.**

Figure 35 illustrated the average cost per day on DK1 training data compared to each RL algorithm. The converged trend demonstrated that our MB-A3C learn and optimize for the best results above others. The cost tends to rise as the agent has no clue how to bid during the first stage. After the network parameters are optimized according to experience, the cost begins to converge.

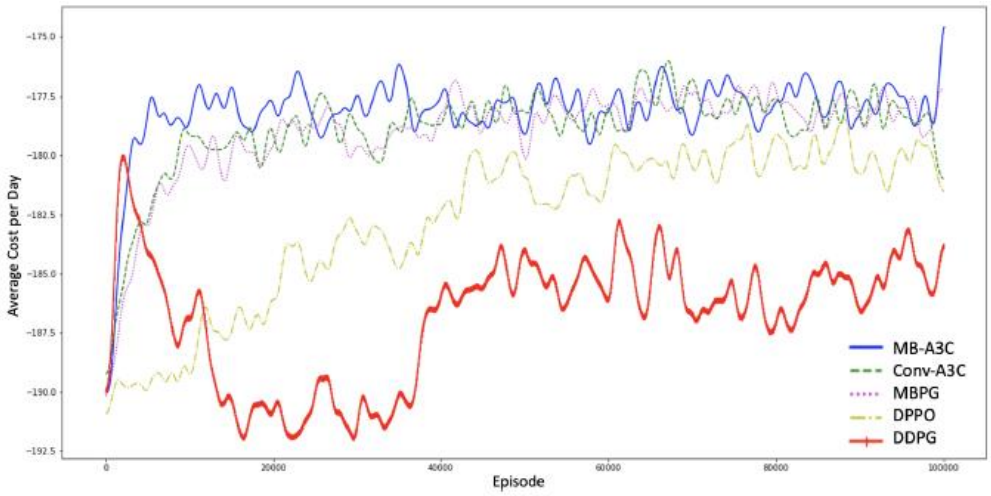


Figure 35. The average cost per day over 100k episodes of training**.**

The model-based deep RL algorithm MB-A3C makes a valid contribution to the strategic bidding of wind energy. MB-A3C combines the advantages of a time-series forecasting model with advanced machine learning methods via (i) the attention-LSTM (ii) Conv-A3C and (iii) the MBRL framework. It is evident that the reduction of average costs per day of bidding demonstrated the superiority of MB-A3C over other RL algorithms used, especially as conducted in the five scenarios of the well-known Nord Pool datasets: Denmark and Sweden. The five scenarios having different adjustments of price ratios for the cost of reserve and dispatch energy were carried out to investigate the potential of MB-A3C. It is significant that MB-A3C outperformed all baselines and performed closely to the upper bound. Such a model is found to provide the lowest average cost per day, which maximizes WPP's revenue.

5.1.5.3. Effect of component removal test

There are three main contributions for utilizing the novel MB-A3C in wind energy bidding:

a. Forecasting model: More accurate forecasting method effect of the model. It is effectively handle time-series and data complexity.
b. Policy model: The original A3C uses neural networks, while ours uses 1D-CNN as its network architecture, which considers both series of

forecasts and previous bidding for wind energy during the previous 24 hours (kernel size) altogether.

c. MBRL framework: To forecast a future cost since there are other factors (e.g., the market clearing price, or spot price) that are not known yet in the testing phase, this is the main contribution module for MBRL.

Figure 36. indicate the WPP's cost having component removal to assess the importance of each following contributions.

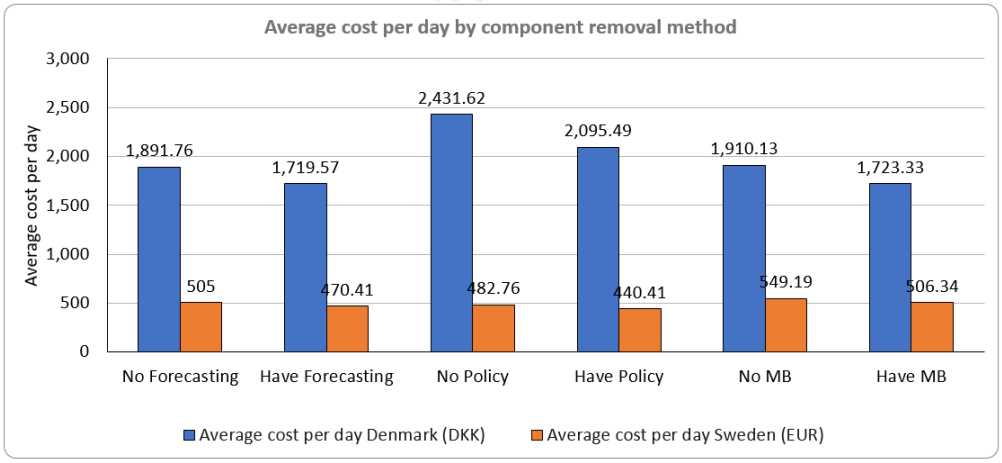


Figure 36. The WPP's average cost per day for component removal analysis; Denmark and Sweden.

The cost reduction of each component removal method is depicted in Table 11. Policy module is the most significant component in both countries, following with model-based framework and forecasting model, respectively.

Table 11. The effect of component removal test by WPP's average cost per day and percent of cost reduction of each.

Method Test	Average cost per day		% Cost Reduction	
	Denmark	Sweden	Denmark	Sweden
No Forecasting	1,891.76	505		
Have Forecasting	1,719.57	470.41	-9.10%	-6.85%
No Policy	2,431.62	482.76		

Method Test	Average cost per day		% Cost Reduction	
	Denmark	Sweden	Denmark	Sweden
Have Policy	2,095.49	440.41	-13.82%	-8.77%
No MB	1,910.13	549.19		
Have MB	1,723.33	506.34	-9.78%	-7.80%

5.2. Experiment on MARL for Energy Trading in the Retail Electricity Market

We conduct the experiment on dataset with scenarios and evaluation which detail is depicted below. The experiment was carried out after an examination of data from Ausgrid's electricity network. The publicly available dataset contains load and rooftop PV generation for 300 residential customers, with load centers encompassing Sydney and the adjacent rural areas. The data was collected over a three-year period; both load and PV generation measurements were taken at 30 min intervals.

5.2.1. Data Description

In Ausgrid's power network area, data was collected from 300 randomly chosen solar customers. Between July 1, 2010, and June 30, 2013, customers were billed on a domestic tariff and a gross metered solar system was installed. Customers were chosen based on a comprehensive set of real-world data gathered from meter readings between July 1, 2010 and June 30, 2011. During the first year, some data quality checks were conducted, thereby removing users who were on the high and low ends of home consumption and solar generation performance. Data from June 1, 2012 to May 31, 2013 were utilized to conduct the experiment. Training of 80% and testing of 20% of the dataset is divided according to time-series split. Tables 12 and 13 focus on the varied types of data matching annual statistics of solar home datasets.

Table 12. Description of solar home electricity data.

