2022-12-14

Application of Internet of Energy in Smart Grids Using Deep Reinforcement Learning

Mohammadi Rouzbahani, Hossein

UNIVERSITY OF CALGARY

Application of Internet of Energy in Smart Grids Using Deep Reinforcement Learning

by

Hossein Mohammadi Rouzbahani

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN ELECTRICAL AND COMPUTER ENGINEERING

CALGARY, ALBERTA

DECEMBER, 2022

Abstract

One of the principal challenges with Smart Grids is the very slow rate of development originating from the lack of investments from the governments and major companies. As a solution, the concept of the Internet of Energy (IoE) is capable of differing the need for massive investments and changing the business model of energy sharing so that end-users participate in the development process. The IoE envisions the next generation of smart grids as a fully interconnected network, including advanced metering infrastructures, distributed energy resources, and bidirectional communication systems.

The first key question is how to convince end-users to participate and invest in upgrading the current power system. The feasibility of any possible solution is linked with the profitability of the whole process by reducing the electricity cost and maximizing the profit of energy trading. Consequently, optimizing operational scheduling and electricity routing are two fundamental problems that need to be addressed. However, the accuracy and originality of data must be guaranteed prior to utilizing it in solving scheduling and routing problems.

The open architecture of the IoE-based smart grid results in manifold security concerns, especially the risk of False Data Injection attacks. The attack may target the technical aspects of a system since fabricating the network's data misleads power scheduling and routing strategies and interrupts the healthy operation of the power system. Also, the high penetration of smart devices in IoE-enabled smart grids, besides decentralization originating from employing renewable resources, faces the power system with intricate optimization problems, including operational scheduling and electricity routing problems. Accordingly, this thesis is on the application of the Internet of Energy in smart grids using Deep Reinforcement Learning, aiming to reduce costs and losses for both generator and consuming sides, considering the correctness of data in the system.

The first objective of this research is to enhance the cyber defense of the Internet of Energy-enabled power systems against False Data Injection attacks. To this end, an intelligent intruder is first developed to generate innovative threats that the model has not previously seen. Moreover, well-known attack strategies are modeled to create passive attacks simultaneously. Next, the quality of the developed attack is examined using the proposed defense algorithm in the literature to demonstrate the necessity of a more powerful attack detection mechanism. Then, a Multi-Layer cyber defense mechanism is developed to detect both passive and active threats.

After guaranteeing the originality and correctness of data, the second objective is optimizing the operational scheduling of all energy components in the system. Accordingly, a novel algorithm named Probabilistic Delayed Double Deep Q-Learning, which is a combination of the tuned version of Double Deep Q-Learning and Delayed Q-Learning has been proposed to optimize energy scheduling problems in IoE-based power systems. This algorithm makes a trade-off between overestimation and underestimation biases, guaranteeing sample complexity by applying a delay in updating the rule.

Finally, to fulfill the last objective, which is optimizing electricity routing, a novel algorithm titled Approximate Reasoning Reward-based Adaptable Deep Double Q-Learning (A2R-ADDQL) is introduced specially to optimize electricity routing in residential units. As a result, both positive and negative biases are reduced compared to other deep Q-Learning-based algorithms. Moreover, the sample complexity of the model is decreased due to utilizing a fuzzy approximate reasoning reward function.

Preface

This manuscript-based Ph.D. thesis contains the results of various studies conducted at the Smart Cyber-Physical (SCPS) Lab at the University of Calgary. The main chapters of this thesis have been published/submitted in top-tier peer-reviewed journals in the field of Electrical and computer engineering. All materials have been reformatted based on the University of Calgary formatting requirements. Here is the list of Journal publications that have been used in this thesis partially or entirely:

1- Hossein Mohammadi Rouzbahani, Hadis Karimipour, Lei Lei, "*A review on virtual power plant for energy management,*" Sustainable Energy Technologies and Assessments, Volume 47, 2021, 101370, ISSN 2213-1388, https://doi.org/10.1016/j.seta.2021.101370.

2- Hossein Mohammadi Rouzbahani, Hadis Karimipour, Lei Lei, " *Optimizing scheduling policy in smart grids using probabilistic Delayed Double Deep Q-Learning (P3DQL) algorithm,*" Sustainable Energy Technologies and Assessments, Volume 53, Part C, 2022, 102712, ISSN 2213-1388, https://doi.org/10.1016/j.seta.2022.102712.

3- Hossein Mohammadi Rouzbahani, Hadis Karimipour, Lei Lei, " *Multi-Layer Defense Algorithm Against Deep Reinforcement Learning-based Intruders in Smart Grids,*" International Journal of Electrical Power and Energy Systems, https://doi.org/10.1016/j.ijepes.2022.108798

4- Hossein Mohammadi Rouzbahani, Hadis Karimipour, Lei Lei, " *Optimizing Resource Swap Functionality in IoE-based Grids Using Approximate Reasoning Reward-based Adjustable Deep Double Q-Learning,*" IEEE Transactions on Consumer Electronics, *(Manuscript under review)*.

Moreover, the following conference papers/book chapters are also outcomes of this thesis that have not been reused directly:

5- H. M. Rouzbahani, H. Karimipour and L. Lei, "*An Ensemble Deep Convolutional Neural Network Model for Electricity Theft Detection in Smart Grids,*" 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2020, pp. 3637-3642, https://doi.org/10.1109/SMC42975.2020.9282837.

6- H. M. Ruzbahani, A. Rahimnejad and H. Karimipour, "*Smart Households Demand Response Management with Micro Grid,*" 2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), 2019, pp. 1-5, https://doi.org/10.1109/ISGT.2019.8791595.

7- Mohammadi Rouzbahani, H., Karimipour, H., Rahimnejad, A., Dehghantanha, A., Srivastava, G. (2020). *"Anomaly Detection in Cyber-Physical Systems Using Machine Learning"*. In: Choo, KK., Dehghantanha, A. (eds) Handbook of Big Data Privacy. Springer, Cham. https://doi.org/10.1007/978-3-030-38557-6_10

8- Rouzbahani, H.M., Bairami, A.H., Karimipour, H. (2021). "*A Snapshot Ensemble Deep Neural Network Model for Attack Detection in Industrial Internet of Things*". In: Karimipour, H., Derakhshan, F. (eds) AI-Enabled Threat Detection and Security Analysis for Industrial IoT . Springer, Cham. https://doi.org/10.1007/978-3-030-76613-9_10

9- Rouzbahani, H.M., Faraji, Z., Amiri-Zarandi, M., Karimipour, H. (2020). "*AI-Enabled Security Monitoring in Smart Cyber Physical Grid*". In: Karimipour, H., Srikantha, P., Farag, H., Wei-Kocsis, J. (eds) Security of Cyber-Physical Systems. Springer, Cham. https://doi.org/10.1007/978-3-030-45541-5_8

10- Mohammadi Rouzbahani, H., Karimipour, H., Srivastava, G. (2020). "Big Data Application for Security of Renewable Energy Resources". Handbook of Big Data Privacy. Springer, Cham. https://doi.org/10.1007/978-3-030-38557-6_11

For all the works reused in this thesis, I was the main author and intellectually responsible for the literature review, idea and concept, dataset selection, analysis, and mathematical modeling, simulation, graphical and tabular results preparation, and the final arrangement. Also, all papers were authored under the supervision of Dr. Hadis Karimipour and Dr. Lei Lei, who were involved throughout the research in forming concepts, identifying the research questions, reviewing the research findings, and editing the manuscripts.

Acknowledgments

Doctoral research is principally a long and highly collaborative process, and this accomplishment would have been inaccessible in the absence of those who supported and encouraged me on this path. There are no proper words to convey my deep gratitude and respect for their efforts.

First and foremost, I would like to express my sincerest gratitude to my esteemed supervisor Dr. Hadis Karimipour, who has patiently and sympathetically supported me throughout this research project. I am extremely grateful for her positive energy, empathetic understanding, and continuous encouragement from the beginning to the end of my Ph.D. study at the University of Calgary.

Additionally, I would like to extend my gratitude to my supervisory committee members, Dr. Lei Lei and Dr. Hamidreza Zareipour, for their priceless comments on enhancing the research at different phases.

Last but definitely not least, I owe the deepest gratitude to my wife, Zahra. Her unconditional trust, timely encouragement, and endless patience helped me get through this arduous period in the most positive way.

Dedicated to:

“Brave Women of Iran”

Table of Contents

List of Tables

List of Figures

Nomenclature

Abbreviation	Definition
GHG	Greenhouse gases
EIA	Energy Information Administration
RES	Renewable Energy Sources
IoE	Internet of Energy
AMI	Advanced Metering Infrastructures
ICT	Information and Communication Technology
EV	Electric Vehicle
ISO	Independent System Operator
ICS-CERT	Industrial Control Systems Cyber Emergency Response Team
FDI	False Data Injection
ER	Energy Router
SE	State Estimation
RL	Reinforcement Learning
SVM	Support Vector Machine
DNN	Deep Neural Network
CNN	Convolutional Neural Network
LSTM	Long-Short-Term Memory
DT	Decision Tree
LP	Linear programming
MILP	Mixed Integer Linear Programming
MINLP	Mixed-integer non-linear programming
GA	Genetic Algorithm
ELPSO	Enhanced Leader Particle Swarm Optimization
CPSO	Cooperative Particle Swarm Optimization
NAA	Natural Aggregation Algorithm
DRL	Deep Reinforcement Learning
DNN	Deep Neural Network

DQL	Deep Q-Learning
DPG	Deep Policy Gradient
DDQL	Double Deep Q-Learning
C-DDPG	Centralized Deterministic DPG
D-DDPG	Distributed Deterministic DPG
NA	Nano Area
NH	Neighborhood
A2R-ADDQL	Approximate Reasoning Reward-based Adaptable Deep Double Q-Learning
TP	True Positive
FP	False Positive
ANN	Artificial Neural Network
DT	Decision Tree
RF	Random Forest
SEDNN	Snapshot Ensemble Deep Neural Network
MA	Micro Area
IF	Isolation Forest
NH	Neighborhoods
WAN	Wide Area Network
PPV	Private PV
SPV	Shared PV
ESS	Electricity Storage System
SoC	State of Charge
G2V	Grid-to-Vehicle
V2G	Vehicle-to-Grid
P2P	Peer-to-Peer
RTP	Real-Time Price
SST	Solid-State Transformers
IEC	International Electrotechnical Commission
ADDQL	Adaptable Deep Double Q-Learning
ToU	Time of Use
CO_2	Carbon dioxide
NO_x	Nitrogen Oxides

Symbols	**Definition**
P^{Act}	Matrix of actual power consumption
P^{False}	Falsified consumption matrix
$T_N^{Snapshot}$	Training time of snapshot
$T_{standard}^{DNN}$	Training time of standard network
$M(r, y_i)$	Marginal distribution
$P(y, r)$	Joint probability distribution
H	Matrix of N nano areas
R	Matrix of shared units
Δ	Activation matrixes for NA
Ψ	Activation matrixes of shared units
δ	Participating status of a micro area
d	Participating status of shared units
Ω	Incentive factor
Φ	Penalty factor
λ	Electricity price
Π	Contract coefficient
θ_1	Lower pricing threshold frame
θ_2	Higher pricing threshold
Φ	A set of NAs joined in the NH
l_t^{wk}	Power loss of from w to k in NH
P_t^{wk}	Transferred power from w to k
Γ_t^*	Exchanged power
ξ_t	Transmission fee coefficient
λ_{ESS}	Efficiency of the ESS
λ_{PV}	Converting efficiency of PV
${}_{out}^{o}C$	Outside temperature
Υ	Contributed WANs in scheduling
l^j	Power consumption of appliance j
E^{App}	State of the appliance
${}_{i}^{NA}E$	State of i^{th} NA

T^{NA}	Current temperature units
$\pm xT^{NA}$	x degree of temperature change
Γ^{EVC}	Electricity price supplied by EVC
$\pi(a_t\|s_t)$	Optimal policy
a_t	Action t
s_t	State t
s'	Next State
γ	Discount factor
R^{total}	Total reward
τ	Agent's trajectory
$\mathrm{E}_{\tau\sim\pi}[.]$	Expected value over τ
r	Reward
φ_i	Penalty of violating criteria i
$\mho$	A set of criteria for shaping reward
ϖ_i^a	Actual quantity of criteria i
ϖ_i^d	Desired quantity of criteria i
α	Learning rate
$Q(s,a)$	Value of action a in state s
Q^A	Q function of approximator A
ζ	Experience buffer
μ	Average reward
$U(-x,+x)$	Uniform distribution
P_i^{sell}	Sold power, from or to i^{th} NH
P_i^{buy}	Bought power, from or to i^{th} NH
P_i^{ex}	Exchanged power with the i^{th} NH
P_i^{sch}	Planned power to be swapped with the i^{th} NH
L_{sw}	Switching loss
T^{op}	Operative temperature in Celsius
R_{DS}	Drain-source resistance

Chapter 1

Introduction and Motivation

1.1. Overview

Electrical energy plays a significant role in economic development and human welfare worldwide [1]. During the past ten years, electricity demand has increased uninterruptedly by an average of 3.1% annually, resulting in growing Greenhouse gases (GHG) emissions and increased need for new energy resources [2]. According to the Energy Information Administration (EIA), 62% of electricity production in the United States is supplied by fossil energies, and 20% of average global growth has happened by these resources [3]. Since fossil fuels are expensive and pollute, Renewable Energy Sources (RESs) are utilized dramatically. Furthermore, under the Paris agreement, many countries are committed to proposing an annual GHG emission reduction plan, which makes utilizing RESs and Electric Vehicles (EVs) indispensable.

High penetration of RESs brings up new challenges due to supply fluctuation, uncertainties imposed by the nature of RESs, and decentralized topology that originates from the wide geographical distribution of energy resources [4]. Moreover, real-time monitoring of energy consumption, generation, and flow, besides dealing with big data generated in the different layers of infrastructures, is vital to enhancing power system performances, including security, reliability [5], and stability [6]. Consequently, conventional energy management systems and cyber defense frameworks are not practical due to transformation in the network's topology and load pattern caused by penetration of new resources, besides utilizing numerous sensors, wireless communication tools, smart appliances, and data acquisition units [7], [8].

As a solution, the Internet of Energy (IoE) can provide an intelligent and secure energy management framework that facilitates the accommodation and coordination of RESs, providing real-time monitoring via embedded bidirectional communication systems. IoE is a cloud-based technology that incorporates all energy components with embedded Advanced Metering Infrastructures (AMI) and Information and Communication Technology (ICT), enabling bidirectional communications and flow of energy [9]. The decentralized structure of IoE-based power systems facilitates the accommodation and integration of RESs, which leads to increased demand-supply reliability. The IoE concept also presents a virtual configuration that allows consumers and customers to participate in the open market regardless of physical distance. Finally, intelligent controls implemented in IoE-based power networks improve energy efficiency, leading to profit maximization. It should be underlined safety regulations that Independent System Operator (ISO) establishes must be maintained during the procedures.

Aside from the numerous advantages of employing IoE, this paradigm is highly vulnerable to malicious attacks due to the broad range of bidirectional interconnection among installed energy components and smart devices [10]. According to the Industrial Control Systems Cyber Emergency Response Team (ICS-CERT) report, roughly 60% of major attacks reported across all sectors have been conducted in the energy section [11]. The distributed pattern of IoE, which allows users to interact and exchange information and energy without central control, leads to various security and privacy challenges and makes IoE-based networks an attractive target for intruders. Security concerns are not limited to cyber layers since the system may be physically manipulated/damaged for electricity theft or sabotage. One of the most critical aspects of cybersecurity concerns in IoE-based networks is the False Data Injection (FDI) attacks, which affect data integrity, resulting in inaccurate forecast and operational scheduling strategies [12]. Likewise, most of the malicious physical activities in the power network aim at electricity theft in order to economic misuse via device tampering or bypassing [13].

The next major challenge is optimizing the operational time of all energy components whereby the end-users can act as price takers (buying electricity from the utility) and price makers (trading electricity among end-users without interference from the utility) by participating in a competitive market and selling their electricity. Moreover, the scheduling framework requires not to be limited to smart homes, home area networks, or grid levels is one of the key challenges in modern power systems. By developing and maintaining a scheduling algorithm, on the one hand, all consumers make a profit. These earnings come from incentives and tariff adjusting (for shifting consumption to off-peak hours), besides selling the surplus energy to the market. On the other hand, utilities can reduce the operation, transmission, and maintenance costs, reducing the need for new investigations. It should be noted that IoE-based electricity network faces technical and economic constraints for optimal operation scheduling in both generator and consumer due to the uncertainties originated by the RESs and costumer welfare and preferences. Therefore, joint techno-economic parameters are required to be addressed at the same time due to the trade-off between technical and economic limitations. Also, coping with different uncertainties, including renewable sources, price, market circumstances, and power demand uncertainties is crucial since these factors can significantly affect forecasting and scheduling procedure.

After guaranteeing data genuineness and optimum operation scheduling, the next challenge is optimizing electricity flows among a group of energy components. The Energy Router (ER) concept has been developed as a compact intelligent power electronic device to control electricity flows among a group of devices. ERs are hybrid AC/DC interfaces that optimize electricity routing strategy by enabling bidirectional power flow in different voltage levels. An electricity routing algorithm is needed to maximize energy efficiency and minimize power losses, enabling end-users to participate in the electricity market and Peer-to-Peer trading. Moreover, the developed algorithm requires taking GHGs emissions amount and environmental concerns into account.

1.2. Literature Review and Gap Identification

Detailed citations and content analysis are presented in this section. Firstly, studies published in top-tier journals are extracted to be analyzed. Then, a comprehensive breakdown is conducted to find the significant research gaps in the literature. Web of Science has been chosen to extract the relevant document for paper mining due to this academic search engine's unique features and strength. The extraction process is not just limited to the main topic, IoE, and three other searches have been conducted considering the keywords related to the objectives and contributions. The query to extract the related papers to the IoE is “TI=("Internet of Energy")) OR TI=("IoE")) OR TI=("Energy Internet")”.

As a result, 581 remained, which are highly cited papers in electrical and computer engineering over the last five years. A significant increase in the number of publications between 2016 and 2022 shows the IoE has attracted much attention over the past few years. Furthermore, the citation number in 2022 is more than ten-fold of 2017. The entire content of the documents has been analyzed to discover the most frequent words and the linkage among them.

Figure 1.1 shows that most keywords are related to energy management terms, optimization, and security. Analyzing the results proves that the IoE publications' trend experienced fast growth, which shows the topic has been investigated in recent releases.

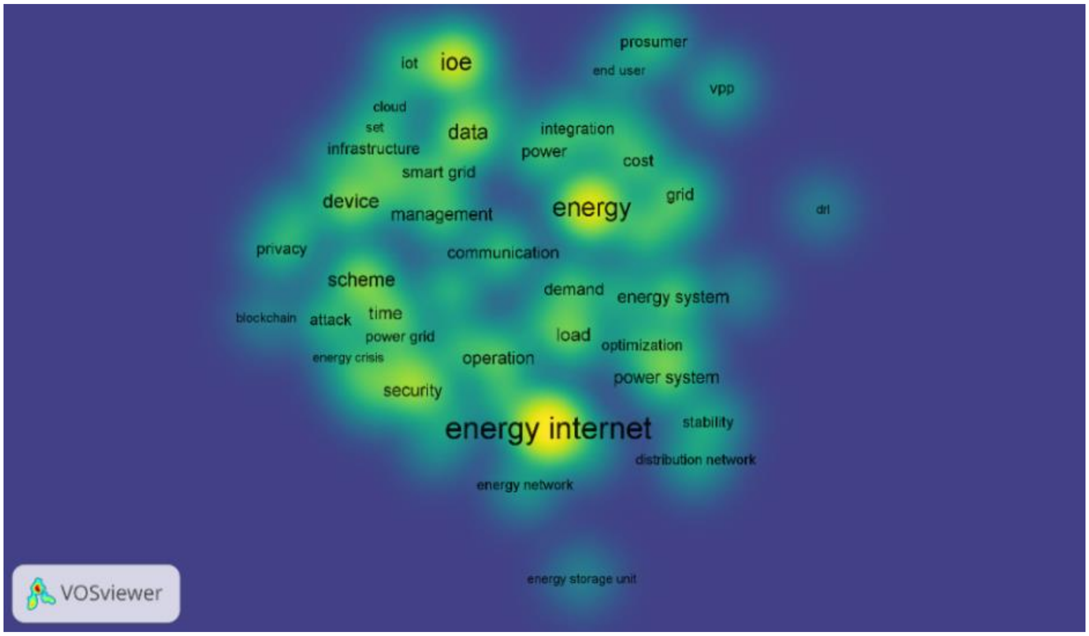


Figure 1-1. The most frequent keywords in the content of publications

1.2.1. Gap Identification-1

FDI attacks are typically identified as falsifying State Estimation (SE) in power systems, and bad-data detection methods are broadly used to detect them [14]. However, despite the fact that the vast majority of the FDIA detection techniques depend on network topology and parameters information, an intruder can still launch an FDI attack into the system without mentioned bits of knowledge [12]. Additionally, relying on power system states makes the detection process impractical in very large-scale networks. Consequently, the fragilities of conventional detection methods become gradually prominent by facing complicated attack strategies deriving from network advancement and utilizing the gigantic quantity of AMIs and communication tools, regardless of SE data.

Developing a cyber defense framework requires a comprehensive insight into attack generation and detection simultaneously. Linear design approaches have been broadly employed in [15] and [16] to generate FDI attacks and target diverse topologies of power systems. The electricity market is the main target of the attacker in [17] by developing a Monte Carlo FDI attack strategy, given that the intruder's knowledge about the topology of the system is insufficient. However, the proposed attack modeling approaches in [18] and [19] necessitate a thorough knowledge of network topology and states. The principal drawback of previously mentioned attack-generating techniques is that a well-developed intelligent detection algorithm can easily prognosticate the attack strategy. Furthermore, once the strategy is disclosed, the attack generation algorithm cannot modify itself to forge new unknown attacks unknown to the detection mechanism. Consequently, a Reinforcement Learning (RL) attack generation mechanism has been developed in [20]–[22], enabling online learning to design an optimal attack policy while dynamically interacting with the environment and continuously improving the strategy. Nonetheless, the main disadvantage of employing a conventional RL algorithm is the curse of dimensionality, which makes the procedure inefficacious, besides from the lack of scalability and generalization.

FDI attack-related studies in the literature are not only concentrated on the attack generation side, and numerous works in the literature have investigated different approaches to detecting FDI attacks in smart grids. Proposed methods in [23]–[26] rely on the admittance perturbations strategy for detecting FDI attacks. Even though the utilized method can reveal stealthy attacks, dependency on network states is still a drawback since these parameters may not be estimated correctly considering meters placement and network topology.

Lately, attack detection strategies employ the transmitted data over communication links among nodes and data centers. Analyzing the generated highly complex big data in the IoE-based grids requires novel and powerful data mining and pattern extraction methods. Consequently, various machine learning techniques have been considerably utilized to design detection frameworks since traditional methods are not capable of feature engineering and finding complex patterns. Various Supervised and semi-supervised learning algorithms, including Support Vector Machine (SVM), Decision Tree (DT), Deep Neural Network (DNN), Convolutional Neural Network (CNN), and Long-Short-Term Memory (LSTM), have been utilized in [27]–[31] to develop a detection structure. These mechanisms were generally developed to detect passive attacks, while active threats are yet a significant concern, and a dynamic threat-hunting layer is needed. Unfortunately, neither the studies mentioned above nor other related works in the context of FDIA detection in IoE-based smart grids present a framework to defend the system from an intelligent intruder who designs attacks adapting to the dynamic environment of the smart grid. Moreover, a multilayer attack detection structure is required to simultaneously detect passive and active threats.

1.2.2. Gap Identification-2

Utilized operational scheduling optimization techniques in the literature are categorized into three classes, including mathematical, heuristic, and learning-based techniques. Conventional methods, including Linear programming (LP) [32]–[34], Mixed Integer Linear Programming (MILP) [35]–[37], and Mixed-integer non-linear programming (MINLP) [38] have been utilized to develop a scheduling algorithm. In the mentioned studies, both consumers and prosumers play the same role in the market as price takers, resulting in a non-transparent market since a large share of the network belongs to household users. Furthermore, the proposed techniques are incapable of dealing with complex non-linear environments besides their incapability to handle big data generated in the IoE-based grids.

Heuristic and mathematical algorithms have widely been employed to optimize operational scheduling. In [39] an operational scheduling algorithm has been developed considering electricity cost and safety using a Genetic Algorithm (GA) based on users' at-home status and awake status. Authors in [40], proposed an Enhanced Leader Particle Swarm Optimization (ELPSO) to schedule home appliances aiming for peak shaving and maximizing monetary profit. At the same time, a Cooperative Particle Swarm Optimization (CPSO) has been developed in [41] to aggregate multiple smart homes as a virtual energy storage system via an optimal appliance scheduling algorithm. In [42], residential energy resources and appliances scheduling algorithm has been proposed for smart homes based on varying electricity tariffs that have been solved using a novel Natural Aggregation Algorithm (NAA)-based approach. Furthermore, a new home energy management system has been developed in [43] using NAA to minimize daily electricity costs, considering the monthly peak power consumption penalty. Heuristics and mathematical algorithms rely on explicit environment models and precise forecasts of various types of uncertainties, including technical, economic, customer preferences and lifestyle, and weather uncertainties, that cannot be fulfilled in real-world problems.

Artificial intelligence and machine learning are promising methods interested in various science branches since, unlike traditional methods, significant expertise is not vital to use these approaches. Furthermore, due to the big data challenges in modern power systems, besides the complexity, velocity, and computational burden of conventional optimization methods, learning-based optimization approaches have been utilized in [44]–[46] to optimize scheduling problems. Despite the fact that supervised learning methods are less subordinate to accurate forecasting data and model uncertainties, they still suffer from limited approximation capability and slow convergence in a large-scale dynamic and decentralized environment [47].

Since scheduling in the new generation of smart grids is a decision-making NP-hard problem, RL is a well-suited algorithm to solve them. The proposed techniques in [48]–[53]

utilized RL to develop an operational scheduling framework due to the capability of learning optimal behavior by making a trade-off between exploration and exploitation. Many challenges are associated with RL, including the curse of dimensionality, lack of scalability, poor generalization, and limited non-linear representation capability that disqualify this method from solving scheduling problems in a real-world smart grid environment [54]. On the other hand, Deep Reinforcement Learning (DRL) algorithms have recently accomplished extraordinary breakthroughs exploiting Deep Neural Network (DNN) strengths, involving the ability to handle unstructured data, no need for feature engineering, and non-linear representation capability, to name a few [55]. Consequently, taking advantage of deep learning strengths, Deep Q-Learning (DQL) swamps RL deficiencies. Different DRL techniques, including DQL, Deep Policy Gradient (DPG), Double DQL (DDQL), Centralized Deterministic DPG (C-DDPG), and Distributed Deterministic DPG (D-DDPG), have been utilized in [56]–[62] to solve the scheduling optimization problem in smart grids. As results show, sample efficiency and usability are still two key concerns in the proposed methods. Moreover, the difference in reward of four reported buildings reveals the model suffers from high variance. It is worth mentioning that policy gradient methods face the risk of trapping in a local optimal; also, a lot of training time is required to reach the global result [63].

Neither formerly mentioned studies nor other related studies in the literature attempted to reduce the positive bias of DQL (originated from using $\max_{a} Q(s, a)$) while improving the negative bias simultaneously. Moreover, the learning efficiency and space and sample complexity of the past works need to be enhanced. Therefore, this thesis aims to develop an algorithm that makes a trade-off between positive and negative bias by synchronously reducing overestimation and underestimation. Additionally, the designed algorithm must be capable of taking advantage of positive and negative biases in case of the need for better exploration and exploitation, respectively. The ultimate objective is to improve learning efficiency while reducing space and sample complexity since these characteristics incredibly affect time-efficient and cost-effective solutions in discrete action spaces and problems with continuous and large action spaces.

1.2.3. Gap Identification-3

Numerous studies have investigated the concept of ER in the present and future generations of power systems. The majority of investigations, including the proposed methods in [64]–[70], tried to optimize the electricity routing problem at the high-level voltage. Utilizing ERs in the high-voltage network is not feasible economically since fulfilling technical and safety requirements in the power transmission/distribution layers is exceedingly costly [71]. However, the high penetration of distributed energy resources in the low-voltage layer necessitates utilizing ERs to meet energy commitments and financial objectives.

Although authors in [72]–[74] developed optimization methods for electricity routing in low and medium voltages local area networks, nevertheless smart home appliances and personal energy units such as exclusive rooftop Photovoltaic (PV) systems and battery storage units have not been considered in the designed algorithms. A few studies have investigated ER-based energy management at the low-voltage and residential levels. However, the utilized methods, including MILP in [75] and fuzzy logic-based hierarchical control strategy in [76], [77] are not able to handle the extremely uncertain environment of the IoE-based grids, which makes mathematical modeling impractical [47]. An additional source of uncertainty also originated from unpredictable customer preferences and conditions. Consequently, using historical data to forecast future electricity generation and demand is not straightforward and makes the predictions inaccurate.

Deep Reinforcement Learning (DRL) has been widely utilized in energy systems thanks to the capability of addressing control and optimization problems model-freely [22]. Moreover, DRL-based methods take advantage of high flexibility and generalization due to not relying on prior knowledge about the system’s topology and information. In [78], a marketing auction mechanism has been developed utilizing DRL to minimize energy costs in microgrids. A Deep Q-Learning (DQL)-based optimal energy management mechanism for an office building has been developed in [79], controlling the energy flow of PV and battery storage. The concept of energy routing centers has been proposed in [80], coordinating multi-energy coupled energy framework. The developed model aims to enhance the conversion flexibility of energy components.

Despite the fact that previously published works in the literature have provided seminal insight into utilizing ERs at the residential level, several significant problems have not been appropriately addressed. Firstly, none of the earlier studies simultaneously deemed all exclusive and shared energy components in Nano Area (NA) and Neighborhood (NH). Furthermore, the proposed routing structures have not supported P2P electricity trading and input power from NHs, which are incredibly imperative and effective in optimizing cost and loss. Ultimately, the hitherto developed routing procedures have not attempted to enhance the algorithm's efficiency while providing a feature to take advantage of positive and negative biases where applicable.

This thesis aims to develop a routing algorithm that simultaneously considers all energy components in the NA and delivered power from NH. Moreover, the designed algorithm must enable and guarantee electricity trading between residential units and the neighborhood besides P2P contracts. Furthermore, the proposed algorithm to solve the optimization problem requires to be efficient and fast in convergence. Ultimately, since overestimation and underestimation, which respectively originate from more exploration and exploitation, are not always destructive, this research's principal objective is to outline boundaries for biases while enabling the capability of adjusting exploration and exploitation.

1.3. Thesis Objectives

The fundamental goal of this research is to optimize the efficiency of IoE-based power networks while guaranteeing the system's cyber security. Consequently, three main objectives are defined below:

- The first objective of this research is to develop a Multi-Layer cyber-defense mechanism to detect and hunt active and passive attacks that can be launched by an intelligent intruder who dynamically interacts with the environment and learns the network topology and parameters. The first objective is addressed in Chapter 2.
- After ensuring data authenticity in the system, the second objective is to optimize the operational scheduling of all energy components in the system to enhance energy efficiency while reducing the cost and GHGs emissions and establishing a competitive market. This objective is addressed in Chapter 3.
- The last objective of this thesis is to establish an electricity routing optimization algorithm to optimize the energy routers' performance improving electricity routing in residential units aiming to reduce monthly average cost, power loss, and GHG emissions. The objective is addressed in Chapter 4.

1.4. Thesis Contributions

Three major contributions are defined in this research, followed by multiple minor contributions, as described below.

- This research's first contribution is developing a Multi-Layer defense algorithm against deep reinforcement learning-based intruders in smart Grids. Consequently, an intelligent intruder as an active attack generator is developed, initialized by modeled passive attacks.

Subsequently, the attack generator algorithm simulates the network environment and creates active attacks. After creating a dynamic attacker, a multilayer defense framework is developed using Snapshot Ensemble Deep Neural Network and an adoptable Deep Auto Encoder network to detect known and unknown threats. The first layer of the detection mechanism is designed to detect passive attacks whose structures have been previously introduced to the system. Finally, the second layer is trained to hunt future unknown attacks based on the real-time information of the network.

- The next contribution of this thesis is optimizing scheduling policy in smart grids using Probabilistic Delayed Double Deep Q-Learning (P3DQL). Consequently, a novel algorithm named Probabilistic Delayed Double Deep Q-Learning is introduced and developed for the first time, which is a combination of the tuned version of Double Deep Q-Learning and Delayed Q-Learning. In this method, the selection order of estimators in DDQL is converted in a probabilistic manner, eliminating the underestimation challenge and aiming to make a trade-off between positive and negative biases. Moreover, the proposed algorithm is modeled as Probably Approximately Correct in Markov Decision Processes (PAC-MDP), enhancing learning efficiency and reducing sample complexity making the algorithm capable of handling problems with a large action space, including scheduling problems in smart grids. Finally, a Multi-Layer scheduling mathematical model with low numerical error is proposed, which comprehensively covers from a single Nano Area (NA) to a Neighborhood (NH) and a Wide Area Network (WAN), including share storage units, PVs, and considering different tariff types and customer preferences. Subsequently, the problem is solved by the P3DQL algorithm.
- The third contribution of this thesis is optimizing resource swap functionality in IoE-based

grids using Approximate Reasoning Reward-based Adjustable Deep Double Q-Learning (A2R-ADDQL) algorithm that is introduced specially to optimize electricity routing in residential units. The proposed algorithm improves efficiency due to the decline in the number of random actions. Furthermore, utilizing the proposed reward function in an RL-based algorithm also leads to a higher convergence speed since the number of state-action pairs is reduced. Furthermore, this electricity routing optimization algorithm is capable of adjusting to the nature of the problem by taking advantage of exploration and exploitation where overestimation and underestimation are favorable, respectively.

1.5. Thesis Structure

This manuscript-based dissertation includes five chapters, and the main findings of this research study are presented in the following three chapters.

Chapter 2 develops a deep Q-Learning-based false data injection attack generator using various possible attack scenarios. Moreover, a two-layer attack detection framework was developed using a snapshot ensemble deep neural network and deep autoencoder to detect passive and active threats, respectively. Additionally, the proposed attack modeling and detection framework are simulated using a combination of ns-3, FNCS, and GridLAB-D simulators. Ultimately, the same setup is modeled based on two different developed algorithms to make a comparison between the performances. A version of this chapter was published in the International Journal of Electrical Power & Energy Systems journal.

