Artificial Intelligence-Powered Energy Management of Reverse Osmosis Desalination Plants

by

Mohammad Amin Soleimanzade

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Mechanical Engineering
University of Alberta

Abstract

The rapid increase in global water and energy demand due to industrialization and population growth is a pressing challenge humankind faces today. Recent estimates indicate that due to population growth and reduction of water supplies, 40% of the global population is struggling with water scarcity, and a 20% increase is predicted for this number by 2025. Furthermore, The global energy demand expanded from 5000 million tons of oil equivalent in 1971 to 11700 million tons of oil equivalent in 2010, and it is predicted that it will increase up to 33% by 2030. The exponential increase in energy demand has exacerbated the greenhouse gas emissions as fossil fuels are mainly used to supply the required energy. The deployment of renewable energy sources, such as solar and wind power, to increase energy supply and diminish the adverse environmental effects of fossil fuels is considered an efficient solution to these problems. Among renewable energy sources, the exploitation of solar power generation has received significant attention and is considered one of the most promising options. However, the intermittent nature of photovoltaic (PV) power brings a huge challenge to PV-powered microgrids and desalination systems. Hence, it is essential to design advanced control techniques capable of coping with this challenge to optimize the performance of PV-driven desalination systems in terms of water production and energy consumption.

This thesis proposes two artificial intelligence-powered energy management systems for a hybrid grid-connected PV-reverse osmosis-pressure retarded osmosis desalination plant. In the first part of the thesis, an intelligent energy management system (IEMS) is developed to maximize the total water production and contaminant removal efficiency while keeping the grid's supplied power as low as possible. To promote the performance of the IEMS, the prediction of PV solar power is performed by three deep neural networks based on convolutional neural networks and long short-term memory neural networks. These networks are designed to perform 5-hour-ahead PV power forecasting, and the model with the smallest error is selected. The IEMS employs the particle swarm optimization (PSO) algorithm to find the optimum solutions of the system for each time step. Four performance indices are defined through which the IEMS is evaluated. The proposed technique results are compared with two benchmark methods, one of which is similar to the IEMS; however, it does not incorporate the PV power predictions. The superiority of the IEMS over the first benchmark is demonstrated by studying three scenarios: two successive sunny days, two successive cloudy days, and 10 days of operation. Moreover, the simulations are executed for different forecast horizons to investigate the effects of this parameter on the optimization results. The impacts of the best-found forecaster errors are also explored by repeating the simulations with the actual PV power data. Finally, the optimization is performed by two other stochastic algorithms: grey wolf optimizer (GWO) and genetic algorithm (GA). It is found that PSO outperforms GWO and GA for solving this optimization problem.

The second part of this thesis proposes a novel deep reinforcement learning-accelerated energy management system for the desalination plant mentioned above. The energy management problem is formulated as a partially observable Markov decision process, and the soft actor-critic (SAC) algorithm is employed as the core of the energy management system. We introduce 1-dimensional convolutional neural networks (1-D CNNs) to the actor, critic, and value function networks of the SAC algorithm to address the partial observability dilemma involved in PV-powered energy systems. The superiority of the proposed CNN-SAC model is verified by comparing its learning performance and simulation results with those of four state-of-the-art deep reinforcement learning algorithms: Deep deterministic policy gradient (DDPG), proximal policy optimization (PPO), twin delayed DDPG (TD3), and vanilla SAC. The results show that the CNN-SAC model outperforms the benchmark methods in terms of effective solar energy exploitation and power scheduling. By conducting ablation studies, the critical contribution of the introduced 1-D CNN is demonstrated, and we highlight the significance of providing historical PV data for substantial performance enhancement. The average and standard deviation of evaluation scores obtained during the last stages of training reveal that the 1-D CNN significantly improves the final performance and stability of the SAC algorithm.

Preface

This thesis is an original work by Mohammad Amin Soleimanzade and investigates the optimum energy management of photovoltaic-powered reverse osmosis desalination systems employing artificial intelligence algorithms.

Chapter 3 and parts of Chapter 1 and 2 of this thesis have been published as M. A. Soleimanzade and M. Sadrzadeh, “Deep learning-based energy management of a hybrid photovoltaic-reverse osmosis-pressure retarded osmosis system,” *Applied Energy*, vol. 293, p. 116959, 2021. I was responsible for the conceptualization, methodology, validation, investigation, and writing the original draft. M. Sadrzadeh was the supervisory author and was involved with concept formation and manuscript composition.

Chapter 4 and parts of Chapter 2 of this thesis are ready for submission to the *Applied Energy*, Title: “Novel data-driven energy management of a hybrid photovoltaic-reverse osmosis desalination system using deep reinforcement learning.”

Dedicated to

My lovely parents

For their sacrifices and support

Acknowledgments

I would like to extend my sincere gratitude to my supervisor, Dr. Mohtada Sadrzadeh, for his exceptional guidance, mentorship, and support. It was a priceless privilege to pursue my M.Sc. degree under his supervision. I am deeply indebted to him for everything he has offered me during these years.

I would like to thank my co-supervisor, Dr. Amit Kumar, for his valuable advice, comments, and suggestions.

I am extremely grateful to my family for their love and never-ending support in every step of my life. I would also like to thank my friends and lab mates, Pooria Karami, Sadegh Aghapour, Sadaf Noamani, Asad Asad, Masoud Rastgar, Behnam Khorshidi, Laleh Shamaeighahfarokhi, Ali Mohammadtabar, Muhammad Islam, for providing their knowledge and assistance.