Column Name	Description	Range	Unit
Customer	Customer ID	1-300	-
Postcode	Postcode location of customer	-	-
Generator Capacity	Solar panel capacity recorded on the application for connection for each customer, which is the solar panels peak power under full solar radiation and tested under standard conditions.	1-9.99	Kilowatt Peak (kWp)
Consumption Category	Two letter code each meaning the following: GC = General Consumption for electricity supplied all the time (primary tariff, either inclining block or time of use rates), excluding solar generation and controlled load supply	0-6.57	Kilowatt Hour (kWh)
	CL = Controlled Load Consumption (Off peak 1 or 2 tariffs)	0-4.09	
	GG = Gross Generation for electricity generated by the	0-4.33	

Column Name	Description	Range	Unit
	solar system with a gross metering configuration, measured separately to household loads		

Table 13. Annual statistics of solar home customers (year 2012-2013).

Description	Mean	Median
Annual consumption; kWh per year	6,387	5,862
Annual gross generation; kWh per year	2,181	1,814
Solar system size (kWp)	2	2
Annual gross generation; kWh/kWp	1,297	1,326

5.2.2. Hyper parameters setting and details

The MB-A3C3 hyper parameters, which are divided into three modules in Figure 24, are specified in Table 14. In addition, the TensorFlow software library and the OpenAI Gym were utilized to assess RL algorithms and provide a customized environment employing P2P energy trade data. The Dyna conceptual framework takes into the account the constraints on such data.

Table 14. MB-A3C3 hyper parameters and details.

Modules	Hyper parameter Setting and Details
Policy model; A3C3-Conv1D	- Data is aggregated to hourly and divided by 24 for normalization.

Modules	Hyper parameter Setting and Details
	- Ten actors, 300 agents in each, with 24 step size each episode due to hourly resolution of datasets. - Learning rate 0.00001 for actor and 0.0001 for critic network. - Discount factor (γ) is 0.01 for advantage estimation and reward discounting. - Tensorflow layer with 10 units, Rectified Linear Unit (ReLU) as activation function for both actor, critic, and communication networks. - The Adaptive Moment Estimation (Adam) optimizer.
Agent's daily trading behavior clustering; DTW	- tslearn, for time series analysis package using machine learning. - Dynamic time warping (DTW) for cluster assignment and barycenter computation into four clusters. - Ten iterations of the k-means algorithm for a single run.
MBRL framework; Multivariate-LSTM	- Data is preprocessed by Min-max normalization. - Two Keras layers with 256 LSTM hidden nodes of each. - The Adam optimizer. - The MSE is evaluated for early stopping during training.

5.2.3. Evaluation

The MB-A3C3 model, which is used to reduce the agent's energy bill, is determined using the reward function. The energy bill, which is the cumulative reward of each episode consisting of 24 steps, is determined by the MB-A3C3 algorithm's action during training to maximize all parameters in A3C3-Conv1D. The innovative approach is then sorely tested to see how well MB-A3C3 performed against the policy model

within the three constraints imposed by the MBRL framework: agent trading behavior clustering, experience gained using the predicted trading quantity, and price.

5.2.4. Experimental results

The overall performance is described first in this section. The rest of the sections show the effects of each proposed module.

5.2.4.1. Overall results

In Table 15, the average community's internal trade, external trade, and net energy bills per day of 8 and 300 households are compared to MARL algorithms.

Table 15. The average community's internal trade, external trade, and net energy bills per day: 8 to 300 households are compared to MARL algorithms.

Algorithm	Internal (kWh) ↑		External (kWh) ↓		Net bills ($) ↓	
	8	300	8	300	8	300
1: Deep Learning based A3C3						
A3C3-FF (baseline)	68.64	241.95	311.86	6,012.49	32.91	738.10
A3C3-Conv	66.32	264.17	310.58	5,956.16	31.18	732.61
A3C3-LSTM	65.58	242.05	313.95	6,114.99	34.59	742.79
2: Model based RL + 3: Agent clustering (one model per cluster)						
MB-A3C3 (LSTM) -Randomly	65.58	272.22	309.07	5,705.82	30.89	730.69
MB-A3C3 (LSTM) -Location-based	66.77	315.89	310.36	5,600.28	31.55	675.18
MB-A3C3 (LSTM) -DTW	72.81	326.51	306.51	5,590.06	29.17	654.95
MB-A3C3 (GRU) -Randomly	67.89	314.44	312.59	5,649.13	33.03	674.10
MB-A3C3 (GRU) -Location-based	66.37	317.43	311.15	5,654.46	32.21	672.61
MB-A3C3 (GRU)-DTW	69.37	321.77	310.71	5,597.87	32.54	672.26
MB-A3C3 (Transformer) -Randomly	69.22	315.68	316.66	5,697.35	33.10	738.86

Algorithm	Internal (kWh) ↑		External (kWh) ↓		Net bills ($) ↓	
	8	300	8	300	8	300
MB-A3C3 (Transformer) -Location-based	69.41	314.78	317.11	5,738.70	32.20	719.60
MB-A3C3 (Transformer) -DTW	70.86	317.80	318.67	5,601.17	32.80	723.73

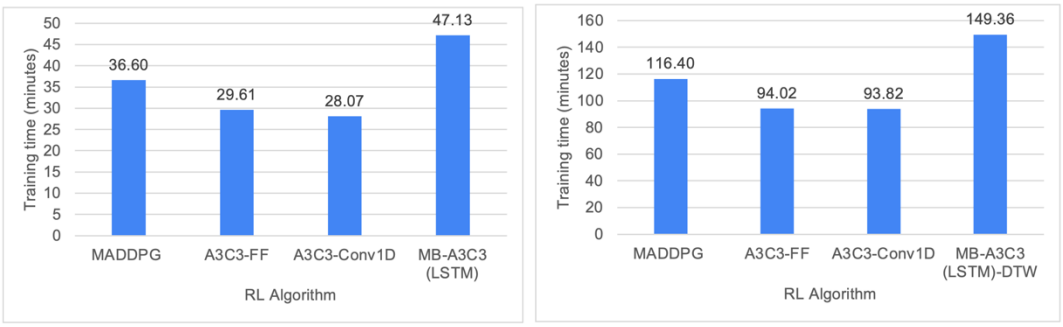
For the multi-agent model, there is one agent per household for the experiment with 8 households (no clustering is applied). For the experiment with 300 households, there is only one agent per cluster. It is assumed that internal trade within communities should increase while external trade directly with the main grid should be reduced by the algorithm.

The baseline MADDPG was extended from 8 to 300 households to ensure the validity of the algorithm. Subsequently, of all the 12 algorithms, the MB-A3C3 (LSTM)-DTW algorithm was found to be the winner ($654.95). As a result, when compared with MADDPG ($789.85), household energy bills are seen to have fallen by more than $100. Energy bills turned out to be 17% lower than trading with the grid ($790.51). At the end of the trading day, the community's net energy bills were greatly reduced via the algorithm. Meanwhile, internal trade increased, and external trade decreased while peak demand for energy dropped from above 600 to 589.26 kW.

As depicted in Figure 37, the training time of the multi-threaded algorithms (A3C3 and MB-A3C3) was compared to the single-threaded (MADDPG) in both 8 (Figure 37a) and extended 300 households (Figure 37b). Although consuming more training time, MB-A3C3 (LSTM)-DTW is efficiently processed by an agent's clustering and environment modeling.

Training time will be 1,767.38 minutes if the number of agents is increased from 8 to 300. (one model per each agent). The result approximated 149.36 minutes when

implemented to the model-based MB-A3C3. Such insights could greatly aid MB-ability A3C3's to support a large number of agents.