Chapter 3 develops and introduces a novel algorithm named Probabilistic Delayed Double Deep Q-Learning, which is a combination of the tuned version of Double Deep Q-Learning and Delayed Q-Learning. The proposed algorithm makes a trade-off between overestimation and underestimation biases, guaranteeing sample complexity by applying a delay in updating the rule.

Finally, the proposed algorithm is tested on three real-world datasets assessing its performance in various benchmarks. A version of this chapter was published in Sustainable Energy Technologies and Assessments journal [47].

Chapter 4 proposes a novel algorithm titled Approximate Reasoning Reward-based Adaptable Deep Double Q-Learning (A2R-ADDQL) that is introduced specially to optimize electricity routing in residential units. As a result, both overestimation and underestimation biases are reduced compared to other deep Q-Learning-based algorithms. Moreover, the sample complexity of the model is decreased due to utilizing a fuzzy approximate reasoning reward function. Ultimately, the proposed algorithm is assessed on a real-world dataset evaluating the findings in several benchmarks. A version of this chapter has been submitted to IEEE Transactions on Consumer Electronics journal.

Finally, general conclusions and recommendations, and suggestions for future studies are presented in Chapter 5.

Chapter 2

Multi-Layer Defense Algorithm Against Deep Reinforcement Learning-based Intruders in Smart Grids

2.1. Introduction

The Internet of Energy (IoE) links energy components, smart metering infrastructure, and Information and Communication Technology (ICT) to overcome emerging challenges using modern energy management techniques and tools [9]. On the one hand, electricity users demand to receive high-quality, reliable, and environment-friendly services with acceptable costs, guaranteeing their security and privacy. On the other hand, access to advanced real-time monitoring and controlling approaches to integrate renewable resources, maximize reliability, and minimize loss is crucial for utilities to provide reliable and secure services with premium quality [47].

Developing an IoE-based smart grid requires installing numerous sensors, wireless communication tools, smart appliances, and data acquisition units. While the open architecture of IoE-based networks, originating from two-way communication infrastructures and myriad internet-based entries, raises vulnerabilities against malicious activities.

False Data Injection Attack (FDIA) is one of the major and most severe threats to the network that endangers data integrity by bypassing the conventional bad data detection mechanisms [12]. The most vulnerable sector against FDIAs is Advanced Metering Infrastructures (AMIs) due to their scale, diversity, and complexity, besides uninterrupted functionality over the communication network [81]. Three main categories of attack layouts have been introduced for

FDIAs, including expert attackers, non-expert attackers, and data-driven attack models. An expert attacker is a professional adversary with complete knowledge of the nature of the system and the network topology, capable of designing an extremely complicated attack. However, a non-expert intruder who has limited knowledge about the system can also create and launch a stealth attack [82]. Finally, data-driven attacks target the network by applying an independent component analysis to learn the system's perception from the correlations of the measured data by AMIs deployed on the physical system [83].

FDIAs are commonly recognized as cyber-attacks on State Estimation (SE) in smart grids, and Bad Data Detection (BDD) methods are widely employed to detect them based on the L_2 norm between the actual and the estimated measurements [84]. However, despite the fact that many of the FDIA detection techniques in the power systems focused on the SE in accordance with the line reactance data and cognizance of network topology, an attacker is still able to target the system by an FDIA in the absence of mentioned bits of knowledge [12]. Furthermore, the fragilities of classic FDIA detection techniques become gradually prominent by facing extremely complicated attacks originating from network advancement and utilizing the gigantic quantity of AMIs and communication tools, regardless of SE data.

FDIAs have been enthusiastically investigated in terms of attack generation and detection at the same time. In this section, the attack simulation techniques are firstly studied, then the defense strategies are investigated. Linear FDIA modeling approaches have been widely utilized to generate and target different topologies of power systems. A linear attack generation technique with an arbitrary mean has been developed in [15] without requiring a zero-mean Gaussian distribution. The proposed attack generation framework leads to an optimal attack approach, addressing a constrained quadratic optimization problem by the Lagrange multiplier technique.

Linear regression with a time stamp has been employed in [16] by filling up the Nan-measurement values in real-time data. This technique focused on remaining stealthy during the attack procedure.

A Monte Carlo FDIA strategy has been proposed in [17], targeting the electricity market. The suggested method assumes that the attacker has an inadequate level of knowledge about the topology of the system while the main aim is monetary profits. Authors in [85] suggested an attack model aiming to launch an inexpensive technique since obtaining the system state is costly. The proposed procedure has been utilized to approximate the system states by employing a small number of power flow parameters or injection measurements. Furthermore, the intruders' system knowledge to analyze and design optimal attack strategies is examined. Despite the designed FDIA model in [85], which assumes that the attacker has partial knowledge of some specific measurements of the power system, the developed FDIA generation in [18] and [19] requires a comprehensive understanding of different parameters.

The main shortcoming of the techniques mentioned above is that a well-designed intelligent defense system can predict the attack strategy effortlessly. Besides, once the attack generation pattern is revealed, the frameworks are not capable of adapting the recent condition to create new unknown attacks. The optimal attack sequences have been generated by the suggested method in [22] using a dynamic game between the attacker and the network based on Reinforcement Learning (RL). Although the proposed optimal attack strategy indicated a satisfactory performance on IEEE 39-bus systems, the intruder can be tricked if the targeted environment is just a simulated substitute system created to engage and delay the attacker. Authors in [21] and [20] proposed RL-based algorithms enabling online learning to design an optimal attack policy. However, the utilized conventional Q-Learning algorithm suffers from the lack of scalability and generalization besides the curse of dimensionality, which makes the algorithm extremely inefficient.

FDIA-related investigations in the literature are not only focused on the attack generation side, and many studies have been conducted on attack detection methods. Using the ex-ante admittance perturbation strategy, a hidden moving target defense approach has been proposed in [23], which the attackers cannot detect. This strategy presumes that the transmission line admittance changes at every SE interval. A state summation strategy has been developed in [24], focusing on sparse attacks specially to protect meters. Then the effectiveness of the proposed method has been investigated regarding the system's scale. Authors in [25] presented a subsequent admittance perturbation strategy based on the differences between the column space of the measurement and attack matrices. Finally, a joint admittance perturbation and meter protection method has been proposed in [26], aiming to increase the accuracy of estimated states under stealthy FDI attacks.

Although the strategies mentioned earlier can precisely detect stealthy FDI attacks, they still rely on all network states that may not be estimated correctly due to meters placement and network topology. Even though physical protection of all utilized assets prevents access to crucial information about the network's topology, limited network information is always available and opens a gate for malicious activities [86]. Moreover, comprehensive physical protection is enormously expensive and impractical, especially in large-scale systems. Authors in [87] show that complete real-time knowledge is not approachable for an attacker in a real case due to inadequate access to most grid facilities. Consequently, most FDIAs occur while network topology and transmission-line admittance values are not utterly clear to the attacker.

Recently, FDIA techniques exploit the transmitted data over communication links among nodes and data centers that lead to generating highly complex big data. Accordingly, machine learning techniques are extensively considered as an attack detection solution since conventional methods are not capable of feature engineering and finding complex patterns [88].

Supervised and semi-supervised learning algorithms based on Support Vector Machine (SVM) have been employed in [27] to develop an attack detection procedure that has been examined on various IEEE test systems. The results show the superiority of the proposed methods over techniques that employ state vector estimation. An optimized extreme learning machine has been proposed in [28] for detecting unobservable FDIAs using a deep learning method-Autoencoder. The proposed technique utilized a combination of differential evolution and an artificial bee colony algorithm to improve the detection performance. Authors in [29] proposed a cyber threat detection approach based on the difference between True Positive (TP) and False Positive (FP) rates. The outcome demonstrates that a combination of event profiling for data preprocessing and Deep Neural Network (DNN) algorithms, including Convolutional Neural Network (CNN) and Long-Short-Term Memory (LSTM), is capable of detecting FDIAs with 6% higher accuracy than conventional machine-learning methods. Moreover, attacks with monetary motivation, such as electricity theft, which is a primary concern for utilities, have been investigated using different machine learning techniques. Although the presented results in [30] demonstrate the superiority of Artificial Neural Network (ANN) over Decision Tree (DT) and Random Forest (RF) for detecting electricity theft as an FDIA, in [31] and [28], it has been shown that CNN-based methods performed a better attack detection rate by a considerable difference.

The earlier techniques were typically designed to generate or detect passive attacks while lagging behind in hunting active threats. Consequently, a defense algorithm is required to detect and hunt threats that can be launched by an intelligent intruder who dynamically interacts with the environment to target the system with active attacks. Unfortunately, neither the studies mentioned above nor other related works in the context of FDIA detection in IoE-based smart grids present a framework to defend the system from an intelligent intruder who designs attacks adapting to the dynamic environment of the smart grid. Moreover, a multilayer attack detection structure is required to simultaneously detect passive and active threats.

Motivated to address the above-mentioned concerns, the main contributions of this chapter are summarized as follows.

I. An intelligent intruder is trained using Deep Q-Learning (DQL) to target the network, taking advantage of online learning by simulating a dummy power system. Moreover, various possible FDIAs are mathematically modeled to initialize the attacker algorithm.

II. As the first layer of the proposed framework, a Snapshot Ensemble Deep Neural Network (SEDNN) algorithm is developed employing the Cosine annealing technique by taking a snapshot once the model hits a local minimum before altering the learning rate. An ensemble of developed snapshots enhances the attack detection performance while reducing the risk of overfitting and computational cost.

III. A Deep Autoencoder-based network with an adaptable reconstruction error threshold is introduced as the active cyber defense to detect future unknown attacks based on the real-time information of the network. Although FDIAs are becoming more complex and intelligent, this active cyber defense makes the proposed framework more reliable in an unsupervised manner.

The remainder of this chapter is organized as follows. Section 2.2 presents the system model. In Section 2.3, the DRL-based attack generation framework is introduced, initialized by the mathematical modeled possible attacks. Section 2.4 presents the structure and algorithms of the proposed attack detection framework. The proposed model and framework are simulated in section 2.5. Finally, section 2.6 concludes this chapter.

2.2. System Model

One of the principal characteristics of an IoE-based smart grid is to provide real-time control and monitoring of physical components anytime and anywhere [89]. As Figure 2.1 illustrates, the architecture of the network model consists of three main layers, including Micro Area (MA), Neighborhood Area (NA), and Wide Area (WA).

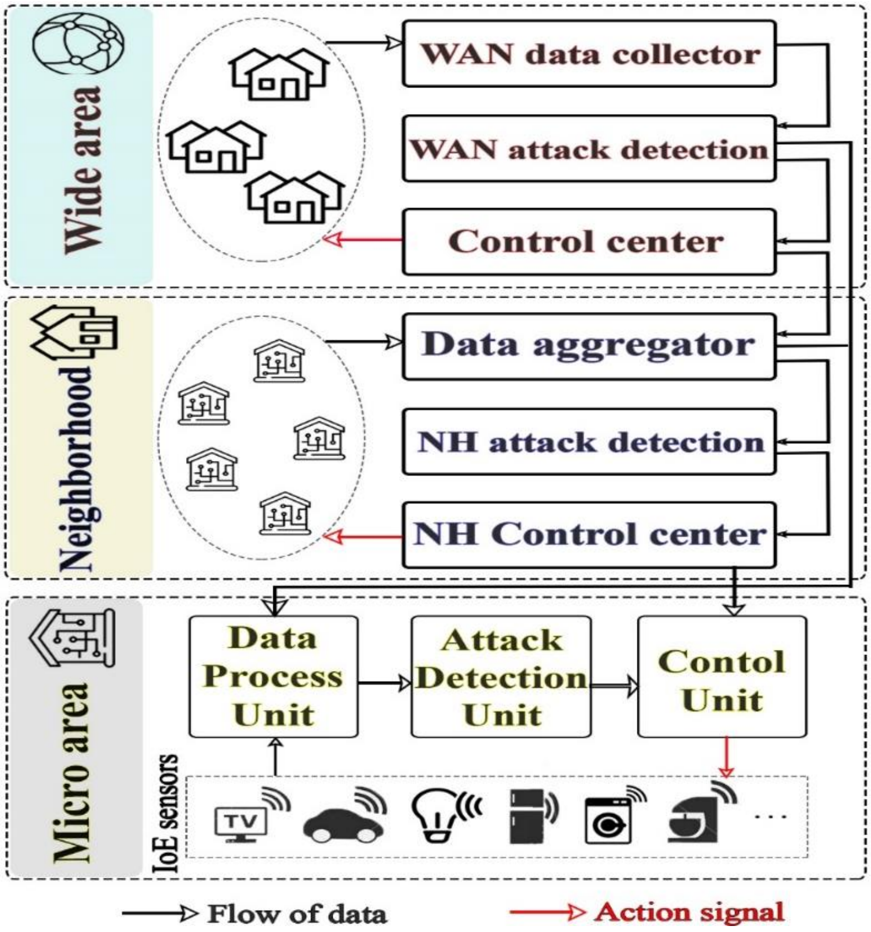


Figure 2-1. The proposed framework of the IoE-based network

Several smart meters, sensors, data concentrators, and AMI headends are placed into MAs over a local bidirectional wireless communication network. Then, an aggregator collects all energy entities' consumption data and sends the information to the attack detection unit. Ultimately, the control unit takes an appropriate action based on the status of the detection unit that shows whether the system is under attack or not.

A group of MAs forms a NA, exchanging electrical energy based on their contract. A neighborhood data aggregator collects the overall data of every participating MA. Next, the utilized attack detection module examines data correctness and declares the attack status to the next unit. The same process takes place at the WA level, considering collected information from two or many NAs. All the embedded sensors report a network parameter according to their assignments. This model takes consumed power as the reported measure with a specific sampling rate in a MA.

The logic is extendable for other parameters and NA and WA in the same way. Equation (2.1) defines the matrix of actual power consumption $P^{Act} \in |\mathbb{R}|^{T\times m}$, where T is the total numbers of time slots (e.g., if reading is reported every 15 minutes, then t=96) and m stands for the number of energy components (all electrical appliances, including solar cells and electric vehicles). The vector $C_j = (c_{1j}, c_{2j}, \cdots, c_{tj})^{Transpose}$ denotes reported daily consumption of appliance j in different time slots, where c_{tj} indicates reported consumed power by sensor j at the specific time slot t.

$$P^{Act}(c) = \begin{bmatrix} c_{11} & \cdots & c_{1m} \\ \vdots & \ddots & \vdots \\ c_{t1} & \cdots & c_{tm} \end{bmatrix} \tag{2.1}$$

Generally, an intruder compromises the integrity of the information by injecting a fake data vector $\alpha \in \mathbb{R}^{n\times m}$. Mathematically, conventional FDIAs are formulated as in (2.2), where P^{False} is the falsified matrix [90].

$$P^{False}(c) = f\left(P^{Act}(c)\right) = P^{Act}(c) + \alpha \tag{2.2}$$

This research considers the capability of node selection in all different locations for the attacker. Also, the intruder can schedule the attack in continuous or many discrete time slots. Accordingly, the formulation of FDIAs is modified in (2.3), where $f(c_{ij}) = \psi_1 c_{ij}^{\beta} + \psi_2 c_{ij}^{\beta-1} + \cdots + \psi_{ij}$ denotes applying function by the attacker on the matrix of measurements, also ψ, and β are constants and $k \in \mathbb{R}$.

$$P^{False}(c) = \begin{bmatrix} f_{11}(c_{11}) & \cdots & f_{1m}(c_{1m}) \\ \vdots & \ddots & \vdots \\ f_{t1}(c_{t1}) & \cdots & f_{tm}(c_{tm}) \end{bmatrix} \tag{2.3}$$

2.3. The proposed DQL-based attack generation framework

This section introduces different parts of the designed framework, including sample library (i.e., initial attacks and normal samples), adversarial attack generator, simulated environment, and actual environment, as indicated in Figure 2-2.

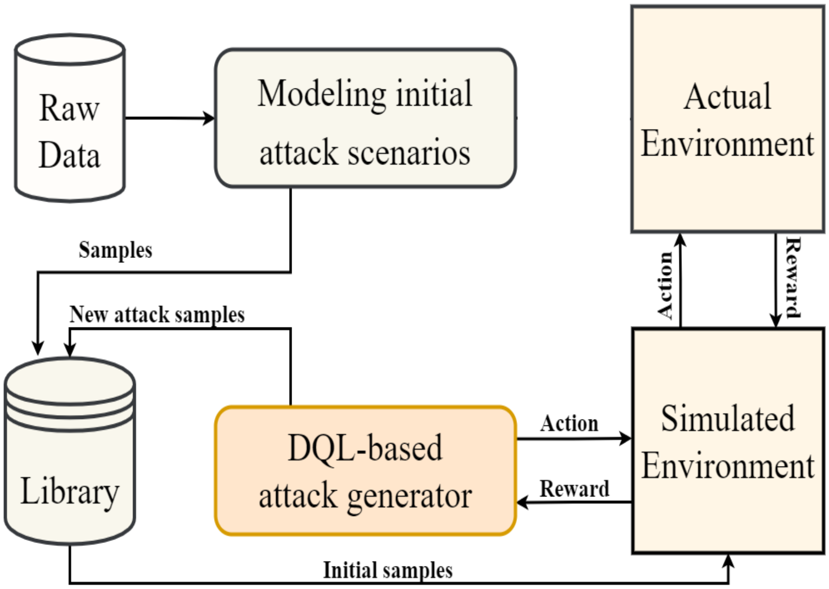


Figure 2-2. The framework of the proposed attack generation method

Primarily, five possible common attacks are mathematically modeled. Then, the DQL-based attack simulator is initialized by the developed attacks and starts training to improve the attack strategy and construct innovative attacks using the simulated environment. The attack library is updated just after discovering a new type of attack that has not been detected before. The entire process of training the attack generator algorithm is as follows.

2.3.1. Step 1: Mathematical modeling of possible FDIA scenarios to train the attack simulator

Five statistically different FDIA scenarios are mathematically modeled to store in the library as classified attacks. These attack scenarios are employed to initialize the DQL algorithm's training process. The attack scenarios are as follows:

i. Node-based attack scenario:

In this scenario, the attacker chooses one or multiple components and targets them regardless of time. Subsequently, corresponding columns of under-attack nodes in $P^{Act}(c)$ are changed. For instance, if the first node is selected by the intruder, then the first column of $P^{Act}(c)$ is changed from $(c_{11}, \dots, c_{t1})^T$ to $(f(c_{11}), \dots, f(c_{t1}))^T$. Equation (2.4) demonstrates alterations of the j^{th} array in the first column of $P^{Act}(c)$ after an attack.

$$P^{False}(C_{j1}) = \psi_{j1} C_{j1}^{\beta} + \psi_{j2} C_{j1}^{\beta-1} + \cdots + \psi_{jk} \tag{2.4}$$

The varying coefficients are defined based on the Joint Probability Distribution Function (JPDF) for each node, considering time. Consumption and time, which are denoted by C and T, are the variables and f_{CT}: $\mathbb{R}^2 \rightarrow \mathbb{R}$ is a nonnegative function so that the JPDF is defined for any set of $\mathbb{Q} \in \mathbb{R}^2$ as in (2.5), where $\{a, b\} \in \mathbb{Q}$.

$$P\{a < C < a + da, b < T < b + db\} = \int_{b}^{b+db} \int_{a}^{a+da} f_{CT}\ da.db \approx f(a,b) da\ db \tag{2.5}$$

Then, the maximum and minimum JPDF for every possible pair of C and T are calculated. Finally, varying coefficients, including ψ, and β are determined, satisfying the inequality in (2.6). Moreover, the deviation degree between actual and injected data distributions is examined using the Chi-square test. Equation (2.7) demonstrates the formula for calculating a Chi-square value χ^2, where observations are classified into k exclusive classes, and O_i and E_i are observed and expected frequencies, respectively. If $\chi^2 > 0.05$, then the test is failed.

$$Min\ P \leq \psi_{i1} C_{i1}^{\beta} + \psi_{i2} C_{i1}^{\beta-1} + \cdots + \psi_{ik} \leq Max\ P \tag{2.6}$$

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i} \tag{2.7}$$

ii. Time-based attack scenario

The second scenario occurs once all nodes are targeted at continuous or multiple discrete time slots. Consequently, the first array of the j^{th} time slot changes as indicated in (2.8). The coefficients are calculated the same as in the previous attack scenario.

$$P^{False}(C_{1j}) = \psi_{1j} C_{1j}^{\beta} + \psi_{2j} C_{2j}^{\beta-1} + \cdots + \psi_{kj} \tag{2.8}$$

iii. Joint node-time-based attack scenario

This scenario combines node-time-based scenarios, and the attacker considers both objectives simultaneously. In addition, the coefficients are set to avoid normality test failure.

iv. Shifting attack scenario

In this setup, the attacker only shifts the time of the reported consumption in one or multiple time slots. Typically, the main aim of this type of attack is to bypass high-priced tariffs during peak hours. No dummy vector is injected into the consumption matrix and just $P^{False}\left(C_{i(j+\Delta)}\right) = P^{Act}\left(C_{ij}\right)$, where Δ stands for the number of shifts in the time slot number.

v. Blind attack scenario

Blind attacks usually arise by amateur attackers intending electricity theft. The attacker has no expertise and randomly injects fake vectors. Predominantly, most injected values are zero to minimize the electricity bill amount.

2.3.2. Step 2: Training the environment simulator

It is challenging to design an optimal attack strategy in the absence of preceding knowledge. The solution is assessing the environment by trial and error, while it is crucial to attack and learn stealthily. Consequently, a dummy environment is needed to avoid revealing the attack strategy during finding the optimal attack strategy.

RL-based algorithms can simulate a system by characterizing agents and states interacting with the environment to learn based on rewards and penalties received through experiences. Q-Learning is one of the most prevalent RL algorithms aiming to maximize the cumulative reward during the learning stage based on all states to find the optimal policy.

The update rule of conventional Q-Learning is indicated in (2.9), where $Q(s, a)$ represents the value of action a_t in state s_t, s' is the next state by the probability of transferring from state s with action a, the learning rate and discount factor are indicated by α and γ, respectively, and r denotes the reward.

$$Q_{t+1}(s, a) \leftarrow Q_t(s, a) + \alpha . \left(r + \gamma \max_{a'} Q(s', a') - Q_t(s, a)\right) \tag{2.9}$$

The major drawback of Q-Learning is the dependency on a look-up table to take the superior action at each state while storing the values of the state-value Q-function. Subsequently, the size of memory required by Q-Learning in medium and large-scale problems is enormous. Since the process is extraordinarily slow and costly due to the required memory and time, DNN plays a crucial role as a function approximator in DQL, where the inputs are the states. The Q-values are calculated as the outputs, focusing on minimizing the loss function as in (2.10), where

μ is the experience buffer containing, and θ represents the parameters of the policy.

$$L(\theta_t) = \mathrm{E}_\mu \left[(Q(s,a;\theta_t) - r_{t+1} - \gamma \max_a Q(s',a;\theta_t))^2 \right] \tag{2.10}$$

The process starts with defining the environment of the problem, including a set of energy components that consume or generate electricity, where the amount of net power is sampled every 15 minutes for each device. Then, samples are randomly selected from the highly imbalanced library with only 8% of attack examples.

States in DQL specify the condition and status of the environment that agents need to observe by interacting with the environment and taking optimal action. The state space is defined as in (2.11), where $l_t^{app_j}$ indicates the status of j^{th} appliance at time slot t that can be electricity load. If $l_t^{app_j}$= 0, the device status is OFF.

$$S_t^{app} = \left\{ l_t^{app_1}, l_t^{app_2}, \dots, l_t^{app_m} \right\} \tag{2.11}$$

The optimal action is a choice that requires to be made by the agent after observing the states to maximize the reward. Equation (2.12) describes the action space of appliances in this environment, where I and D actions stand for the increase and decrease of the consumption load, respectively, C is an action to shut down the device, H denotes hold (no action is needed), and M indicates the number of appliances.

$$A^{app} = \{I, D, C, H\}, \forall app \in [1, M] \tag{2.12}$$

The selected samples are input into the dummy environment as the agent, while the $\max_{a'} Q(s', a')$ is predicted as the output. Then, the DRL environment (i.e., the actual environment in this stage) receives the current state and the corresponding action to generate the reward. This process continues to minimize the loss function while predicting the parameters, constraints, and network topology of the simulated environment.

2.3.3. Step 3: Generating innovative FDIAs

After eliminating the risk of revealing, the second DQL algorithm acts as an attack generator. Accordingly, the reward function is modified in this stage to distinguish the newly created attack from the previously modeled ones. Furthermore, since the attack generation section freely targets different sections of the dummy environment on various time slots, the attacker's information is not limited to local information or specific time slots. Also, there is no restriction on cooperation and communication with the dummy environment, and providing feedback allows the attacker to define the optimal policy and improve it constantly. The training process starts by initializing the algorithm parameters, as in Table 2-1.

Table 2-1. DQL Agent and the Neural Network Parameters

Description	Value
Number of iterations	1000
Gamma	0.95
Epsilon	0.90
Batch size	500
Decay rate	0.98
Number of hidden layers	4
Number of hidden units per layer	120
Activation function	ReLu

The process is briefed in Algorithm 2.1, where Pr^{mis} denotes the probability of mis-scored, D is the replay buffer, and α^{lr} indicates the learning rate.

```
Algorithm 2.1: Attack generation algorithm process
Initialize the parameter of the dummy environment
Initialize parameters as in Table 1
Input $D$ to capacity $C^{rep}$, minibatch $k^{rep}$, $\alpha^{lr}$
Inputs $S, A, \gamma, n, \epsilon$
```

```
**for** episode = 1, M **do**
    randomly generate a sample of sates
    initialize sequences ${S_1}^i$
    store transition in $D$ at each episode
    create (q_value_list = [Batch size, Action size])
    compute loss as in equation (2.9)
    get the similarity results from the dummy environment
    compare the classification labels $l$
    **if** $l$: $True$: *#it is similar to attack examples*
        **set** $reward = 2$
            Compare $Pr^{mis}$
            **if** ${Pr^{mis}}_{t+1} > 1.1\ ({Pr^{mis}}_t)$: *#the created attack is not innovative*
                **set** $new - reward = reward - 1$
            **if** ${Pr^{mis}}_{t+1} < 0.9(\ {Pr^{mis}}_t)$**:** *#the created attack is innovative*
                **set** $new - reward = reward + 3$
        **return** reward
    **if** $l$: $False$:
        **set** $reward = 0$ *#it is like normal examples*
        **return** reward
    set $y_j$, then calculate the error
    perform gradient descent
**end**
```

2.4. The Proposed Multilayer Attack Detection Framework

Mathematical modeling of different attack scenarios illustrated that a professional intruder could design a series of attacks that can pass conventional FDIA detection frameworks. Moreover, a broad attack detection strategy is required to detect overlooked threats since the network environment is exceptionally dynamic, and adversaries are capable of planning progressively complex and intelligent attacks.

This chapter proposes a multilayer attack detection framework that combines supervised and unsupervised learning algorithms. Figure 2.3 shows that a SEDNN attack detection algorithm analyzes real-time reported information to find any malicious activities using the predefined and classified attack models in a library. Then, normal data is inserted into a Deep Auto Encoder (DAE) based unsupervised classifier to discover any possible abnormality. The developed DAE network takes advantage of an adaptable reconstruction error threshold. After detecting an attack, the library is updated to reduce detection time and cost in the future.

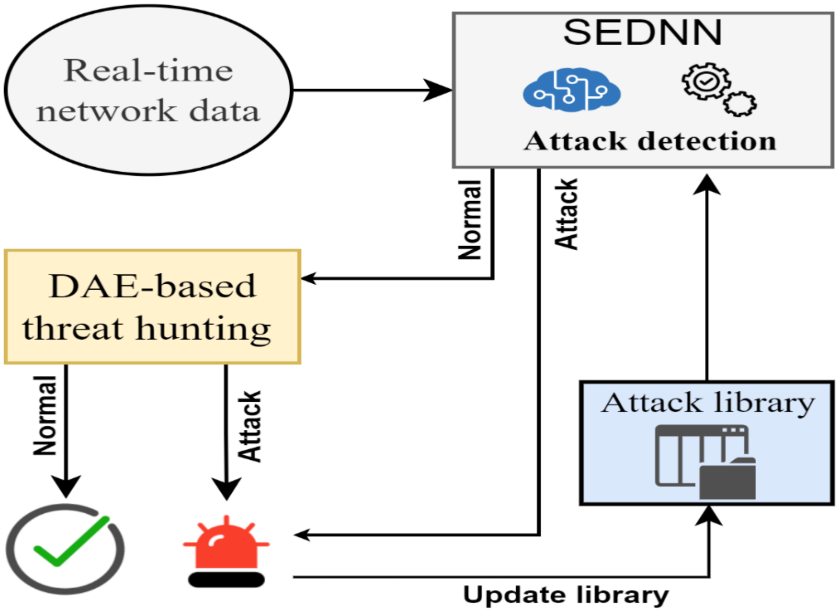


Figure 2-3. A schematic of the proposed attack detection algorithm

2.4.1. The proposed Snapshot Ensemble Deep Neural Network (SEDNN) algorithm to detect passive attacks

Ensemble learning expresses the method of training and combining multiple machine learning algorithms aiming to enhance predictive performance [91]. The ensemble architecture of neural networks is more precise and robust than a single model due to the abilities stemming from this method, including overfitting avoidance, concept drifting, and dimensionality reduction.

The main disadvantage of the ensemble method is that training multiple DNN models is a costly process due to the extensive computational burden. Also, the best model among all trained models usually beats the ensemble method. Consequently, a snapshot ensemble that develops multiple models from a single training process is introduced as the solution. This technique combines different models' predictions while saving models during the training phase and employing them to create an ensemble setup [92]. Furthermore, the learning rate used during the training stage is aggressively altered using the Cosine annealing technique defining the initial learning rate and the number of training epochs to avoid similarity among models. In DNNs' training process, the learning rate generally decreases after several epochs, resulting in a reduction of validation loss. Hence, the risk of overfitting is remarkably increased, which needs to be addressed.

The Cosine annealing method fluctuates the learning rate from a maximum to approximately zero, letting the algorithm converge to a different solution. Equation (2.13) formulates the learning rate α in the Cosine annealing procedure, where α_0 denotes the initial learning rate; t is the iteration number, T stands for the total iteration number, and M denotes the number of cycles [93].

$$\alpha(n) = \frac{\alpha_0}{2}(\cos\left(\pi \times \left[\frac{T}{M}\right]^{-1} \times mod\left(t, \left[\frac{T}{M}\right]\right) + 1\right) \tag{2.13}$$

Once the model hits a local minimum considering the validation loss, a snapshot of the model is taken, and the parameters are saved. Then, the learning rate is increased, as mentioned above, to start the training cycle of the second snapshot. An ensemble model can be developed after training N models, while the number of snapshots is defined based on the total training time of all models. Equation (2.14) defines the process of selecting N (i.e., the number of snapshots), where $T_i^{Snapshot}$, and $T_{standard}^{DNN}$ define the training time of snapshot number i and the training time of a standard DNN, respectively.

$$T_N^{Snapshot} = T_{standard}^{DNN} - \sum_{i=0}^{N-1} T_i^{Snapshot} \tag{2.14}$$

2.4.2. Developing a Deep Auto Encoder (DAE) network to detect active attacks

Even though the previous layer is trained with numerous attack samples created by the DQL-based attack generator, there might still be unknown attacks that are capable of passing the passive attack detection layer. Accordingly, a threat-hunting layer is required to exnhance the detection rate [94]. Furthermore, since the algorithm needs to detect unknown attacks, the model must be developed by unsupervised techniques.

Deep autoencoders are feed-forward multilayer neural networks consisting of an input layer, one or multiple hidden layers, and an output layer, aiming to learn data reconstructions. As a data-compression model, DAE maps the original data into a reduced-dimension representation and rebuilds the data from compressed information via a pair of encoders and decoders. In addition, the ability to discover correlations among data features makes DAEs capable of detecting FDIAs in an unsupervised manner.

Equation (2.15) shows how the encoder maps the original d dimensional vector $(x_1, x_2, \dots, x_n)^T$ to λ number of neurons in the hidden layer h, reducing the dimension ($\lambda < n$), where h_i is the activation of the i^{th} neuron; W denotes the encoder weight matrix, b and σ stand for input bias vector and nonlinear transformation function, respectively [95]. The decoder in (2.16) reconstructs back the hidden layer to the original space. The critical point in this model is minimizing reconstruction error, which is given in (2.17).

$$h_i = \sigma\left(\sum_{k=1}^{n}(W_{ik} \times x_k) + b_i\right) \tag{2.15}$$

$$y_i = \sigma\left(\sum_{k=1}^{\lambda}(W_{ik} \times h_k) + b_i\right) \tag{2.16}$$

$$error = \tfrac{1}{n}\textstyle\sum_{i=1}^{n}(x_i - y_i)^2 \tag{2.17}$$

A flat reconstruction error threshold may result in a vulnerable detection structure or even false alarms due to the dynamic nature of the attacks created by the DQL-based attack generator. The procedure for developing the adaptable DAE layer is demonstrated in Figure 2.4.

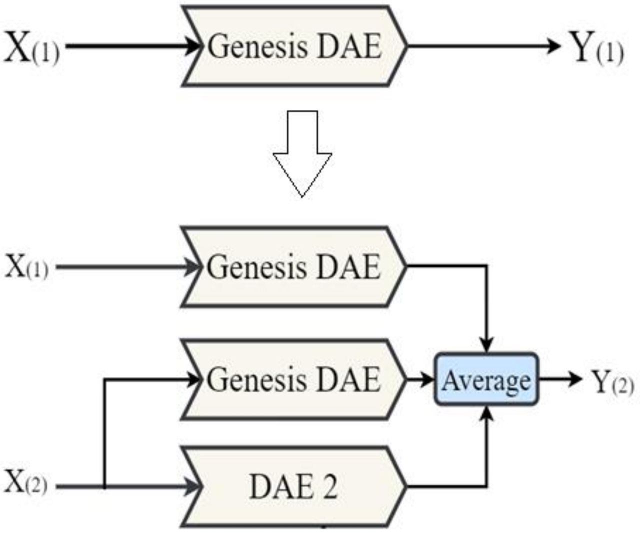


Figure 2-4. Schematic of DAE network

After training each training stage, the residuals $r_k = |x_k - y_k|$ are calculated to estimate the probability distribution of the outputs and the residuals using the Radial Basis Function (RBF) kernel. Then, the marginal distribution $M(r, y_i)$ is determined as shown in (2.18), where $P(y, r)$ denotes the joint probability distribution.

$$P(y_i, r) = M(r, y = y_i) \times \int_{-\infty}^{+\infty} P(y_i, r)\, dr \tag{2.18}$$

Next, a critical point is estimated for each y_i considering the upper and lower levels of y, where $y^{uper} = 1.15 \times y$ and $y^{lower} = 0.85 \times y$. After defining a critical function and making it constant between the defined upper and lower levels, the process is done. The proposed multilayer FDIA detection framework is summarized in Figure 2.5.