Table of Contents

List of Tables

List of Figures

Abbreviations

AI	Artificial intelligence
CNN	Convolutional neural network
DDPG	Deep deterministic policy gradient
DDQN	Double deep Q-network
DNN	Deep neural network
ECP	External concentration polarization
ED	Electro-Dialysis
FH	Forecast horizon
GA	Genetic algorithm
GHG	Greenhouse gas
GWO	Grey wolf optimizer
ICP	Internal concentration polarization
IEMS	Intelligent energy management system
IMF	Intrinsic mode function
inRSE	Integral normalized mean square error
LSTM	Long short-term memory
MAE	Mean absolute error
MDP	Markov decision process
MSE	Mean squared error
PI	Performance index
POMDP	Partially observable Markov decision processes
PPO	Proximal policy optimization
PRO	Pressure retarded osmosis
PSO	Particle swarm optimization
PV	Photovoltaic
ReLU	Rectified linear unit
RMSE	Root mean square error
RNN	Recurrent neural network
RO	Reverse osmosis
SAC	Soft actor-critic
SOC	State of charge
TD3	Twin delayed deep deterministic policy gradient
VMD	Variational mode decomposition

1 Introduction

1.1 Motivation

While nearly 70% of Earth's surface is covered by water, only a small fraction of it is freshwater. The rapid industrialization, urbanization, and population growth have also led to an increase in freshwater consumption, which made the situation even more severe [1]–[3]. The global annual water demand is approximately 4600 billion m^3 [4], and despite the availability of natural water resources such as rainfall, 40% of the global population is facing water shortage, which will increase up to 60% by 2025 [5]. Consequently, to cope with the water scarcity dilemma, the only viable options are seawater desalination and treatment, recycle and reuse of wastewater. The water desalination processes can be classified into two primary types: membrane processes and thermal processes. In the membrane processes, such as electro-dialysis (ED) and reverse osmosis (RO), a permselective membrane is used to remove contaminants from water while in thermal processes, such as multi-effect distillation, vapor compression distillation, and multi-stage flash distillation, phase change plays the main role in clean water production [2]. Desalination processes consume 75.2 TWh annually, which is roughly 0.4% of global electricity usage. Moreover, the desalination techniques powered by fossil fuels produce 76 million tons of CO_2 per year, and it is estimated to reach 218 million tons per year by 2040 [6]. Hence, in addition to exhibiting appropriate performance in terms of water recovery and water quality, the desalination methods must be energy and cost-efficient in the long-run, while minimizing the CO_2 footprint.

Nowadays, RO dominates the water desalination market as it uses significantly lower energy than rival distillation processes [2], [7], [8]. The total specific energy consumption (considering both electrical and thermal energy utilization) of multi-stage flash distillation, multi-effect distillation, and thermal vapor compression are 19.58-27.25 kWh/m^3, 14.45-21.35 kWh/m^3, and 16.26 kWh/m^3, respectively, whereas RO consumes 5 kWh/m^3 on average for seawater desalination [9]. In this process, the driving force is the transmembrane pressure, provided by a high-pressure pump to overcome the osmotic pressure difference across the membrane [10]. RO is capable of treating a vast range of solute concentrations in water at a reasonable cost compared to other techniques. It can also provide higher water recovery rates than multi-effect and multi-stage flash distillation [11]. Despite these advantages, the RO process, similar to other desalination processes, suffers from environmental drawbacks such as brine discharge and greenhouse gas (GHG) emissions. While RO has lower emissions than thermal-based desalination methods [6], its emission is still roughly 1.8 times higher than a typical ED seawater desalination plant [12]. The reason lies behind the higher applied pressure in RO as compared to ED that leads to more GHG emissions due to increased power consumption. The carbon footprint for RO seawater desalination plants was reported to be in the range of 0.4-6.7 kg CO_2eq/m^3 [13]. The highest and lowest carbon footprints are associated with RO plants powering with fossil fuels (e.g., coal, oil, and natural gas) and renewable energies (e.g., wind and geothermal), respectively. Hence, it is of great significance to replace conventional fossil fuels with clean energy sources to reduce their impact on the environment.

The integration of photovoltaic (PV) systems with RO is one of the most common hybridization methods that is currently used when it comes to powering the RO systems by renewable energy sources [14]–[18]. The PV-RO systems are considered ideal hybrid systems for coastal areas that do not have access to grid electricity [19]. The intermittent nature of PV power, however, brings a huge challenge to PV-powered desalination systems, which makes the effective exploitation of solar power rather complex and severely impacts energy efficiency. The energy nexus between different domains such as water is crucial for enhancing the sustainability of different sectors and minimizing GHG emissions [20]. Energy and water are two indispensable elements of modern economics [21]. The provision of clean water and sanitation requires access to modern energy systems. To achieve the access-to-water-for-all goal, more energy will be needed to treat saline and brackish water, and a shift towards energy-intensive water projects is expected in the next 25 years [22]. Hence, these challenges and goals necessitate the design of advanced methods and algorithms to optimize the performance of PV-driven RO plants and effectively utilize solar energy.

1.2 Thesis Objectives

Microgrids can pave the way for the effective integration of renewable energy sources into the power grid, offering various advantages such as sustainability, flexibility, reliability, and improvement of efficiency [23], [24]. They can be regarded as small-scale and self-supporting networks that can be powered using on-site generation sources and operate either autonomously or in grid-connected mode [25]. In microgrids, energy management systems play a critical role in balancing power generation and consumption between distributed energy sources, loads, and energy storage devices to promote reliability and sustainability [26], [27]. As discussed in the previous section, the exploitation of solar energy for powering RO desalination systems has received significant attention. However, their intermittent nature poses a difficult challenge to the energy management problem of these systems and microgrids. Incorporating energy storage systems into PV-driven microgrids can mitigate the impacts of power fluctuations [28]. Despite that, the efficiency of microgrids comprised of energy storage devices is highly vulnerable to the effectiveness of the battery scheduling process performed by the energy management systems [29].

To cope with the challenges mentioned above, this thesis investigates the optimum energy management of a grid-connected PV-powered hybrid desalination plant comprised of RO and pressure retarded osmosis (PRO) using artificial intelligence (AI) algorithms. PRO is a green technology to harvest electricity from the salinity gradient between two water sources which has demonstrated less periodic behavior compared to other renewable energy sources [30], [31]. Also, the PRO system in the RO-PRO configuration can dilute the RO concentrate and reduce the environmental impacts of brine discharge. To solve the energy management problem of the hybrid PV-RO-PRO desalination system, two methods based on deep learning techniques and deep reinforcement learning algorithms are proposed. The primary goal of these models is to solve a multi-objective optimization problem consisting of three objectives: minimization of supplied power from the external grid, maximization of water production, and maximization of contaminant removal efficiency. The minimization of imported power from the main grid improves the hybrid desalination system's independent performance, and maximization of water production and contaminant removal efficiency ensures the high quality and amount of potable water produced using the RO system.