(a) 8 Households (b) 300 Households

Figure 37. Training time (min) of each RL algorithms by number of households.

In Figure 38, the community's average energy bill per day for the training set is presented. The reward of the five RL algorithms' convergence during the training phrase is depicted to illustrate the superior performance of MB-A3C3 (LSTM)-DTW over other algorithms; providing faster convergence and lower energy bills. When trading within a community, the algorithm is optimized under certain constraints and environments. An agent's energy bill is reduced by having a price incentive scheme in the algorithm. It is seen that the reward tends to be lower as agents have no knowledge or experience of how to trade during the first stage. After the training phase, the optimized network parameters, which result from multi-threaded mechanisms, deep learning networks, agents' clustering, and environmental models, efficiently lower the community's energy bills, as shown by the green line in the graph.

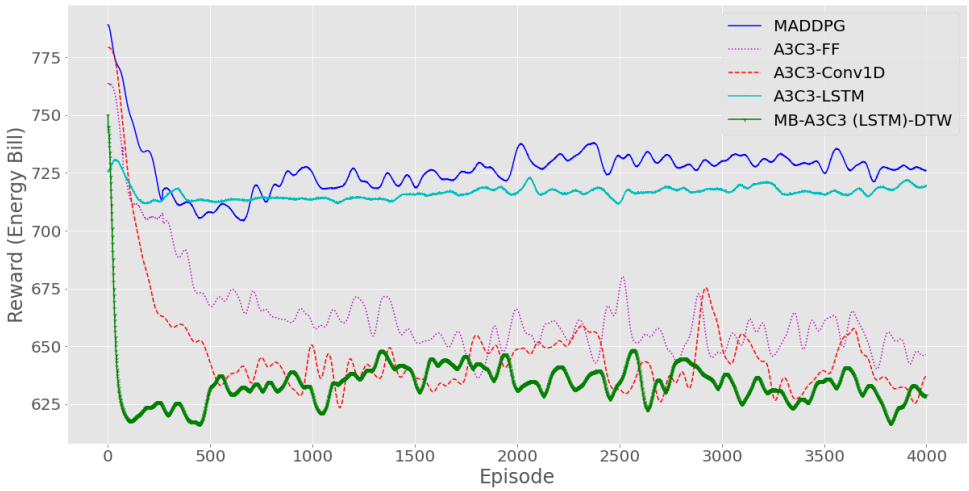


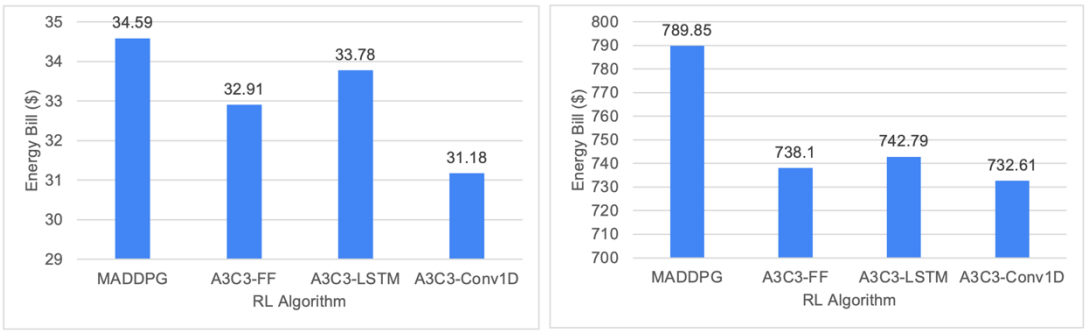
Figure 38. The community's energy bill per day over 4,000 episodes of training; a lower bill is preferred.

It is proved that less external trade and more internal trade can contribute in lower energy bills. When trading within a community, the algorithm is tuned for trading constraints: purchasing less and selling more at local prices rather than grid prices. An agent's energy bill is reduced as profits and expenses increase.

5.2.4.2. Effect of multithreaded and deep learning in policy model

In this section, we aim to show that A3C3-FF outperforms the baseline (MADDPG). Also, the performance can be further improved by applying deep learning techniques (Conv1D) rather than the feed-forward architecture (FF).

From Table 15, it is noted that the performance of the A3C3-Conv1D model is superior to that of the single-threaded MADDPG, having a 9.86% (from 34.59 to 31.18) and 7.25% (from 789.85 to 732.61) of energy bill reduction in 8 and 300 households, respectively. Moreover, by comparing results with different network architectures in Figure 39, the A3C3-Conv1D algorithm outperforms A3C3-FF and A3C3-LSTM, revealing the lowest energy bills in both 8 (Figure 38a) and extended 300 households (Figure 38b).



(a) 8 Households (b) 300 Households

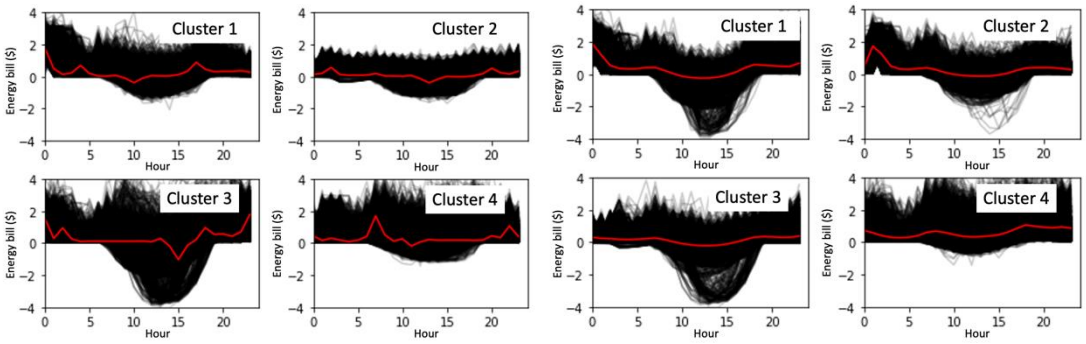
Figure 39. Community's net energy bills of MARL algorithms by number of households.

Because LSTM is typically used to analyse given sequences of data, CNN outperforms LSTM when a policy model evaluates the correlation between observations in a limited timestep to take appropriate action. CNN is designed to leverage "spatial correlation" in data.

5.2.4.3. Effect of agent's trading behavior time series clustering

Figure 39 compares 300 agents' clustering methods: DTW and Euclidean. As projected in Figure 40, DTW is denser than the Euclidean method having a more different centroid (red line) and superior silhouette score of 0.23, while the Euclidean has a score of 0.17.

As shown in Figure 41, three clustering methods were inspected: 1) randomly selected from eight households, 2) location-based, and 3) DTW to observe the results affected by clustering techniques in MB-A3C3 (LSTM).



(a) DTW (b) Euclidean

Figure 40. Agent's daily trading behavior clustering results of four clusters; the red line is the centroid of each cluster.