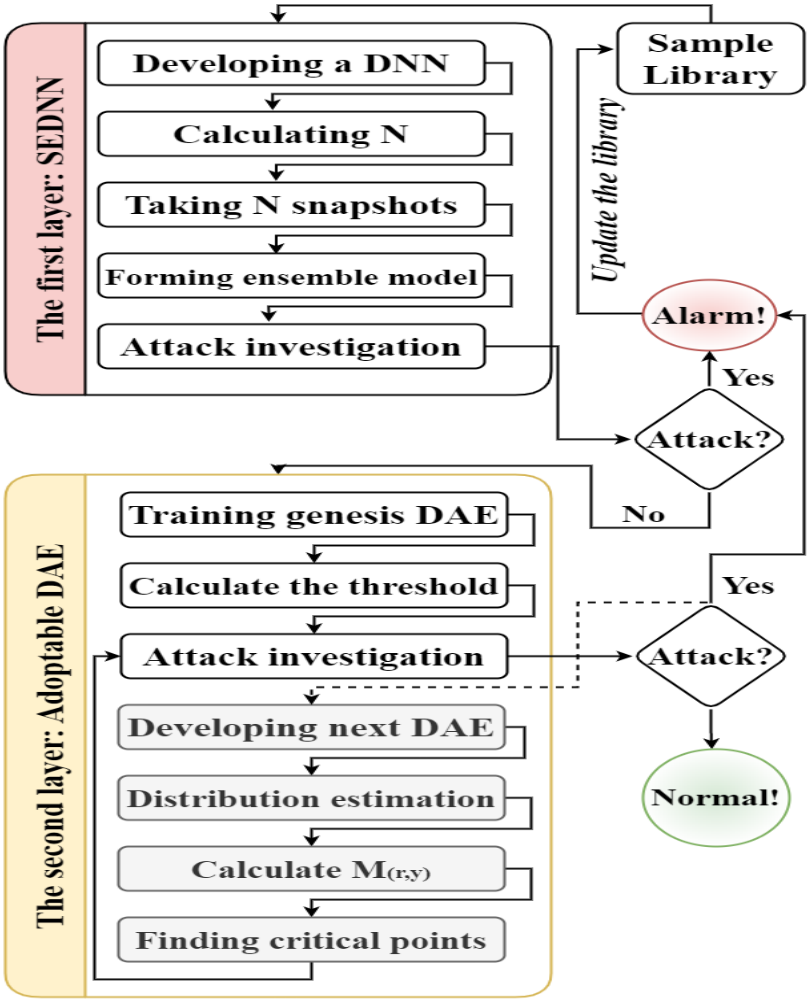


Figure 2-5. The procedure of the proposed attack detection layer

2.5. Results And Evaluations

This section first investigates the quality of the generated attacks by the DQL-based attack simulator after initializing the mathematically modeled attack scenarios demonstrating that the created attacks are able to pass the presented defense algorithms in the literature. Then, the performance of both the active and passive layers is evaluated. All experiments are performed on a subset of the Pecan Street dataset, which is available in the Non-Intrusive Load Monitoring Toolkit (NILMTK) format [89]. Finally, a simulation examination demonstrates the feasibility, necessity, and practical outcome of the proposed FDIA detection algorithms.

2.5.1. Qualification of the developed attack generator

After initializing the DQL-based attack simulator with the five sample scenarios, the training process continued for 1000 iterations. Figure 2.6 indicates the percentage of different attack scenarios during the training process. It should be noted that the allotment of all attack scenarios, including the generated attack by the intruder simulator, is just under 8%, and 92% of samples are normal.

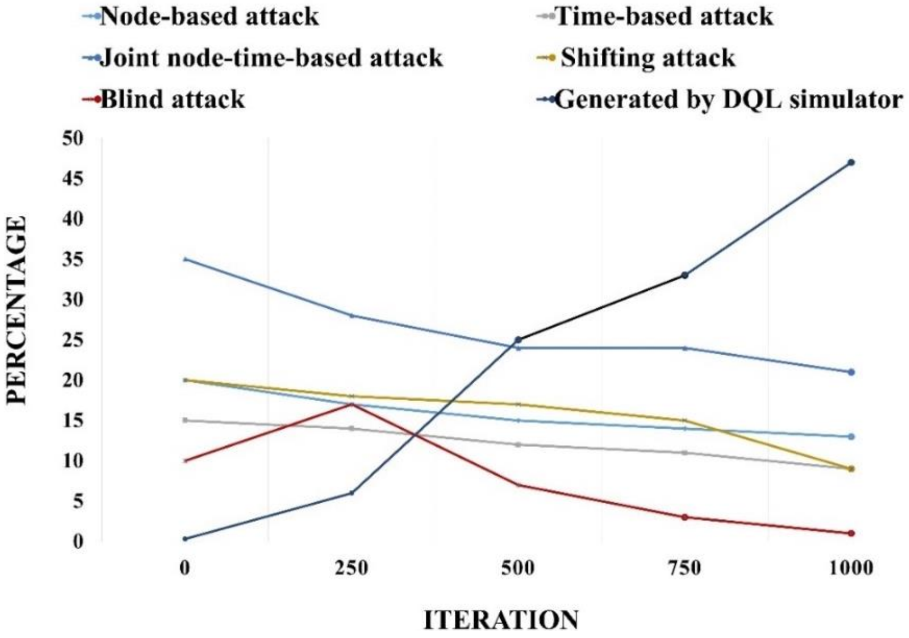


Figure 2-6. The percentage of different attack scenarios during the training stage

Three proposed FDIA detection frameworks published in top-tier journals during the last three years are selected to show their performances against the proposed attack generator framework. Artificial Neural Network (ANN), Decision Tree (DT), and Random Forest (RF) have been employed in [30] to determine attacks and anomalies in IoT sensors. Additionally, two different CNN-based mechanisms have been developed in [96] and [97], focusing on FDIAs. After some minor justifications to make the codes compatible with the dataset based on the proposed attack scenarios, the simulations show that the generated attacks can pass the detection systems. Table 2-2 demonstrates that the preceding defense frameworks cannot detect the proposed attack models with a reasonable performance. Since the dataset is highly imbalanced, the accuracy does not reflect the performance of the algorithm. Accordingly, three important metrics (i.e., Recall, Precision, and F-1 score), which are not affected by the asymmetry of the dataset, are reported to illustrate the preciseness of the model. The recall is the number of correctly positive detected attacks (TP) divided by the sum of TP and the number of samples that are falsely labeled as normal (FN). Finally, as shown in (2.19) to (2.21),different metrics including Precision, Recall and the f-1 score are formulated [98], [99]. Consequently, a new attack detection is required to detect all possible attack scenarios besides hunting unknown threats.

$$Precision = \frac{True\ positive}{True\ positive + False\ positive} \tag{2.19}$$

$$Recall = \frac{True\ positive}{True\ positive + False\ negative} \tag{2.20}$$

$$f1 - score = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{2.21}$$

Table 2-2. Performance of the recently developed FDIA detection frameworks against the proposed attack scenarios

Model	f1-score	Precision	Recall
Combined CNN [95]	0.4942	0.4548	0.5412
CNN-LSTM [26]	0.5186	0.5119	0.5255
WDCNN [97]	0.3004	0.2945	0.3066
EDNN [30]	0.4627	0.4704	0.4554
RF [30]	0.2924	0.2852	0.2998

2.5.2. Performance of the first layer: SEDNN

An ensemble of ten single models is developed using the Cosine annealing technique. The proposed SEDNN is developed using 60%, 15%, and 25% of data for training, validation, and testing, respectively. Three hidden layers are defined for each model where the number of epochs is 120, and the Cosine annealing learning rate cycling is 5. The batch size and learning rate are set at 256 and 0.01, respectively. This model uses the ReLU activation function while the drop-out rate is 0.3. Also, an SGD optimizer with a momentum of 0.90 is used in the model. Although the final attack detection accuracy of the first layer of the proposed framework is 96.9%, since the dataset is highly imbalanced with just 9% attack samples, f1-score, Precision, and Recall are reported to clarify the algorithm's performance.

Table 2.3 summarizes the results and compares the performance of the developed SEDNN and other techniques, including CNN_LSTM, random bagging Ensemble of DNN (EDNN), Wide Deep CNN (WDCNN), and RF. Additionally, the superiority proposed algorithm in terms of f1-score, Precision, and Recall is investigated, making a comparison with works in [31] and [97].

Table 2-3. Performance Comparison of different methods

Model	f1-score	Precision	Recall
The proposed SEDNN	0.9566	0.9631	0.9502
CNN-LSTM [31]	0.8967	0.9044	0.8892
WDCNN [97]	0.8953	0.9001	0.8906
EDNN [31]	0.91879	0.9372	0.9011
RF [31]	0.7339	0.7424	0.7256

Moreover, Table 2.4 demonstrates the performance of all algorithms mentioned above in detecting each attack scenario. As illustrated, although the developed SEDNN algorithm performs better in detecting attacks than other investigated methods, the novel attacks generated by the DQL-based intruder simulator (DQL-attacks) still escape the first defense layer. Accordingly, the second layer is vital to identify the attacks that the algorithm has not detected before.

Table 2-4. The detection rate of different techniques based on the attack type

	The detection rate of each attack scenario by:			
Attack type	SEDNN	CNN-LSTM	WDCNN	EDNN
Node-based	97 %	83 %	85 %	90 %
Time-based	97 %	85 %	84 %	89 %
Joint node-time-based	96 %	92 %	91 %	88 %
Shifting	93 %	89 %	86 %	88 %
Blind	99 %	98 %	97 %	98 %
DQL-attacks	**71 %**	**63 %**	**64 %**	**68 %**

2.5.3. Performance of the second layer: DAE

The normal data that passes through the first layer is then injected into the second layer, aiming to detect any unknown threat. In this stage, the data splitting procedure assigns 65% and 15% of the entire dataset to the training and validation stages, while 20% of the data is remained to test the developed model. Four hidden layers are embedded, while the number of neurons is reduced layer by layer based on the comparison factor. Drop-out is also utilized at the rate of 0.15, mitigating the risk of overfitting and improving generalization error. An Adam optimizer is utilized to compile the DAE, and the learning rate and batch size are set at 0.001 and 512, respectively. Finally, the algorithm calculates validation errors for the first training round to define a threshold.

The fixed threshold is set as shown in (2.22), where Interquartile Range (IQR) stands for the interquartile range. Once the test error exceeds the reconstruction error threshold, the model sends an attack signal. Then, the threshold is adjusted, as mentioned in the previous section.

$$\tau = Median + \frac{3 \times IQR}{2} \tag{2.22}$$

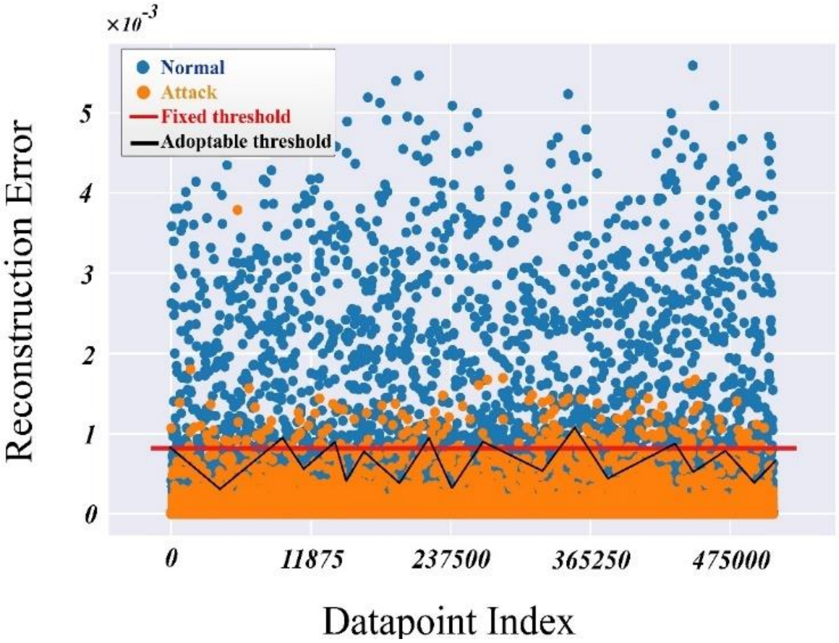


Figure 2-7. Reconstruction error distribution

The threat-hunting layer is trained 500 epochs while the validation test is monitored to avoid overfitting. Figure 2.7 shows the reconstruction error of 475,450 observations. The least FP rate obtains when τ is 0.815×10^{-3}. The same setup is then trained to utilize the adoptable reconstruction error.

Later, the model is tested by launching the total number of 43,480 FDIAs at diverse time slots during midnight, morning off-peak hours, midday and afternoon peak hours, and mid-load hours, with various false data injection magnitudes. The FP rate is a critical metric of attack detection in smart grids due to the severe economic and forensic consequences of a mistaken alarm. The FP rate of the proposed threat hunting layer is 0.097, along with the model accuracy of 98.82%, indicating outstanding performance.

Finally, Table 2.5 makes a performance comparison among the second layer of the proposed method, with a flexible Bayes Classifier [100], and two well-known anomaly detection algorithms, including One-Class Support Vector Machine (OCSVM) and Isolation Forest (IF).

Table 2-5. Performance comparison among the above-mentioned methods in general and against DQL-based attacks

Model	Accuracy	FP (%)	Detecting DQL-attacks(%)
The proposed Adoptable-DAE	0.9882	0.97	96.245
The proposed Fixed -DAE	0.9433	1.45	89.761
Flexible Bayes classifier	Not reported	1.92	64.245
OCSVM	0.8249	9.21	43.861
IF	0.7516	13.41	39.458

2.5.4. Smart Grid Performance Under Attack

Various system parameters are investigated in normal and under-attack situations illuminating the effect of employing the proposed defense mechanism from the technical and economic points of view. Ten units are randomly selected to investigate four different scenarios over 24 hours. Both defense layers are deactivated in the first scenario to perceive FDIAs' potency. In the second and third scenarios, just the SEDNN and the DAE attack detection algorithms are active, respectively. Finally, full protection employing both developed algorithms is provided to the system in the fourth scenario. Table 2.6 summarizes the results indicating the peak load (kW), Peak to Average Ratio (PAR), and profit reduction of the utilities. The absence of attack detection mechanisms results in an average reducing the electricity bill for the attacker to 43%, which is not noticeable in conventional inspections. Consequently, the net profit of the power supplier dropped 76%. The absence of the proposed framework results in chaos in power scheduling and routing, especially in the neighborhood area, affecting peer-to-peer electricity trading among the end-users.

Table 2-6. Effect of employing the developed defense mechanisms

Scenario	Peak (kW)	PAR reduction	Profit reduction
Normal operation	6.024	-	-
Scenario 1	2.419	93%	76%
Scenario 2	3.957	58%	41%
Scenario 3	5.193	37%	26%
Scenario 3	5.933	0	1.1 %

2.5.5. Network Parameters

This section investigates a real-world network simulation to indicate the network's performance that operates with the developed frameworks. The communication network and smart grid structure are modeled using ns-3 and GridLAB-D, respectively, while the Framework for Network Co-Simulation (FNCS) operates as an integrator between both simulators. Furthermore, various necessary communication and grid configurations are outlined and appended into a preprocessing module.

All created attack scenarios are scheduled and stored in a library specifying their target. As the heart of the simulator, a model engine manages and executes all the processes. Moreover, after developing the simulator, the proposed attack detection framework is embedded into the model engine to discover its performance in a real-world environment. Figure 2.8 shows the architecture of the simulator and simulation parameters. Furthermore, two neighborhoods are created as a NA, ensuring the system's scalability. The first neighborhood contains fourteen MA, and the rest eleven belong to the second one.

Two identical network topologies are also created using the developed algorithms in [31] and [97] as the pair model to compare network performances, including throughput and delay. Network throughput is a metric that indicates the amount of successfully transmitted data between transceivers in a timespan. Additionally, the average time of receiving the entire information at the end node is network delay.

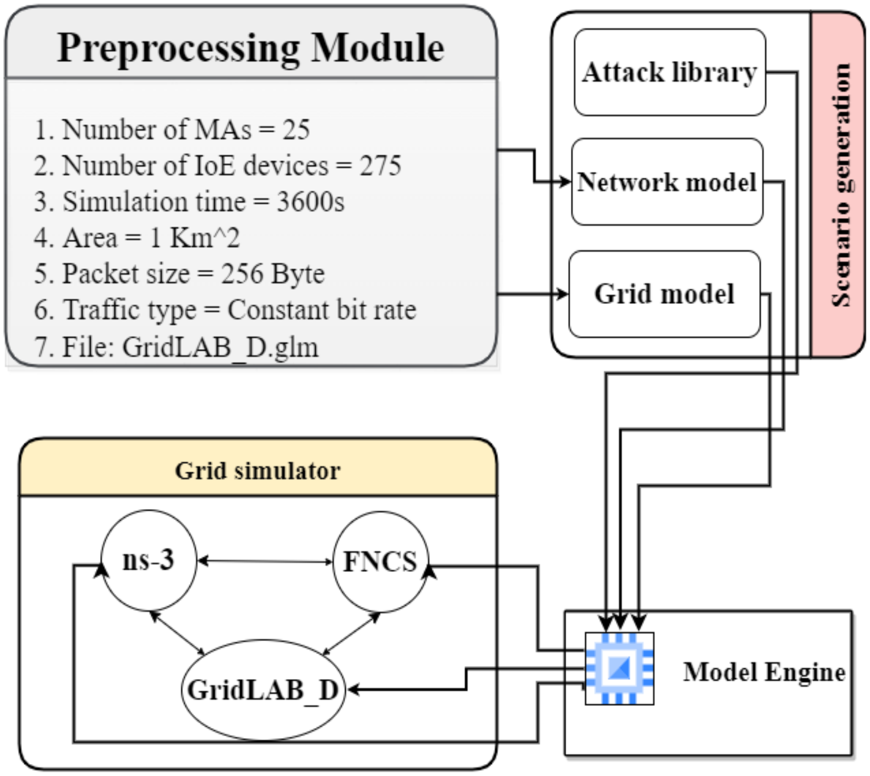


Figure 2-8. The structure of the GridAttackSim simulator

As Table 2.7 demonstrates, since the proposed method employs a DQL-based attack generation engine, most of the possible FDIAs have been classified as the detection framework at the training stage, resulting in better network performance.

Table 2-7. Network parameters improvement with different algorithms

Method	Data rate (pkts/sec)	Throughput (kbps)	Delay (ms)
The proposed Model (SEDNN + DAE)	2	252	126
	6	271	164
	9	364	197
CNN-LSTM [101]	2	185	258
	6	231	308
	9	262	361
WDCNN [102]	2	192	212
	6	219	278
	9	269	349

2.6. Conclusion

This chapter has developed a deep Q-Learning-based false data injection attack generator using various possible attack scenarios. Moreover, a two-layer attack detection framework was developed using a snapshot ensemble deep neural network and deep autoencoder to detect passive and active threats, respectively. The first layer indicated an outstanding performance with an accuracy are 98.02%. The second layer that was responsible for threat hunting could detect unknown attacks where the FP rate was remarkably low at 2.9%. Ultimately, the proposed attack modeling and detection framework were simulated using a combination of ns-3, FNCS, and GridLAB-D simulators. Additionally, the same setup was modeled based on two different developed algorithms to make a comparison between the performances. The result showed the proposed framework's superiority in network throughput and end-to-end delay.

Chapter 3

Optimizing Scheduling Policy in Smart Grids Using Probabilistic Delayed Double Deep Q-Learning (P3DQL) Algorithm

3.1. Introduction

Demand for electrical energy is increasing dramatically worldwide [103]. Since the consumption of fossil fuels is increasing at 1.16% per year [104], replacing conventional power plants with renewable energy resources is inevitable, which requires novel approaches. One of the key challenges in power systems is efficiently harmonizing demand and generation. Although economic dispatch and unit commitment are two well-investigated problems aiming to deal with the concern mentioned above, high penetration of distributed resources, Electric Vehicles (EVs), advanced metering infrastructures, and telecommunication networks open new challenges and perspectives in addressing scheduling problems [105], [106]. Furthermore, playing an active role by consumers in Demand Response (DR) programs makes utilities capable of orchestrating the balance between demand and supply sides, especially during peak hours, to prevent running expensive power plants [107].

Conventional methods fail to solve new scheduling problems in smart grids due to complex non-linear environments, in addition to their incapability of processing big data generated by different components of the smart grids [108]. Likewise, heuristics and mathematical algorithms rely on explicit environment models and precise forecasts of various types of uncertainties, including technical, economic, and weather uncertainties, that cannot be fulfilled in real-world problems.

Despite the fact that supervised learning methods are less subordinate to accurate forecasting data and model uncertainties, they still suffer from limited approximation capability and slow convergence in a large-scale dynamic and decentralized environment [47]. However, since scheduling in the new generation of smart grids is a decision-making NP-hard problem, Reinforcement Learning (RL) is a well-suited algorithm to solve them due to the capability of learning optimal behavior by making a trade-off between exploration and exploitation [109].

Artificial intelligence 2 (AI 2.0) and the Internet of Energy (IoE) are advanced concepts that play crucial roles in the ease of solving scheduling problems in fully interconnected large-scale networks with dynamic and uncertain environment models. While an IoE framework collects data from all energy kits providing intelligent real-time monitoring and dynamic control, AI 2.0 combines data-driven and experience-based engines by integrating natural and artificial intelligence to make the optimal decision [110]. RL and Deep Reinforcement Learning (DRL), which are two well-known subcategories of AI 2.0, demonstrate outstanding performance that exceeds human intelligence. For example, in March 2016, the world Go champion was defeated by the AlphaGo algorithm, which had been developed by Google DeepMind based on DRL [111].

Subsequently, a large volume of research has been conducted to solve scheduling optimization problems in the new generation of smart grids, where all energy components are fully interconnected, collecting and sharing big data over an IoE framework to be analyzed by AI 2.0 family algorithms. A model-free RL algorithm has been proposed in [49], guarantying data confidentiality and consumers' privacy. In [48], a multi-agent optimizing scheme has been developed based on RL for solving routing and scheduling problems. The authors in [53] suggested a data-driven DRL-based scheduling algorithm for microgrid energy optimization. Remani et al. [52] presented an RL-based solution to minimize electricity costs for the end-users considering Photovoltaic (PV) generation uncertainty. In [51], a model-free Q-Learning framework has been developed to schedule the operational time of home appliances considering a rooftop PV system and an energy storage unit. The authors in [50] developed a Q-Learning algorithm to schedule multiple appliances in a smart home, preserving customer welfare and preferences.

Many challenges are associated with RL, including the curse of dimensionality, lack of scalability, poor generalization, and limited non-linear representation capability that make this method disqualified from solving scheduling problems in a real-world smart grid environment [54]. On the other hand, DRL algorithms have recently accomplished extraordinary breakthroughs exploiting Deep Neural Network (DNN) strengths, involving the ability to handle unstructured data, no need for feature engineering, and non-linear representation capability, to name a few [55]. Consequently, taking advantage of deep learning strengths, Deep Q-Learning (DQL) swamps RL deficiencies.

DQL has been utilized in [62] and [61], specifying the optimal scheduling policy to schedule heterogeneous virtual machines' workflow. David et al. [60] proposed a DQL-based solution learning optimal policy for the operation of energy entities in a microgrid to minimize cost. In [59], a scheduling problem aiming to address the supply-demand mismatch in microgrid energy trading has been solved using DRL. A model-free DQL approach has been introduced in [58] to determine an optimal strategy for real-time scheduling of EV charging. The authors in [48] compared DQL and Deep Policy Gradient (DPG) methods, where DPG demonstrated better performance. The superiority of DPG over DQL in this optimization problem originated from overestimation bias in Q-Learning. As a solution, Double DQL (DDQL) reduces the positive bias of Q-Learning, which has not been investigated in [57]. Chung et al. [56] proposed a DPG algorithm for scheduling the operation of home appliances while preserving their privacy. Two versions of DPG, including Centralized Deterministic DPG (C-DDPG) and Distributed Deterministic DPG (D-DDPG), have been developed while comparing results with a stochastic weight averaging algorithm. As results show, sample efficiency and usability are still two key concerns in the proposed methods. Moreover, the difference in reward of four reported buildings reveals the model suffers from high variance. It is worth mentioning that policy gradient methods face the risk of trapping in a local optimal; also, a lot of training time is required to reach the global result [63].

Neither formerly mentioned papers nor other related studies in the literature attempted to reduce the positive bias of DDQL while improving the negative bias simultaneously. Moreover, the learning efficiency and space and sample complexity of the past works need to be enhanced. Therefore, this chapter aims to develop an algorithm that makes a trade-off between positive and negative bias by synchronously reducing overestimation and underestimation. Additionally, the designed algorithm must be capable of taking advantage of positive and negative biases in case of the need for better exploration and exploitation, respectively. The ultimate objective is to improve learning efficiency while reducing space and sample complexity since these characteristics incredibly affect time-efficient and cost-effective solutions in discrete action spaces and problems with continuous and large action spaces. Consequently, the main contributions of this chapter are summarized as follows.

I. For the first time, a novel DRL-based algorithm named Probabilistic Delayed Double Deep Q-Learning (P3DQL) is specially developed for the scheduling problem, adapting to smart grids' environment. The proposed algorithm addresses efficiency and bias challenges simultaneously.

II. The selection order of estimators in DDQL is converted in a probabilistic manner, eliminating the underestimation challenge. Accordingly, a trade-off between positive and negative biases is made.

III. The developed algorithm is modeled as a Probably Approximately Correct in Markov Decision Processes (PAC-MDP), enhancing learning efficiency and reducing sample complexity making the algorithm capable of handling problems with a large action space, including scheduling problems in smart grids.

IV. A multi-layer scheduling mathematical model with low numerical error is proposed, which comprehensively covers from a single Nano Area (NA) to a Neighborhood (NH) and a Wide Area Network (WAN), including share storage units, PVs, and considering different tariff types and customer preferences. Subsequently, the problem is formulated as a Markov decision process (MDP) and solved by the P3DQL algorithm.

The remaining of this chapter is structured as follows. Sections 3.2 and 3.3 present the system mathematical and MDP model of the problem. The proposed P3DQL algorithm is introduced in section 3.4. Results are provided in section 3.5. Finally, a brief conclusion is presented in section 3.6.

3.2. Mathematical Formulation of the Proposed Multi-Layer Model

The proposed model is an autonomous and dynamic framework enabling optimal scheduling at different levels, including NAs, NHs, and WANs. Before diving into formulating the optimization problem, the methodology process is briefed in Figure 3.1. All components are interconnected in three different layers taking advantage of a bi-directional communication enabled by the IoE. A micro area is a smart home that consists of various appliances as well as a Private PV (PPV) cell and/or EV(s). Consequently, two or more units can form a neighborhood that includes an Electricity Storage System (ESS), Shared PV (SPV) modules, and an Electric Vehicle Charging-station (EVC). Finally, a wide network area can be composed of several neighborhoods.

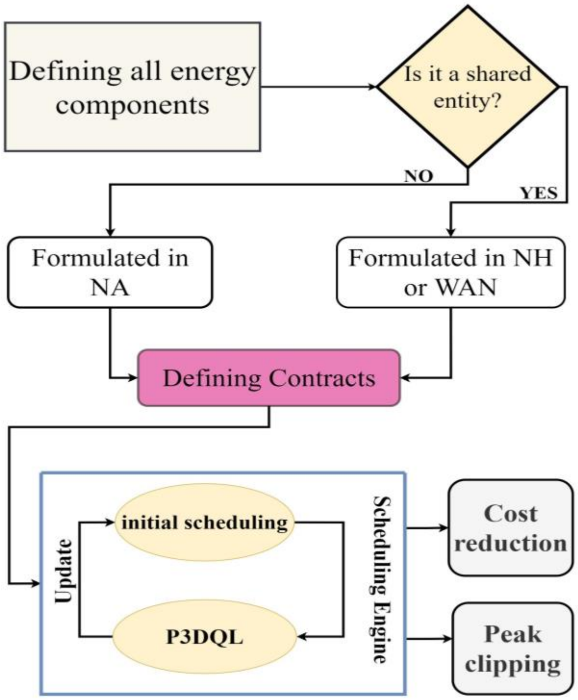


Figure 3-1. Optimal scheduling process

The proposed model has a highly dynamic environment in terms of forming layers. Each NA can participate in one or multiple NHs simultaneously, while the configuration may change in different time slots based on their contracts for electricity trading. Likewise, each ESS, SPV, and EVC can be exploited by one or more NHs. Moreover, each WAN is formed by two or more NHs. Equations (3.1) and (3.2) define the matrix of WAN contracts $W_{M\times 1}$ in a specific time slot, where $H \in \mathbb{N}^{N\times 1}$ denotes a set of N participated smart homes, $R \in \mathbb{N}^{3\times 1}$ stands for the matrix of shared unit vectors demonstrating the type and capacity of the modules, also $\Delta \in \mathbb{N}^{M\times N}$ and $\delta \in \mathbb{N}^{M\times 3}$ are activation matrixes for micro areas and shared units, respectively.

$$[W]_{M\times 1} = ([\Delta]_{M\times N} \times [H]_{N\times 1}) + ([\delta]_{M\times 3} \times [R]_{3\times 1}) \tag{3.1}$$

$$\begin{bmatrix} w_1 \\ \vdots \\ w_M \end{bmatrix} = \begin{bmatrix} \Delta_{11} & \cdots & \Delta_{1m} \\ \vdots & \ddots & \vdots \\ \Delta_{m1} & \cdots & \Delta_{mn} \end{bmatrix} \times \begin{bmatrix} h_1 \\ \vdots \\ h_n \end{bmatrix} + \begin{bmatrix} \delta_{11} & \delta_{12} & \delta_{13} \\ \vdots & \vdots & \vdots \\ \delta_{m1} & \delta_{m2} & \delta_{m3} \end{bmatrix} \times \begin{bmatrix} ESS \\ SPV \\ EVC \end{bmatrix} \tag{3.2}$$

The objective is to reduce consumption during peak hours, reducing peak-to-average ratio and load variance while minimizing costs at all levels. Let $RP_t = \Omega_t \Phi_t \Pi_t \lambda_t$ represents the retail price in time slot t under the given incentive/penalty factor Ω, tiered pricing factor Φ, and regular price of electricity λ. Additionally, the contract coefficient Π is defined based on the terms and conditions of energy trading between units during a specific period of time ($\Pi = 1$ means there is no contract). Equations (3.3) and (3.4) describe the incentive and penalty factors, where P_t^{cum} is the cumulative consumption up to time frame t, also lower and higher pricing threshold of net load are defined by θ_1^t and θ_2^t , respectively. It should be noted that incentive and penalty factors and both thresholds are estimated to calculate the real-time price considering the contract coefficient.

$$\Omega_t = \begin{cases} \overline{\Omega} & \overline{\Omega} > 1 \overset{if}{\Rightarrow} t \in peak\ hours \\ \underline{\Omega} & 0 < \underline{\Omega} < 1 \overset{if}{\Rightarrow} t \in off - peak \\ 1 & \underline{\Omega} = 1 \ \overset{if}{\Rightarrow} t \in mid - load\ hours \end{cases} \tag{3.3}$$

$$\Phi_t = \begin{cases} \underline{\Phi} & 0 < \underline{\Phi} < 1 \overset{if}{\Rightarrow} P_t^{cum} < \theta_1^t \\ 1 & \Phi = 1 \overset{if}{\Rightarrow} \theta_1^t \le P_t^{cum} \le \theta_2^t \\ \overline{\Phi} & \overline{\Phi} > 1 \quad \overset{if}{\Rightarrow} P_t^{cum} > \theta_2^t \end{cases} \tag{3.4}$$

3.2.1. Scheduling problem formulation in NA

There are three types of costs in each NA that need to be considered and formulated, including monetary cost f_{11}, instruments degradation cost f_{12}, and the consumer discomfort cost f_{13}. Consequently, the formulation of the primary objective functions of the NA level is as follows.

$$min(F_1) \tag{3.5}$$

$$F_1 = (f_{11}, f_{12}, f_{13}) \tag{3.6}$$

$$f_{11} = \sum_{t=1}^{T} (\sum_{j=0}^{m} (RP_t)\mathrm{P}_{jt} + {}^{NA}_{y}\xi_t\, {}^{NA}_{y}l_t(\Gamma_t^i - \Gamma_t^e)) \tag{3.7}$$

$$f_{12} = \sum_{j=0}^{m} D_j\big(\check{T}; \vartheta, k\big) = \sum_{j=0}^{m} \frac{\kappa_j}{\vartheta_j} (\frac{\check{T}_j}{\vartheta_j})^{\kappa_j - 1} . e^{-(\frac{\check{T}_j}{\vartheta_j})^k} \tag{3.8}$$

$$f_{13} = \sum_{j=0}^{m} \eta_j{}^{(\sum_{t=1}^{T} |\zeta_t^d - \zeta_t^a|)} + \sum_{t=1}^{T} |{}^{o}_{a}c_t - {}^{o}_{d}c_t| \tag{3.9}$$

The cost minimization problem at the NA level must also fulfill the following constraints:

$$\textstyle\sum_{t=1}^{T} \sum_{j=1}^{m} P_{jt} = L_{total}, \forall t \in \mathbb{N}, \forall j \in \mathbb{N} \tag{3.10}$$

$$0 \leq \left|\Gamma_t^i - \Gamma_t^e\right| \leq L_{total}, \forall t \in \mathbb{N}, \Gamma_t \in \mathbb{R} \tag{3.11}$$

$$D_j\big(\check{T}; \vartheta, k\big) > 0, \forall \check{T} \in \mathbb{N}, \forall \vartheta \in \mathbb{R}^+, \forall \kappa \in \mathbb{R}^+ \tag{3.12}$$

$$\xi_t > 1,\ \eta_j > 1, \forall t = [1{:}T] \in \mathbb{N}, \forall j = [1{:}m] \in \mathbb{N} \tag{3.13}$$

$$\zeta_t^d \in \{0,1\}, \zeta_t^a \in \{0,1\}, \zeta_t^d \neq \zeta_t^a\ , \forall t = [1{:}T] \in \mathbb{N} \tag{3.14}$$

The first objective f_{11} aims to minimize the bill considering; the imported power Γ_t^i, the exported power Γ_t^e, and the transmission fee coefficient ${}^{NA}_{y}\xi_t$ and loss coefficient ${}^{NA}_{y}l_t$ at each time frame t in y^{th} NA, where T and m are the total number of time slots and appliances, respectively. Besides, ${}^{NA}_{y}L_{total}$ is the total power consumption of NA number y, and P_{jt} indicates the power consumption of j^{th} device during each time slot. The second objective focuses on the degradation cost of electrical devices based on the Weibull distribution function [112]. Accordingly, $\check{T}_j$ defines the cumulative time of operating j^{th} appliance, ϑ is the Weibull scale parameter, and κ stands for the Weibull shape parameter. The third objective is interested in discomfort minimization, taking operational delay and desired indoor temperature into account at the same time. Also, η indicates the importance of on-time operation for each appliance, considering the difference between the desired operational status ζ_t^d and actual status ζ_t^a.