1.3 Thesis Outline

According to the goals and objectives of the thesis, the remainder of this dissertation is organized as follows. In Chapter 2, the essential preliminary concepts and methods regarding RO, PRO, deep learning, and reinforcement learning are provided. Also, a literature review on the energy management of water desalination systems is presented to point out the research gaps and shortcomings of previous studies. In Chapter 3, the first energy management system we propose based on deep learning techniques and metaheuristic optimization algorithms for the PV-RO-PRO system is described. The simulations results of this model are compared with different benchmark methods to evaluate its effectiveness. Chapter 4 details the deep reinforcement learning algorithm we develop to solve the control problem of the hybrid desalination system. The performance of this model is compared with state-of-the-art deep reinforcement learning algorithms to verify its superiority. Finally, in Chapter 5, key findings and results are highlighted to conclude the thesis and provide future research directions.

2 Background and Literature Review

2.1 Background

2.1.1 Reverse Osmosis

RO is a pressure-driven process that separates dissolved contaminants from water. The exerted pressure on the saline water must dominate the osmotic pressure so that water passes through the membrane to the permeate side. The RO membrane modules are divided into three types: hollow fiber, spiral-wound modules, and plate-and-frame [32]. Spiral-wound membrane modules are utilized in the present study as they provide a higher packing density and lower operational and capital costs compared to other modules [32]. For the mathematical formulation of RO spiral-wound modules, a model proposed by Sundaramoorthy et al. is used [33]. Sundaramoorthy et al. considered the spatial variations of pressure, mass transfer coefficient, flow rate, and solute concentration in the feed channel. More importantly, the severe effect of concentration polarization on permeation properties was not neglected in their study. The concentration polarization phenomenon occurs due to the accumulation of solute on the membrane surface, resulting in the concentration difference between the solution adjacent to the membrane surface and the bulk solution that reduces both water flux and solute rejection [33]. The proposed model is based on the solution-diffusion mechanism through which water and solute flux can be calculated as follows [33]:

$$J_w = A_w(\Delta P - \Delta\pi) \tag{2.1}$$

$$J_s = B\left(C_m - C_p\right) \tag{2.2}$$

where J_w is the water flux, J_s is the solute flux, A_w is the membrane permeability, B is the solute permeability, ΔP is the transmembrane pressure, $\Delta\pi$ is the osmotic pressure difference across the membrane, C_m is the solute concentration on the membrane surface at the feed side, and C_p is the solute concentration in the permeate. The osmotic pressure can be estimated through Van't Hoff equation:

$$\pi = iCRT \tag{2.3}$$

where i is the Van't Hoff factor, C is the concentration, R is the universal gas constant, and T is the temperature. Also, by writing the mass balance for the solute on a control volume surrounding the membrane thickness, the concentration polarization modulus is derived as follows [34]:

$$\frac{C_m - C_p}{C_f - C_p} = \exp\left(\frac{J_w}{k}\right) \tag{2.4}$$

where C_f is the bulk concentration of the feed solution and k is the mass transfer coefficient. Based on Equation 2.1, the permeate flux is a function of the pressure difference between the feed and permeate solutions, and thus, the change in the feed-side pressure along the membrane surface must be found to calculate the local permeate flux. For this purpose, Darcy's law is utilized to calculate the pressure loss inside the feed channel [33]:

$$\frac{dP_f(x)}{dx} = -\frac{\mu}{k_m A_c} Q = -fQ \tag{2.5}$$

where P_f is the pressure inside the feed channel, μ is the dynamic viscosity of the fluid, k_m is the permeability of the medium, A_c is the cross-sectional area of the channel, Q is the volumetric flow rate inside the feed channel, and f is the friction parameter of the feed channel, which is found experimentally. Obtaining the solute concentration in the permeate, the rejection percentage is calculated by the following equation:

$$R = \left(1 - \frac{C_p}{C_{f_i}}\right) \times 100 \tag{2.6}$$

where C_{f_i} is the solute concentration of the feed solution at the inlet.

The proposed model by Sundaramoorthy et al. [33] solves a system of equations to calculate flow rate and concentration of permeate as well as pressure, concentration, and flow rate of retentate, when the properties of the RO membrane (e.g., membrane dimensions, water, and solute permeability, and feed channel friction parameter), and flow rate, pressure, and concentration of feed solution at the inlet are given.

2.1.2 Pressure Retarded Osmosis

PRO is a membrane process capable of extracting the salinity gradient energy as one of the renewable energy sources [2]. The main purpose of PRO is to produce power through a salinity gradient between a high-concentration solution (draw solution) and a low-concentration solution (feed solution). To evaluate the performance of a PRO system in terms of energy generation, the power density is defined as follows:

$$PD = J_w \Delta P \tag{2.7}$$

where J_w is the water flux through the membrane, and ΔP is the transmembrane pressure.

Similar to RO, different modules exist for PRO that includes hollow fiber, flat sheet, and spiral-wound modules. The hollow fiber modules demonstrated a better performance regarding power density than the flat sheet and spiral-wound membranes [35]. These modules can be operated in inner- and outer-selective configurations [36]. In the inner-selective configuration, the draw solution is pumped into the lumen side and the feed solution is introduced into the shell, whereas, in the outer-selective one, the draw and feed solutions enter the shell and tube, respectively. The flow direction in these modules can be co-current or counter-current. The counter-current mode is reported to provide a better performance in terms of energy harvesting compared to the co-current one [37]. Given that, in this study, a PRO system equipped with hollow fiber modules with an inner-selective configuration that operates in the counter-current mode is considered.

Wan and Chung proposed a mathematical model for inner-selective PRO hollow fiber membranes [38]. The primary advantage of this model is that it considers the detrimental impacts of both internal concentration polarization (ICP) and reverse solute flux. The following equations are used to calculate water and solute fluxes in this model:

$$J_w = A\left(\mathit{\Delta}\pi_{effective} - \mathit{\Delta}P\right) \tag{2.8}$$

$$J_s = \frac{BM_W}{iRT}\left[\frac{J_w}{A} + \Delta P\right] \tag{2.9}$$

where A is the membrane permeability, $\Delta\pi_{effective}$ is the effective osmotic pressure difference, B is the salt permeability, and M_W is the molecular weight of the solute. The effective osmotic pressure difference is calculated as follows:

$$\mathit{\Delta}\pi_{effective} = \frac{\pi_d - \pi_f \exp\left(\frac{J_w S}{D_A}\right)}{1 + \frac{B}{J_w}\left[\exp\left(\frac{J_w S}{D_A}\right) - 1\right]} \tag{2.10}$$

where S is the structural parameter of the membrane, and π_d and π_f are the osmotic pressure of draw and feed solutions, respectively. Since the pressure drop inside the fibers is significant, the Hagen–Poiseuille equation is used to account for this [38]:

$$\Delta P_{lumen} = \frac{128\,\mu\,Q_d\,L}{\pi\,d_i^4} \tag{2.11}$$

where Q_d is the flow rate of the draw solution, and L and d_i are the length and inner diameter of fibers, respectively. It should be noted that pressure drop inside the shell is neglected. The performance of the PRO process can be predicted by solving the equations mentioned above through the finite element method to find the pressure and flow rate of the diluted draw solution when the pressure, flow rate, and concentration of the draw and feed solutions, as well as membrane properties, are given [38].