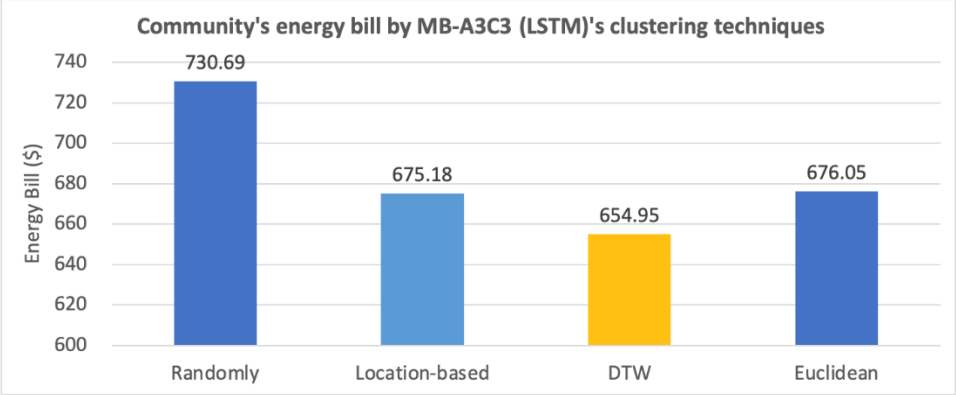


Figure 41. Community's energy bill of each forecasting and clustering techniques in MB-A3C3 (LSTM) from 300 households.

The algorithm utilizes forecasted values from centralized data from the clustering model as one of the states for the policy model. After being optimized during training, the policy will act due to its experience considering other factors. If incorrect categorization occurs in clustering, forecasted values will not be as accurate as before because it expects to get accurate forecasting values based on behavior.

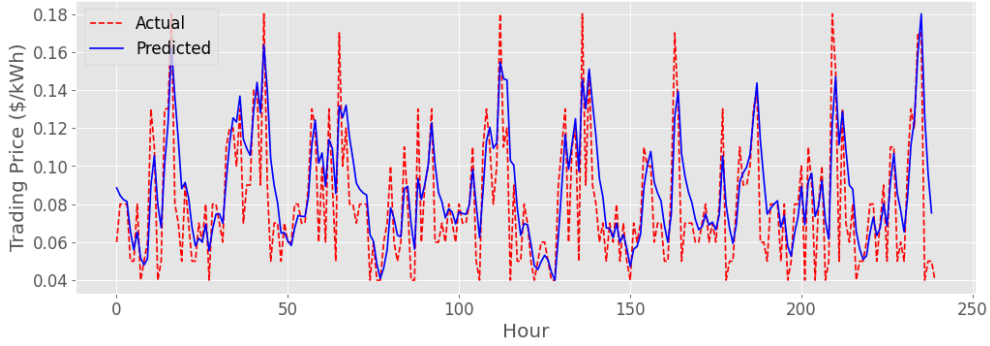
5.2.4.4. Effect of forecasting models in MB-MADRL framework

In Table 16, it is seen that the multivariate-LSTM provides superior root mean square error (RMSE) on testing sets over GRU and the transformer in both trading price and quantity forecasting model.

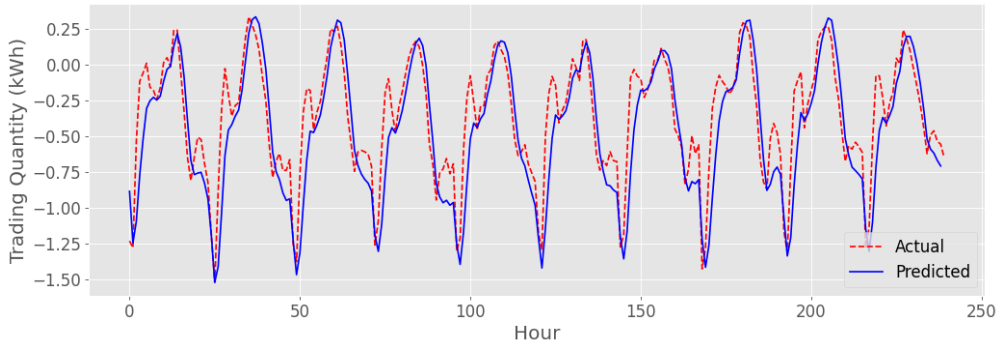
Table 16. The performance of forecasting models in terms of RMSE (lower is preferred). There are two measures: predicted trading price and predicted trading quantity. Boldface refers to the winner.

Method	RMSE of predicted price	RMSE of predicted quantity
Multivariate-LSTM	0.0344	0.0263
GRU	0.0582	0.0379
Transformer	0.0412	0.0290

In Figure 42, trading price and quantity forecasting results are depicted. According to the accurate forecast, the policy model can learn to act and minimize energy bills more efficiently. As illustrated in Table 14, MB-A3C3 (LSTM)-DTW outperforms other algorithms by providing higher internal trade, lower external trade, and reduced community energy bills in both 8 and 300 households.



(a) The actual and predicted trading price.



(b) The actual and predicted trading quantity.

Figure 42. Results are shown for actual and predicted trading price and quantity using the multivariate-LSTM model for the first 240 timesteps of the testing set.

5.2.4.5. Effect of MB-A3C3 in each households

Another benefit of P2P energy trading in communities is that each participant's net energy bill is reduced as each household attempts to trade their energy with others. In Table 17, each households' average energy bills per day from P2P energy trading compared to directly grid trading is illustrated.

Table 17. The average energy bills per day per household of MB-A3C3 compared to grid.

Algorithm	Internal trade (kWh)	External trade (kWh)	Net bills ($)		
			Min.	Max.	Average
Grid	-	26.50	-0.02	0.33	2.64
MB-A3C3 (LSTM)-DTW	1.09	18.63	-0.02	0.29	2.18 (17% reduction)

In Figure 43, the histogram of each household is shown to indicate the average energy net bills. Most households have successfully reduced their energy bills.

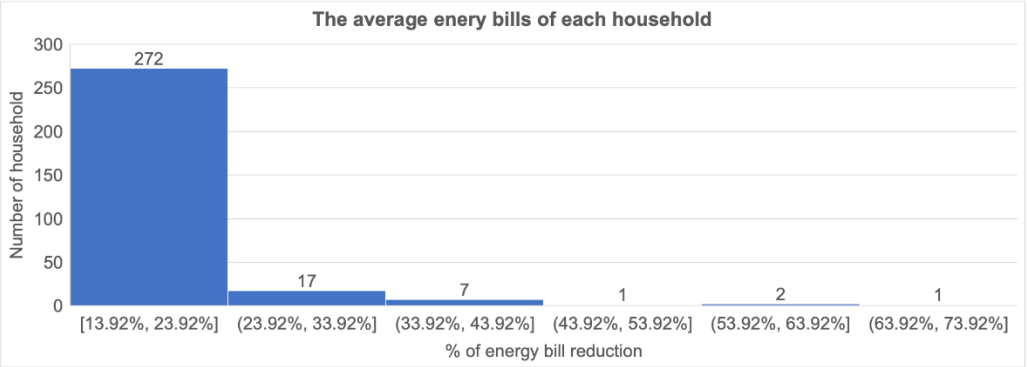


Figure 43. Histogram of the average energy bills of each households trading their energy with MB-A3C3.

5.2.4.6. Effect of component removal test

There are three main contributions for utilizing the novel MB-A3C3 in P2P energy trading:

d. Multi-agent deep reinforcement learning (MADRL): This technique can effectively handle data complexity and many agents.
e. Agent's daily trading behavior clustering: Having insignificant consideration on prosumers' trading behavior in previous research, this research problem is addressed by classifying prosumers into clusters based on their daily trading habits.
f. Model-based framework: MADRL has been enhanced with the model-based concept called "MB-MADRL". It aims to manipulate the lack of local knowledge by allowing agents to generate a model of their environments.