3.2.2. Scheduling problem formulation in NH

The mathematical scheduling model of an NH is presented, comprising multiple NAs h_i and shared units $R^j \in \mathbb{N}^{3\times1}$, where i and j are the IDs of participated NAs and set of RES in the given NH. Since all NAs are considered as optimized energy components in the prior level, they are inspected as black boxes with hidden internal mechanisms that just swap electricity with the NH. The first objective f_{21} intends to minimize transaction costs among all participated units, also other NHs and WANs. Minimizing ESS loss P_{ESS}^{loss}, and PV loss P_{PV}^{loss} [113] are the key goals of the second objective, f_{22}. It should be pointed out that ESSs involve both fixed and mobile electricity storage (EV). The third objective f_{23} minimizes EVC cost considering various potential generation sources using a non-linear cost function based on the output power of the unit (${P_{EVC}}_t$), while the fourth objective investigates the maintenance and degradation costs f_{24}. Ultimately, the scheduling problem at the NH level is formulated as follows.

$$min(F_2) \tag{3.15}$$

$$F_2 = (f_{21}, f_{22}, f_{23}, f_{24}) \tag{3.16}$$

$$f_{21} = \sum_{t=1}^{T}(\sum_{w\in\Phi}\sum_{k\in\Phi} {}^{NH}_{z}l_t^{wk}.P_t^{wk}) + {}^{NH}_{y}\xi_t\,{}^{NH}_{y}l_t\Gamma_t^* \tag{3.17}$$

$$f_{22} = \sum_{t=1}^{T}\big(-\alpha({\lambda_{ESS}}_t^{\alpha} - 1)\big){P_{bat}}_t + (1 - {\lambda_{PV}}_t)\hat{A}_{pv}\, I_t\tau_t \tag{3.18}$$

$$f_{23} = \sum_{t=1}^{T}\beta_1 {P_{EVC}}_t^{\;2} + \beta_2 {P_{EVC}}_t + \beta_3 \tag{3.19}$$

$$f_{24} = \sum_{r\in R} D_r\big(\check{T};\vartheta,k\big) = \sum_{r\in R} \frac{\kappa_r}{\vartheta_r}\big(\frac{\check{T}_r}{\vartheta_r}\big)^{\kappa_r-1}.\, e^{-(\frac{\check{T}_r}{\vartheta_r})^k} \tag{3.20}$$

Equations (3.21) to (3.28) illustrate the constraints that must be satisfied in the scheduling problem.

$$\varPhi \in \{h_1,\cdots,h_n\} \cup \{RES_i, PV_i, EVC_i\}, \forall n \in \mathbb{N},, \forall j \in \mathbb{N} \tag{3.21}$$

$$0 \leq {}^{NH}_{z}l^{wk}_{t} < 1, \{{}^{NH}_{y}\xi_t, {}^{NH}_{y}l_t\} \geq 1, \qquad \varGamma^*_t \geq 0,, \forall t = [1{:}T] \tag{3.22}$$

$$\lambda_{ESS_t} < 1, \lambda_{PV_t} < 1, \forall t = [1{:}T] \in \mathbb{N} \tag{3.23}$$

$$SOC_{ESS_{t+1}} \leq SOC_{ESS_t} + \big(-\alpha(\lambda_{ESS_t}^{\ \alpha} - 1)\big)P_{bat_t}, \forall t = [1{:}T] \in \mathbb{N} \tag{3.24}$$

$$SOC_{ESS_{min}} \leq SOC_{ESS_t} \leq SOC_{ESS_{max}} \tag{3.25}$$

$$\tau_t = 1 - 0.005({}_{outside}{}^{o}c_t - 25), t = [1{:}T] \in \mathbb{N} \tag{3.26}$$

$$\beta_j > 0, P_{EVC_t} \geq 0, j \in \{1,2,3\},, t = [1{:}T] \in \mathbb{N} \tag{3.27}$$

$$D_r\big(\check{T};\vartheta,k\big) > 0, \forall \check{T} \in \mathbb{N}, \forall \vartheta \in \mathbb{R}^+, \forall \kappa \in \mathbb{R}^+, r \in R \tag{3.28}$$

Where Φ is a set of NAs joined in the NH, ${}^{\mathrm{NH}}_{\mathrm{z}}\mathrm{l}^{\mathrm{wk}}_{\mathrm{t}}$ stands for the power loss of swapping electricity from w to k in z^{th} NA, $\mathrm{P}^{\mathrm{wk}}_{\mathrm{t}}$ demonstrates the quantity of transferred power between entities during time slot t. Moreover, ${}^{\mathrm{NH}}_{\mathrm{y}}\xi_{\mathrm{t}}$, ${}^{\mathrm{NH}}_{\mathrm{y}}\mathrm{l}_{\mathrm{t}}$, and Γ^*_{t} are transmission fee coefficient, loss coefficient, and exchanged power from/to the NH, respectively. Furthermore, $\lambda_{\mathrm{ESS_t}}$ denotes the overall efficiency coefficient of the ESS, $\lambda_{\mathrm{PV_t}}$ represents the overall efficiency of the PV, α symbolizes the charging ($\alpha = 1$) and discharging ($\alpha = -1$) status, $\mathrm{SOC_{ESS_t}}$ defines the state of charge of the storage at time slot t. Finally, ${}_{\mathrm{outside}}{}^{\mathrm{o}}\mathrm{c}$ indicates the outside temperature in Celsius, $\mathrm{P_{EVC_t}}$ stands for the total power of the charging station at time slot t, and $\mathrm{D_r}$ denotes the degradation of each $\mathrm{r} \in \{\mathrm{RES}, \mathrm{PV}, \mathrm{EVC}\}$.

3.2.3. Scheduling problem formulation in WAN

From the perspective of the WAN level, all different types of costs and losses are previously optimized considering layers' requirements and their subdivision entities. Consequently, scheduling problems in WAN layers focused on minimizing electricity trading costs, as given in Equation (3.29), where Y is the set of contributed WANs in the scheduling optimization plan.

$$min\sum_{t=0}^{T}(\sum_{w\in \Upsilon}\sum_{k\in \Upsilon} {}^{WAN}_{s}\xi_t . {}^{WAN}_{s}l_t^{wk} . P_t^{wk}) \tag{3.29}$$

3.3. MDP Formulation of the Proposed Multi-Layer Model

The MDP formulation of the multi-layer power system model is presented in this section. The proposed formulation is configured in three different stages whereby each level is considered as a single energy entity in the next level. Initially, five major controllable appliances, including Air Conditioner (AC), Washing Machine (WM), Dryer (DY), Dishwasher (DW), and EV, are deemed in a smart house as a NA. Equation (3.30) illustrates the state space of a NA.

$$s^{NA}{}_t = \{l^{AC}{}_t, l^{WM}{}_t, l^{DY}{}_t, l^{DW}{}_t, l^{EV}{}_t, E^{PV}{}_t, RP_t\} \tag{3.30}$$

Where $l^j{}_t$ denotes the power consumption of the appliance j during time slot t, $E^{PV}{}_t$ represents the state of the PV, including internal dispensing or exporting to the upper level, and RP_t indicates the retail price at t. From an NH point of view , ${}^{NA}_{i}E_t$ is the state of i^{th} NA defines whether the NA exports or imports electricity during t. Likewise, states of ESS, SPV, and EVC are defined as ${}^{ESS}_{i}E_t$, ${}^{SPV}_{i}E_t$, and ${}^{EVC}_{i}E_t$, respectively. Finally, from the WAN standpoint, the state space contains the energy status of each NH ${}^{NH}_{i}E_t$, besides the swapping price between i^{th} and j^{th} NH ${}^{W}_{ij}RP_t$. Equations (3.31) and (3.32) illustrate the state space of NHs and WANs, respectively.

$$s^{NH}{}_t = \left\{ {}^{NA}_{1}E_t, \dots, {}^{NA}_{i}E_t, \ \ {}^{ESS}_{i}E_t, {}^{SPV}_{i}E_t, {}^{EVC}_{i}E_t, {}^{N}_{ij}RP_t \right\} \tag{3.31}$$

$$s^{WAN}{}_t = \left\{ {}^{NH}_{1}E_t, \dots, {}^{NH}_{i}E_t, {}^{W}_{ij}RP_t \right\} \tag{3.32}$$

The optimal action of each entity depends on the dedicated agent to maximize its reward.

Equation (33.3) defines the action space of non-adjustable appliances, including WM, DY, and DW.

$$A^{app} = \{On, Off, Hold\}, \forall app \in \{WM, DY, DW\} \tag{3.33}$$

EVs are equipped with Grid-to-Vehicle (G2V) and Vehicle-to-Grid (V2G) technologies. Correspondingly, the action space of EVs is as follows.

$$A^{EV} = \{G2V, V2G, Disconect\} \tag{3.34}$$

The action space of AC is discretized into $2n + 1$ levels where n is the temperature degrees of freedom given by the user. The agent is authorized to decrease or increase the temperature by n degrees, one degree at a time, as Equation (3.35) shows. T^{NA} denotes the current temperature, and $^{\pm}xT^{NA}$ indicate x degree of temperature change. The same logic applies to the ESS and EVC as indicated in Equations (3.36) and (3.37) where E^{ESS} is the energy level of the shared unit, and q defines the number of permitted changes in the SOC once at a time, Γ^{EVC}denotes the price of supplied electricity by EVC, and $+r\Gamma^{EVC}$ stands for increasing price by r unit of currency.

$$A^{AC} = \{-nT^{NA}, \dots, -1T^{NA}, T^{NA}, +1T^{NA}, \dots, +nT^{NA}\} \tag{3.35}$$

$$A^{ESS} = \{-qE^{ESS}, \dots, -1E^{ESS}, E^{ESS}, +1E^{ESS}, \dots, +qE^{ESS}\} \tag{3.36}$$

$$A^{EVC} = \{\mathit{\Gamma}^{EVC}, +1\mathit{\Gamma}^{EVC}, \dots, +r\mathit{\Gamma}^{EVC}, Off\} \tag{3.37}$$

Both private and shared PV agents follow the same action space, Equation (3.35) explains, where the system is capable of switching the terminus consumption from internal to external (IE) or from external to internal (EI) units. Furthermore, the PV will be detached once output power drops under the economic power threshold. The action space of the rest of the entities, including the entire NA, NH, and WAN, also follow the same method.

$$A^{Z} = \{IE, EI, Off\}, \forall Z \in \{PV, SPV, NA, NH, WAN\} \tag{3.38}$$

The key objective of the problem is to find the optimal policy $\pi(a_t|s_t)$, maximizing the total expected reward as follows.

$$\max_{\pi} J(\pi) = \mathrm{E}_{\tau\sim\pi}\left[\sum_{t=o}^{T-1} {R^{total}}_{t} . \gamma^{t}\right] \tag{3.39}$$

Where $\tau = \{s_0, a_0, s_1, \dots, a_{T-1}, s_T\}$ denotes the agent's trajectory, $\mathrm{E}_{\tau\sim\pi}[.]$ represents the expected value over τ, the probability of transition from s_{t+1} to s_t by action a_t defines by the probability transition function $P(s_{t+1}|s_t, a_t)$, $\gamma \in (0,1]$ is the discount factor indicating the importance of future rewards. Moreover, ${R^{total}}_{t}$ indicates the overall reward, defined independently for each level, as shown in Equations (3.40) to (3.42).

$$R_{NA}^{total} = \sum_{t=0}^{T} \sum_{i\in D} {r_i}_t \,, D \in \{app, EV, AC, PV\} \tag{3.40}$$

$$R_{NH}^{total} = \sum_{t=0}^{T} \left(\sum_{i\in NA} {r_i}_t + \sum_{j\in G} {r_j}_t\right), G \in \{ESS, SPV, EVC\} \tag{3.41}$$

$$R_{NH}^{total} = \sum_{t=0}^{T} \sum_{i\in NH} {r_i}_t \tag{3.42}$$

Ultimately, this chapter presents a comprehensive reward function that applies to all entities based on the feasible conditions indicated in Equation (40).

$$r_t = \begin{cases} -(RP_t\Psi_t + \sum\limits_{i\in\mho} \varphi_i \left|\varpi_i^a - \varpi_i^d\right|) & \varpi_i^a \neq \varpi_i^d \\ -RP_t\Psi_t & \varpi_i^a = \varpi_i^d \end{cases} \tag{3.43}$$

Where $\mho$ is a set of criteria including cost, delay, loss, degradation, temperature, and monetary profit, ϖ_i^a and ϖ_i^d denote the actual and preferred quantity of criteria i, respectively, φ_i represent the penalty factor of violating i^{th} criteria.

3.4. The Proposed Probabilistic Delayed Double Deep Q-Learning (P3DQL) Algorithm

Different stages of the P3DQL algorithm development are defined in this section. The first step is tuning the early version of DDQL, making a trade-off between overestimation and underestimation biases. The second stage aims to enhance sample complexity and learning proficiency by applying a delay in updating rules considering the nature of the problem guaranteeing efficiency. Finally, combining applied methods in the previous stages forms a Probably Approximately Correct (PAC) algorithm with an adjustable trade-off between positive and negative biases.

Primarily the transition from Q-Learning to the early version of DDQL is investigated to illustrate the significant concerns that require to be addressed by P3DQL. The classic Q-Learning relies on a look-up table taking the superior action at each state while storing the values of the state-value Q-function. Unfortunately, besides the sluggish process, the quantity of required memory has made the Q-Learning algorithms worthless for real-world problems. DNN plays a role as a function approximator in DQL, where the inputs are the states, and the Q-values are calculated as the outputs, focusing on minimizing the loss function, as shown in Equation (3.44).

$$L(\theta_t) = (Q\left(s, a; {\theta_t}^{pred}\right) - r_{t+1} - \gamma \max_a Q(s', a; {\theta_t}^{target}))^2 \tag{3.44}$$

Where $Q(s,a)$ represents the value of action a_t in state s_t, μ is the experience buffer containing, r denotes the reward, ${\theta_t}^{pr}$ and ${\theta_t}^{tar}$ are . Also, the update of Q-Learning is as follows.

$$Q_{t+1}(s,a) \leftarrow Q_t(s,a) + \alpha_t(s,a).\widehat{TD}_{t+1}(s,a) \tag{3.45}$$

$$\widehat{TD}_{t+1}(s,a) = r(s,a) + \gamma \max_{a'} Q(s',a') - Q_t(s,a) \tag{3.46}$$

Where s' is the next state by the probability of transferring from state s with action a, $P(s,a,s')$→[0,1], α_t denotes the learning rate that controls the velocity of adaptivity to randomness in the rewards and transitions.

Q-Learning is stricken with a significant positive bias originating from applying the max operator $\max_a Q(s',a;\theta_t)$ in the update rule. Correspondingly, DDQL was introduced by Van-

Hasselt [114], approximating the Q-values using two independent estimators to update Q^A and Q^B follows.

$$\begin{cases} Q^A\,(s,a) \leftarrow Q^A\,(s,a) + \alpha\big(r + \gamma Q^B\,(s',\ a^*) - Q^A\,(s,a)\big) \\ Q^B\,(s,a) \leftarrow Q^B\,(s,a) + \alpha\big(r + \gamma Q^A\,(s',\ b^*) - Q^B\,(s,a)\big) \end{cases} \tag{3.47}$$

Where $a^* = argmax_a Q^A\,(s',a)$ to update the first Q-function Q^A, $b^* = argmax_a Q^B\,(s',a)$ updating Q^B. It should be noted that each update takes place using the value of the mutual Q-function for the next state s', eliminating overestimation bias.

Nonetheless, the original and different variants of the DQL algorithm, including Clipped-DQL [115] and Weighted-DQL [116] are still associated with high underestimation bias which is not desirable in many problems and may lead to unsatisfactory performance. A developed algorithm to deal with scheduling problems in an IoE-based smart grid is needed to address two critical concerns as follows.

3.4.1. Step 1: Making a trade-off between underestimation and overestimation biases

Inductive bias in a learning algorithm is a prior probability that shows the preference of an event before running the test. Fitting inductive bias with the nature of the problem is a crucial issue in accomplishing a proper generalization performance. In simple words, overestimation and underestimation are not necessarily unfavorable in essence, and making a trade-off between these biases enhances the program's functioning. Overestimation bias leads to more exploration, while underestimation bias results in higher exploitation.

To address the above-mentioned concern selecting the estimators in a probabilistic manner for updating the value function is proposed. The algorithm takes advantage of two separated unbiased estimators Q^A and Q^B that can be selected for the updating rule by the probability of δ and 1-δ, respectively. As a hyperparameter, δ adjusts the positive and negative biases based on the problem environment.

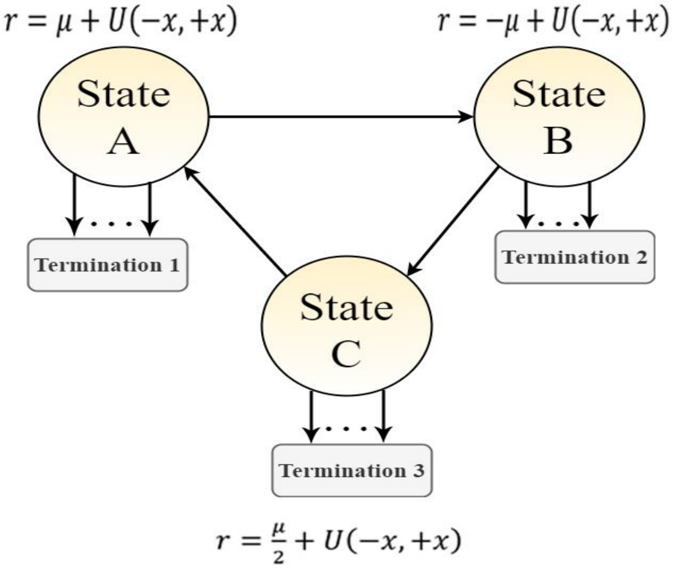


Figure 3-2. An example of episodic MDP with three non-terminal states

As Figure 3.2 illustrates, a plain MDP model has three states (i.e., A, B, and C) inspired by the user's consumption statuses, including normal, above normal, and less than normal. Each state has ω+1 actions so that one action transitions to the next state with a deterministic reward r=0, and ω actions transition to the terminate state with a stochastic reward from states A, B, and C, which are clarified by Equations (3.48) to (3.50), respectively. Where μ denotes the average reward, and U(-x,+x) is a uniform distribution used to determine ζ.

$$r = \mu + U(-x, +x) = \mu + \zeta \tag{3.48}$$

$$r = -\mu + U(-x, +x) = -\mu + \zeta \tag{3.49}$$

$$r = \frac{\mu}{2} + U(-x, +x) = 0.5\mu + \zeta \tag{3.50}$$

Suppose $\mu > 0$, the stochastic regions from state A to termination and from state B to termination become high value and low value, respectively. In this condition, overestimation bias improves exploration in state A resulting in a better performance. Also, states C takes advantage of overestimation with a less slope. On the other hand, when $\mu < 0$, underestimation bias helps regarding taking action after state B and hurts others. This arrangement shows that eliminating overestimation and underestimation in many circumstances is not the case by itself, and making a trade-off based on the problem and environment is required.

3.4.2. Step 2: Turning the tuned version into a Probably Approximately Correct (PAC) algorithm

Simultaneous improvement in exploration and exploitation abilities leads to the high complexity of the model, especially sample complexity, which identifies the quantity of time and experience that the model requires to perform near optimal. Additionally, model-free algorithms (e.g., DDQL) lean on observed reward regardless of state transmissions resulting in higher sample complexity.

As a comprehensive RL-based solution, the proposed model needs to be developed as a PAC algorithm by applying a delay in the update rule achieving near-optimal performance while bounding the sample complexity. The designed algorithm restricts the sample complexity to be $O(S^2A)$ with $P[error \leq \epsilon] \geq 1 - \delta$. Where S is state space, A represents action space, δ denotes the confidence parameter, and ε is the error parameter. The key point is that the estimator Q-value updates after n attempts when Equation (3.51) is fulfilled [117].

$$Q_t(s,a) - \left(\frac{1}{n}\sum_{i=1}^{n} r_{w_i} + \gamma \max_a Q_{w_i}(s_{w_i}, a)\right) \geq 2\epsilon \tag{3.51}$$

After the n^{th} attempt the update takes place as shown in Equation (3.52) where n denotes the number of attempts, s_{w_i} and r_{w_i} are i^{th} recent next states and reward, respectively.

$$Q_{t+1}(s,a) = \frac{1}{n}\sum_{i=1}^{n} r_{w_i} + \gamma \max_a Q_{w_i}(s_{w_i}, a) + \epsilon \tag{3.52}$$

3.4.3. Step 3: Forming the P3DQL algorithm

This chapter proposes the P3DQL algorithm, which is a combination of a new variant of DDQL and Delayed Q-Learning. First, the original version of DDQN is well-tuned, making a logical trade-off between positive and negative biases. Next, the algorithm developed into a PAC algorithm by applying a delay in the update rule that bounds sample complexity. Although choosing the estimator for the updating rule in a probabilistic manner may cause higher problem complexity, developing the set of rules as a PAC algorithm guarantees superior learning efficiency.

As Algorithm 3.1 shows, the probabilistic DDQL takes advantage of two separated unbiased estimators. The update rules for both Q-functions (i.e., Q^A and Q^B) adhere to update parameter β, where the probability of $\beta = 1$ is δ, likewise $Pr(\beta = 0) = 1 - \delta$. Then, the selection parameter ψ defines whether Q^A or Q^B be updated. Hyperparameter δ adjusts positive and negative biases considering the problem environment so that the bigger δ results in higher underestimation. Contrariwise, lower δ is utilized when more exploration leads to better performance.

After initializing Q^A and Q^B, two temporal buffer functions $U^A\ (s, a)$ and $U^B\ (s, a)$ store n recent updates while the counter $l(s, a)$ defines the number of occurred updates. Each Q-function update is allowed only once Learning Flag $LF(s, a)$ is true. On the condition that no update takes place after a specific length of time, $LF(s, a)$ turns into false.

Algorithm 3.1: P3DQL algorithm

```
Input reply buffer $D$ to capacity $C^{rep}$, $\tau << 1$, minibatch $k^{rep}$, learning rate $\alpha^{lr}$, discount factor $\eta$, period $\Delta^{rep}$, reward decay $\varepsilon$
Initialize $Q^A$ and $Q^B$
Inputs $S, A, \gamma, n, \epsilon$
for episode = 1, M do
    initialize sequences ${S_1}^i$
    store transition in $D$ at each episode
    set $y_j$, then calculate the error
    perform gradient descent
for all $(s, a)$
```

```
    $Q^A\ (s,a) \leftarrow (1-\gamma)^{-1}$        //Q-value estimated by A
    $Q^B\ (s,a) \leftarrow (1-\gamma)^{-1}$        //Q-value estimated by B
    $U^A\ (s,a) \leftarrow 0$                      //attempted updates by A
    $U^B\ (s,a) \leftarrow 0$                      //attempted updates by A
    $t(s,a) \leftarrow 0$                          //time of the last update
    $l(s,a) \leftarrow 0$                          //update counter
    $LF(s,a) \leftarrow true$                      //learning flag
end
choose $\beta^j = \{0,1\}$                         //j is iteration number
$Pr\left(\beta^j = 1\right) = \delta$ and $Pr\left(\beta^j = 0\right) = 1-\delta$
    if  $\beta^j = 1$
        $\beta^{j+1} = 0$
    else
        $\Pr\left(\beta^{j+1} = 1\right) = \delta$
$t^*(s,a) \leftarrow 0$                            //time of the most recent Q change
for $t = i$                                        //$i \in \mathbb{N}$
choose $a$, based on $Q^A\ (s,.)$ And $Q^B\ (s,.)$, observe $r$, $s'$
randomly choose $\psi = \{1,2\}$
    if $\psi$ =1
        update A
    else
        update B
if $\psi$ =1 and $LF(s,a)\ = true$
     $\boldsymbol{a}^* = \max_{a'} Q^A(s,a')$
    $U^A\ (s,a) \leftarrow U^A\ (s,a) + \gamma\left(\beta Q^B\ (s',\ a^*) + (1-\beta) Q^A\ (s',\ a^*)\right)$
    $l(s,a) \leftarrow l(s,a) + 1$
    if  $l(s,a) = n$
        if $Q^A\ (s,a) - \frac{1}{n} U^A\ (s,a) \geq 2\epsilon$
            $Q^A\ (s,a) \leftarrow \frac{1}{n} U^A\ (s,a) + \epsilon$
        else if $t(s,a) \geq t^*$
                $LF(s,a) \leftarrow false$
        end
        $t(s,a) \leftarrow t, U^A\ (s,a) \leftarrow 0, l(s,a) \leftarrow 0$
    end
if $\psi = 0$ and $LF(s,a)\ = true$
     $\boldsymbol{b}^* = \max_{a'} Q^B(s,a')$
    $U^B\ (s,a) \leftarrow U^B\ (s,a) + \gamma\left(\beta Q^A\ (s',\ a^*) + (1-\beta) Q^B\ (s',\ a^*)\right)$
    $l(s,a) \leftarrow l(s,a) + 1$
    if  $l(s,a) = n$
        if $Q^B\ (s,a) - \frac{1}{n} U^B\ (s,a) \geq 2\epsilon$
            $Q^B\ (s,a) \leftarrow \frac{1}{n} U^B\ (s,a) + \epsilon$
        else if $t(s,a) \geq t^*$
```

```
                $LF(s,a) \leftarrow false$
            end
            $t(s,a) \leftarrow t, U^B\ (s,a) \leftarrow 0, l(s,a) \leftarrow 0$
        end
else if
        $t(s,a) \leq t^*$
        $LF(s,a) \leftarrow True$
end if
end for
```

The proposed method establishes a bias margin between underestimation and overestimation bounds. Both positive and negative biases are shrunk at the same time considering the following statements.

Lemma 3.1. *Let $V = \{v_1, v_2, \dots, v_n\}$ be a set of values and let $\alpha = min\{v_1, v_2, \dots, v_n\}$, $\beta = max\{v_1, v_2, \dots, v_n\}$, and $v_j > v_{j+1}$. Then, $\alpha \leq \frac{1}{n}\sum_{i=1}^{n} v_i \leq \beta$ and $\alpha \leq \mathrm{E}\{V_i\} \leq \beta$.*

Lemma 3.2. *Let $V = \{v_1, v_2, \dots, v_n\}$ be a set of values and let $\alpha = min\{v_1, v_2, \dots, v_n\}$, $\beta = max\{v_1, v_2, \dots, v_n\}$, $Y = \{V\} - \alpha$, $W = \{V\} - \beta$. Also, $Pr\ (x_i{}^{\Omega})$ is the probability of $x \in \{\Omega | \Omega = \{V, Y, W\}\}$. Then, $Pr\ (x_i{}^{\mathrm{Y}}) \max_i\{Y_i\} + Pr\ (x_i{}^{\mathrm{W}}) \max_i\{W_i\} < \beta$, and $\alpha < Pr\ (x_i{}^{\mathrm{Y}}) \min_i\{Y_i\} + Pr\ (x_i{}^{\mathrm{W}}) \min_i\{W_i\}$.*

These two lemmas are used to prove Theorem 3.1, which indicates the tuned version improves both negative and positive biases simultaneously. The theorem is as follows:

Theorem 3.1. *Let $V = \{v_1, v_2, \dots, v_n\}$ be a set of values, the expected value is maximized by N, and minimized by M, which are two subsets of V ($i.e., N \subseteq V, and\ M \subseteq V$), and $N = \{i \big| E\{V_i\} = max_j V_j\}$, also $M = \{k \big| E\{V_k\} = min_j V_j\}$. Let $T^A = \{\tau_1^A, \tau_2^A, \dots, \tau_n^A\}$ and $T^B = \{\tau_1^B, \tau_2^B, \dots, \tau_n^B\}$ be two unbiased estimators so that $E\{V_i\} = E\{T^A{}_i\} = E\{T^B{}_i\}$, also let $\overline{a}$ and $\underline{a}$ be two elements that maximize and minimize T^A. If θ_u and θ_o are lower and upper bound of bias in original DQN, then*

$$\min_i E\{V_i\} = \theta_u < E\{T^B{}_{\underline{a}}\} < E\{T^B{}_{\overline{a}}\} < \theta_o = \max_i E\{V_i\}$$

Proof. If $\underline{a} \in M$, and $\beta = 1$, subsequently $E\{T^B{}_{\underline{a}}\} = \frac{1}{n}\sum_{i=1}^{n} \vartheta_i$, where $\theta_u = \vartheta_1 = \min_i E\{V_i\}$, and $\vartheta_{j+1} \geq \vartheta_j$. Then, $\min_i E\{V_i\} = \theta_u < E\{T^B{}_{\underline{a}}\}$. If $\underline{a} \in N$, and $\beta = 0$, then $\theta_u < \vartheta_1$, and consequently $\min_i E\{V_i\} = \theta_u \ll E\{T^B{}_{\underline{a}}\}$. In the same way, if $\overline{a} \in N$, and $\beta = 1$, accordingly $\{T^B{}_{\overline{a}}\} = \frac{1}{n}\sum_{i=1}^{n} \vartheta_i$, where $\theta_o = \vartheta_n = \max_i E\{V_i\}$, and $\vartheta_j \geq \vartheta_{j-1}$. Afterward $E\{T^B{}_{\underline{a}}\} < \max_i E\{V_i\} = \theta_o$. If $\overline{a} \in N$, and $\beta = 0$, then $\{T^B{}_{\overline{a}}\} = \frac{1}{n}\sum_{i=1}^{n} \vartheta_i$, where $E\{T^B{}_{\underline{a}}\} \ll \theta_o = \max_i E\{V_i\}$.

3.5. Simulation Results

The proposed P3DQL algorithm is evaluated using three large real-world datasets. The data used in the first two case studies have been collected by Pecan Street [118] in Austin and New York, and the latter case study investigates the electricity demand of multiple smart houses in Germany, provided in the DEDDIAG dataset [119]. All datasets contain PV generation and electricity consumption by every single device in different smart homes recorded for over three years every 15 minutes. Additionally, weather information and electricity retail prices are collected from the national weather service and energy companies in both countries, respectively. The simulation is tested using Python 3.9.7 on a standard system with an Intel Core i7-97580H CPU with 16.0 GB of RAM.

3.5.1. Multi-Layer Grid Setup: Case studies 1 and 2

Both the first (Austin) and the second (New York) case studies consist of 75 smart homes as NAs, where each NA includes major appliances, as mentioned in section 3, along with a 1 kW PV system and an EV. A group of NAs forms an NH besides a set of ESS, SPV, and EVC. Consequently, five

NHs are modeled where several NAs participate in multiple NHs. Similarly, two WANs are formed by multiple NHs. Table 3.1 clarifies the structure and components of different layers, where $i \in \{1 \rightarrow 75\}$ and $j \in \{A \overset{to}{\rightarrow} E\}$ indicates the ID of participated NAs, and NHs, respectively

Table 3-1. The structure and components in case studies 1 and 2

Layer	**Components**	**ESS (kWh)**	**SPV (kW)**	**EVC (kW)**
All NAs	All major devices	-	-	-
NH-1	$i \in \{1 \rightarrow 15\}$	8.2	10.4	15.8
NH-2	$i \in \{14 \rightarrow 27\}$	10.7	13.4	17.5
NH-3	$i \in \{28 \rightarrow 42\} \cup \{2\}$	9.9	12.0	11.4
NH-4	$i \in \{43 \rightarrow 60\}$	9.9	6.2	12.5
NH-5	$i \in \{61 \rightarrow 75\} \cup \{29\}$	10.1	6.2	11.4
WAN-1	$j \in \{A, B, C\}$	34.7	38.9	46.4
WAN-2	$j \in \{B, D, E\}$	35.5	40.1	52.6

All ESSs are Zinc bromide flow batteries with titanium electrodes, where $SOC_{ESS_{min}} = 1 - 2\ kWh$, and $SOC_{ESS_{max}} = 12 - 20\ kWh$, and $\lambda_{bat_t} = 0.75$. All ESSs are 5.2kW solar kit Canadian 400 black with Enphase micro-inverter or 6.2kW solar kit Q-Cells with Generac hybrid inverter. Finally, diesel generators supply EVCs, where quadratic coefficients of the generation cost function $\beta_1 = 0.001\$/(kW)^2$, $\beta_2 = 0.042\ \$/kW$, and $\beta_3 = 0.4\ \$/h$. It should be noted that incentive factor Ω and penalty factor Φ are defined based on the historical consumption of each user. Also, all units' loss and transmission fee coefficients are set based on standard values in the real-world power system and electricity market.

3.5.2. Multi-Layer Grid Setup: Case study 3

The third case study contains recordings of 15 smart homes over three years in Germany. Three NHs and two WANs have been formed, as indicated in Table 3.2.