In addition to ICP, external concentration polarization (ECP) can affect the performance of PRO membranes. In this phenomenon, the permeation of water from the low-concentration side to the high-concentration side dilutes the draw solution close to the active layer of the PRO membrane, and consequently, reduces the water flux. To take the impact of ECP into account, the following equation is utilized for the calculation of the effective osmotic pressure [39]:

$$\Delta\pi_{effective} = \frac{\pi_d \exp(-\frac{J_w}{k}) - \pi_f \exp\left(\frac{J_w S}{D_A}\right)}{1 + \frac{B}{J_w}\left[\exp\left(\frac{J_w S}{D_A}\right) - \exp(-\frac{J_w}{k})\right]} \tag{2.12}$$

The mass transfer coefficient, k, in this equation can be calculated by the following empirical equation [37]:

$$Sh = 1.62\left(Re\,Sc\frac{d_i}{L}\right)^{0.33} \tag{2.13}$$

where Sh, Re, and Sc are the Sherwood number, Reynolds number, and Schmidt number of the lumen and L is the fiber length. **Figure 2.1** shows the flowchart of the PRO process simulation.

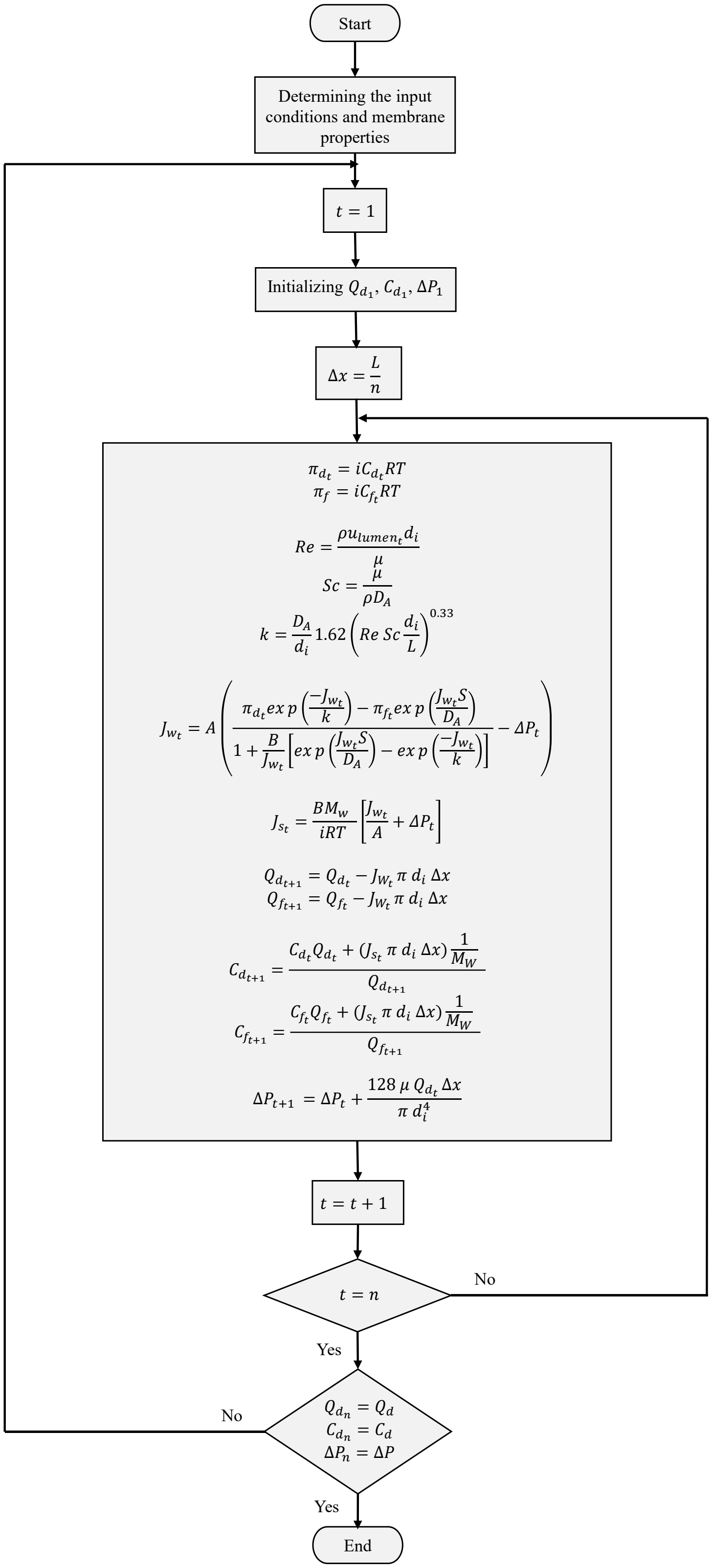
Start
Determining the input conditions and membrane properties
$t = 1$
Initializing Q_{d_1}, C_{d_1}, ΔP_1
$\Delta x = \frac{L}{n}$
$\pi_{d_t} = iC_{d_t}RT$
$\pi_f = iC_{f_t}RT$
$Re = \frac{\rho u_{lumen_t} d_i}{\mu}$
$Sc = \frac{\mu}{\rho D_A}$
$k = \frac{D_A}{d_i} 1.62 \left(Re\, Sc \frac{d_i}{L} \right)^{0.33}$
$J_{w_t} = A \left(\frac{\pi_{d_t} exp\left(\frac{-J_{w_t}}{k}\right) - \pi_{f_t} exp\left(\frac{J_{w_t} S}{D_A}\right)}{1 + \frac{B}{J_{w_t}} \left[exp\left(\frac{J_{w_t} S}{D_A}\right) - exp\left(\frac{-J_{w_t}}{k}\right) \right]} - \Delta P_t \right)$
$J_{s_t} = \frac{B M_W}{iRT} \left[\frac{J_{w_t}}{A} + \Delta P_t \right]$
$Q_{d_{t+1}} = Q_{d_t} - J_{W_t} \pi\, d_i\, \Delta x$
$Q_{f_{t+1}} = Q_{f_t} - J_{W_t} \pi\, d_i\, \Delta x$
$C_{d_{t+1}} = \frac{C_{d_t} Q_{d_t} + (J_{s_t} \pi\, d_i\, \Delta x) \frac{1}{M_W}}{Q_{d_{t+1}}}$
$C_{f_{t+1}} = \frac{C_{f_t} Q_{f_t} + (J_{s_t} \pi\, d_i\, \Delta x) \frac{1}{M_W}}{Q_{f_{t+1}}}$
$\Delta P_{t+1} = \Delta P_t + \frac{128\, \mu\, Q_{d_t} \Delta x}{\pi\, d_i^4}$
$t = t + 1$
$t = n$
No
Yes
$Q_{d_n} = Q_d$
$C_{d_n} = C_d$
$\Delta P_n = \Delta P$
No
Yes
End