Figure 44. indicate the community's net energy bill of 8 and 300 households having component removal to assess the importance of each following contributions.

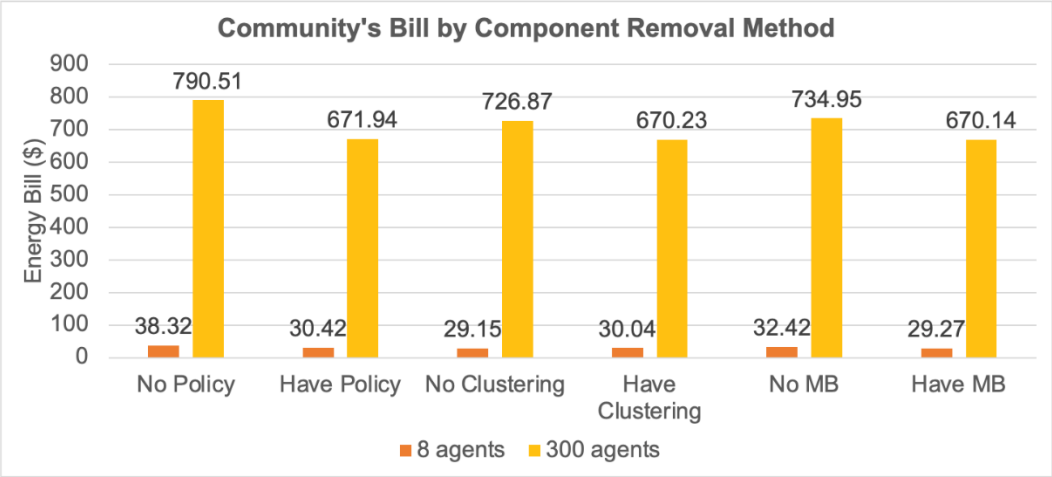


Figure 44. The community's net energy bill of 8 and 300 households for component removal analysis.

The cost reduction of each component removal method is depicted in Table 18. Policy module is the most significant component in both 8 and 300 agents, following with model-based framework and agents' clustering, respectively. If we remove policy module, it means no trading occurs. Agents trading their energy to main grid with ToU and FiT pricing. The cost has not been dropped; however, it is quite close since there is no clustering for 8 agents in agents' clustering test. For MBRL framework removal test, we investigate by randomly select trading previous day value in 300 agents. The cost is also reduced in both.

Table 18. The effect of component removal test by community's net energy bill and percent of cost reduction of each.

Method Test	Community's net bill ($)		Cost Reduction	
	8 agents	300 agents	8 agents	300 agents
No Policy	38.32	790.51		
Have Policy	30.42	671.94	-20.62%	-15.00%
No Clustering	29.15	726.87		

Method Test	Community's net bill ($)		Cost Reduction	
	8 agents	300 agents	8 agents	300 agents
Have Clustering	30.04	670.23	3.05%	-7.79%
No MB	32.42	734.95		
Have MB	29.27	670.14	-9.72%	-8.82%

5.3. Discussion

For the implementation of the real-world P2P trading environment, there are issues found interesting for further discussion.

5.3.1. MBRL with forecasting model

As investigated in Chapter 4.2, the MBRL framework begins by collecting environmental data and training the model to forecast. It is a requirement for MBRL that the forecasting model be accurate to ensure precise information for agents. The algorithm must be able to utilize the productive information to optimize the reward for the community's energy bill.

5.3.2. Number of k in clustering method

The clustering method was introduced to reduce the number of forecasting models (one model per cluster), assuming that homes in the same cluster behave similarly. Since it is quite costly to develop a forecasting model separately for each household (a total of 300 households), three clustering techniques were tested to determine the winner: random matching, location-based clustering, and k-means (DTW) clustering.

The results of clustering depend on the number of clusters (k); bias-variance trade-off determines the cluster number. If overfitting is taken into consideration, a large cluster will produce a tiny bias while a small number of clusters will produce a

minor variation (sometimes favorable for generalization or interpretation) and is typically great for prediction. In Figure 45, the community's energy bill and peak demand for 300 households diverge between k = 4 and 7; between k = 8 and 10, a tight race begins. It is projected that if k is increased to 300, the result will remain the same while requiring a significant amount of computational resources. It is significant that the winner of the selected number of clusters (k = 4) exhibits the lowest energy bill and peak demand.

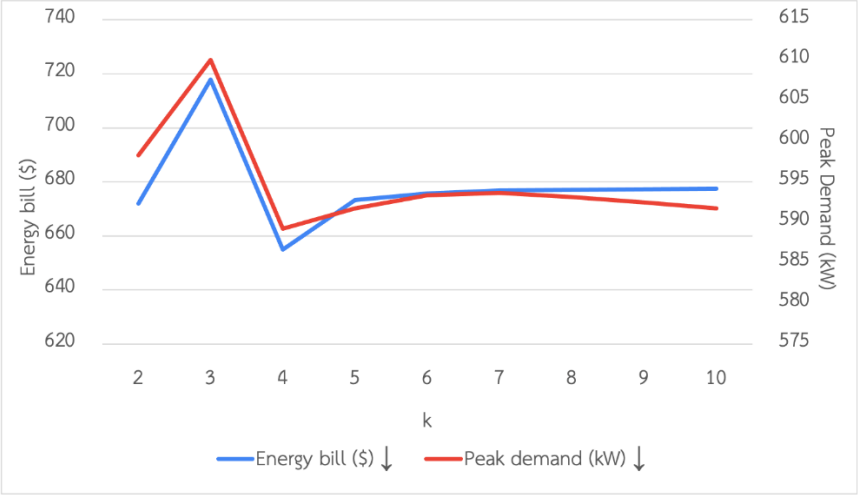


Figure 45. The inspection of energy bill and peak demand from k = 2 to 10 using the winner's clustering method (k-means (DTW)).

5.3.3. The variety of each households' trading method

It is thought to influence the model if some houses use their own method, such as ZI. These behaviors are then collected and trained in a recurrent manner to increase the experience for more suggestions. Trading behavioral information via different methods, such as buying without trust, at will, or based on experience, will make the model easier to categorize. Thus, the model provides more accurate forecasting.

5.3.4. The scenario of P2P energy trading

The manager-based energy market allows it to trade energy both inside the community and, if necessary, with other communities, but this market structure

requires gathering data on each market participant to create and resolve a centralized optimization problem. P2P energy trade, on the other hand, might be an excellent design for a society that is self-sufficient in terms of energy because it is decentralized, adaptable, and privacy-preserving. Meanwhile, P2P energy trading can ensure the network's security through ex post security verification.

For the purposes of our investigation, the algorithm functions as a centralized system, gathering data from each agent, filtering out the house ID, and then modeling the trading information for each household. After obtaining those forecasting values, each household, which is considered to have no information about the source of the data, continues trading their energy to reduce their energy bills. The scenario is centralized-decentralized, combining the benefits of both a manager-based and a P2P energy market.