Table 3-2. The structure and components of the third case study

Layer	Components	ESS (kWh)	SPV (kW)	EVC (kW)
All NAs	All major devices	-	-	-
NH-1	$i \in \{1 \rightarrow 6\}$	3.3	4.2	6.5
NH-2	$i \in \{4 \rightarrow 7\} \cup \{1\}$	3.8	4.8	4.8
NH-3	$i \in \{5 \rightarrow 10\} \cup \{2\}$	4.7	5.2	5.6
NH-4	$i \in \{11 \rightarrow 15\}$	3.2	3.9	5.4
WAN-1	$NH - 1\ and\ NH - 2$	13.8	15.6	18.5
WAN-2	$NH - 3\ and\ NH - 4$	14.2	16.2	19.9

3.5.3. Verification of Optimization Algorithm

Before training the algorithm, the optimization method requires to be validated in terms of numerical error. It is worth mentioning that numerical errors may originate from rounding errors due to machine representation format, tolerance errors, and truncation errors in integrations and approximations, just to name a few. The Rosenbrock function is utilized as a performance test problem in this study, as defined in (3.53).

$$f(x_n) = \sum_{1}^{n-1} (100({x_i}^2 - x_{i+1}) + (1 - x_i)^2) \tag{3.53}$$

Where $x = (x_1, \dots, x_N) \in \mathbb{R}^N$, and it has only one optimal solution $f = 0$ at (1,1), once $N = 2$. It should be noted that the function is scaled as required, restraining the global optimum in the scope of inputs.

The optimization method and objective function formulations are tested over 15 runs, estimating the possibility of converging the actual minimum to within $f < 0.01$. The same test is also applied to two other methods from literature. The results indicated $P(f < 0.01) = 0.9896$ in the proposed P3DQL algorithm, which is 11% and 9% better than methods in [52] and [51], respectively.

3.5.4. P3DQN Setup

The prioritized experience replay technique is utilized to eliminate instability caused by the significant correlation between actions and states. The size of the replay buffer is 10^6, while the number of training sets processed during each stochastic gradient descent update is $k^{rep} = 64$, which is the minibatch size. Also, ε-greedy increment $\varepsilon = 0.995$, decay step is 50, the learning rate is $\alpha^{\mathrm{lr}} = 0.015$, and discount factor $\eta = 0.9$.

It should be noted that a four-layer deep neural network is designed to simulate the model under given system parameters, historical consumption, retail price, weather data, and solar PV generation. The number of inputs and outputs are defined based on the total number of time slots during a day which is 96, while the number of neurons is taken by trial and error to select the best fit number. An Adam optimizer is utilized with the defined learning rate α^{lr}. The number of neurons in each layer is 1500, and the activation function is ReLU. Ultimately, running the algorithm using different δ indicates that $Pr(\beta = 1) = 0.65$ performs better than other combinations to solve this specific scheduling problem.

3.5.5. Training Process; Biasness, Error, and Speed of convergence

Before investigating the effect of the algorithm on the load curve and cost in each grid layer, the training results are provided. Adapting the agents to the environment is a progressive procedure that is more fluctuating at the beginning steps due to many random choices obtaining a better exploration. Figure 3.3 shows the reward trend during the training stage for the first case study, where after 300 episodes, the agent learns to behave near the optimal policy. The reward for a NA is demonstrated to simplify the explanation since other layers also follow the same trends. Although oscillation in the P3DQN is higher than other tested algorithms initially, it achieves a higher reward and demonstrates more stability in terms of learning and reward.

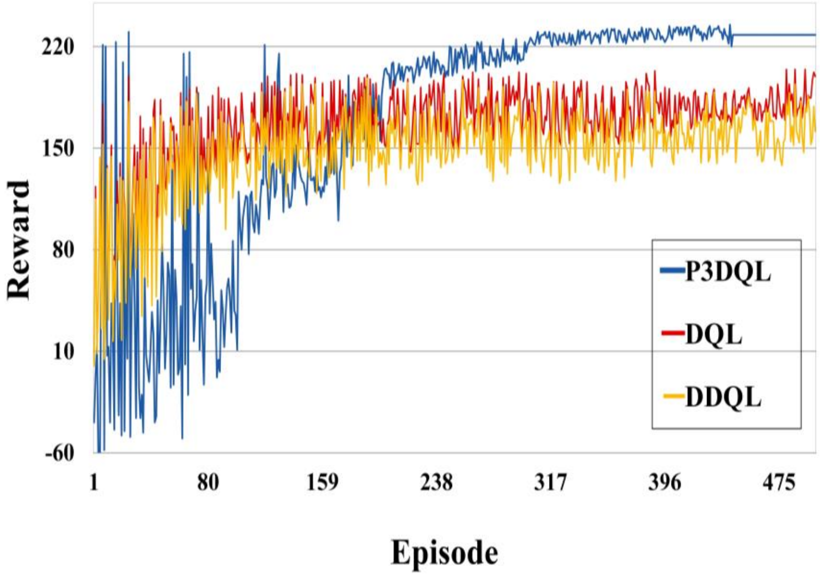


Figure 3-3. The rewards of three algorithms during training

Table 3.3 indicates that while DDQL and DDQ-ESS [120] have reduced the overestimation of DQL [57], underestimation was still a concern. Consequently, the proposed P3DQL algorithm decreased the underestimation bias while addressing positive bias. All four algorithms were trained in different layers under exact conditions comparing average expected and actual returns after ten random seeds.

Table 3-3. Result comparison among different methods

Method	Case Study	Expected return	Actual return	Error %
P3DQL	#1	178.7124	182.3781	-0.0205
	#2	177.0648	178.0226	-0.0054
	#3	162.3189	161.9403	+0.0021
DDQL	#1	111.3924	113.0334	-1.4732
	#2	120.1414	121.5809	-1.1982
	#3	94.0915	95.3455	-1.3328
DDQ-ESS [120]	#1	109.5485	111.1128	-1.4280
	#2	107.8547	109.3729	-1.4074
	#3	101.3214	102.5266	-1.1895
DQL [57]	#1	110.7838	110.375	+0.3704
	#2	106.2429	105.847	+0.3741
	#3	99.2980	98.9922	+0.3090

Additionally, after 412 episodes (on average for the three case studies), the training stage is completed, and there will be no more updates by the estimators. The results show that the sample efficiency of the P3DQL has been enhanced by 41%, 33%, and 34%, comparing DQL, DDQL, and DDQ-ESS, respectively.

Table 4 compares the iterative performance of different models (average of three case studies) over twenty runs with three different pairs of discount factors and learning rates.

Table 3-4. Comparison of execution time and Variance reduction rate on target values

Method	#Epoch	Average Reward, $\pm$ σ	Running Time
P3DQL	40	0.8458, 0.0291	0.9823
	160	0.7124, 0.0113	0.9683
	290	0.9011, .0060	0.9217
	420	1.0000 , 0.0055	0.9012
DDQL	40	0.7624, 0.0212	0.9831
	160	0.8121, 0.01423	0.9712
	290	0.8005, 0.0102	0.9619
	420	0.8131, 0.0098	0.9408
DQL [57]	40	0.7941, 0.03412	1
	160	0.8275, 0.0294	0.9732
	290	0.8281, 0.0288	0.9924
	420	0.8451, 0.0292	0.9820

The discount factor has also been increased continuously, guaranteeing reaching the optimal policy. Even though less discount factor leads to faster convergence, achieving optimal policy is the priority. As the results demonstrate, the average reward in the developed P3DQL is higher than other techniques, reducing the standard deviation (σ). Additionally, the proposed algorithm is faster than other investigated methods since the execution time of the P3DQL is less than other approaches and reaching the optimal policy is attainable by at least 19% fewer epochs. Finally, it should be noted that the reported numbers are normalized by the measured scale of corresponding runs.

3.5.6. Stability and sensitivity of P3DQL

This section first evaluates the stability of the proposed P3DQL algorithm, and then a sensitivity test is conducted to identify the most sensitive features and parameters. Dynamic systems analysis defines a stable method as one in which the outcome slightly changes under perturbations or remains unaltered [121]. Accordingly, the stability of the developed model is examined based on the distribution and characteristic analysis using Population Stability Index (PSI) and Characteristic Stability Index (CSI) metrics, respectively.

The effect of data distribution drift is investigated using PSI, whereby two different samples of a population are bucketed to compare changes in the predicted variable. Initially, a reference population is selected, and the scoring variables are sorted in descending order. Then, ten bins of data are created by cutting the scale of the variable into units of the same size after ranking the order of the preference scores. Subsequently, the cutoff points used to create the bins for In-sample are applied to the new distribution. Lastly, the distribution shifting of variables between two samples is calculated as in (3.54):

$$PSI = \left((\%actual - \%expected) * \ln \frac{\%actual}{\%expected} \right) \tag{3.54}$$

$PSI < 0.1$ indicates minor population bias and excellent stability of the model, while $0.1 < PSI < 0.2$ shows that the population has slightly changed, but still no immediate action is required. Ultimately, if PSI is larger than 0.2, the model should be retrained or redesigned. The results demonstrate that in all three case studies, the PSI is less than 0.1, where $PSI^{case\ study\ 1} = 0.0688$, $PSI^{case\ study\ 2} = 0.0451$, and $PSI^{case\ study\ 3} = 0.07315$. Accordingly, the existing model is utterly stable and requires no action.

The impact of feature distribution is investigated through CSI by the same procedure as

PSI. The only difference is related to the bucketing process, where each feature is employed to generate the bins instead of slicing the data using the predicted variable. The results show the highest calculated PCI for all features in all three case studies is 0.0891, which still falls under the stability threshold.

Sensitivity analysis is also conducted to investigate the significance and impact of each input variable on the system’s output using the Pearson correlation coefficient and Sobol’s variance-based sensitivity index. Moreover, a ranking of their influence is presented by systematically altering the model’s hyperparameters.

Table 3.5 indicates the ranking of input variables based on two sensitivity analysis methods. The Pearson coefficient indicates the normalized correlation between the selected feature and the output varying from -1 to +1, denoting total negative and positive linear correlations, respectively. Similarly, Sobol’s sensitivity index determines the contribution of each input parameter based on the decomposition of the model output variance that can be attributed to inputs or sets of inputs. The results show that PV, RP, and AC are the most critical features. However, the correlations in both tests are considerably low. Global insight into hyper-parameter influence is investigated using Sobol's indices. The results demonstrate that the discount factor and error parameter (threshold of updating rule) are the most sensitive parameters. Increasing the discount factor results in more accuracy in obtaining the optimal policy (more exploration) while reducing the convergence speed. Conversely, discount factor diminution leads to better exploitation and a speedy process. The proposed model can adjust the discount factor amount where overestimation or underestimation is advantageous. Moreover, the error parameter plays a key role in sample efficiency, where a rise improves the efficiency endangering the accuracy.

Table 3-5. The structure and components of the third case study

Variable	Index		Ranking		Overall Ranking
	Pearson	Sobol'	Pearson	Sobol'	
PV	-0.211	0.0245	1	2	1
RP	-0.195	0.0301	3	1	2
AC	0.197	0.0122	2	4	3
WM	0.092	0.0083	5	5	5
EV	0.143	0.0195	4	3	4
DW	0.084	0.0012	6	6	6
DY	0.023	0.0009	7	7	7

3.5.7. Numerical Results: Load Profile

The P3DQN algorithm is tested to discover the outcomes for different grid layers in terms of peak reduction and flatting the load curve. The trained algorithm is utilized to schedule the energy entities over four separate weeks in different seasons, taking weather and irradiance differences during a year into account, considering holidays and weekend effects besides the effect of seasonal appliances.

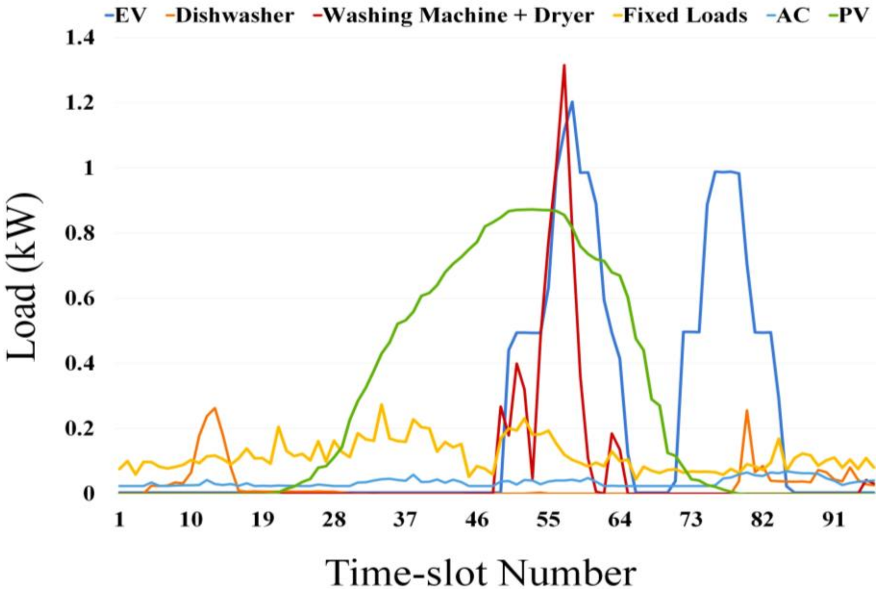


Figure 3-4. Load curve of different components in an NA

Figure 3.4 shows the average load curve of major shiftable appliances, fixed loads, and power generated by the solar system in NA number 14 of the first case study before applying the proposed scheduling algorithm. Table 3.6 summarizes the average results of three case studies, where %PAPR represents the percentage of reduction in Peak to Average Power Ratio (PAPR). Utilizing P3DQN reduces peak load by 27.9% in July, 8.7% in October, 11.8% in February, and 14.1% in April. Then, the algorithm applies to the NH-4 of the first case study, examining the effect of shared battery storage, shared PV, and EV charging station in two different scenarios.

Table 3-6. Load profile improvement with different algorithms

Layer	Metric	Initial	P3DQL	DDQL	DQL
NA	Peak (kW)	8.043	5.8137	6.670	7.237
	Variance	0.784	0.687	0.771	0.801
	% PAPR	-	12.2 %	11.5 %	8.3%
NH	Peak (kW)	32.858	23.557	26.979	30.125
	Variance	2.756	2.447	2.592	2.699
	% PAPR	-	10.8%	10.1 %	7.6%
WAN	Peak (kW)	31.750	22.532	27.953	31.081
	Variance	15.586	13.277	14.119	14.473
	% PAPR	-	10.3 %	8.9 %	6.9%

In the first scenario with 6 NAs, the peak load decreases by 25.1%, 7.9%, 11.4%, and 13% in July, October, February, and April, respectively. The second scenario, including 6 NAs, a 6 kW ESS, a 6.2 kW SPV, and a 10 kW EVC shows outstanding performance in terms of peak reduction in July by 28.2% reduction. In the same way, the algorithm is tested on WAN-2, which is a combination of NH-2, NH-4, and NH-5.

3.5.8. Numerical Results: Cost

This section analyzes the performance of the P3DQL in terms of cost reduction, taking into account three different tariff types: flat tariff, Time of Use (ToU), and Real-Time Pricing (RTP). Since the flat tariff scenario assumes that the electricity price does not change during the day, the only electricity scheduling feature is managing the generated power by the solar system, and the effect of utilizing any algorithm is slight.

The average cost deduction after applying the developed algorithm to all three case studies is indicated in Figure 3.5, where both ToU and RTP tariff types demonstrate outstanding outcomes. To conclude, the cost is deducted by almost 29% in different layers.

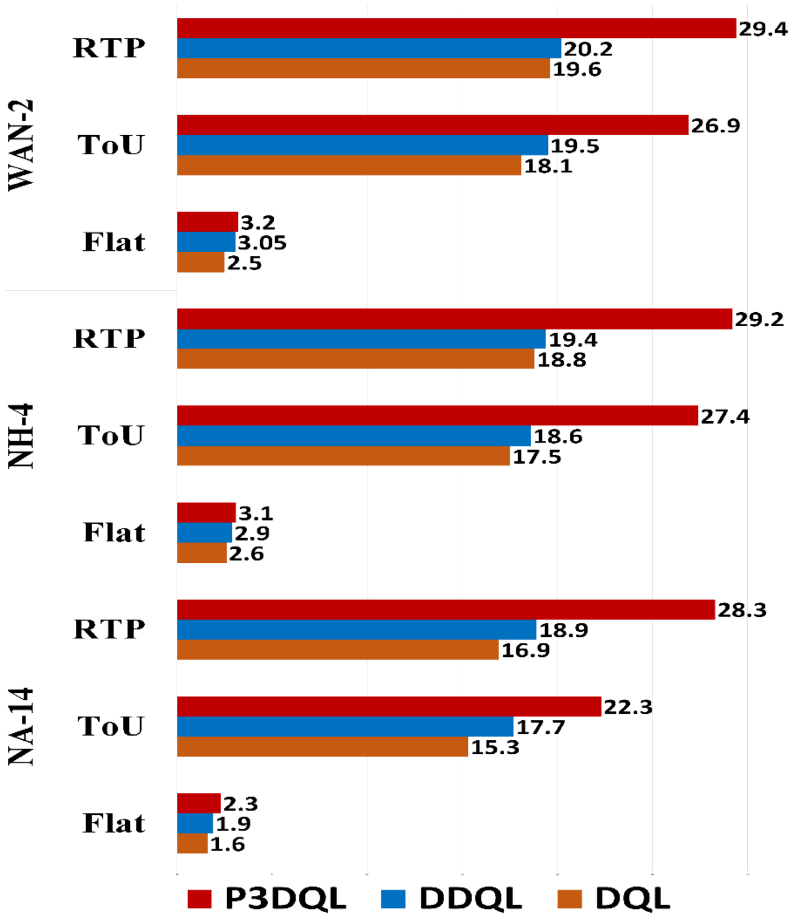


Figure 3-5. Different scenarios of cost reduction

3.5.9. Results Comparison

This section compares various methods, including DQL-based, DPG-based algorithms, and LSTM as modern methods, and one of the most used traditional methods for solving scheduling problems, which is Mixed Integer Linear Programming (MILP). Since the state-action space of the proposed method is discrete, the biases and convergence speed results are compared with models of the same nature. As indicated in Table 3.4, the proposed P3DQL algorithm simultaneously improved overestimation in DQL [57] and underestimation in DDQL and DDQ-ESS [120]. Moreover, Table 3.4 demonstrated that the standard deviation of the average reward of the P3DQL algorithm is less than other techniques, while the running time is lower.

Furthermore, the proposed P3DQL is compared with other algorithms, whether the state-action space is discrete or continuous. As Figure 3.6 shows, the proposed algorithm reduces the peak by 29.4%, which is almost 16% higher than MILP [122], and LSTM [123] and approximately 14% better than DQL [57], and DDPG [56]. It should be noted that the performance of the proposed method is also better than the suggested algorithm in DPG [57]. However, the deep policy gradient is less capable regarding sample efficiency and running time. Finally, the P3DQL reduced the cost by 30.1% which is 4.7%, 10%, and 17.8% higher than DPG [57], DQL [57], and DDPG [56], respectively.

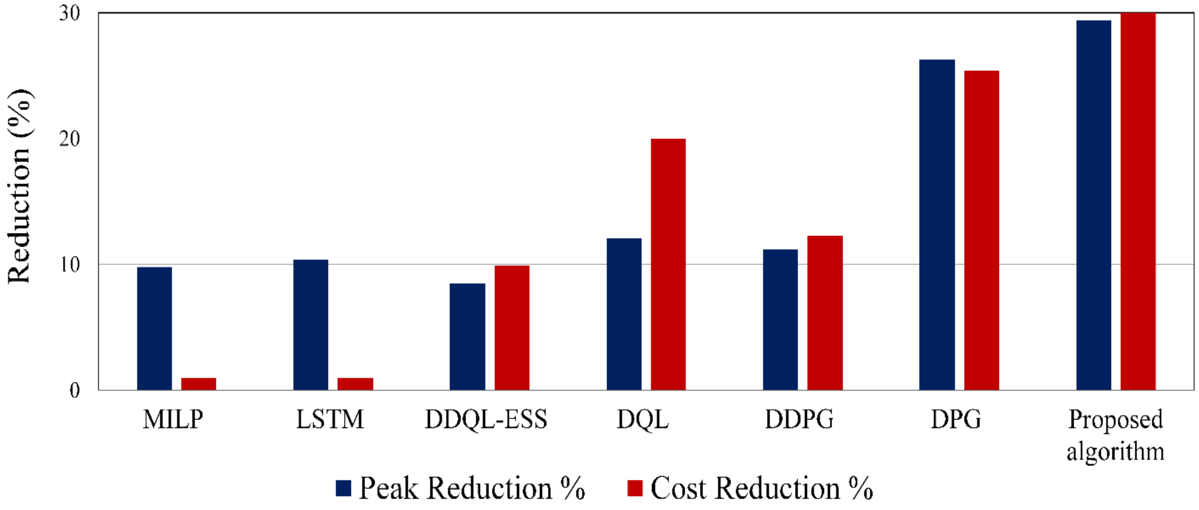


Figure 3-6. Result comparison among different methods

3.6. Conclusion

This chapter proposed a novel model-free Q-Learning-based algorithm for scheduling various energy components in a Multi-Layer IoE-enabled smart grid. The proposed algorithm can address the problem in each layer independently or by considering three layers as a whole, where the level of participation is predefined based on users' preferences. Also, the privacy of users is preserved by the utility, who as the principal owner of the network provides subscription-based services to the customers in a decentralized manner.

Traditional scheduling methods cannot address this problem since comprehensive modeling of the environment is practically impossible due to various types of uncertainties in price, weather conditions, solar irradiation, and customers' behavior and preferences. Moreover, the number of state-action pairs is extremely large and tabular methods are inefficient in terms of time and sample complexity. Likewise, DDQN and DPG-based algorithms suffer from underestimation bias and discretization issues, respectively. Furthermore, stability and learning efficiency are two crucial metrics that need to be considered. Accordingly, the P3DQL algorithm has been proposed making an adjustable trade-off between positive and negative biases ensuring high efficiency besides lower complexity of the algorithm.

The proposed model has been tested on a large real-world dataset validating the algorithm's effectiveness regarding peak clipping, reducing PAPR, and cost reduction in different grid layers. Comparing the results with other applied methods on the same dataset in the literature indicated the superiority of the P3DQL by 28.2% peak clipping, 12.9% PAPR reduction, and 29.4% cost saving.

In future work, we plan to develop an electricity routing mechanism in IoE-based networks containing multiple AC and DC energy components beside different rectifiers and inverters. A key challenge will be the algorithm's execution time, which is crucial in reaching the best performance. Last but not least, defining the optimum reward function will be challenging, considering the limited action space that needs to be addressed with an innovative solution.

Chapter 4

Optimizing Resource Swap Functionality in IoE-based Grids Using Approximate Reasoning Reward-based Adjustable Deep Double Q Learning

4.1. Introduction

The Internet of Energy (IoE) concept emerged to monitor, control, and respond to the ever-growing electricity demand in the present and expected future power systems, integrating grid components with the internet [124]. This modern energy paradigm requires an innovative control strategy addressing the impacts of renewable energy resources, which are naturally uncertain while enhancing the routing mechanism to reduce loss and increase the system's efficiency. Moreover, IoE aims to facilitate Peer-to-Peer (P2P) energy trading, especially at the residential level, increasing the monetary benefits of demand-side management programs [107].

Energy Router (ER) is a compact intelligent power electronic device that is known as the core of the IoE. Optimizing energy management strategy by enabling bidirectional power flow among several devices is the leading capability of ERs. In other words, ERs are hybrid AC/DC interfaces that integrate energy components to harmonize demand and supply while maximizing efficiency from technical and economic points of view. By expediting and easing the integration of various resources and devices, besides providing flexible and cost-effective power flow management, ERs indicate tremendous capability in future power systems.

ERs are categorized into three different types, including grid level, microgrid level, and user level, based on location-dependent tasks [125]. In a hierarchical control mechanism, the primary control stage is positioned at the user level to optimize electricity flow among energy components based on the Real-Time Price (RTP), resource availability, customer preferences, technical limitations, and trade contracts with other parties. The secondary control layer aims to decrease the voltage and frequency deviation at the microgrid level while considering electricity swapping commitments and stability requirements. Finally, the tertiary control layer employs the optimal scheduling algorithm accomplishing the final stage of loss minimization and efficiency maximization by dispatching power flow at the distribution network level. Due to the crucial role of Solid-State Transformers (SST) at the grid level, utilized routers in the distribution system are mostly known as SST-ER.

Numerous studies have been carried out investigating the concept of ER in the present and future generations of power systems. The optimal placement of energy storage units at the microgrid level has been investigated in [70], aiming to improve the system's stability. The proposed method relies on the structure preserving energy function in medium and high voltage networks, focusing on three-phase inverters. The optimum location selection of charging stations has been researched in [69], comparing greedy heuristic and diffusion-based solutions. The outcomes indicated that the greedy heuristic method requires fewer charging stations, while the diffusion-based design resulted in less transmission loss. In [68], a modeling method has been presented to assess the stability of converter-dominated islanded AC microgrids based on droop controllers with a phase-locked loop. Two energy routing control strategies have been designed in [67] for multi-energy interconnected systems. The first strategy focuses on minimizing the loss of local absorption in a monopoly market, while the latter tends to diminish the transmission loss in a competitive market. Ant colony optimization has been utilized in [66] to discover the best possible path with minimum loss among producer and consumer units in the power system.

Additionally, a particle swarm optimization algorithm has been developed to determine the amount of energy that needs to be collected from each producer, considering customer satisfaction. An optimal selection of source and routing path strategies has been developed in [65] based on International Electrotechnical Commission (IEC) 61850 communication routing algorithm in microgrids. The proposed method successfully discovers the lowest loss route in a microgrid based on the least objective function value. Ultimately, the Authors [64] designed a minimum loss model for ERs, solving the minimum loss problem on a real-time scale. The proposed method is capable of handling multiple energy storage devices with high efficiency in a cooperative manner.

Utilizing ERs in the high-voltage network is not feasible economically since fulfilling technical and safety requirements in the power transmission/distribution layers is exceedingly costly [71]. However, the high penetration of distributed energy resources in the low-voltage layer necessitates utilizing ERs to meet energy commitments and financial objectives.

A weighted routing algorithm for a local area network has been designed in [74] following Graph theory. The proposed framework includes renewable energy sources, Electric Vehicles (EVs), and energy storage units only at the neighborhood level. Furthermore, a multi-terminal ER design has been proposed in [73] and [72], interconnecting and coordinating multiple microgrids simultaneously. Nevertheless, smart home appliances and personal energy units, such as exclusive rooftop Photovoltaic (PV) systems and battery storage units, have not been investigated in the previously mentioned papers.

A few studies have investigated ER-based energy management at the low-voltage and residential levels. Mixed-Integer Linear Programming (MILP) has been employed to design an optimal routing and control scheme for a single household and at a neighborhood level in [75]. A fuzzy logic-based hierarchical control strategy has been planned in [76] and [77] to reduce energy costs by enabling plug-and-play connections, and optimizing the utilization of renewable energy sources, respectively. Authors in [126] developed a home energy router accommodating AC/DC powered load, aiming to reduce the number of power converters.

Many serious challenges are associated with the methods studied earlier, which conventional methods are not able to address. First and foremost, the nature of renewable energy resources is extremely uncertain, and power delivery fluctuates continuously, which makes mathematical modeling impractical [47]. An additional source of uncertainty also originated from unpredictable customer preferences and conditions. Consequently, using historical data to forecast future electricity generation and demand is not straightforward and makes the predictions inaccurate.

Deep Reinforcement Learning (DRL) has been widely utilized in energy systems thanks to the capability of addressing control and optimization problems model-freely [22]. Moreover, DRL-based methods take advantage of high flexibility and generalization due to not relying on prior knowledge about the system's topology and information. In [77], a marketing auction mechanism has been developed utilizing DRL to minimize energy costs in microgrids. A Deep Q-Learning (DQL)-based optimal energy management mechanism for an office building has been developed in [78], controlling the energy flow of PV and battery storage. The concept of energy routing centers has been proposed in [79], coordinating multi-energy coupled energy framework. The developed model aims to enhance the conversion flexibility of energy components.

Despite the fact that previously published works in the literature have provided seminal insight into utilizing ERs at the residential level, several significant problems have not been appropriately addressed. Firstly, none of the earlier studies simultaneously deemed all exclusive and shared energy components in Nano Area (NA) and Neighborhood (NH). Furthermore, the proposed routing structures have not supported P2P electricity trading and input power from NHs, which are incredibly imperative and effective in optimizing cost and loss. Ultimately, the hitherto developed routing procedures have not attempted to enhance the algorithm's efficiency while providing a feature to take advantage of positive and negative biases where applicable.

Therefore, this chapter aims to develop a routing algorithm that simultaneously considers all energy components in the NA and delivered power from NH. Moreover, the designed algorithm must enable and guarantee electricity trading between residential units and the neighborhood besides P2P contracts. Furthermore, the proposed algorithm to solve the optimization problem requires to be efficient and fast in convergence. Ultimately, since overestimation and underestimation, which respectively originate from more exploration and exploitation, are not always destructive, this chapter's principal objective is to outline boundaries for biases while enabling the capability of adjusting exploration and exploitation. Consequently, the main contributions of this study are listed as follows:

I. A comprehensive ER structure at the residential level is proposed, supported by an optimal scheduling program that includes all AC/DC electricity devices and bidirectional power flow with the main grid and NH based on their delivery commitments during the day. This structure supports P2P trading.

II. An Adaptable Deep Double Q-Learning (ADDQL) algorithm is developed, which is capable of adjusting to the nature of the problem by taking advantage of exploration and exploitation where overestimation and underestimation are favorable, respectively.

III. An Approximate Reasoning Reward function is designed to optimize routing strategy in

NAs to improve efficiency due to declining in the number of random actions. Utilizing the proposed reward function in an RL-based algorithm also leads to a higher convergence speed since the number of state-action pairs is reduced.

IV. An Approximate Reasoning Reward-based Adaptable Deep Double Q-Learning (A2R-ADDQL) is proposed for optimizing the resource swap functionality of ERs to reduce the loss and greenhouse gasses emissions while improving monetary profits. The developed algorithm is specially designed for routing optimization problems at the residential level and benefits from the dynamic bias setting and efficiency of ADDQL and AR3L, respectively.

This chapter is arranged as follows. Section 4.2 presents the proposed ER architecture while formulating the routing optimization problem. Section 4.3 introduces the proposed A2R-ADDQL algorithm. Results are provided in section 4.4. Finally, a brief conclusion is presented in section 4.5.

4.2. System description and formulation

This chapter proposes a schedule-synchronized ER following and completing our previous research in [126], where an operation DRL-based scheduling algorithm was proposed supporting three different layers, including NA, NH, and Wide Area Network (WAN). Since the routing algorithm is technically and economically feasible in the NA layers, the proposed ER architecture is focused on this level, as indicated in Figure 4.1.

NAs are characteristically composed of various loads, PV systems, Electricity Storage Systems (ESS), and two bidirectional power lines from the main grid and the NH. Loads are categorized into four types: AC-fixed loads, AC-flexible loads, DC-fixed loads, and DC-flexible loads. Fixed loads demand must be delivered instantly, while flexible loads can be controlled by shifting or adjusting through a scheduling mechanism.

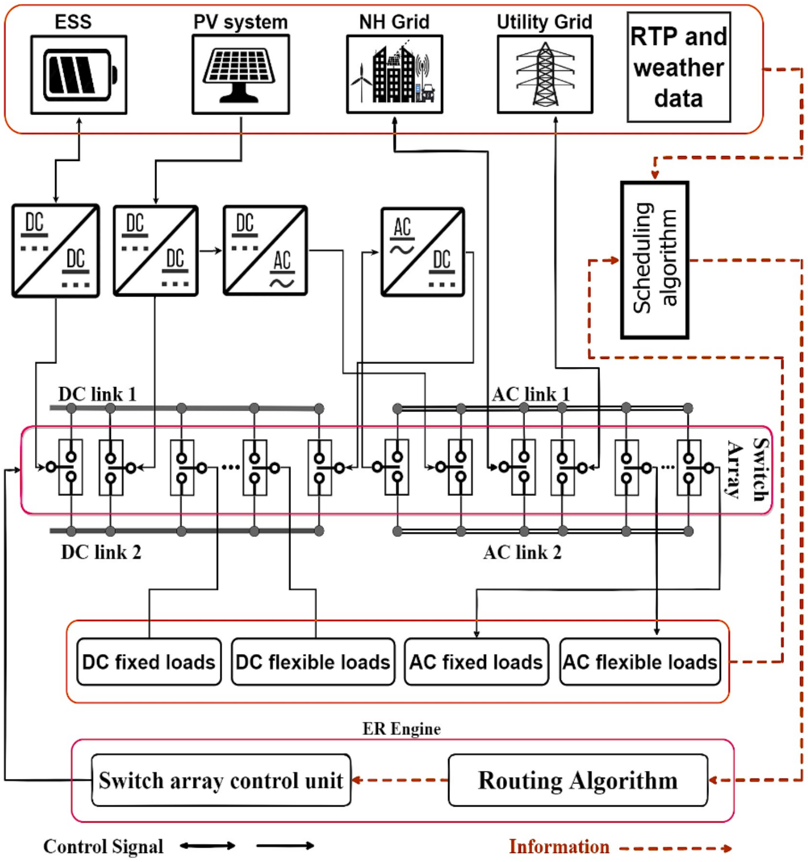


Figure 4-1. The proposed ER architecture

The switch array, planted between AC and DC, enables coupling every single load with any possible resource regardless of whether the voltage is AC or DC. This efficacious attribute is facilitated by utilizing multiple bidirectional converters. The ER engine encompasses the routing algorithm and the switch array control units. Finally, the scheduling algorithm backs the routing framework to decide the optimum way of connecting loads to the best available electricity source. Consequently, the proposed ER architecture operates under four primary functionality modes as follows:

i. **Mode 1:** Both utility and NH grids (external resources) are connected to the NA in this mode, whereas the PV and ESS (internal resources) are also available. The desired action in this mode is to maximize utilizing the external sources due to monetary considerations or even internal resource failure or inadequacy.

ii. **Mode 2:** While the system still tries to maximize the utilization of external resources, the ESS is in charging mode.

iii. **Mode 3:** All resources are available without specific priority, whereas ESS charging is undesirable and deleterious.

iv. **Mode 4:** Either external resources are disconnected (islanding mode), or the system aims to minimize their deployment while maximizing PV and ESS utilization. One of the AC links is deactivated in this mode reducing loss, whilst both DC links are still operational.