Figure 2.1. Flowchart of simulating the PRO process

2.1.3 Battery Energy Storage System

The battery energy storage system enables the proposed energy management systems to efficiently schedule the power consumption or production of the hybrid PV-RO-PRO system units. Moreover, to improve the lifetime of this device, we restrict the storage of energy and do not allow the methods to charge the batteries beyond 80% of their capacity or reduce the energy level by less than 20%. This constraint is considered since operating at charge levels close to the maximum or minimum state of charge (SOC) leads to severe degradation of batteries [40]. The stored energy in the battery system at each time slot t can be calculated as follows [41]:

$$E_B^{(t)} = \begin{cases} E_B^{(t-1)} - \dfrac{1}{\eta_D} P_B^{(t-1)} \Delta t, & P_B^{(t-1)} > 0 \\ E_B^{(t-1)} - \eta_C P_B^{(t-1)} \Delta t, & P_B^{(t-1)} \leq 0 \end{cases} \tag{2.14}$$

where E_B is the stored energy in the battery system, η_D is the discharging efficiency, P_B is the power of the battery, Δt is the time interval (one hour in this study), and η_C is the charging efficiency. As it can be observed in Equation 2.14, positive values of P_B indicate that batteries are being discharged, and negative ones show that the battery system operates in the charging mode.

2.1.4 Variational Mode Decomposition

The variational mode decomposition (VMD) technique, which was proposed by Dragomiretskiy and Zosso in 2014, decomposes the original signal or time series into several modes. Indeed, it exhibits better robustness to the noise of the time series compared to other mode decomposition methods [42]. This model can artificially determine the number of modes and prevent issues regarding the mode aliasing problem [43]. By decomposing time series through the VMD technique, the signal is divided into k sub-signals. Each mode or sub-signal is formulated as follows:

$$u_k(t) = A_k(t) \cos \phi_k(t) \tag{2.15}$$

where $u_k(t)$ is the k-th mode, $A_k(t)$ is the amplitude function, and $\phi_k(t)$ is the phase function. To calculate modes and center frequencies ω, the following optimization problem needs to be solved [42]:

$$\min_{u_k, \omega_k} \sum_k \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] \exp(-j \omega_k t) \right\|_2^2 \tag{2.16}$$

$$\text{Subject to } \sum_k u_k(t) = f(t)$$

where f is the original signal, * denotes convolution, and δ is the Dirac distribution. Using these equations, k modes are obtained, k-1 of which are intrinsic mode functions (IMFs), and the last one is the residue.

2.1.5 Convolutional Neural Networks

Convolutional neural networks (CNNs) are one of the deep learning models that have been inspired by animals' visual cortex [44]. As the name of this model implies, they utilize the convolution mathematical operation, which makes them powerful tools to extract the spatial features of images [45]. CNNs are employed in various fields of research, such as object and text recognition [44]. The typical structure of CNNs is comprised of convolutional layers, pooling layers, and dense layers [45]. The input data of these models can be one-dimensional, such as one-dimensional time series, two-dimensional, such as images, and even n-dimensional. Furthermore, the input data can have one or multiple channels; for instance, black and white images have one channel, but colorful images consist of three channels. Various layers of CNNs are explained as follows.

In the convolutional layer, for 2-D CNNs, the input data (e.g., a 2-D tensor) is convolved with convolution kernels or filters, which extract local features and create the feature maps. Convolution kernels are 2-D matrices containing several weights that are found during the training process. The number of produced feature maps in each convolutional layer is equal to the number of that layer's kernels. The following equation is used to calculate the feature maps:

$$y_j = f\left(\sum_i x_i * W_j + b_j\right) \tag{2.17}$$

where y_j is the j$^{\text{th}}$ output feature map, x_i is the i$^{\text{th}}$ channel of the input data, W_j is the j$^{\text{th}}$ kernel weights, and b_j is the j$^{\text{th}}$ bias. Function f is the activation function of this layer (e.g., rectified linear unit (ReLU). The operation of the convolution is demonstrated in **Figure 2.2**. As can be observed, the input data is convolved with a 3×3 kernel producing a 4×4 feature map. To calculate a pixel of the feature map, the input data covered by the kernel are pointwise multiplied by the kernel's weights and summed subsequently.