5.3.5. The seasonality in time-series data

Seasonality is a characteristic of a time series in which the data flows through predictable and recurring changes on a yearly basis. Seasonal refers to any predictable variation or pattern that recurs or repeats over the course of a year. In time-series forecasting, one of the key frontiers in deep learning is the attention mechanism, which was created to enhance performance on longer input sequences. The fundamental goal is to enable the decoder to access encoder data selectively when decoding. This is accomplished by creating a unique context vector for each time step of the decoder, computing it in function of the most recent hidden state and of all the hidden states of the encoder, and assigning trainable weights to each of them.

In this way, the attention mechanism provides the various input sequence components with varying degrees of priority while paying closer attention to the inputs that are more pertinent. The model's name is explained by this). The attention weights also provide the attention mechanism the advantage of being simpler to understand than other deep learning models, which are frequently referred to as "black boxes" because they lack the ability to explain their results.

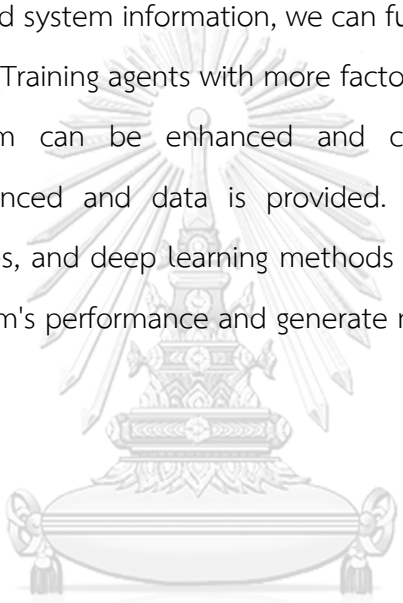
CHAPTER VI

CONCLUSION

In this dissertation, we propose a novel model-based deep reinforcement learning for two scales of energy markets: wholesale and retail. The model-based deep RL algorithm MB-A3C contributes significantly to wind energy strategic bidding in the wholesale energy market. Through (i) the attention-LSTM, (ii) Conv-A3C, and (iii) the MBRL framework, the MB-A3C utilizes a combination of a time-series forecasting model with advanced machine learning approaches. The superiority of MB-A3C over other RL algorithms was conclusively demonstrated by the reduction in average costs per day of bidding, especially in the five scenarios of the well-known Nord Pool datasets: Denmark and Sweden. The five scenarios having different adjustments of price ratios for the cost of reserve and dispatch energy were carried out to investigate the potential of MB-A3C. It is significant that MB-A3C outperformed all baselines and performed closely to the upper bound. Such a model is found to provide the lowest average cost per day, which maximizes WPP's revenue. For P2P energy trading in the retail energy sector, a model-based multi-agent deep reinforcement learning algorithm called MB-A3C3 is presented. Firstly, the baseline A3C3 was enhanced by using the 1D convolutional network. Secondly, RL can support a large number of households (agents) by clustering those houses based on their trading behaviors using dynamic time warping (DTW). Thirdly, the environment was forecasted using multivariate LSTM; this is called model-based RL. Besides, both the multivariate-LSTM and CNN network are seen to improve multi-agent deep reinforcement learning. For large-scale households, the time-series clustering strategy based on trading behavior was utilized as an agent-based model. The experiment was conducted on the Ausgrid data set based on 300 households in NSW, Australia. Results demonstrate that our MB-A3C3, being less time-consuming and less complex, proved to be superior to other RL

algorithms, producing costs 17% lower than traditional grid trading. It is significant that MB-A3C3 leveraged internal trading between households, thereby decreasing external trading under the grid's price incentives and constraints. Herein, the algorithms are seen to potentially aid in reducing customers' electricity bills.

Further research must investigate various regulations to embrace more real-world scenarios of electricity consumers, producers, and power system operators to create more opportunities for energy trading. Moreover, by adding other related factors, e.g., weather and system information, we can further improve the approach to make it more accurate. Training agents with more factors can provide more optimized policies. The algorithm can be enhanced and customized if energy storage components are enhanced and data is provided. More appropriate algorithms, optimization approaches, and deep learning methods will be included and designed to improve the algorithm's performance and generate more realistic results.



REFERENCES

1. Nagbe, K., J. Cugliari, and J. Jacques, *Short-Term Electricity Demand Forecasting Using a Functional State Space Model.* Energies, 2018. **1**.
2. Lisi, F. and I. Shah, *Forecasting next-day electricity demand and prices based on functional models.* Energy Systems, 2020. **11**.
3. Shah, I., et al., *Short-Term Electricity Demand Forecasting Using Components Estimation Technique.* Energies, 2019.
4. Weron, R., *Electricity price forecasting: A review of the state-of-the-art with a look into the future.* International Journal of Forecasting, 2014. **30**.
5. Kristiansen, T., *Forecasting Nord Pool day-ahead prices with an autoregressive model.* Energy Policy, 2012. **49**: p. 328–332.
6. Mulaosmanovic, M. and E. Ali. *SHORT-TERM ELECTRICITY PRICE FORECASTING ON THE NORD POOL MARKET*. 2017.
7. Roungkvist, J., P. Enevoldsen, and G. Xydis, *High-Resolution Electricity Spot Price Forecast for the Danish Power Market.* Sustainability, 2020. **12**: p. 4267.
8. Nowotarski, J. and R. Weron, *Recent advances in electricity price forecasting: A review of probabilistic forecasting.* Renewable and Sustainable Energy Reviews, 2017. **81**.
9. Panapakidis, I. and A. Dagoumas, *Day-ahead electricity price forecasting via the application of artificial neural network based models.* Applied Energy, 2016. **172**: p. 132-151.
10. Kolberg, J.K. and K. Waage. *Artificial Intelligence and Nord Pool s intraday electricity market Elbas : a demonstration and pragmatic evaluation of employing deep learning for price prediction : using extensive market data and spatio-temporal weather forecasts*. 2018.
11. Chinnathambi, R., et al., *A Multi-Stage Price Forecasting Model for Day-Ahead Electricity Markets.* Forecasting, 2018. **1**: p. 3.
12. Karabiber, O. and G. Xydis, *Electricity price forecasting in the Danish day-ahead market using the TBATS, ANN and ARIMA methods.* Energies, 2019. **12**: p. 928.

13. Rantonen, M. and J. Korpihalkola, *Prediction of Spot Prices in Nord Pool's Day-Ahead Market Using Machine Learning and Deep Learning*. 2020. p. 676-687.
14. Foster, J., X. Liu, and S. McLoone, *Load forecasting techniques for power systems with high levels of unmetered renewable generation: A comparative study.* IFAC-PapersOnLine, 2018. **51**: p. 109-114.
15. Yan, K., et al., *Short-Term Solar Irradiance Forecasting Based on a Hybrid Deep Learning Methodology.* Information, 2020. **11**: p. 32.
16. Li, P., X. Wang, and J. Yang, *Short-term Wind Power Forecasting Based on Two-stage Attention Mechanism.* IET Renewable Power Generation, 2020. **14**.
17. Yang, J.J., et al., *A deep reinforcement learning method for managing wind farm uncertainties through energy storage system control and external reserve purchasing.* International Journal of Electrical Power & Energy Systems, 2020. **119**: p. 105928.
18. Zhang, F. and Q. Yang, *Energy Trading in Smart Grid: A Deep Reinforcement Learning-based Approach*. 2020. 3677-3682.
19. Jia, S., et al., *A Deep Reinforcement Learning Bidding Algorithm on Electricity Market.* Journal of Thermal Science, 2020. **29**.
20. Zhang, G., et al., *A data-driven approach for designing STATCOM additional damping controller for wind farms.* International Journal of Electrical Power & Energy Systems, 2020. **117**: p. 105620.
21. Liu, Y., et al., *Deep Reinforcement Learning Approach for Autonomous Agents in Consumer-centric Electricity Market*. 2020. 37-41.
22. Kim, J.-G. and B. Lee, *Automatic P2P Energy Trading Model Based on Reinforcement Learning Using Long Short-Term Delayed Reward.* Energies, 2020. **13**: p. 5359.
23. Longoria, G., A. Davy, and L. Shi, *Subsidy-Free Renewable Energy Trading: A Meta Agent Approach.* IEEE Transactions on Sustainable Energy, 2019. **PP**: p. 1-1.
24. Zhou, S., et al., *Combined heat and power system intelligent economic dispatch: A deep reinforcement learning approach.* International Journal of Electrical Power & Energy Systems, 2020. **120**: p. 106016.
25. Mnih, V., et al., *Asynchronous Methods for Deep Reinforcement Learning.* 2016.