The designed Probabilistic Delayed Double Deep Q-Learning (P3DQL) algorithm in our prior work provides operation scheduling of all components in four described operational modes, as demonstrated in [126]. Straightway the next challenge is how to respond and what route to deliver the demanded power to maximize monetary profit while minimizing losses and environmental pollution.

4.2.1. Operational Monetary Costs

Different types of costs are considered to optimize routing monetary profit, including electricity purchasing fees from the utility grid f_{11}, and power exchange cost between NA and NHs f_{12}.

Utility grid electricity tariffs are contained within the fixed rate, Time of Use (ToU), and RTP In this study, the Wiener process is utilized to approximate the trade price, especially in the RTP tariff that is highly under the influence of the stochastic characteristics of the electricity price in the utility grid. Accordingly, the cost of exchanging electricity from the utility grid is indicated in (4.1), where $price_t^{his}$ defines the historical price at the time t, ${P^{G/ex}}_t$ denotes the exchanged power with the main grid, d_t indicates drift rate in price, σ (standard deviation) stands for degree of volatility, and function N is normal distribution with 95% confidence interval.

$$f_{11} = \sum_{t=1}^{T} P^{G/ex}{}_{t}\left(price_{t}^{his} + d_{t} + N(0, \sigma\sqrt{t})\right) \tag{4.1}$$

The power exchange amount between NA and NHs entirely depends on the activation coefficient of the scheduling algorithm ($\alpha \in \{0,1\}$), as demonstrated in (4.2). P_{i}^{ex} is the exchanged power with the i^{th} NH, P_{i}^{sch} indicates planned power by the scheduling algorithm to be swapped with the i^{th} NH. If $\alpha = 1$, the exchanged power quantity strictly follows the scheduling algorithm. Conversely, $\alpha = 0$ means there is no power swap commitment, and NA can sell or buy electricity from or to i^{th} NH, which is indicated by P_{i}^{sell} and P_{i}^{buy}, respectively. It should be noted that any single individual unit is considered as NH in P2P energy trading since the price and amount of exported electricity are the only factors that need to be deemed.

$$P_{i}^{ex}(t) = \alpha\left(P_{i}^{sch}(t)\right) + (1-\alpha)\left(\ P_{i}^{buy}(t) - P_{i}^{sell}(t)\right) \tag{4.2}$$

The embedded ESS and PV provide available power in each NA for selling to other parties, as indicated in (4.3). P_{disch}^{ess} and P_{chg}^{ess} stand for provided and drained power by the ESS in discharging and charging modes, respectively, while λ_{ESS} symbolizes the efficiency of the ESS. Also, P^{pv} denotes the output power of the PV system.

$$P_{i}^{sell}(t) = P^{pv}(t) + \left(\lambda_{ESS}(t)\left(P_{disch}^{ess}(t) - P_{chg}^{ess}(t)\right)\right) \tag{4.3}$$

Ultimately, Equation (4.4) illustrates the power exchange cost between NA and NHs, where Γ_{jt} is the electricity swap rate with i^{th} NH during the time slot t. Also, T and ω denote the total number of time slots and NHs.

$$f_{12} = \sum\nolimits_{t=1}^{T} \sum_{j=1}^{\omega} \Gamma_{jt}\ P_{i}^{ex}(t) \tag{4.4}$$

4.2.2. Power Losses

Power losses that originate from the connection lines among NHs and WANs have formerly been taken into account by the developed scheduling algorithm in our previous work. Also, the losses caused by the resistance of wires in the NAs are neglectable. The significant power loss in NAs occurs during energy conversion processes by the utilized converters in the proposed architecture.

The efficiency of inverters and converters varies based on the quantity of their output power, whereas the highest efficiency appears at a point between the mid and the highest output power. Consequently, an independent efficiency function $f_i(\delta(t))$ is defined for each convert device based on $\delta(t)$, which is the fraction of the occurring and the rated output powers. Therefore, properly routing the available power among the activated devices minimizes the total conversion losses f_{21}, formulated in (4.5).

$$f_{21} = \sum_{t=1}^{T} \sum_{i=1}^{n} p_i(t)\big(1 - f_i(\delta(t))\big) \tag{4.5}$$

In Equation (4.5), the number of conversion devices is denoted by n, while $p_i(t)$ is the amount of shared power with the i^{th} device at time t.

The utilized switching array is another source of power losses in the proposed architecture. All the switches are assumed to be assembled using Metal Oxide Semiconductor Field Effect Transistor (MOSFET) modules to reduce power loss while improving switching speed. Total power loss L_T in the switch array is the sum of the switching loss L_{sw} and resistance loss L_{res}, as shown in (4.6), where k_t is the number of operational MOSFETs at time t. For simplicity's sake, resistance losses in the high and low side MOSFETs are assumed to be equal.

$$f_{22} = \sum_{t=1}^{T} k_t(L_{sw}(t) + L_{res}(t)) \tag{4.6}$$

$L_{sw}(t)$ and $L_{res}(t)$ are expressed in Equations (4.7) and (4.8), respectively. In Equation (4.7), V_{in_t} denotes input voltage, I_{l_t}denotes load current, f_t indicates the number of switching during time slot t, and Δt is switching time, respectively. In Equation (4.8), T^{op} in operative temperature in Celsius, R_{DS} stands for drain-source resistance, and $c < 1$ is a constant, which is defined based on the MOSFET type.

$$L_{sw}(t) = V_{in_t} I_{l_t} (\Delta t) f_t \tag{4.7}$$

$$L_{res}(t) = c T^{op} R_{DS} I_{l_t} \tag{4.8}$$

4.2.3. Environmental Cost

Carbon dioxide (CO_2) and Nitrogen Oxides (NO_x) are the primary greenhouse gasses emitted through power generation by conventional power systems. Consequently, supplying from the utility grid results in higher environmental costs. $f_{CO_2}(t)$ and $f_{NO_x}(t)$, which are CO_2 and NO_x emission functions, are defined based on the power generation time since emission levels during off-peak and on-peak hours are significantly different due to the type of utilized power stations. Accordingly, the environmental cost function is formulated in (4.9), where T denotes the total number of time slots, and $\alpha_t \in \{0,1\}$ stands for the status of connecting to the utility grid at time slot t.

$$f_{31} = \sum_{t=1}^{T} \frac{1}{T} \alpha_t \big(f_{CO_2}(t) + f_{NO_x}(t) \big) \tag{4.9}$$

4.2.4. Optimization Problem Formulation

The objective function of the routing optimization problem aims to minimize operational costs, power losses, and environmental costs. Thus, the optimal electricity routing in a NA is formulated as (4.10).

$$min \quad \left(f_{31} + \sum\nolimits_{i,j\in\{1,2\}} f_{ij}\right) \tag{4.10}$$

The routing optimization as a minimization must also fulfill the following constraints:

$$\mu - 2\sigma < N(\mu, \sigma) \; < \mu + 2\sigma \tag{4.11}$$

$$\sum_{t=1}^{T} P^{pv}(t) + \left(P_{disch}^{ess}(t) - P_{chg}^{ess}(t)\right) + P^{G/ex}(t) + P_{EX}^{NH}(t) + L_{total}^{appliances} = 0 \tag{4.12}$$

$$0 < \lambda_{ESS}(t) < 1, \; \forall t = [1{:}T] \in \mathbb{N} \quad 0 \tag{4.13}$$

$$SOC_{charge}^{ess} \leq SOC_{t}^{ess} \leq SOC_{discharge}^{ess} \tag{4.14}$$

$$P_{disch}^{ess}(t) = 0, \forall t \in \left\{SOC_{t}^{ess} < SOC_{charge}^{ess}\right\} \tag{4.15}$$

$$P_{chg}^{ess}(t) = 0, \forall t \in \left\{SOC_{t}^{ess} > SOC_{discharge}^{ess}\right\} \tag{4.16}$$

$$P^{pv}(t) = \hat{A}_{pv}\lambda_{pv}H^{r}(0.875 - 0.005{}_{out}^{o}c_{t}) \tag{4.17}$$

$$\mathit{\Gamma}_{jt}^{com} \leq \mathit{\Gamma}_{jt} \leq \mathit{\Gamma}_{jt}^{max}, \forall j = [1{:}\,\omega] \in \mathbb{N} \tag{4.18}$$

$$\sum\nolimits_{t=1}^{T} p_{i}(t) = \; P_{total}, 0 < p_{i} < p_{max} \tag{4.19}$$

$$0 \leq \delta(t) \leq 1, f_{i}\big(\delta(t)\big) < 1, \forall t = [1{:}T] \in \mathbb{N} \tag{4.20}$$

$$k_{t} \in \{0{,}1\}, \; f_{t} \in \mathbb{N}, t = [1{:}T] \in \mathbb{N} \tag{4.21}$$

$$f_{CO_2}(t) = 28.33 \, P_{in}^{UG}(t), f_{NO_x}(t) = \; 23.04 \, P_{in}^{UG}(t) \tag{4.22}$$

Where, P_{EX}^{NH} is the total swapped electricity with all NHs that NA has an energy trading contract with, $L_{total}^{appliances}$ points out the total demand for consumer devices, SOC_{t}^{ess} defines the State of Charge (SOC) of the storage at time slot t, $\hat{A}_{pv}$ is the total area of the panel (m^2), H^{r} defines annual average solar radiation on tilted panels, λ_{pv} denotes solar panel yield, Γ_{jt}^{com} is the NA electricity exchange commitment defined by the scheduling algorithm, Γ_{jt}^{max} is maximum electricity swap capacity, and P_{in}^{UG} indicates the quantity of received electricity from the utility grid.

4.3. The architecture of The Proposed ARRA-DDQL Routing Algorithm in the residential sector

The early version of DQL suffers from a large overestimation bias caused by utilizing $\max_a Q(s^{'}, a)$ in the updating rule, as indicated in (4.23).

$$Q_{t+1}(s,a) \leftarrow Q_t(s,a) + \alpha_t \left(r_t + \gamma \max_a Q(s',a) - Q_t(s,a)\right) \tag{4.23}$$

In Equation (4.23), $Q_t(s,a)$ denotes the value of action a in state s, r_t defines the reward, $\alpha_t \in [0,1]$ is the learning rate, and $\gamma \in [0,1)$ indicates the discount factor. Consequently, the Double Deep Q-Learning (DDQL) algorithm was developed using two independent estimators to approximate Q-value reducing the positive bias [127]. However, high underestimation bias is still a major concern associated with the original version of DDQL since either the Q^A or Q^B is updated randomly while the Q-value of the other approximator is used instead of the corresponding one. Equation (4.24) summarize the process, where a^* and b^* are $argmax_a Q^A\,(s',a)$and $argmax_a Q^B\,(s',a)$, using in updating Q^A and Q^B, respectively.

$$\begin{cases} Q^A\,(s,a) \leftarrow Q^A\,(s,a) + \alpha\big(r + \gamma(Q^B\,(s',\,a^*) - Q^A\,(s,a)\big) \\ Q^B\,(s,a) \leftarrow Q^B\,(s,a) + \alpha\big(r + \gamma(Q^A\,(s',\,b^*) - Q^B\,(s,a)\big) \end{cases} \tag{4.24}$$

Therefore, the P3DQL algorithm was proposed in our previous work [126], focusing on the scheduling problems in an IoE-based smart grid to address the underestimation challenge in the DDQL algorithm. Although still in this technique, Q^A or Q^B is randomly chosen to be updated, the probability of utilizing $Q^B\,(s',\,a^*)$ or $Q^A\,(s',\,b^*)$ in each updating rule can be defined based on the nature of the problem. Accordingly, the update rules were modified as shown in (4.25) and (4.26), where β_1^i and β_2^i are coefficients that define whether the corresponding or the mutual approximators' Q-value is selected in the i^{th} update, respectively. It should be noted that,

$Pr(\beta_1^i = 1) = \delta^{update}$ and $Pr(\beta_1^i = 0) = 1 - \delta^{update}$, while $\beta_1^i + \beta_2^i = 1$.

$$Q^A \leftarrow Q^A \ + \alpha \left(r + \gamma \left(\beta_1^i Q^B \,(s',\, a^*) + \beta_2^i Q^A \,(s',\, b^*) \right) - Q^A \ \right) \tag{4.25}$$

$$Q^B \leftarrow Q^B \ + \alpha \left(r + \gamma \left(\beta_1^i Q^A \,(s',\, b^*) + \beta_2^i Q^B \,(s',\, a^*) \right) - Q^B \ \right) \tag{4.26}$$

The main challenge in the P3DQL algorithm is associated with the initialization of δ, which is fixed until convergence. However, once the target value shifts, δ needs to be updated to increase the model's total performance and stability, especially in problems with small state-action space like the routing optimization problem is NAs. Furthermore, the probability of selecting the estimator to update either Q^A or Q^B must be adopted with the learning process instead of randomly updating.

4.3.1. Step 1: Developing the ADDQL algorithm (Adjusting update rules considering the positive and negative biases)

This section firstly introduces the probability of selecting i^{th} estimator δ_i^{est}, which indicates which estimator is selected for the next update. Whereby $\delta_1^{est} = 1$ and $\delta_2^{est} = 1$ indicate the first (Q^A) and the second (Q^B) estimator's values need to be updated, respectively. It should be noted that $\delta_1^{est} = 1 - \delta_2^{est}$, while $\delta_i^{est} \neq 1$ indicates no preference, and an approximator is chosen for the next update in a random manner.

Each estimator tends to select the action with the highest corresponding value. As Figure 4.2 illustrates, the maximum estimated value by the first approximator appears in action a_m, which is underestimated since the expected Q value is less than the target one. Although the second estimator has not reached its peak value, the underestimation bias of this approximator is more destructive at a_m. Furthermore, the second estimator suffers from a positive bias (b^+) in action a_n, while the estimated value by the first approximator is still less than the target value.

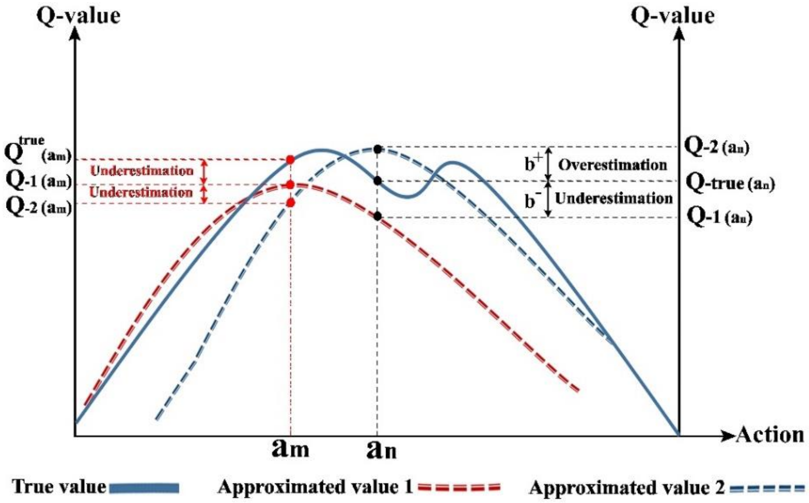


Figure 4-2. Comparing estimated and true values

The proposed ADDQL algorithm employs a target network to prevent spiraling around by exploiting the predicted value of this network to backpropagate through the main network. After each N step, δ^{est} is reinitialized as in (4.27), where $Q^i = f_i(a)$ of the i^{th} estimator, and $\Delta Q^i = Q^i - Q^{target}$.

$$\begin{cases} if\ \left\{\frac{\partial f_1}{\partial a} > 0, \quad \frac{\partial f_2}{\partial a} > 0, \quad |\Delta Q^1| > |\Delta Q^2|\right\}, & then: \quad \delta_2^{est} = 1 \\ if\ \left\{\frac{\partial f_1}{\partial a} > 0, \quad \frac{\partial f_2}{\partial a} > 0, \quad |\Delta Q^1| = |\Delta Q^2|\right\}, & then: \ 0 < \delta_{1,2}^{est} < 1 \\ if\ \left\{\frac{\partial f_1}{\partial a} > 0, \quad \frac{\partial f_2}{\partial a} > 0, \quad |\Delta Q^2| > |\Delta Q^1|\right\}, & then: \quad \delta_1^{est} = 1 \\ if\ \left\{\frac{\partial f_1}{\partial a} < 0, \quad \frac{\partial f_2}{\partial a} > 0, \quad \forall\, \Delta Q^i \in \mathcal{R}\right\}, & then: \quad \delta_2^{est} = 1 \\ if\ \left\{\frac{\partial f_1}{\partial a} > 0, \quad \frac{\partial f_2}{\partial a} < 0, \quad \forall\, \Delta Q^i \in \mathcal{R}\right\}, & then: \quad \delta_1^{est} = 1 \end{cases} \tag{4.27}$$

Moreover, in the proposed ADDQL algorithm, $Pr\left(\beta_1^i = 1\right) = \delta^{update}$ is not fixed during the training stage. The algorithm adjusts δ^{update} after each update in the target network, as indicated in (4.28).

$$\begin{cases} if\ \{\ \delta_1^{est} = 1, \Delta Q^1 > 0\}, & then: \quad \delta^{update} = 1 \\ if\ \{\ \delta_1^{est} = 1, \Delta Q^1 \leq 0\}, & then: \quad \delta^{update} = 0 \\ if\ \{\ \delta_2^{est} = 1, \Delta Q^2 > 0\}, & then: \quad \delta^{update} = 1 \\ if\ \{\ \delta_2^{est} = 1, \Delta Q^2 \leq 0\}, & then: \quad \delta^{update} = 0 \end{cases} \tag{4.28}$$

The proposed ADDQL algorithm enhances the performance of the P3DQL algorithm in terms of both positive and negative biases since δ^{est} and δ^{update} are updated in each N step.

Lemma 4.1. *Let* $V = \{v_1, v_2, \dots, v_k\}$ *be a set of values while* $\mu^W = \{\mu_1^W, \mu_2^W, \dots, \mu_k^W\}$, *and* $\mu^Z = \{\mu_1^Z, \mu_2^Z, \dots, \mu_k^Z\}$ *are two unbiased approximators updating* F^W *and* F^Z. *Also,* $\max_v F^W = F^W(v_n)$, *and* $\max_v F^Z = F^Z(v_m)$. *Then, positive and negative biases are* $b^+ = F^W - F^{target}$, *and* $b^- = F^{target} - F^Z$, *respectively.*

Lemma 4.2. *Let* $V = \{v_1, v_2, \dots, v_n\}$ *be a set of random variables and,* $G(v_i) = G, G_{min} = min\{g_1, g_2, \dots, g_n\}$, $G_{max} = max\{g_1, g_2, \dots, g_n\}$, $b_i = |G_i - G_{min}|$, $b^- = \{G\} - G_{min}$, *and* $b^+ = G_{max} - \{G\}$. *Also,* $Pr\,({x_i}^{\mu})$ *is the probability of* $x \in \{\mu | \mu = \{G, G_{min}, G_{max}\}\}$. *Then:*

$$\begin{cases} Pr\,({x_i}^{G_{min}}) \max\limits_i \left\{G_{min_i}\right\} + Pr\,({x_i}^{G_{max}}) \max\limits_i \left\{G_{max_i}\right\} < G_{min} \\ G_{max} < Pr\,({x_i}^{G_{min}}) \min\limits_i \left\{G_{min_i}\right\} + Pr\,({x_i}^{G_{max}}) \min\limits_i \left\{G_{max_i}\right\} \end{cases}$$

Theorem 4.1 illuminates that the proposed ADDQL algorithm simultaneously enhances negative and positive biases in the P3DQL algorithm. Subsequently, Lemmas 4.1 and 4.2 are utilized to prove the theorem.

Theorem 4.1. *Let* $\theta^A = \{\theta_1^A, \theta_2^A, \dots, \theta_l^A\}$, *and* $\theta^B = \{\theta_1^B, \theta_2^B, \dots, \theta_l^B\}$ *be two unbiased approximators updating* Q^A *and* Q^B *values in the P3DQL algorithm so that* $E\left\{{Q^A}_i\right\} = E\left\{{Q^B}_i\right\}$, *and* $Q_i = f_i(a_i = i^{th} action)$. *Let* $\overline{a_z}$ *and* $\underline{a_z}$ *be two elements that maximize and minimize* $\theta^{z \in \{A,B\}}$. *Also, let action values be* $Q = \{q_1, q_2, \dots, q_l\}$, *while* $Q_{max} = \{i \middle| E\{Q_i\} = max_j Q_j\}$, *and* $Q_{min} = \{k \middle| E\{Q_k\} = min_j Q_j\}$ *are two subsets of* Q. *If* ψ_u *and* ψ_o *are lower and upper bound of bias in the P3DQL, then* $\min\limits_i E\{Q_i\} = \psi_u < E\left\{{\theta^z}_{\underline{a_z}}\right\} < E\left\{{\theta^z}_{\overline{a_z}}\right\} < \psi_o = \max\limits_i E\{Q_i\}$.

Proof. *If* $\frac{\partial f_B}{\partial a} > 0$, *and* $\delta_A^{est} = 1$, *then* $|\Delta Q^B| > |\Delta Q^A|$. *since* $E\left\{\max\limits_a \{Q(a) +\right.$

$\varDelta Q^B(a)\}\Big\} \geq E\Big\{\max_a \{Q(a) + \varDelta Q^{z\in\{A,B\}}(a)\}\Big\}$. *Consequently,* $E\{\,Q^B_{max}(a)\,\} = E\Big\{\max_a\{Q(a)\ +\ \varDelta Q^2(a)\}\Big\} \geq E\Big\{\max_a\{Q(a) + \varDelta Q^1(a)\}\Big\} = E\{\,Q^A_{max}(a)\,\}$. *Ultimately, the quantity of the positive bias heretofore is less than the overestimation in the P3DQL algorithm when* $\delta^{est}_A \neq 1$. *Moreover, if* $\delta^{est}_1 = 1$, *and* $\varDelta Q^1 > 0$, *then* $Q^B\,(s',\,a^*) \leq Q^A\,(s',\,b^*)$. *Therefore, when* $\delta^{update} = 1$, *then* $\min_i E\{Q_i\} <\ E\{Q^A\}$. *Accordingly, the underestimation bias is reduced in contrast with the P3DQL algorithm. The same logic applies to the other conditions in Equations (4.27) and (4.28).*

4.3.2. Step 2: Forming an approximated reasoning-based reward function for routing optimization in NAs

The environment gives the reward signal to evaluate the quality of agents in taking action. However, a precise reward function predominantly results in a higher complexity in the RL-based solutions while demanding flexible constraints for optimality [128]. Accordingly, fuzzy approximate reasoning is utilized in this chapter to derive rules which are not precisely matched with the base rules [129].

Primarily the system's states are described as the inputs of the fuzzy interface system to start the fuzzification process. The state-space is characterized by the monetary profit of electricity swap, PV power availability, and battery SOC. Monetary Profit (MP) is defined as the profit from using external sources, including UG and NHs. Accordingly, to reduce the complexity of the problem, MP is divided into two levels as in (4.29), where ϕ is the threshold of the desired gain.

$$MP^{index} = \begin{cases} MP^{positive} & if\ MP \geq\ \phi\ \$/kW \\ MP^{negative} & if\ MP <\ \phi\ \$/kW \end{cases} \tag{4.29}$$

Moreover, the availability index of the PV output power is outlined based on the timeslot, which illustrates the quantity of irradiation. Thus, $PV^{availabe}$, and $PV^{unavailabe}$ are indices determining the availability and unavailability of PV power, respectively. Finally, three different levels classify the SOC index, whereby $SOC_t^{ess} < SOC_{charge}^{ess}$ means the battery level is low (SOC^{low}), $SOC_{charge}^{ess} \leq SOC_t^{ess} \leq SOC_{discharge}^{ess}$ indicates the battery level is at an acceptable condition and no immediate charging is required, and $SOC_t^{ess} > SOC_{discharge}^{ess}$ denotes that the battery is full and ready for discharging. Figure 4.3 depicts the membership functions of inputs.

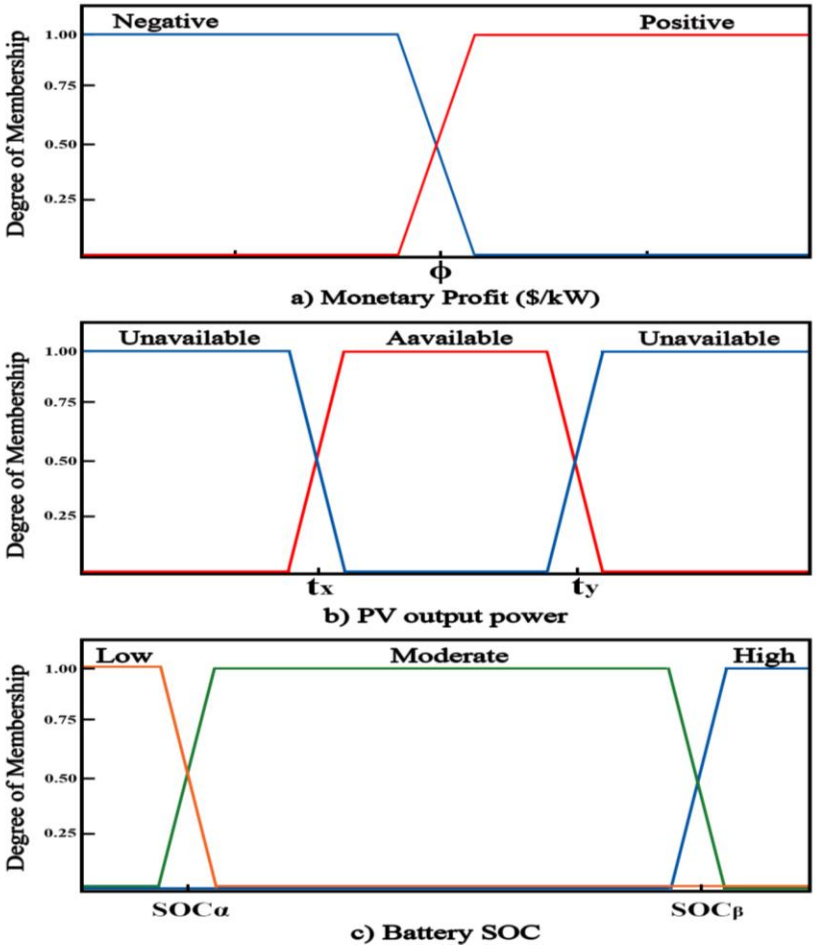


Figure 4-3. Membership functions

Four main operational modes (M1 to M4) of the system were introduced in section 2. Fuzzy rules are defined to infer and select one of these modes as an output, acquiring the functionality of each switch. Consequently, the action space is described as: $A = \{Mode 1, Mode\ 2, Mode\ 3, Mode\ 4\}$. The fuzzy reward set is defined as Awful (A), Bad (B), Good

(G), and Perfect (P), evaluating the quality of the agent's decision. Table 4.1 illustrates the fuzzy rules utilized as the link between fuzzification and defuzzification units to select the best output based on the inputs.

Table 4-1. Fuzzy rules

	Inputs			**Mode number**			
# Rule	MP^{index}	PV^{index}	SOC^{index}	**1**	**2**	**3**	**4**
1	positive	available	Low	G	P	A	A
2	positive	available	Moderate	G	G	B	B
3	positive	available	High	G	B	G	G
4	positive	unavailable	Low	P	P	A	A
5	positive	unavailable	Moderate	P	G	B	A
6	positive	unavailable	High	P	B	G	A
7	negative	available	Low	B	A	P	G
8	negative	available	Moderate	B	A	G	P
9	negative	available	High	A	B	P	P
10	negative	unavailable	Low	B	A	G	G
11	negative	unavailable	Moderate	A	B	G	G
12	negative	unavailable	High	A	A	P	P

Three special preventive measures are specified to reduce frequent switching, which may cause damage to the devices and appliances. The first limitation rule is that each action must obey the operational schedule planned by the P3QL algorithm. Then there must be no switching in the first timeslot of the operational time of appliances unless when the scheduling algorithm issues a turning-off signal. Finally, once the bidirectional converter between AC and DC links is activated, it remains functional for a minimum of two timeslots. It should be noted that the length of each timeslot is set based on the nature and conditions of the system.

4.3.3. Step 3: Developing the A2R-ADDQL algorithm

The proposed A2R-ADDQL in step 1 untangles the fixed $\boldsymbol{\delta}^{update}$ bug in the P3DQL algorithm while mitigating positive and negative biases at the same time. In the next step, as a provision for reducing the complexity of the problem, a fuzzy approximate reasoning-based reward function has been designed especially for the routing optimization problem in NAs. As Algorithm 4.1 illustrates, the proposed A2R-ADDQL algorithm is a version of the proposed ADDQL algorithm that takes advantage of a fuzzy approximate reasoning-based reward function.

```
Algorithm 1: P3DQL algorithm
Initialize network $Q$
Initialize target network $Q^*$
Initialize experience replay memory $D$ to capacity $C^{rep}$
Initialize $Q^A$ and $Q^B$
Initialize $\delta^{update}, \delta_A^{est}$, and $\delta_A^{est}$
Input  minibatch $k^{rep}$,  learning rate $\alpha^{lr}$, discount factor $\eta$, period $\Delta^{rep}$, reward decay $\varepsilon$
Inputs $S, A, \gamma, n, \epsilon$
for episode = 1, M do
      initialize sequences ${S_1}^i$
      store transition in $D$ at each episode
for all $(s, a)$
    $Q^A\ (s, a) \leftarrow (1-\gamma)^{-1}$      //Q-value estimated by A
    $Q^B\ (s, a) \leftarrow (1-\gamma)^{-1}$      //Q-value estimated by B
    $t(s, a) \leftarrow 0$                        //time of the last update
end
repeat
       choose $\beta^j = \{0,1\}$              //j is iteration number
       $Pr(\beta^j = 1) = \delta^{update}$ and $Pr(\beta^j = 0) = 1 - \delta^{update}$
           if  $\beta^j = 1$
              $\beta^{j+1} = 0$
           else
        $\Pr(\beta^{j+1} = 1) = \delta^{update}$
        observe the states
        select the rule number
        find the mode number
        adjust the reward r
        if  update $Q^A$, then:
                $Q^A \leftarrow Q^A\ + \alpha\left(r + \gamma\left(\beta_1^i Q^B\ (s',\ a^*) + \beta_2^i Q^A\ (s',\ b^*)\right) - Q^A\ \right)$
        if  update $Q^B$, then:
                $Q^B \leftarrow Q^B\ + \alpha\left(r + \gamma\left(\beta_1^i Q^A\ (s',\ b^*) + \beta_2^i Q^B\ (s',\ a^*)\right) - Q^B\ \right)$
        end if
    update $\delta^{update}$:
        if $\{\delta_A^{est} = 1, \Delta Q^A > 0\}$ then: $\delta^{update} = 1$
        if $\{\delta_A^{est} = 1, \Delta Q^A \leq 0\}$ then: $\delta^{update} = 0$
        if $\{\delta_B^{est} = 1, \Delta Q^B > 0\}$ then: $\delta^{update} = 1$
        if $\{\delta_B^{est} = 1, \Delta Q^B \leq 0\}$ then: $\delta^{update} = 0$
    update $\delta^{set}$:
        if $\left\{\frac{\partial f_A}{\partial a} > 0,\ \frac{\partial f_B}{\partial a} > 0,\ |\Delta Q^A| > |\Delta Q^B|\right\}$ then: $update\ Q^B$
        if $\left\{\frac{\partial f_A}{\partial a} > 0,\ \frac{\partial f_B}{\partial a} > 0,\ |\Delta Q^A| = |\Delta Q^B|\right\}$ then: $0 < \delta_{A,B}^{est} < 1$
        if $\left\{\frac{\partial f_A}{\partial a} > 0,\ \frac{\partial f_B}{\partial a} > 0,\ |\Delta Q^B| > |\Delta Q^A|\right\}$ then: $update\ Q^A$
        if $\left\{\frac{\partial f_A}{\partial a} < 0,\ \frac{\partial f_B}{\partial a} > 0,\ \forall\ \Delta Q^i \in \mathcal{R}\right\}$ then: $update\ Q^B$
        if $\left\{\frac{\partial f_A}{\partial a} > 0,\ \frac{\partial f_B}{\partial a} < 0,\ \forall\ \Delta Q^i \in \mathcal{R}\right\}$ then: $update\ Q^A$
    perform gradient descent and calculate the loss
    update the target network parameters
until end
```

4.4. Case study and results

The proposed A2R-ADDQL algorithm is assessed using a large real-world dataset collected by Pecan Street [130]. Seventy-five smart homes from New York and Austin are selected to conduct the evaluations. PV generation and power consumption of all energy components are reported every 15 minutes over three years. This chapter presumes that some home appliances, including washing machine, EV, water heater, dish washer, computers, LED lights, and Television, are directly compatible with DC voltage. Furthermore, retail electricity prices are collected from The U.S. Energy Information Administration (EIA) [131]. The simulation is tested using Python, 3.9.7, and MATLAB 2022, on a standard system with an Intel Core i7-97580H CPU with 16.0 GB of RAM.

All ESSs are Tesla Powerwall 1, which is a 3.3 kW wall-mounted battery system with a rechargeable lithium-ion battery pack and an internal bidirectional DC/DC converter. Additionally, the round-trip efficiency of the intended ESS is 92.5%, while $SOC_\alpha = 30\%$, $SOC_\alpha = 70\%$ and

$\lambda_{bat_t} = 0.90$. The solar system contains $4 \times 340\ W$ Canadian solar panels and a 'MVL 3K 24V U' hybrid solar inverter with an operating voltage range of 30-100V DC, converting 24V DC to 110V-120V AC, 50Hz/60Hz frequency.

Table 4-2. Devices and switches data

Appliance	Power (kW)	Schedulable	#Switch
Refrigerator	1.27	×	A-1
Water heater	5.3	✓	D-1
Pumps	3.1	✓	A-2
Television	0.15	×	D-2
EV	7.2	✓	D-3
Stove/rice cooker	2.1	×	A-3
Toaster	0.8	×	A-4
Iron	1.5	✓	A-5
Coffee maker	0.9	×	A-6
Washing Machine	1.4	✓	D-4
Dryer	3.9	✓	A-7
Dishwasher	1.7	✓	D-5
Computers	0.4	×	D-6
Air conditioner	2.9	✓	A-8
Microwave	0.9	×	A-9
Fans	0.5	×	D-7
Lighting	0.6	×	D-8
Other	0.4	×	A10

Table 2 indicates the average nominal power of all appliances in under-study residential units besides the connected switch number. Moreover, the third column illustrates whether the operational time of the device is scheduled by the developed P3DQL algorithm in [126]. Finally, the allocated switch for each appliance is indicated in the last column, where the first character demonstrates the switch type (A stands for AC, and D denotes DC voltage), and the latter one defines the switch number in the designated category. It should be noted that S-1 and S-2 are switches that connect DC and AC links via a bidirectional converter.