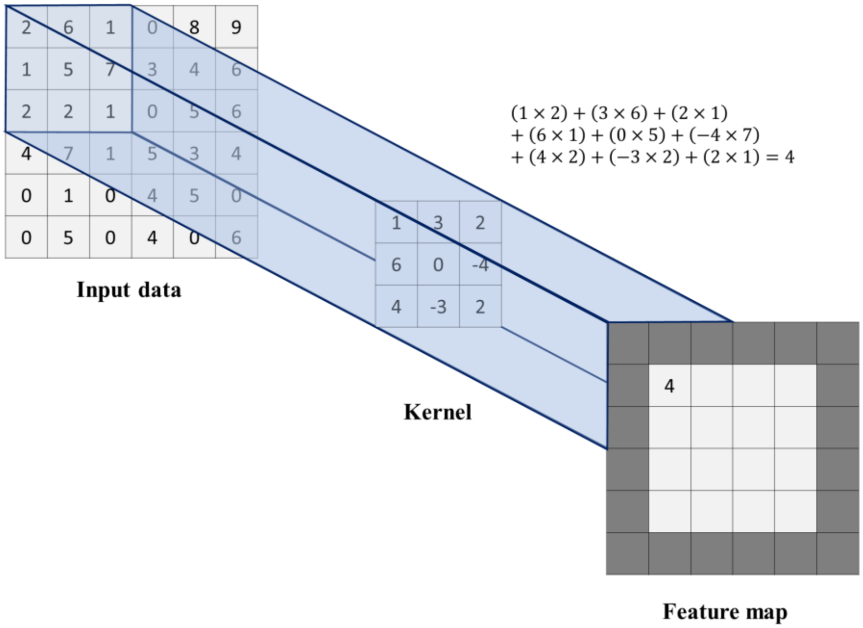
2 6 1 0 8 9
1 5 7 3 4 6
2 2 1 0 5 6
4 7 1 5 3 4
0 1 0 4 5 0
0 5 0 4 0 6
Input data
$(1 \times 2) + (3 \times 6) + (2 \times 1) + (6 \times 1) + (0 \times 5) + (-4 \times 7) + (4 \times 2) + (-3 \times 2) + (2 \times 1) = 4$
1 3 2
6 0 -4
4 -3 2
Kernel
4
Feature map

Figure 2.2. Convolution operation in CNNs (the bias was assumed to be zero)

In the pooling layer, the feature maps acquired from the previous convolutional layer are downsampled, which impedes over-fitting at the expense of losing a small portion of data. There are a variety of pooling methods, such as max-pooling, average-pooling, and mixed pooling [46]. For instance, in the average-pooling method, a 2-D block (e.g., 2×2) moves in the horizontal and vertical directions, calculates the average of the covered data points, and finally stores it in another tensor called pooled feature map. This operation is carried out for all feature maps calculated in the previous convolutional layer.

After the pooling layer, the pooled feature maps are converted into a one-dimensional output through a flatten layer and given to one or multiple fully connected or dense layers. In a dense layer, the inputs, or outputs of the previous dense layer, are connected to all neurons of that dense layer with a specific weight. In each neuron, the weighted sum of inputs, plus a bias, is calculated and is passed through an activation function (e.g., ReLU).

2.1.6 Long Short-Term Memory Neural Networks

Recurrent neural networks (RNNs) are powerful models for extracting temporal features and are capable of coping with sequential data. However, the major problem of traditional RNNs is gradient disappearance or gradient explosion, which makes the training process unfeasible [47]. To overcome this problem, long short-term memory (LSTM) neural network was proposed by

Hochreiter and Schmidhuber [48]. These networks are comprised of several LSTM cells, as shown in **Figure 2.3**. As can be observed, the inputs of each LSTM cell are the input vector X_t at timestep t, hidden layer output h_{t-1} at timestep $t-1$, and cell state C_{t-1} at the former timestep. Given the inputs, the input gate i_t, output gate O_t, and forget gate f_t can be found as follows [47]:

$$i_t \ = \sigma\left(W_i . [h_{t-1}, x_t] + \ b_i\right) \tag{2.18}$$

$$O_t \ = \sigma\left(W_o . [h_{t-1}, x_t] + \ b_o\right) \tag{2.19}$$

$$f_t \ = \sigma\left(W_f . [h_{t-1}, x_t] + \ b_f\right) \tag{2.20}$$

where W and b are the weights and bias vectors, and subscripts i, o, and f refer to input gate, output gate, and forget gate, respectively. To find the hidden layer output h_t and cell state C_t at timestep t, the temporary cell state $\widetilde{C_t}$ must be calculated:

$$\widetilde{C_t} \ = \tanh(W_c . [h_{t-1}, x_t] + \ b_c) \tag{2.21}$$

$$C_t \ = \ f_t \otimes \ C_{t-1} \ \oplus \ i_t \otimes \ \widetilde{C_t} \tag{2.22}$$

$$h_t \ = \ O_t \ \otimes \tanh(C_t) \tag{2.23}$$

where σ and $tanh$ denote the sigmoid and hyperbolic tangent functions, and $\oplus$ and $\otimes$ represent pointwise sum and multiplication, respectively.

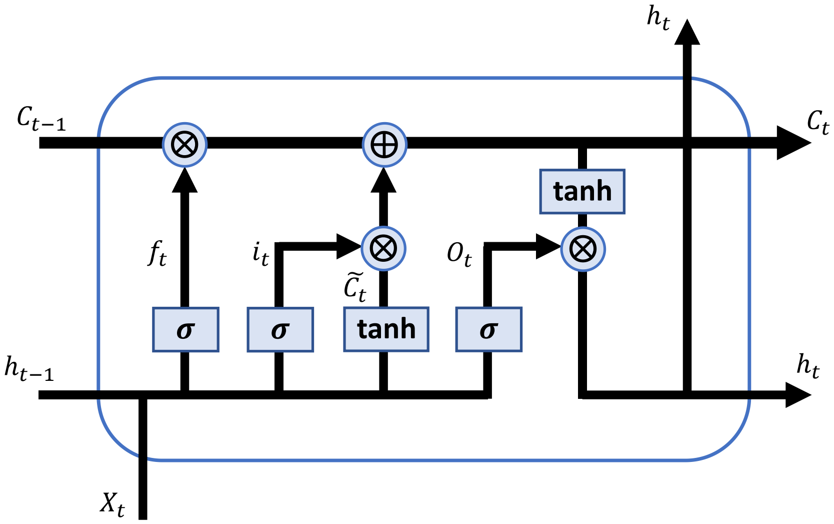

Figure 2.3. Structure of LSTM cells

2.1.7 Particle Swarm Optimization

The particle swarm optimization (PSO) algorithm [49] is a stochastic optimization approach that exhibits a robust and efficient performance despite its simple algorithm [50]. Moreover, a few parameters need to be adjusted in this algorithm, making it attractive among the population-based algorithms [51]. The PSO algorithm distributes several particles in a D-dimensional search space; thus, the position of each particle is, in fact, a D-dimensional vector. During each iteration, particles explore the search space to find the possible optimum solutions. These solutions are then evaluated by the cost function. The position of each particle is iteratively updated by its velocity vector, which depends on three vectors: the previous velocity vector, the distance between the particle current position and the personal best-found solution, and the distance between the particle current position and the best-found solution in the entire population. This technique allows particles to communicate with each other and to move towards the best solution found by the particle and the best-found solution within the population. The following equations are utilized to adjust the particles' position:

$$\vec{v}_i^{(t+1)} = \omega(t)\vec{v}_i^{(t)} + c_1\vec{r}_1^{(t)} \otimes \left(\vec{p}_i^{(t)} - \vec{x}_i^{(t)}\right) + c_2\vec{r}_2^{(t)} \otimes \left(\vec{G}^{(t)} - \vec{x}_i^{(t)}\right) \tag{2.24}$$

$$\vec{x}_i^{(t+1)} = \vec{x}_i^{(t)} + \vec{v}_i^{(t+1)} \tag{2.25}$$

where t is the iteration number, $\vec{v}_i$ is the i^{th} particle velocity vector, c_1 and c_2 are acceleration coefficients, $\vec{r}_1$ and $\vec{r}_2$ are D-dimensional vectors whose components are uniformly distributed numbers between 0 and 1, $\vec{x}_i$ is the i^{th} particle position, $\vec{p}_i$ is the best solution found by the i^{th} particle, $\vec{G}$ is the global best solution, and ω denotes the inertia weight.

2.1.8 Reinforcement Learning

Reinforcement learning is a subfield of machine learning that deals with solving the control problem of dynamically changing systems or environments. Reinforcement problems can be regarded as a discrete-time stochastic control process, called the Markov decision process (MDP), which is defined by the tuple $(\mathcal{S}, \mathcal{A}, \wp, r)$. $\mathcal{S}$ is the state space which is comprised of all possible states (s) that represent the information required for describing the environment. $\mathcal{A}$ is the action space containing all possible actions that the decision-maker, known as the agent in the context of reinforcement learning, can take to control the environment's decision variables [52]. $\wp$ is the state transition probability that models the uncertainties involved in the transition of the environment to the next state s_{t+1} given the current state s_t and action a_t determined by the agent [53]. Finally, r is the reward emitted by the environment on each transition, which conveys the objective of the control process to the agent. The general control framework of reinforcement learning is illustrated in **Figure 2.4**. At each timestep, the agent takes action according to the current state of the environment. The action is executed in the environment altering the system's state and moving the agent to the new state. The environment sends a reward signal to the agent based on the new state as a performance evaluation metric. The agent aims to discover an optimal policy to maximize the aggregated reward in every episode it encounters through interacting with the environment. Specifically, the agent learns a policy that maximizes the cumulative discounted reward called the return [54]:

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{2.26}$$

where G is the return, r is the obtained reward, and $\gamma \in [0, 1]$ is the discount factor that represents the relative importance of future rewards against the current ones. Small values of γ indicate that the agent should focus on the current rewards, while high values of γ (close to one) imply that the actions taken by the agent must result in the maximization of future rewards as well. This definition evinces the primary difference between reinforcement learning and conventional machine learning since reinforcement learning algorithms aim to maximize future rewards in addition to the immediate ones [40]. The performance of the agent during its interactions with the environment can be evaluated using two closely related value functions:

$$V_\pi(s) = \mathbb{E}[G(t) \mid s_t = s] = \mathbb{E}[r_{t+1} + \gamma \mathbb{E}[V_\pi(s')] \mid s_t = s] \tag{2.27}$$

$$Q_\pi(s, a) = \mathbb{E}[G(t) \mid s_t = s, a_t = a] = \mathbb{E}[r_{t+1} + \gamma \mathbb{E}[Q_\pi(s', a')] \mid s_t = s, a_t = a] \tag{2.28}$$

where π is the policy, V_π is the state-value function under policy π, $\mathbb{E}[.]$ is the expectation operator, s' is the next state of the environment, Q_π is the action-value function under policy π, a is the action taken at the current timestep, and a' is the corresponding action of the next state. The state-value function $V_\pi(s)$ represents the expected return of a given state, and the action-value function $Q_\pi(s, a)$ indicates the expected return of taking action a when the agent is in state s.