26. Cao, D., et al., *Bidding strategy for trading wind energy and purchasing reserve of wind power producer – A DRL based approach.* International Journal of Electrical Power & Energy Systems, 2020. **117**: p. 105648.
27. Lin, L., et al., *Deep Reinforcement Learning for Economic Dispatch of Virtual Power Plant in Internet of Energy.* IEEE Internet of Things Journal, 2020. **PP**: p. 1-1.
28. Guan, J., et al., *A parallel multi-scenario learning method for near-real-time power dispatch optimization.* Energy, 2020. **202**: p. 117708.
29. Sutton, R., *Dyna, an integrated architecture for learning, planning, and reacting.* ACM SIGART Bulletin, 1995. **2**.
30. Wang, X. and T. Dietterich, *Model-based Policy Gradient Reinforcement Learning*. 2003. 776-783.
31. Pong, V., et al., *Temporal Difference Models: Model-Free Deep RL for Model-Based Control.* 2018.
32. Xu, H., et al., *Algorithmic Framework for Model-based Reinforcement Learning with Theoretical Guarantees*. 2018.
33. Kostmann, M. and W. Härdle, *Forecasting in Blockchain-Based Local Energy Markets.* Energies, 2019. **12**: p. 2718.
34. Tushar, W., et al., *Peer-to-Peer Trading in Electricity Networks: An Overview.* IEEE Transactions on Smart Grid, 2020. **PP**: p. 15.
35. Hayes, B., S. Thakur, and J. Breslin, *Co-simulation of Electricity Distribution Networks and Peer to Peer Energy Trading Platforms.* International Journal of Electrical Power & Energy Systems, 2019. **115**.
36. Vithanage, V., et al., *A review on Multi-Agent system based energy management systems for micro grids.* AIMS Energy, 2019. **7**: p. 924-943.
37. Xu, Y., et al., *Deep Reinforcement Learning and Blockchain for Peer-to-Peer Energy Trading among Microgrids*. 2020. 360-365.
38. Gao, G., Y. Wen, and D. Tao, *Distributed Energy Trading and Scheduling Among Microgrids via Multiagent Reinforcement Learning.* IEEE Transactions on Neural Networks and Learning Systems, 2022: p. 1-15.

39. Ghasemi, A., et al., *A Multi-Agent Deep Reinforcement Learning Approach for a Distributed Energy Marketplace in Smart Grids*. 2020. 1-6.
40. Xiaohan, F., et al., *Multi-Agent Reinforcement Learning Approach for Residential Microgrid Energy Scheduling.* Energies, 2019. **13**: p. 123.
41. Christensen, M., C. Ernewein, and P. Pinson, *Demand Response through Price-setting Multi-agent Reinforcement Learning*. 2020.
42. Munir, M., et al., *A Multi-Agent System Toward the Green Edge Computing with Microgrid*. 2019.
43. Nguyen, T., N.D. Nguyen, and S. Nahavandi, *Multi-Agent Deep Reinforcement Learning with Human Strategies*. 2019. 1357-1362.
44. Simoes, D., N. Lau, and L. Reis, *Multi Agent Deep Learning with Cooperative Communication.* Journal of Artificial Intelligence and Soft Computing Research, 2020. **10**: p. 189-207.
45. Christianos, F., L. Schäfer, and S. Albrecht, *Shared Experience Actor-Critic for Multi-Agent Reinforcement Learning*. 2020.
46. Nguyen, N.D., et al., *A Visual Communication Map for Multi-Agent Deep Reinforcement Learning*. 2020.
47. Yang, J., et al., *Deep Mean Field Games for Learning Optimal Behavior Policy of Large Populations.* 2017.
48. Yang, Y., et al., *Mean Field Multi-Agent Reinforcement Learning*, in *Proceedings of the 35th International Conference on Machine Learning*, D. Jennifer and K. Andreas, Editors. 2018, PMLR: Proceedings of Machine Learning Research. p. 5571--5580.
49. Zhou, S., et al., *Multi-Agent Mean Field Predict Reinforcement Learning*. 2020. 625-629.
50. Wang, B., J. Xie, and N. Atanasov, *Coding for Distributed Multi-Agent Reinforcement Learning*. 2021.
51. Canese, L., et al., *Multi-Agent Reinforcement Learning: A Review of Challenges and Applications.* Applied Sciences, 2021. **11**: p. 4948.
52. Witt, C., et al., *Deep Multi-Agent Reinforcement Learning for Decentralized Continuous Cooperative Control*. 2020.

53. Khorasany, M., *Market Design for Peer-to-Peer Energy Trading in a Distribution Network with High Penetration of Distributed Energy Resources*. 2019.
54. Cui, J., et al., *Optimal Electricity Allocation Model Under China's Planning-Market Double-Track Mechanism Considering Bidding Game of Generation Companies.* Frontiers in Energy Research, 2021. **9**.
55. Bellman, R., *A Markovian Decision Process.* Journal of Mathematics and Mechanics, 1957. **6**(5): p. 679-684.
56. Telser, L., *Dynamic Programming and Markov Processes Ronald A. Howard.* Journal of Political Economy, 1961. **69**: p. 296-297.
57. Bertsekas, D., *Dynamic Programming and Optimal Control*. 1995.
58. Sutton, R. and A. Barto, *Reinforcement Learning: An Introduction.* IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council, 1998. **9**: p. 1054.
59. Hazeghi, K. and M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Journal of the American Statistical Association, 1995. **90**: p. 392.
60. Soares, T., P. Pinson, and H. Morais, *Wind offering in energy and reserve markets.* Journal of Physics: Conference Series, 2016. **749**: p. 012021.
61. Otterlo, M. and M. Wiering, *Reinforcement Learning and Markov Decision Processes.* Reinforcement Learning: State of the Art, 2012: p. 3-42.
62. Bradford, A., *Reinforcement Learning: An Introduction Richard S. Sutton and Andrew G. Barto.* 2021.
63. Khorasany, M., Y. Mishra, and G. Ledwich, *Market Framework for Local Energy Trading: A Review of Potential Designs and Market Clearing Approaches.* IET Generation Transmission & Distribution, 2018. **12**: p. 5899 – 5908.
64. Soto, E.A., et al., *Peer-to-peer energy trading: A review of the literature.* Applied Energy, 2021. **283**: p. 116268.
65. Alam, M.R., M. St-Hilaire, and T. Kunz, *Peer-to-peer energy trading among smart homes.* Applied Energy, 2019. **238**: p. 1434-1443.
66. Aitzhan, N. and D. Svetinovic, *Security and Privacy in Decentralized Energy Trading Through Multi-Signatures, Blockchain and Anonymous Messaging*

Streams. IEEE Transactions on Dependable and Secure Computing, 2016. **PP**: p. 1-1.