4.4.1. A2R-ADDQL Setup

This chapter utilizes a target network to make the training more stable by updating neural networks' parameters to make Q-value nearby the desired outcome. The size of the replay buffer is set to 10^6, while the minibatch size equals 32. Furthermore, the agent history length and target network update frequency are 4 and 10000, respectively. Moreover, ε-greedy increment $\text{ε} = 0.99$, decay step is 50, the learning rate is $\alpha^{\mathrm{lr}} = 0.001$, and discount factor $\eta = 0.99$.

It should be noted that the number of inputs and outputs of the constructed four-layer deep neural network are specified considering the total number of time slots during a day. Every 24 hours contains a total of ninety-six timeslots of 15 minutes. Also, the number of employed neurons (1500 in each layer) is specified by trial and error to choose the best fit. An RMSprop optimizer is utilized to limit the oscillations in the vertical direction. Consequently, the convergence speed is improved due to selecting a larger learning rate and step in the horizontal direction. The gradient momentum used by RMSprop is set to 0.95, the constant added to the squared gradient momentum equals 0.01, while the activation function is ReLU. Finally, the probability of selecting estimator δ^{set} is initialized to 0.5, while $Pr\left(\beta_1^i = 1\right) = \delta^{update}$ equals 0.65 at the beginning.

4.4.2. Training process

The proposed A2R-ADDQL algorithm appears more fluctuated than the P3DQL and the original version of DDQL, as shown in Figure 4.4. It is perfectly obvious that early oscillations originated from the difference in the initialized and optimum values of δ^{set} and δ^{update}. After 150 epochs, the proposed A2R-ADDQL impeccably learns to behave near the optimal policy by achieving a higher reward.

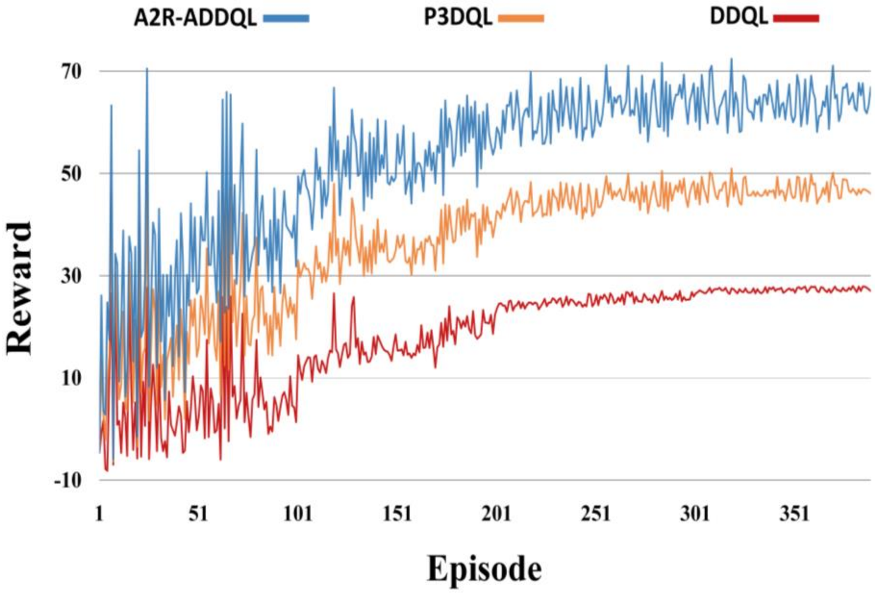


Figure 4-4. The rewards of three algorithms during training

Table 4.3 indicates an illustration of the Q-matrix of both estimators during the training process. For example, suppose the current state index is seven, the next one is ten, and the selected action is M-3. If $\delta_A^{est} = 1$ and $\delta^{update} = 1$, then $Q^B\ (s',\ a^*)$=4.22, and the Q-value vector is updated as $Q(M1{:}\,M4) = \{3.41, 3.08, 4.22, 3.72\ \}$. Updating continues until the agent meets all state-state pairs and different modes' values converge.

Table 4-3. Example of Q values

	Q^A				Q^B			
State	**M 1**	**M 2**	**M 3**	**M 4**	**M 1**	**M 2**	**M 3**	**M 4**
#7	3.14	2.74	**4.16**	3.78	2.97	2.28	**4.22**	3.69
#8	2.54	2.21	2.98	**3.58**	2.66	2.17	3.12	**3.51**
#9	2.66	2.94	**4.31**	4.18	2.83	3.12	4.14	**4.42**
#10	3.41	3.08	**3.84**	3.72	3.63	3.29	**4.03**	3.96

To determine the status of switches, the output of the fuzzy interface system at each timeslot is required first, as shown in Figure 4.5. Then the selected mode is interpreted to identify the connectivity of the switches either to link 1 or 2, considering the voltage type. For example, on 1st

July, at time slot number sixty-three (from 4:30 to 5:00 P.M.), the states are $\{MP^{positive}, PV^{availabe}, SOC^{High}\}$ and the Q-value vector of the outputs is {3.75, 2.62, 3.77, 3.81}. Accordingly, the selected mode is M-4 which aims to maximize utilizing internal resources. Consequently, the ESS and PV are connected to DC-link 1 and 2, respectively, and both the utility grid and NH are connected to AC-link 1. The refrigerator and computers are connected to DC-link 2, while Television and lighting are fed by DC-link 1. The Air conditioner and EV are connected to AC-link 1, while the DC-link 1 and AC-link 1 are coupled via S-1 and S-2.

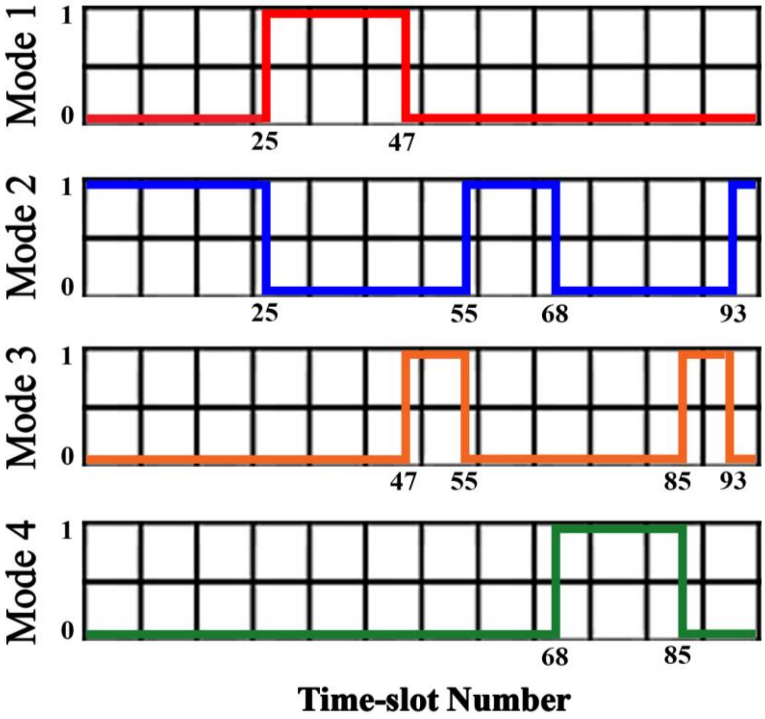


Figure 4-5. Actions taken in during a day (96 timeslots)

4.4.3. Complexity, Biases, and Speed of Convergence

The number of experiences an agent takes during the training to effectively learn a target function directly correlates with the sample complexity of the developed model. The replay memory stores all agent interactions with the environment, and the sample complexity is defined by O(|S||A||R|), where |S| denotes the number of states, |A| stands for the number of actions, and |R| is the number of rewards for each state-action pair. The proposed A2R-ADDQL algorithm turns the reward function to a non-dominant term in the complexity function. Consequently, the complexity of the A2R-ADDQL algorithm is reduced compared to the P3DQL [126] and DDQL [127] algorithms, considering the type of reward function.

The expected and actual returns are calculated to find positive or negative error amounts and evaluate the bias level in the proposed algorithm. As Table 4.4 illustrates, even though the P3DQL reduces positive and negative biases in the DQL and DDQL algorithms, the proposed algorithm performs even better than the P3DQL by a +0.001 error in return estimation. It should be mentioned that all algorithms are trained in the same environment and conditions, measuring average expected and actual returns after ten random seeds.

Moreover, the average reward and its standard deviation over fifteen runs with three different pairs of discount factors and learning rates are reported in Table 4.4. Although a smaller discount factor increases convergence speed, it has been heightened continuously during the tests to ensure the optimal policy. As the results demonstrate, the proposed algorithm reduces the standard deviation while increasing the average reward. Furthermore, the converges speed of the A2R-ADDQL is higher than the P3DQL, DDQL, and DQL by 1.2%, 6.93%, and 10.81%, respectively. It should be pointed out that the reported numbers of the average reward, standard deviation, and running time get normalized by the measured scale of corresponding runs.

Table 4-4. Performance comparison of different algorithms

Metrics/Methods	**The proposed** **A2R-ADDQL**	**P3DQL** [126]	**DDQL** [127]	**DQL** [132]
Expected return	81.36	74.31	64.69	62.45
Actual return	81.44	74.08	63.74	62.77
Error (%)	+0.001	+0.003	-1.491	+0.51
Average reward	1	0.9880	0.9307	0.8919
σ	0.0061	0.0073	0.0148	0.0295
Running Time	0.9037	0.9218	0.9842	1

4.4.4. Numerical Results: Power loss, Monetary costs, and Environmental pollution

This section analyzes the performance of the developed routing algorithm in two scenarios: 1) a P3DQL scheduling algorithm supports the routing algorithms, and 2) the routing algorithms operate solely. Moreover, the outcomes of both scenarios are compared with the result of applying algorithms in [127] and [133] on the same dataset with an identical environment. All tests are conducted during the first ten days of June, and the results are the average outcome of twenty selected units. Figure 4.5 indicates that the proposed algorithm performs better than the earlier developed methods in the literature. The monthly average cost is reduced by 24.9% and 29.1% compared with the DDQL [127] and Intelligent Power Router (IPR) [133] algorithms, respectively. Furthermore, the developed algorithm decreases power loss and greenhouse gases emission by 0.72 kW and 3.91 kg per month.

Table 4-5. Performance comparison of different methods

	Cost **($/month)**	**Loss** **(kW/month)**	**Emission** **(kg/month)**
A2R-ADDQL + P3DQL	72.65	1.33	4.93
A2R-ADDQL	104.95	1.69	7.64
IPR [133] + P3DQL	102.34	1.85	6.84
IPR	144.27	1.77	8.18
DDQL [127]+ P3DQL	96.78	1.56	6.61
DDQL	134.56	1.73	8.05

4.5. Conclusion

This chapter proposed a novel algorithm named Approximate Reasoning Reward-based Adaptable Deep Double Q-Learning (A2R-ADDQL) that was developed to optimize the energy routers' performance. The designed ER architecture contains various loads, PV systems, Electricity Storage systems (ESS), and two bidirectional power lines from the main grid and the NH.

The developed algorithm reduced the sample complexity of the model due to utilizing a fuzzy approximate reasoning reward function which reduces the number of random actions. Moreover, the developed algorithm can adjust to the nature of the problem by taking advantage of exploration and exploitation where overestimation and underestimation are favorable, respectively.

The results indicated that the proposed routing mechanism performs outstandingly since the monthly cost dropped by 41%, which is 24.9% better than utilizing other methods. Moreover, the power loss and greenhouse gases emission were reduced by 29.1% and 3.91 kg per month, respectively.

Chapter 5

Conclusion

5.1. Concluding Remarks

This thesis aims to make the application of IoE feasible to address the slow rate of development in smart grids. The critical point is how to make end-users able to reduce their electricity costs and also recognize the surplus electricity to trade in the market. The practicability of any possible solution is associated with the profitability of the entire process by decreasing the electricity expense and increasing the profit of energy exchange. Subsequently, optimizing operational scheduling and electricity routing are two underlying problems that need to be addressed. However, one of the crucial prerequisites in this regard is guaranteeing data integrity and correctness. Accordingly, the originality of data must be guaranteed before using it in solving scheduling and routing problems.

The originality of the power consumption and generation data is investigated in chapter 2. Accordingly, an intelligent intruder is first developed to generate innovative threats that the model has not previously seen. Moreover, well-known attack strategies are modeled to generate passive attacks simultaneously. Next, the quality of generated attack is examined using the proposed defense algorithm in the literature to demonstrate the necessity of a more powerful attack detection mechanism. Then, a Multi-Layer cyber defense mechanism is developed to detect both passive and active threats. The first layer takes advantage of the ensemble method in machine learning. The ensemble architecture of neural networks is more precise and robust than a single model due to the abilities stemming from this method, including overfitting avoidance, concept drifting, and dimensionality reduction. However, the main disadvantage of the ensemble method is that training multiple DNN models is costly due to the extensive computational burden. Also, the best model among all trained models usually beats the ensemble method.

Consequently, a snapshot ensemble that develops multiple models from a single training process is introduced as the solution. This technique combines different models' predictions while saving models during the training phase and employing them to create an ensemble setup. Even though the previous layer is trained with numerous attack samples created by the DQL-based attack generator, there might still be unknown attacks that are capable of passing the passive attack detection layer. Accordingly, a threat-hunting layer is required to enhance the detection rate. Furthermore, since the algorithm needs to detect unknown attacks, the model must be developed with unsupervised techniques. Deep autoencoders are feed-forward Multi-Layer neural networks consisting of an input layer, one or multiple hidden layers, and an output layer, aiming to learn data reconstructions. As a data-compression model, DAE maps the original data into a reduced-dimension representation and rebuilds the data from compressed information via a pair of encoders and decoders. In addition, the ability to discover correlations among data features makes DAEs capable of detecting FDIAs in an unsupervised manner. Besides a real-world simulation, performance evaluation proves that the proposed framework can successfully detect both passive and active FDIAs.

Once the correctness of data is guaranteed, the next two steps are optimizing scheduling and routing problems. In chapter 3, challenges originating from the high penetration of smart devices, decentralized networks, and new topologies of power systems are investigated. Operation scheduling of energy components is one of the principal problems that must be addressed. However, engaging with big data produced by the interconnected infrastructures, besides the high dimensional and uncertain environment, make traditional methods incapable of addressing these problems since exact modeling of the environment under uncertainties is impracticable. While learning-based methods suffer from excessive complexity and the curse of dimensionality, Deep Reinforcement Learning has recently successfully handled highly complex scheduling problems. However, biases and model efficiency are two primary considerations that need more investigation. Positive and negative biases lead to better exploration and exploitation, respectively, and their harmony, considering model efficiency, results in a better outcome.

Accordingly, a novel algorithm named Probabilistic Delayed Double Deep Q-Learning, which is a combination of the tuned version of Double Deep Q-Learning and Delayed Q-Learning, is proposed to optimize energy scheduling problems in IoE-based power systems. This algorithm makes a trade-off between overestimation and underestimation biases, guaranteeing sample complexity by applying a delay in updating the rule. Finally, the proposed algorithm is tested on three real-world datasets assessing its performance in various benchmarks. The results indicate that the developed model is thoroughly stable since both population and characteristic stability indices are less than 0.1 in all case studies. The average model's error is 0.028 showing the exactitude of the model while the running time is lower than other examined methods. Utilizing the developed algorithm results in an 11.1% reduction in the average power ratio. Consequently, the peak load decreased from 8.043 kW to 5.8137 kW, resulting in a 30.1% cost reduction.

Chapter 4 studies the Energy Router (ER) concept as a compact intelligent power electronic device. ERs are crucial in maximizing energy efficiency, minimizing loss and costs, and addressing growing electricity demand. However, optimizing electricity routing in the residential sector has not been well investigated. Moreover, complex modeling of the energy components besides the uncertain environment made the conventional methods impotent in tackling these problems.

Consequently, this chapter proposes a novel algorithm titled Approximate Reasoning Reward-based Adaptable Deep Double Q-Learning (A2R-ADDQL) that is introduced specially to optimize electricity routing in residential units. As a result, both overestimation and underestimation biases are reduced compared to other deep Q-Learning-based algorithms. Moreover, the sample complexity of the model is decreased due to utilizing a fuzzy approximate reasoning reward function. Ultimately, the proposed algorithm is assessed on a real-world dataset evaluating the findings in several benchmarks. The results indicate that the proposed model is unbiased while the convergence speed is higher than other analyzed techniques. Additionally, monthly average cost and power loss are lowered by 24.9% and 29.1% more than other techniques. Finally, the proposed algorithm reduces greenhouse gases emission by 3.91 kg per month.

References

[1] R. Waheed, S. Sarwar, and C. Wei, “The survey of economic growth, energy consumption and carbon emission,” *Energy Reports*, vol. 5, pp. 1103–1115, Nov. 2019, doi: 10.1016/J.EGYR.2019.07.006.

[2] H. M. Ruzbahani and H. Karimipour, “Optimal incentive-based demand response management of smart households,” *Conference Record - Industrial and Commercial Power Systems Technical Conference*, vol. 2018-May, pp. 1–7, May 2018, doi: 10.1109/ICPS.2018.8369971.

[3] “Annual Energy Outlook 2022 - U.S. Energy Information Administration (EIA).” https://www.eia.gov/outlooks/aeo/ (accessed Sep. 27, 2022).

[4] S. R. Sinsel, R. L. Riemke, and V. H. Hoffmann, “Challenges and solution technologies for the integration of variable renewable energy sources—a review,” *Renew Energy*, vol. 145, pp. 2271–2285, Jan. 2020, doi: 10.1016/J.RENENE.2019.06.147.

[5] A. Rostami, M. Mohammadi, and H. Karimipour, “Reliability Assessment of Cyber-Physical Generation System,” *Iranian Journal of Science and Technology, Transactions of Electrical Engineering 2022*, pp. 1–10, Nov. 2022, doi: 10.1007/S40998-022-00566-6.

[6] K. Wang *et al.*, “A survey on energy internet: Architecture, approach, and emerging technologies,” *IEEE Syst J*, vol. 12, no. 3, pp. 2403–2416, Sep. 2018, doi: 10.1109/JSYST.2016.2639820.

[7] H. M. Rouzbahani, H. Karimipour, A. Rahimnejad, A. Dehghantanha, and G. Srivastava, “Anomaly detection in cyber-physical systems using machine learning,” *Handbook of Big Data Privacy*, pp. 219–235, Mar. 2020, doi: 10.1007/978-3-030-38557-6_10/COVER.

[8] H. M. Rouzbahani, H. Karimipour, A. Dehghantanha, and R. M. Parizi, “Blockchain applications in power systems: A bibliometric analysis,” *Advances in Information Security*, vol. 79, pp. 129–145, 2020, doi: 10.1007/978-3-030-38181-3_7/COVER.

[9] N. Sheykhi, A. Salami, J. M. Guerrero, G. D. Agundis-Tinajero, and T. Faghihi, “A comprehensive review on telecommunication challenges of microgrids secondary control,” *International Journal of Electrical Power & Energy Systems*, vol. 140, p. 108081, Sep. 2022, doi: 10.1016/J.IJEPES.2022.108081.

[10] H. M. Rouzbahani, H. Karimipour, and L. Lei, “Multi-layer defense algorithm against deep reinforcement learning-based intruders in smart grids,” *International Journal of Electrical Power & Energy Systems*, vol. 146, p. 108798, Mar. 2023, doi: 10.1016/J.IJEPES.2022.108798.

[11] “Cyber Attacks and Energy Infrastructures: Anticipating Risks | IFRI - Institut français des relations internationales.” https://www.ifri.org/en/publications/etudes-de-lifri/cyber-attacks-and-energy-infrastructures-anticipating-risks (accessed Sep. 27, 2022).

[12] H. Karimipour and V. Dinavahi, “On false data injection attack against dynamic state estimation on smart power grids,” *2017 5th IEEE International Conference on Smart Energy Grid Engineering, SEGE 2017*, pp. 388–393, Sep. 2017, doi: 10.1109/SEGE.2017.8052831.

[13] P. Jokar, N. Arianpoo, and V. C. M. Leung, “Electricity theft detection in AMI using customers’ consumption patterns,” *IEEE Trans Smart Grid*, vol. 7, no. 1, pp. 216–226, Jan. 2016, doi: 10.1109/TSG.2015.2425222.

[14] H. Guo, J. Sun, and Z. H. Pang, “Stealthy false data injection attacks with resource constraints against multi-sensor estimation systems,” *ISA Trans*, vol. 127, pp. 32–40, Aug. 2022, doi: 10.1016/J.ISATRA.2022.02.045.

[15] R. Nawaz, M. A. Shahid, I. M. Qureshi, and M. H. Mehmood, “Machine learning based false data injection in smart grid,” *Proceedings - 2018, IEEE 1st International Conference on Power, Energy and Smart Grid, ICPESG 2018*, pp. 1–6, Jun. 2018, doi: 10.1109/ICPESG.2018.8384510.

[16] H. Badrsimaei, R. A. Hooshmand, and S. Nobakhtian, “Monte-Carlo-based data injection attack on electricity markets with network parametric and topology uncertainties,” *International Journal of Electrical Power & Energy Systems*, vol. 138, p. 107915, Jun. 2022, doi: 10.1016/J.IJEPES.2021.107915.

[17] A. Anwar, A. N. Mahmood, and M. Pickering, “Modeling and performance evaluation of stealthy false data injection attacks on smart grid in the presence of corrupted measurements,” *J Comput Syst Sci*, vol. 83, no. 1, pp. 58–72, Feb. 2017, doi: 10.1016/J.JCSS.2016.04.005.

[18] G. Dán and H. Sandberg, “Stealth attacks and protection schemes for state estimators in power systems,” *2010 1st IEEE International Conference on Smart Grid Communications, SmartGridComm 2010*, pp. 1–6, 2010, doi: 10.1109/SMARTGRID.2010.5622046.

[19] S. Paul and Z. Ni, “A Study of Linear Programming and Reinforcement Learning for One-Shot Game in Smart Grid Security,” *Proceedings of the International Joint Conference on Neural Networks*, vol. 2018-July, Oct. 2018, doi: 10.1109/IJCNN.2018.8489202.

[20] Y. Chen, S. Huang, F. Liu, Z. Wang, and X. Sun, “Evaluation of reinforcement learning-based false data injection attack to automatic voltage control,” *IEEE Trans Smart Grid*, vol. 10, no. 2, pp. 2158–2169, Mar.

2019, doi: 10.1109/TSG.2018.2790704.

[21] Z. Ni and S. Paul, "A Multistage Game in Smart Grid Security: A Reinforcement Learning Solution," *IEEE Trans Neural Netw Learn Syst*, vol. 30, no. 9, pp. 2684–2695, Sep. 2019, doi: 10.1109/TNNLS.2018.2885530.

[22] J. Tian, R. Tan, X. Guan, and T. Liu, "Enhanced hidden moving target defense in smart grids," *IEEE Trans Smart Grid*, vol. 10, no. 2, pp. 2208–2223, Mar. 2019, doi: 10.1109/TSG.2018.2791512.

[23] Y. Li and Y. Wang, "State summation for detecting false data attack on smart grid," *International Journal of Electrical Power & Energy Systems*, vol. 57, pp. 156–163, May 2014, doi: 10.1016/J.IJEPES.2013.11.057.

[24] C. Liu, J. Wu, C. Long, and D. Kundur, "Reactance Perturbation for Detecting and Identifying FDI Attacks in Power System State Estimation," *IEEE Journal on Selected Topics in Signal Processing*, vol. 12, no. 4, pp. 763–776, Aug. 2018, doi: 10.1109/JSTSP.2018.2846542.

[25] C. Liu, H. Liang, T. Chen, J. Wu, and C. Long, "Joint Admittance Perturbation and Meter Protection for Mitigating Stealthy FDI Attacks against Power System State Estimation," *IEEE Transactions on Power Systems*, vol. 35, no. 2, pp. 1468–1478, Mar. 2020, doi: 10.1109/TPWRS.2019.2938223.

[26] M. Ozay, I. Esnaola, F. T. Yarman Vural, S. R. Kulkarni, and H. V. Poor, "Machine Learning Methods for Attack Detection in the Smart Grid," *IEEE Trans Neural Netw Learn Syst*, vol. 27, no. 8, pp. 1773–1786, Aug. 2016, doi: 10.1109/TNNLS.2015.2404803.

[27] L. Yang, Y. Li, and Z. Li, "Improved-ELM method for detecting false data attack in smart grid," *International Journal of Electrical Power & Energy Systems*, vol. 91, pp. 183–191, Oct. 2017, doi: 10.1016/J.IJEPES.2017.03.011.

[28] J. Lee, J. Kim, I. Kim, and K. Han, "Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles," *IEEE Access*, vol. 7, pp. 165607–165626, 2019, doi: 10.1109/ACCESS.2019.2953095.

[29] M. Hasan, M. M. Islam, M. I. I. Zarif, and M. M. A. Hashem, "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches," *Internet of Things*, vol. 7, p. 100059, Sep. 2019, doi: 10.1016/J.IOT.2019.100059.

[30] M. Nazmul Hasan, R. N. Toma, A. al Nahid, M. M. Manjurul Islam, and J. M. Kim, "Electricity Theft Detection in Smart Grid Systems: A CNN-LSTM Based Approach," *Energies 2019, Vol. 12, Page 3310*, vol. 12, no. 17, p. 3310, Aug. 2019, doi: 10.3390/EN12173310.

[31] S. Khemakhem, M. Rekik, and L. Krichen, "Double layer home energy supervision strategies based on demand response and plug-in electric vehicle control for flattening power load curves in a smart grid," *Energy*, vol. 167, pp. 312–324, Jan. 2019, doi: 10.1016/J.ENERGY.2018.10.187.

[32] K. Mitra and G. Dutta, "A two-part dynamic pricing policy for household electricity consumption scheduling with minimized expenditure," *International Journal of Electrical Power & Energy Systems*, vol. 100, pp. 29–41, Sep. 2018, doi: 10.1016/J.IJEPES.2018.01.028.

[33] E. Shirazi and S. Jadid, "Cost reduction and peak shaving through domestic load shifting and DERs," *Energy*, vol. 124, pp. 146–159, Apr. 2017, doi: 10.1016/J.ENERGY.2017.01.148.

[34] X. Ran and K. Liu, "Robust Scatter Index Method for the Appliances Scheduling of Home Energy Local Network with User Behavior Uncertainty," *IEEE Trans Industr Inform*, vol. 15, no. 7, pp. 4129–4139, Jul. 2019, doi: 10.1109/TII.2019.2897126.

[35] I. Y. Joo and D. H. Choi, "Optimal household appliance scheduling considering consumer's electricity bill target," *IEEE Transactions on Consumer Electronics*, vol. 63, no. 1, pp. 19–27, Feb. 2017, doi: 10.1109/TCE.2017.014666.

[36] D. M. Minhas and G. Frey, "Modeling and Optimizing Energy Supply and Demand in Home Area Power Network (HAPN)," *IEEE Access*, vol. 8, pp. 2052–2072, 2020, doi: 10.1109/ACCESS.2019.2962660.

[37] Y. F. Du, L. Jiang, Y. Z. Li, J. Counsell, and J. S. Smith, "Multi-objective demand side scheduling considering the operational safety of appliances," *Appl Energy*, vol. 179, pp. 864–874, Oct. 2016, doi: 10.1016/J.APENERGY.2016.07.016.

[38] B. Hussain, Q. U. Hasan, N. Javaid, M. Guizani, A. Almogren, and A. Alamri, "An Innovative Heuristic Algorithm for IoT-Enabled Smart Homes for Developing Countries," *IEEE Access*, vol. 6, pp. 15550–15575, Feb. 2018, doi: 10.1109/ACCESS.2018.2809778.

[39] J. Zhu, Y. Lin, W. Lei, Y. Liu, and M. Tao, "Optimal household appliances scheduling of multiple smart homes using an improved cooperative algorithm," *Energy*, vol. 171, pp. 944–955, Mar. 2019, doi: 10.1016/J.ENERGY.2019.01.025.

[40] F. Luo, Z. Y. Dong, Z. Xu, W. Kong, and F. Wang, "Distributed residential energy resource scheduling with renewable uncertainties," *IET Generation, Transmission and Distribution*, vol. 12, no. 11, pp. 2770–2777, Jun. 2018, doi: 10.1049/IET-GTD.2017.1136/CITE/REFWORKS.

[41] F. Luo, W. Kong, G. Ranzi, and Z. Y. Dong, "Optimal home energy management system with demand

charge tariff and appliance operational dependencies," *IEEE Trans Smart Grid*, vol. 11, no. 1, pp. 4–14, Jan. 2020, doi: 10.1109/TSG.2019.2915679.

[42] V. Pilloni, A. Floris, A. Meloni, and L. Atzori, "Smart Home Energy Management Including Renewable Sources: A QoE-Driven Approach," *IEEE Trans Smart Grid*, vol. 9, no. 3, pp. 2006–2018, May 2018, doi: 10.1109/TSG.2016.2605182.

[43] A. Alahyari, M. Ehsan, and M. S. Mousavizadeh, "A hybrid storage-wind virtual power plant (VPP) participation in the electricity markets: A self-scheduling optimization considering price, renewable generation, and electric vehicles uncertainties," *J Energy Storage*, vol. 25, p. 100812, Oct. 2019, doi: 10.1016/J.EST.2019.100812.

[44] J. Faraji, A. Ketabi, H. Hashemi-Dezaki, M. Shafie-Khah, and J. P. S. Catalao, "Optimal day-ahead self-scheduling and operation of prosumer microgrids using hybrid machine learning-based weather and load forecasting," *IEEE Access*, vol. 8, pp. 157284–157305, 2020, doi: 10.1109/ACCESS.2020.3019562.

[45] P. Moutis and N. D. Hatziargyriou, "Decision trees aided scheduling for firm power capacity provision by virtual power plants," *International Journal of Electrical Power & Energy Systems*, vol. 63, pp. 730–739, Dec. 2014, doi: 10.1016/J.IJEPES.2014.06.038.

[46] H. M. Rouzbahani, H. Karimipour, and L. Lei, "A review on virtual power plant for energy management," *Sustainable Energy Technologies and Assessments*, vol. 47, p. 101370, Oct. 2021, doi: 10.1016/J.SETA.2021.101370.

[47] M. A. Lopes Silva, S. R. de Souza, M. J. Freitas Souza, and A. L. C. Bazzan, "A reinforcement learning-based multi-agent framework applied for solving routing and scheduling problems," *Expert Syst Appl*, vol. 131, pp. 148–171, Oct. 2019, doi: 10.1016/J.ESWA.2019.04.056.

[48] Q. Zhang, K. Dehghanpour, Z. Wang, and Q. Huang, "A Learning-Based Power Management Method for Networked Microgrids under Incomplete Information," *IEEE Trans Smart Grid*, vol. 11, no. 2, pp. 1193–1204, Mar. 2020, doi: 10.1109/TSG.2019.2933502.

[49] M. Khan, J. Seo, and D. Kim, "Real-Time Scheduling of Operational Time for Smart Home Appliances Based on Reinforcement Learning," *IEEE Access*, vol. 8, pp. 116520–116534, 2020, doi: 10.1109/ACCESS.2020.3004151.

[50] S. Lee and D. H. Choi, "Reinforcement Learning-Based Energy Management of Smart Home with Rooftop Solar Photovoltaic System, Energy Storage System, and Home Appliances," *Sensors 2019, Vol. 19, Page 3937*, vol. 19, no. 18, p. 3937, Sep. 2019, doi: 10.3390/S19183937.

[51] T. Remani, E. A. Jasmin, and T. P. I. Ahamed, "Residential Load Scheduling with Renewable Generation in the Smart Grid: A Reinforcement Learning Approach," *IEEE Syst J*, vol. 13, no. 3, pp. 3283–3294, Sep. 2019, doi: 10.1109/JSYST.2018.2855689.

[52] Y. Ji *et al.*, "Data-Driven Online Energy Scheduling of a Microgrid Based on Deep Reinforcement Learning," *Energies 2021, Vol. 14, Page 2120*, vol. 14, no. 8, p. 2120, Apr. 2021, doi: 10.3390/EN14082120.

[53] W. Cui and W. Yu, "Scalable Deep Reinforcement Learning for Routing and Spectrum Access in Physical Layer," *IEEE Transactions on Communications*, vol. 69, no. 12, pp. 8200–8213, Dec. 2021, doi: 10.1109/TCOMM.2021.3113948.

[54] V. H. Bui and W. Su, "Real-time operation of distribution network: A deep reinforcement learning-based reconfiguration approach," *Sustainable Energy Technologies and Assessments*, vol. 50, p. 101841, Mar. 2022, doi: 10.1016/J.SETA.2021.101841.

[55] H. M. Chung, S. Maharjan, Y. Zhang, and F. Eliassen, "Distributed Deep Reinforcement Learning for Intelligent Load Scheduling in Residential Smart Grids," *IEEE Trans Industr Inform*, vol. 17, no. 4, pp. 2752–2763, Apr. 2021, doi: 10.1109/TII.2020.3007167.

[56] E. Mocanu *et al.*, "On-Line Building Energy Optimization Using Deep Reinforcement Learning," *IEEE Trans Smart Grid*, vol. 10, no. 4, pp. 3698–3708, Jul. 2019, doi: 10.1109/TSG.2018.2834219.

[57] Z. Wan, H. Li, H. He, and D. Prokhorov, "Model-Free Real-Time EV Charging Scheduling Based on Deep Reinforcement Learning," *IEEE Trans Smart Grid*, vol. 10, no. 5, pp. 5246–5257, Sep. 2018, doi: 10.1109/TSG.2018.2879572.

[58] X. Lu, X. Xiao, L. Xiao, C. Dai, M. Peng, and H. V. Poor, "Reinforcement Learning-Based Microgrid Energy Trading With a Reduced Power Plant Schedule," *IEEE Internet Things J*, vol. 6, no. 6, pp. 10728–10737, Dec. 2019, doi: 10.1109/JIOT.2019.2941498.

[59] D. Domínguez-Barbero, J. García-González, M. A. Sanz-Bobi, and E. F. Sánchez-Úbeda, "Optimising a Microgrid System by Deep Reinforcement Learning Techniques," *Energies 2020, Vol. 13, Page 2830*, vol. 13, no. 11, p. 2830, Jun. 2020, doi: 10.3390/EN13112830.