67. Guerrero, J., et al., *Towards a transactive energy system for integration of distributed energy resources: Home energy management, distributed optimal power flow, and peer-to-peer energy trading.* Renewable and Sustainable Energy Reviews, 2020. **132**: p. 27.
68. Friedman, D. and J. Rust, *The Double Auction Market: Institutions, Theories and Evidence.* 1993. **14**.
69. Qiu, D., et al., *Multi-Agent Reinforcement Learning for Automated Peer-to-Peer Energy Trading in Double-Side Auction Market*. 2021. 2913-2920.
70. Alabdullatif, A., E. Gerding, and A. Perez-Diaz, *Market Design and Trading Strategies for Community Energy Markets with Storage and Renewable Supply.* Energies, 2020. **13**: p. 972.
71. Wooldridge, M., *An Introduction to MultiAgent Systems / M.J. Wooldridge.* 2022.
72. Zhang, D., X. Han, and C. Deng, *Review on the research and practice of deep learning and reinforcement learning in smart grids.* CSEE Journal of Power and Energy Systems, 2018. **4**: p. 362-370.
73. Quan, W., et al., *Comparative Study of CNN and LSTM based Attention Neural Networks for Aspect-Level Opinion Mining*. 2018. 2141-2150.
74. Vaswani, A., et al., *Attention Is All You Need.* 2017.
75. Jing, R., *A Self-attention Based LSTM Network for Text Classification.* Journal of Physics: Conference Series, 2019. **1207**: p. 012008.
76. Wang, Y., et al., *Attention-based LSTM for Aspect-level Sentiment Classification*. 2016. 606-615.
77. Janner, M., et al., *When to Trust Your Model: Model-Based Policy Optimization*. 2019.
78. Lu, Y. and K. Yan, *Algorithms in Multi-Agent Systems: A Holistic Perspective from Reinforcement Learning and Game Theory*. 2020.
79. Nowe, A., P. Vrancx, and Y.-M. De Hauwere, *Game Theory and Multi-agent Reinforcement Learning*. 2012. p. 30.

80. Juliani, A., *Simple Reinforcement Learning with Tensorflow Part 8: Asynchronous Actor-Critic Agents (A3C)*, in *Emergent // Future*. 2017.
81. Heess, N., et al., *Emergence of Locomotion Behaviours in Rich Environments.* 2017.
82. Perera, A.T.D. and P. Kamalaruban, *Applications of reinforcement learning in energy systems.* Renewable and Sustainable Energy Reviews, 2021. **137**: p. 110618.
83. An, Z., et al., *High dimensional quantum optimal control with Reinforcement Learning*. 2020.
84. Anfu, G., et al., *An Autonomous Path Planning Model for Unmanned Ships Based on Deep Reinforcement Learning.* Sensors, 2020. **20**: p. 426.
85. Lowe, R., et al., *Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments.* 2017.
86. Charbonnier, F., T. Morstyn, and M.D. McCulloch, *Scalable multi-agent reinforcement learning for distributed control of residential energy flexibility.* Applied Energy, 2022. **314**.
87. Shih, S.-Y., F.-K. Sun, and H.-y. Lee, *Temporal pattern attention for multivariate time series forecasting.* Machine Learning, 2019. **108**.
88. Zhang, X., et al., *AT-LSTM: An Attention-based LSTM Model for Financial Time Series Prediction.* IOP Conference Series: Materials Science and Engineering, 2019. **569**: p. 052037.
89. Abbasimehr, H. and R. Paki, *Improving time series forecasting using LSTM and attention models.* Journal of Ambient Intelligence and Humanized Computing, 2022. **13**: p. 1-19.
90. Qiu, D., et al., *Scalable coordinated management of peer-to-peer energy trading: A multi-cluster deep reinforcement learning approach.* Applied Energy, 2021. **292**: p. 116940.
91. Chabchoub, Y. and C. Fricker, *Classification of the vélib stations using Kmeans, Dynamic Time Wraping and DBA averaging method.* 2014 International Workshop on Computational Intelligence for Multimedia Understanding, IWCIM 2014, 2015.

92. Khalid, K. and R. Sudirman, *Dynamic Time Wraping, Speech: Current Features & Extraction Methods*. 2008.

93. Gold, O. and M. Sharir, *Dynamic Time Warping and Geometric Edit Distance: Breaking the Quadratic Barrier.* ACM Transactions on Algorithms, 2018. **14**: p. 1-17.

94. Javed, A., B. Lee, and D. Rizzo, *A Benchmark Study on Time Series Clustering*. 2020.

95. Thorndike, R., *Who belong in the family?* Psychometrika, 1953. **18**: p. 267-276.

96. Rousseeuw, P., *Silhouettes: A graphical aid to the interpretation and validation of cluster analysis.* Journal of Computational and Applied Mathematics, 1987. **20**: p. 53-65.

97. Alcaraz, R., F. Hornero, and J.J. Rieta, *Dynamic time warping applied to estimate atrial fibrillation temporal organization from the surface electrocardiogram.* Medical Engineering & Physics, 2013. **35**(9): p. 1341-1348.

98. Aghabozorgi, Sr., A.S. Shirkhorshidi, and T. Wah, *Time-series clustering - A decade review.* Information Systems, 2015. **53**.

99. *See market data for all areas*. Available from: https://www.nordpoolgroup.com/Market-data1/#/nordic/table.

VITA

NAME	Manassakan Sanayha
DATE OF BIRTH	5 August 1989
PLACE OF BIRTH	Bangkok, Thailand
INSTITUTIONS ATTENDED	Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University
HOME ADDRESS	101/2 Village no. 5, Bang Kruai-Sai Noi 9 Alley, Bang Kruai-Sai Noi Rd., Bang Kruai Subdistrict, Bang Kruai District, Nonthaburi 11130
PUBLICATION	M. Sanayha and P. Vateekul, "Model-Based Approach on Multi-Agent Deep Reinforcement Learning with Multiple Clusters for Peer-To-Peer Energy Trading," in IEEE Access, 2022, doi: 10.1109/ACCESS.2022.3224460.
	Sanayha, Manassakan & Vateekul, Peerapon. (2022). Model-based deep reinforcement learning for wind energy bidding. International Journal of Electrical Power & Energy Systems. 136. 107625. 10.1016/j.ijepes.2021.107625.
	Sanayha, Manassakan & Vateekul, Peerapon. (2019). Remaining Useful Life Prediction Using Enhanced Convolutional Neural Network on Multivariate Time Series Sensor Data. Walailak Journal of Science and Technology (WJST). 16. 669-679. 10.48048/wjst.2019.4144.
	Sanayha, Manassakan & Vateekul, Peerapon. (2017). Fault detection for circulating water pump using time series

forecasting and outlier detection. 193-198. 10.1109/KST.2017.7886095.