[60] Y. Wang *et al.*, "Multi-objective workflow scheduling with deep-Q-network-based multi-agent reinforcement learning," *IEEE Access*, vol. 7, pp. 39974–39982, 2019, doi:

10.1109/ACCESS.2019.2902846.

[61] S. Lee and Y. H. Lee, “Improving Emergency Department Efficiency by Patient Scheduling Using Deep Reinforcement Learning,” *Healthcare 2020, Vol. 8, Page 77*, vol. 8, no. 2, p. 77, Mar. 2020, doi: 10.3390/HEALTHCARE8020077.

[62] D. Wang and M. Hu, “Deep Deterministic Policy Gradient With Compatible Critic Network,” *IEEE Trans Neural Netw Learn Syst*, 2021, doi: 10.1109/TNNLS.2021.3117790.

[63] M. Xia, M. Chen, and Q. Chen, “Rule-based energy buffer strategy of energy router considering efficiency optimization,” *International Journal of Electrical Power & Energy Systems*, vol. 125, p. 106378, Feb. 2021, doi: 10.1016/J.IJEPES.2020.106378.

[64] S. M. Suhail Hussain, M. A. Aftab, F. Nadeem, I. Ali, and T. S. Ustun, “Optimal Energy Routing in Microgrids with IEC 61850 Based Energy Routers,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 6, pp. 5161–5169, Jun. 2020, doi: 10.1109/TIE.2019.2927154.

[65] S. Hebal, D. Mechta, S. Harous, and M. Dhriyyef, “Hybrid Energy Routing Approach for Energy Internet,” *Energies 2021, Vol. 14, Page 2579*, vol. 14, no. 9, p. 2579, Apr. 2021, doi: 10.3390/EN14092579.

[66] Y. Du, X. Yin, J. Lai, Z. Ullah, Z. Wang, and J. Hu, “Energy optimization and routing control strategy for energy router based multi-energy interconnected energy system,” *International Journal of Electrical Power & Energy Systems*, vol. 133, p. 107110, Dec. 2021, doi: 10.1016/J.IJEPES.2021.107110.

[67] R. Wang, Q. Sun, D. Ma, and Z. Liu, “The Small-Signal Stability Analysis of the Droop-Controlled Converter in Electromagnetic Timescale,” *IEEE Trans Sustain Energy*, vol. 10, no. 3, pp. 1459–1469, Jul. 2019, doi: 10.1109/TSTE.2019.2894633.

[68] P. Yi, T. Zhu, B. Jiang, R. Jin, and B. Wang, “Deploying Energy Routers in an Energy Internet Based on Electric Vehicles,” *IEEE Trans Veh Technol*, vol. 65, no. 6, pp. 4714–4725, Jun. 2016, doi: 10.1109/TVT.2016.2549269.

[69] Q. Sun, B. Huang, D. Li, D. Ma, and Y. Zhang, “Optimal Placement of Energy Storage Devices in Microgrids via Structure Preserving Energy Function,” *IEEE Trans Industr Inform*, vol. 12, no. 3, pp. 1166–1179, Jun. 2016, doi: 10.1109/TII.2016.2557816.

[70] B. Liu *et al.*, “An AC-DC Hybrid Multi-Port Energy Router with Coordinated Control and Energy Management Strategies,” *IEEE Access*, vol. 7, pp. 109069–109082, 2019, doi: 10.1109/ACCESS.2019.2933469.

[71] J. Yu, L. Xiao, Z. Hu, Y. Zhao, and J. Nie, “Multi-Terminal Energy Router and Its Distributed Control Strategy in Micro-grid Community Applications,” *2020 Asia Energy and Electrical Engineering Symposium, AEEES 2020*, pp. 1028–1033, May 2020, doi: 10.1109/AEEES48850.2020.9121396.

[72] Y. Liu, Y. Fang, and J. Li, “Interconnecting Microgrids via the Energy Router with Smart Energy Management,” *Energies 2017, Vol. 10, Page 1297*, vol. 10, no. 9, p. 1297, Aug. 2017, doi: 10.3390/EN10091297.

[73] R. Wang, J. Wu, Z. Qian, Z. Lin, and X. He, “A Graph Theory Based Energy Routing Algorithm in Energy Local Area Network,” *IEEE Trans Industr Inform*, vol. 13, no. 6, pp. 3275–3285, Dec. 2017, doi: 10.1109/TII.2017.2713040.

[74] N. G. Paterakis, O. Erdinc, I. N. Pappi, A. G. Bakirtzis, and J. P. S. Catalao, “Coordinated Operation of a Neighborhood of Smart Households Comprising Electric Vehicles, Energy Storage and Distributed Generation,” *IEEE Trans Smart Grid*, vol. 7, no. 6, pp. 2736–2747, Nov. 2016, doi: 10.1109/TSG.2015.2512501.

[75] Y. Liu, J. Li, Y. Wu, and F. Zhou, “Coordinated Control of the Energy Router-Based Smart Home Energy Management System,” *Applied Sciences 2017, Vol. 7, Page 943*, vol. 7, no. 9, p. 943, Sep. 2017, doi: 10.3390/APP7090943.

[76] Y. Liu, X. Chen, Y. Wu, K. Yang, J. Zhu, and B. Li, “Enabling the Smart and Flexible Management of Energy Prosumers via the Energy Router with Parallel Operation Mode,” *IEEE Access*, vol. 8, pp. 35038–35047, 2020, doi: 10.1109/ACCESS.2020.2973857.

[77] E. Foruzan, L. K. Soh, and S. Asgarpoor, “Reinforcement Learning Approach for Optimal Distributed Energy Management in a Microgrid,” *IEEE Transactions on Power Systems*, vol. 33, no. 5, pp. 5749–5758, Sep. 2018, doi: 10.1109/TPWRS.2018.2823641.

[78] G. Shi, D. Liu, and Q. Wei, “Echo state network-based Q-learning method for optimal battery control of offices combined with renewable energy,” *IET Control Theory and Applications*, vol. 11, no. 7, pp. 915–922, Apr. 2017, doi: 10.1049/IET-CTA.2016.0653/CITE/REFWORKS.

[79] D. L. Wang, Q. Y. Sun, Y. Y. Li, and X. R. Liu, “Optimal Energy Routing Design in Energy Internet with Multiple Energy Routing Centers Using Artificial Neural Network-Based Reinforcement Learning Method,” *Applied Sciences 2019, Vol. 9, Page 520*, vol. 9, no. 3, p. 520, Feb. 2019, doi: 10.3390/APP9030520.

[80] A. Hansen, J. Staggs, and S. Shenoi, “Security analysis of an advanced metering infrastructure,” *International Journal of Critical Infrastructure Protection*, vol. 18, pp. 3–19, Sep. 2017, doi: 10.1016/J.IJCIP.2017.03.004.

[81] H. M. Rouzbahani, Z. Faraji, M. Amiri-Zarandi, and H. Karimipour, “AI-Enabled Security Monitoring in Smart Cyber Physical Grids,” *Security of Cyber-Physical Systems*, pp. 145–167, 2020, doi: 10.1007/978-3-030-45541-5_8.

[82] V. Krishnan and F. Pasqualetti, “Data-Driven Attack Detection for Linear Systems,” *IEEE Control Syst Lett*, vol. 5, no. 2, pp. 671–676, Apr. 2021, doi: 10.1109/LCSYS.2020.3005102.

[83] H. Karimipour and V. Dinavahi, “Extended Kalman Filter-Based Parallel Dynamic State Estimation,” *IEEE Trans Smart Grid*, vol. 6, no. 3, pp. 1539–1549, May 2015, doi: 10.1109/TSG.2014.2387169.

[84] R. Deng, P. Zhuang, and H. Liang, “False Data Injection Attacks Against State Estimation in Power Distribution Systems,” *IEEE Trans Smart Grid*, vol. 10, no. 3, pp. 2871–2881, May 2019, doi: 10.1109/TSG.2018.2813280.

[85] H. M. Rouzbahani, H. Karimipour, and L. Lei, “An Ensemble Deep Convolutional Neural Network Model for Electricity Theft Detection in Smart Grids,” *Conf Proc IEEE Int Conf Syst Man Cybern*, vol. 2020-October, pp. 3637–3642, Oct. 2020, doi: 10.1109/SMC42975.2020.9282837.

[86] M. A. Rahman and H. Mohsenian-Rad, “False data injection attacks with incomplete information against smart power grids,” *GLOBECOM - IEEE Global Telecommunications Conference*, pp. 3153–3158, 2012, doi: 10.1109/GLOCOM.2012.6503599.

[87] D. Xue, X. Jing, and H. Liu, “Detection of false data injection attacks in smart grid utilizing elm-based ocon framework,” *IEEE Access*, vol. 7, pp. 31762–31773, 2019, doi: 10.1109/ACCESS.2019.2902910.

[88] M. Rana, “Architecture of the internet of energy network: An application to smart grid communications,” *IEEE Access*, vol. 5, pp. 4704–4710, 2017, doi: 10.1109/ACCESS.2017.2683503.

[89] H. Karimipour and V. Dinavahi, “Robust Massively Parallel Dynamic State Estimation of Power Systems Against Cyber-Attack,” *IEEE Access*, vol. 6, pp. 2984–2995, Dec. 2017, doi: 10.1109/ACCESS.2017.2786584.

[90] A. Yazdinejad, M. Kazemi, R. M. Parizi, A. Dehghantanha, and H. Karimipour, “An ensemble deep learning model for cyber threat hunting in industrial internet of things,” *Digital Communications and Networks*, Sep. 2022, doi: 10.1016/J.DCAN.2022.09.008.

[91] H. M. Rouzbahani, A. H. Bahrami, and H. Karimipour, “A Snapshot Ensemble Deep Neural Network Model for Attack Detection in Industrial Internet of Things,” *AI-Enabled Threat Detection and Security Analysis for Industrial IoT*, pp. 181–194, 2021, doi: 10.1007/978-3-030-76613-9_10.

[92] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, “Snapshot Ensembles: Train 1, get M for free,” *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, Apr. 2017, doi: 10.48550/arxiv.1704.00109.

[93] A. Yazdinejad, B. Zolfaghari, A. Dehghantanha, H. Karimipour, G. Srivastava, and R. M. Parizi, “Accurate threat hunting in industrial internet of things edge devices,” *Digital Communications and Networks*, Sep. 2022, doi: 10.1016/J.DCAN.2022.09.010.

[94] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, “Autoencoder-based network anomaly detection,” *Wireless Telecommunications Symposium*, vol. 2018-April, pp. 1–5, May 2018, doi: 10.1109/WTS.2018.8363930.

[95] D. Yao, M. Wen, X. Liang, Z. Fu, K. Zhang, and B. Yang, “Energy Theft Detection with Energy Privacy Preservation in the Smart Grid,” *IEEE Internet Things J*, vol. 6, no. 5, pp. 7659–7669, Oct. 2019, doi: 10.1109/JIOT.2019.2903312.

[96] Z. Zheng, Y. Yang, X. Niu, H. N. Dai, and Y. Zhou, “Wide and Deep Convolutional Neural Networks for Electricity-Theft Detection to Secure Smart Grids,” *IEEE Trans Industr Inform*, vol. 14, no. 4, pp. 1606–1615, Apr. 2018, doi: 10.1109/TII.2017.2785963.

[97] A. N. Jahromi, H. Karimipour, and A. Dehghantanha, “An ensemble deep federated learning cyber-threat hunting model for Industrial Internet of Things,” *Comput Commun*, vol. 198, pp. 108–116, Jan. 2023, doi: 10.1016/J.COMCOM.2022.11.009.

[98] A. Alabassi, A. N. Jahromi, H. Karimipour, A. Dehghantanha, P. Siano, and H. Leung, “A Self-Tuning Cyber-Attacks’ Location Identification Approach for Critical Infrastructures,” *IEEE Trans Industr Inform*, vol. 18, no. 7, pp. 5018–5027, Jul. 2022, doi: 10.1109/TII.2021.3133361.

[99] M. Cui, J. Wang, and B. Chen, “Flexible Machine Learning-Based Cyberattack Detection Using Spatiotemporal Patterns for Distribution Systems,” *IEEE Trans Smart Grid*, vol. 11, no. 2, pp. 1805–1808, Mar. 2020, doi: 10.1109/TSG.2020.2965797.

[100] M. Nazmul Hasan, R. N. Toma, A. al Nahid, M. M. Manjurul Islam, and J. M. Kim, “Electricity Theft Detection in Smart Grid Systems: A CNN-LSTM Based Approach,” *Energies 2019, Vol. 12, Page 3310*, vol. 12, no. 17, p. 3310, Aug. 2019, doi: 10.3390/EN12173310.

[101] Z. Zheng, Y. Yang, X. Niu, H. N. Dai, and Y. Zhou, “Wide and Deep Convolutional Neural Networks for Electricity-Theft Detection to Secure Smart Grids,” *IEEE Trans Industr Inform*, vol. 14, no. 4, pp. 1606–1615, Apr. 2018, doi: 10.1109/TII.2017.2785963.

[102] A. Azadeh and S. Tarverdian, “Integration of genetic algorithm, computer simulation and design of experiments for forecasting electrical energy consumption,” *Energy Policy*, vol. 35, no. 10, pp. 5229–5241, Oct. 2007, doi: 10.1016/J.ENPOL.2007.04.020.

[103] G. Aydin, “The Modeling and Projection of Primary Energy Consumption by the Sources,” *http://dx.doi.org/10.1080/15567249.2013.771716*, vol. 10, no. 1, pp. 67–74, Jan. 2014, doi: 10.1080/15567249.2013.771716.

[104] H. Chamandoust, S. Bahramara, and G. Derakhshan, “Day-ahead scheduling problem of smart micro-grid with high penetration of wind energy and demand side management strategies,” *Sustainable Energy Technologies and Assessments*, vol. 40, p. 100747, Aug. 2020, doi: 10.1016/J.SETA.2020.100747.

[105] F. Li, J. Qin, and W. X. Zheng, “Distributed Q-Learning-Based Online Optimization Algorithm for Unit Commitment and Dispatch in Smart Grid,” *IEEE Trans Cybern*, vol. 50, no. 9, pp. 4146–4156, Sep. 2020, doi: 10.1109/TCYB.2019.2921475.

[106] H. M. Ruzbahani, A. Rahimnejad, and H. Karimipour, “Smart Households Demand Response Management with Micro Grid,” *2019 IEEE Power and Energy Society Innovative Smart Grid Technologies Conference, ISGT 2019*, Feb. 2019, doi: 10.1109/ISGT.2019.8791595.

[107] A. Alarifi, A. Ali AlZubi, O. Alfarraj, and A. Alwadain, “Automated control scheduling to improve the operative performance of smart renewable energy systems,” *Sustainable Energy Technologies and Assessments*, vol. 45, p. 101036, Jun. 2021, doi: 10.1016/J.SETA.2021.101036.

[108] J. Li, W. Shi, N. Zhang, and X. Shen, “Delay-Aware VNF Scheduling: A Reinforcement Learning Approach with Variable Action Set,” *IEEE Trans Cogn Commun Netw*, vol. 7, no. 1, pp. 304–318, Mar. 2021, doi: 10.1109/TCCN.2020.2988908.

[109] Y. Pan, “Heading toward Artificial Intelligence 2.0,” *Engineering*, vol. 2, no. 4, pp. 409–413, Dec. 2016, doi: 10.1016/J.ENG.2016.04.018.

[110] D. Zhang, 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*, vol. 4, no. 3, pp. 362–370, Sep. 2018, doi: 10.17775/CSEEJPES.2018.00520.

[111] A. Lassetter and E. Cotilla-Sanchez, “Exponential modeling of equipment degradation in the grid for more reliable contingency analysis,” *SEST 2021 - 4th International Conference on Smart Energy Systems and Technologies*, Sep. 2021, doi: 10.1109/SEST50973.2021.9543166.

[112] D. M. Minhas and G. Frey, “Modeling and Optimizing Energy Supply and Demand in Home Area Power Network (HAPN),” *IEEE Access*, vol. 8, pp. 2052–2072, 2020, doi: 10.1109/ACCESS.2019.2962660.

[113] H. van Hasselt, A. Guez, and D. Silver, “Deep Reinforcement Learning with Double Q-learning,” *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pp. 2094–2100, Sep. 2015, doi: 10.48550/arxiv.1509.06461.

[114] S. Fujimoto, H. van Hoof, and D. Meger, “Addressing Function Approximation Error in Actor-Critic Methods,” *35th International Conference on Machine Learning, ICML 2018*, vol. 4, pp. 2587–2601, Feb. 2018, Accessed: Jan. 20, 2022. [Online]. Available: https://arxiv.org/abs/1802.09477v3

[115] Z. Zhang, Z. Pan, and M. J. Kochenderfer, “Weighted double Q-learning,” *IJCAI International Joint Conference on Artificial Intelligence*, vol. 0, pp. 3455–3461, 2017, doi: 10.24963/IJCAI.2017/483.

[116] A. L. Strehl, L. Lihong, E. Wiewiora, J. Langford, and M. L. Littman, “PAC model-free reinforcement learning,” *ACM International Conference Proceeding Series*, vol. 148, pp. 881–888, 2006, doi: 10.1145/1143844.1143955.

[117] L. Certification, “Pecan Street Dataport,” 2020. https://www.pecanstreet.org/dataport/

[118] M. Wenninger, A. Maier, and J. Schmidt, “DEDDIAG, a domestic electricity demand dataset of individual appliances in Germany,” *Sci Data*, vol. 8, no. 1, p. 176, 2021, doi: 10.1038/s41597-021-00963-2.

[119] Q. Zhang, M. Lin, L. T. Yang, Z. Chen, S. U. Khan, and P. Li, “A Double Deep Q-Learning Model for Energy-Efficient Edge Scheduling,” *IEEE Trans Serv Comput*, vol. 12, no. 5, pp. 739–749, Sep. 2019, doi: 10.1109/TSC.2018.2867482.

[120] J. Shair, X. Xie, W. Liu, X. Li, and H. Li, “Modeling and stability analysis methods for investigating subsynchronous control interaction in large-scale wind power systems,” *Renewable and Sustainable Energy Reviews*, vol. 135, p. 110420, Jan. 2021, doi: 10.1016/J.RSER.2020.110420.

[121] T. Buechler, F. Pagel, T. Petitjean, M. Draz, and S. Albayrak, “Optimal Energy Supply Scheduling for a Single Household: Integrating Machine Learning for Power Forecasting,” *Proceedings of 2019 IEEE PES Innovative Smart Grid Technologies Europe, ISGT-Europe 2019*, Sep. 2019, doi: 10.1109/ISGTEUROPE.2019.8905536.

[122] J. T. Meyer, L. A. Agrofoglio, J. Clement, Q. Liu, O. Yurdakul, and S. Albayrak, "Multi-objective residential electricity scheduling based on forecasting generation and demand via LSTM," *IEEE PES Innovative Smart Grid Technologies Conference Europe*, vol. 2020-October, pp. 270–274, Oct. 2020, doi: 10.1109/ISGT-EUROPE47291.2020.9248784.

[123] S. M. Suhail Hussain, F. Nadeem, M. A. Aftab, I. Ali, and T. S. Ustun, "The Emerging Energy Internet: Architecture, Benefits, Challenges, and Future Prospects," *Electronics 2019, Vol. 8, Page 1037*, vol. 8, no. 9, p. 1037, Sep. 2019, doi: 10.3390/ELECTRONICS8091037.

[124] X. Shi, Y. Xu, and H. Sun, "A Biased Min-Consensus-Based Approach for Optimal Power Transaction in Multi-Energy-Router Systems," *IEEE Trans Sustain Energy*, vol. 11, no. 1, pp. 217–228, Jan. 2020, doi: 10.1109/TSTE.2018.2889643.

[125] S. A. Y. Mustufa and D. Brunelli, "Home Energy Router for smart consumer grids," *Proceedings - 2015 International Symposium on Smart Electric Distribution Systems and Technologies, EDST 2015*, pp. 505–509, Nov. 2015, doi: 10.1109/SEDST.2015.7315260.

[126] H. M. Rouzbahani, H. Karimipour, and L. Lei, "Optimizing scheduling policy in smart grids using probabilistic Delayed Double Deep Q-Learning (P3DQL) algorithm," *Sustainable Energy Technologies and Assessments*, vol. 53, p. 102712, Oct. 2022, doi: 10.1016/J.SETA.2022.102712.

[127] H. van Hasselt, A. Guez, and D. Silver, "Deep Reinforcement Learning with Double Q-learning," *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pp. 2094–2100, Sep. 2015, doi: 10.1609/aaai.v30i1.10295.

[128] P. Kofinas, G. Vouros, and A. I. Dounis, "Energy management in solar microgrid via reinforcement learning using fuzzy reward," *https://doi.org/10.1080/17512549.2017.1314832*, vol. 12, no. 1, pp. 97–115, Jan. 2017, doi: 10.1080/17512549.2017.1314832.

[129] A. Yazdinejad, A. Dehghantanha, R. M. Parizi, G. Srivastava, and H. Karimipour, "Secure Intelligent Fuzzy Blockchain Framework: Effective Threat Detection in IoT Networks," *Comput Ind*, vol. 144, p. 103801, Jan. 2023, doi: 10.1016/J.COMPIND.2022.103801.

[130] "Dataport – Pecan Street Inc." https://www.pecanstreet.org/dataport/ (accessed Aug. 29, 2022).

[131] "Homepage - U.S. Energy Information Administration (EIA)." https://www.eia.gov/ (accessed Aug. 29, 2022).

[132] H. van Hasselt, "Double Q-learning," *Adv Neural Inf Process Syst*, vol. 23, 2010.

[133] J. M. Rodriguez-Bernuz, E. Prieto-Araujo, F. Girbau-Llistuella, A. Sumper, R. Villafafila-Robles, and J. A. Vidal-Clos, "Experimental validation of a single phase Intelligent Power Router," *Sustainable Energy, Grids and Networks*, vol. 4, pp. 1–15, Dec. 2015, doi: 10.1016/J.SEGAN.2015.07.001.

Appendix A

Copyright Permission Letters

To Whom It May Concern:

I, Hadis Karimipour, hereby grant permission to ‘Hossein Mohammadi Rouzbahani’ to reuse the below articles in his thesis titled “Application of Internet of Energy in Smart Grids Using Deep Reinforcement Learning”.

1. Hossein Mohammadi Rouzbahani, Hadis Karimipour, Lei Lei, “A review on virtual power plant for energy management”, Sustainable Energy Technologies and Assessments, Volume 47, 2021, 101370, ISSN 2213-1388, https://doi.org/10.1016/j.seta.2021.101370.

2. Hossein Mohammadi Rouzbahani, Hadis Karimipour, Lei Lei, “Optimizing scheduling policy in smart grids using probabilistic Delayed Double Deep Q-Learning (P3DQL) algorithm”, Sustainable Energy Technologies and Assessments, Volume 53, Part C, 2022, 102712, ISSN 2213-1388, https://doi.org/10.1016/j.seta.2022.102712.

3. Hossein Mohammadi Rouzbahani, Hadis Karimipour, Lei Lei, “Multi-Layer Defense Algorithm Against Deep Reinforcement Learning-based Intruders in Smart Grids”, International Journal of Electrical Power and Energy Systems. (Accepted)

4. Hossein Mohammadi Rouzbahani, Hadis Karimipour, Lei Lei, “Optimizing Resource Swap Functionality in IoE-based Grids Using Approximate Reasoning Reward-based Adjustable Deep Double Q-Learning”, IEEE Transactions on Consumer Electronics. (Submitted)

I agree to the terms outlined in the University of Calgary Non-Exclusive Distribution License.
I am aware that all University of Calgary Theses are also achieved by the Library and Archives Canada (LAC), and the University of Calgary Theses may be submitted to ProQuest.

Date:

Signature:

To Whom It May Concern:

I, Lei Lei, hereby grant permission to ‘Hossein Mohammadi Rouzbahani’ to reuse the below articles in his thesis titled “Application of Internet of Energy in Smart Grids Using Deep Reinforcement Learning”.

1. Hossein Mohammadi Rouzbahani, Hadis Karimipour, Lei Lei, “A review on virtual power plant for energy management”, Sustainable Energy Technologies and Assessments, Volume 47, 2021, 101370, ISSN 2213-1388, https://doi.org/10.1016/j.seta.2021.101370.

2. Hossein Mohammadi Rouzbahani, Hadis Karimipour, Lei Lei, “Optimizing scheduling policy in smart grids using probabilistic Delayed Double Deep Q-Learning (P3DQL) algorithm”, Sustainable Energy Technologies and Assessments, Volume 53, Part C, 2022, 102712, ISSN 2213-1388, https://doi.org/10.1016/j.seta.2022.102712.

3. Hossein Mohammadi Rouzbahani, Hadis Karimipour, Lei Lei, “Multi-Layer Defense Algorithm Against Deep Reinforcement Learning-based Intruders in Smart Grids”, International Journal of Electrical Power and Energy Systems. (Accepted)

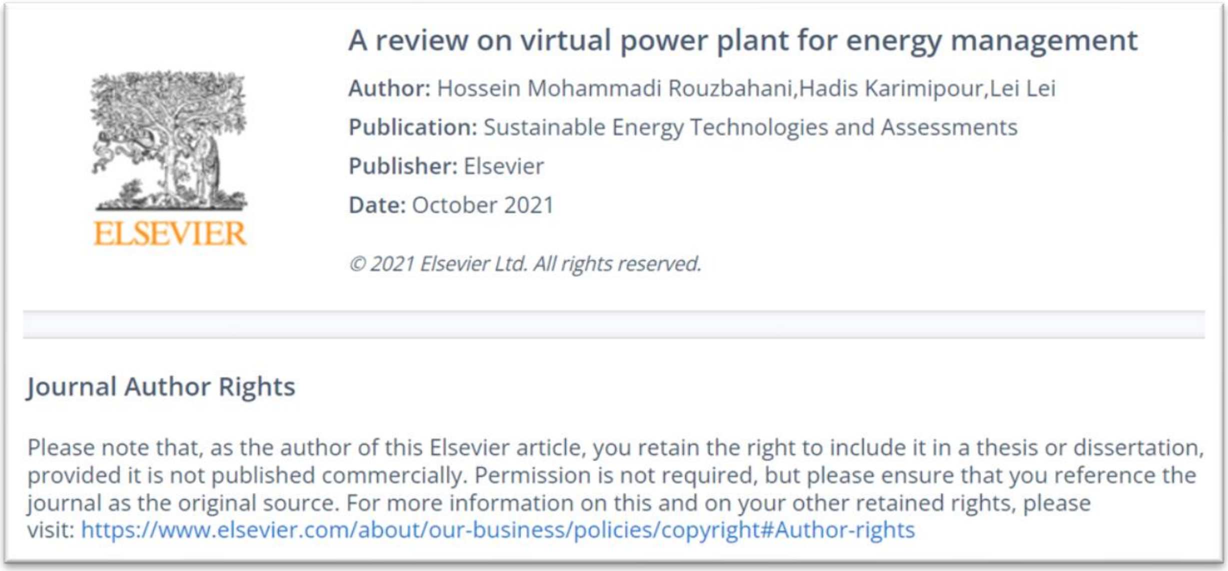
4. Hossein Mohammadi Rouzbahani, Hadis Karimipour, Lei Lei, “Optimizing Resource Swap Functionality in IoE-based Grids Using Approximate Reasoning Reward-based Adjustable Deep Double Q-Learning”, IEEE Transactions on Consumer Electronics. (Submitted)

I agree to the terms outlined in the University of Calgary Non-Exclusive Distribution License.

I am aware that all University of Calgary Theses are also achieved by the Library and Archives Canada (LAC) and the University of Calgary Theses may be submitted to ProQuest.

Date:

Signature:



A review on virtual power plant for energy management

Author: Hossein Mohammadi Rouzbahani,Hadis Karimipour,Lei Lei

Publication: Sustainable Energy Technologies and Assessments

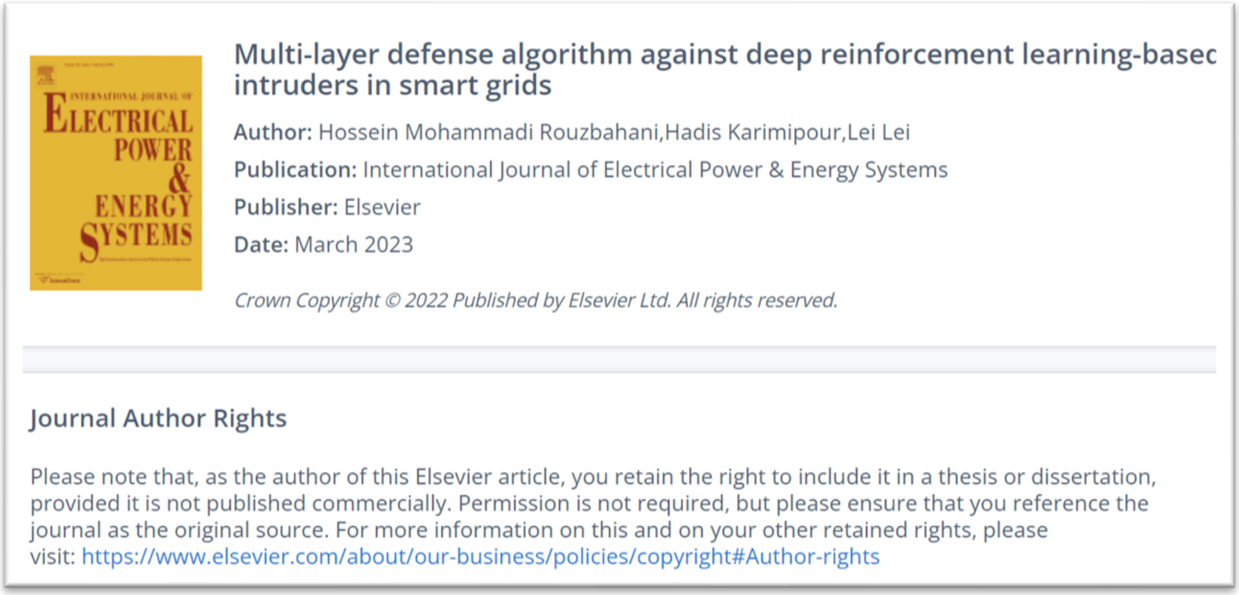
Publisher: Elsevier

Date: October 2021

© 2021 Elsevier Ltd. All rights reserved.

Journal Author Rights

Please note that, as the author of this Elsevier article, you retain the right to include it in a thesis or dissertation, provided it is not published commercially. Permission is not required, but please ensure that you reference the journal as the original source. For more information on this and on your other retained rights, please visit: https://www.elsevier.com/about/our-business/policies/copyright#Author-rights



Multi-layer defense algorithm against deep reinforcement learning-basec intruders in smart grids

Author: Hossein Mohammadi Rouzbahani,Hadis Karimipour,Lei Lei

Publication: International Journal of Electrical Power & Energy Systems

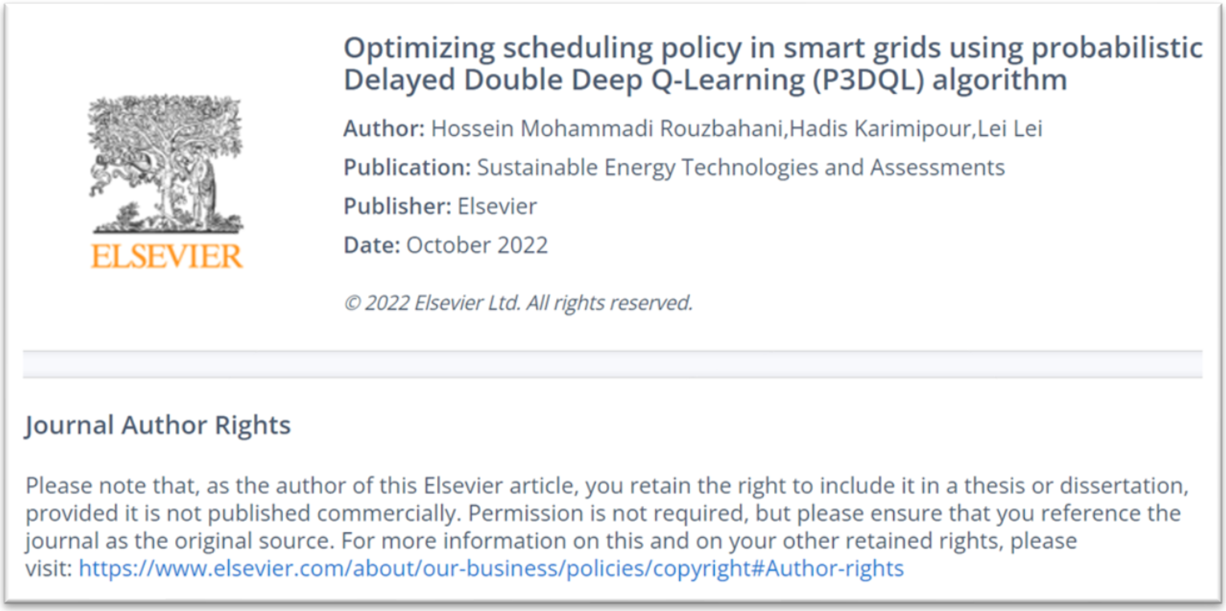
Publisher: Elsevier

Date: March 2023

Crown Copyright © 2022 Published by Elsevier Ltd. All rights reserved.

Journal Author Rights

Please note that, as the author of this Elsevier article, you retain the right to include it in a thesis or dissertation, provided it is not published commercially. Permission is not required, but please ensure that you reference the journal as the original source. For more information on this and on your other retained rights, please visit: https://www.elsevier.com/about/our-business/policies/copyright#Author-rights



Optimizing scheduling policy in smart grids using probabilistic Delayed Double Deep Q-Learning (P3DQL) algorithm

Author: Hossein Mohammadi Rouzbahani,Hadis Karimipour,Lei Lei

Publication: Sustainable Energy Technologies and Assessments

Publisher: Elsevier

Date: October 2022

© 2022 Elsevier Ltd. All rights reserved.

Journal Author Rights

Please note that, as the author of this Elsevier article, you retain the right to include it in a thesis or dissertation, provided it is not published commercially. Permission is not required, but please ensure that you reference the journal as the original source. For more information on this and on your other retained rights, please visit: https://www.elsevier.com/about/our-business/policies/copyright#Author-rights