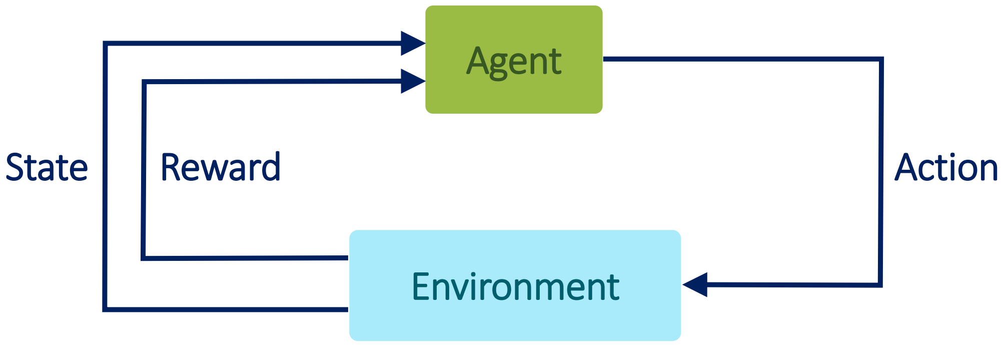

Figure 2.4. Interactions between the agent and environment

2.2 Literature Review

Although extensive research has been carried out on the modeling of the RO process and sizing of PV-powered RO systems, energy management of these systems received less attention. Ali et al. [55] proposed an energy management system based on fuzzy logic for an isolated battery-less PV-wind driven RO desalination unit. The genetic algorithm was applied on a hand-made fuzzy logic-based method to find the optimum parameters of the fuzzy inference system and maximize the water production. The optimized fuzzy inference system increased the freshwater production by 3.3% during the fall with respect to the proposed hand-made system. Xavier et al. [56] explored the energy management of a hybrid PV-wind powered RO desalination system modeled by a quasi-static model. The optimum dispatching strategy for determining the optimum value of shared power between different pumps was found by using a standard nonlinear programming method so that the required time for filling a superior water tank becomes minimum and also the pumps operate in an energy-efficient manner. Kyriakarakos et al. [57] developed a fuzzy logic-based energy management system for a microgrid consisting of PV array, wind turbine, proton exchange membrane fuel cell and electrolyzer, RO unit, battery system, and metal hydride tank. The proposed energy management system was used to find the optimum size of the microgrid. The design with the lowest operating costs and investments, as the optimum microgrid, was found through the particle swarm optimization algorithm. To evaluate the developed method, the same analysis was carried out by an ON-OFF strategy, and it was concluded that the fuzzy logic energy management system controls different components in a more effective manner than the ON-OFF approach due to the reduction in the size of microgrid units. In another study, Kyriakarakos et al. introduced a fuzzy cognitive map-based energy management system that could be operated in the variable load mode [58]. The designed method used the produced PV power, battery bank state of charge, and predicted PV power for the next step as the inputs. The final results demonstrated that water production improved by upgrading to the variable load operation mode. As stated earlier, the weather-dependent nature of renewable energy sources impacts the systems' performance; hence, developing an efficient energy management system requires a high-accuracy forecaster [41]. So far, PV power forecasting either was not executed or just was performed with a small forecast horizon, which is inadequate for optimum energy management. Moreover, none of the previous studies carried out research on the optimum power scheduling of hybrid desalination systems consisting of PV, RO, and PRO. Although PRO can provide the desalination system with a clean energy source, the integration of RO with PRO complicates the power allocation and optimization process. As a consequence, there is a lack of information about the forecaster-based energy management systems for PV-powered RO-PRO desalination units.

In Chapter 3 of the thesis, we propose an intelligent energy management system (IEMS) for a grid-connected PV-RO-PRO desalination system. PRO-powered systems are leveraged from a lower power intermittency level compared to other renewable energy sources such as wind power [31]. The IEMS incorporates the predicted PV power for the next five hours into the optimization process through which the optimum operating conditions of the hybrid system are obtained. To perform 5-hour-ahead solar power forecasting, three deep neural networks (DNNs) based on CNNs and LSTM networks are developed, and the model with the highest accuracy is selected. For the optimization of the system, the PSO algorithm is employed, and its performance is compared with the grey wolf optimizer (GWO) [59] and genetic algorithm (GA) [60]. In order to evaluate the IEMS, two benchmark methods are introduced, and their performance is compared based on four defined performance indices. To the best of our knowledge, the energy management of a hybrid PV-RO-PRO system by means of solar power predictors has not been investigated previously.

In addition to forecaster-based models, reinforcement learning algorithms have demonstrated enormous potential for energy management and operation optimization of energy systems [61], [62]. Model-free reinforcement learning algorithms can cope with uncertain parameters without requiring any prior knowledge or model for renewable power generation devices and have become an effective tool for the management of energy storage systems in microgrids [61], [63]. Moreover, after the training process, they can detect the optimum or near-optimum decisions within several milliseconds [63], making them a powerful asset for real-time control problems. The integration of deep neural networks into reinforcement learning algorithms has led to a new field of research, called deep reinforcement learning, which has demonstrated better performance than standard reinforcement learning in solving complex problems [61], [64]. Kofinas et al. [65] developed a multi-agent energy management system based on fuzzy Q-Learning techniques to control energy flows of a microgrid consisting of a PV system, fuel cell, diesel generator, desalination plant, and electrolyzer. The proposed energy management system aimed to minimize the utilization of the diesel generator and ensure the power balance between the microgrid units. The simulation results indicated that the power balance between consumption and production was almost stabilized to zero, and a low level of energy was not supplied. In a recent study, Zhang et al. [66] utilized the proximal policy optimization (PPO) algorithm to solve the energy management problem of a microgrid comprised of a wind turbine, PV system, battery storage system, RO plant, and diesel generator. The primary objective of the developed model was minimizing costs associated with operation, pollution, and battery storage systems. The performance of the PPO algorithm was compared with that of three baseline methods: stochastic programming, double deep Q-network (DDQN), and deep deterministic policy gradient (DDPG). The comparison results demonstrated that PPO outperformed the baseline algorithms and reduced the costs by up to 14.17%. Although these studies provide insight into the energy management of water desalination plants, none of them addressed the partial observability associated with uncertainties of PV power generation. In partially observable MDPs, the observations that the agent has access to do not provide adequate information about the environment, and advanced methods are needed to tackle this problem. Moreover, due to the uncertainties of PV power generation, the agent cannot observe the current output of the PV system [67], posing yet another challenge to the energy management problem that previously mentioned studies failed to take into account. Furthermore, the RO experimental data indicate that the flow rate and quality of the freshwater product are affected by the transmembrane pressure and the saline water flow rate [68]. However, in the previous papers, the desalination plants have been controlled only via their power consumption, which is not an accurate approach for modeling these systems.

In Chapter 4 of this dissertation, we solve the energy management problem of the PV-RO-PRO system by developing a deep reinforcement learning-accelerated energy management system based on the soft actor-critic (SAC) algorithm [69]. To cope with the partial observability dilemma (caused by the PV system) and address the shortcomings of previous studies, we formulate the problem as a partially observable MDP and provide the SAC algorithm with a history of PV data. To make the interpretation of PV power time series less challenging and promote the SAC algorithm performance, we introduce a 1-dimensional CNN (1-D CNN) to the actor, critic, and value networks of the SAC model. We call this novel algorithm, whose function approximators are modified to extract information favorable to value, critic, and actor networks, the CNN-SAC algorithm. It should be pointed out that no mathematical model is used for the output power of the PV system, and the CNN-SAC algorithm observes only the PV power time series. We carry out ablation studies to substantiate the partial observability of PV-driven systems and the significance of providing the algorithm with a history of previously encountered observations. The critical role of the introduced 1-D CNN in extracting essential information from PV power time series is also investigated. Since in most studies in the literature cutting-edge actor-critic methods such as twin delayed deep deterministic policy gradient (TD3) [70] and SAC were not applied [62], we benchmark the CNN-SAC algorithm against four state-of-the-art deep reinforcement learning models: DDPG [71], PPO [72], vanilla SAC, and TD3. The comparison between the CNN-SAC algorithm and benchmark methods is made by analyzing their learning performance and examining their simulation results in different case studies. Additionally, we compare the performance of the CNN-SAC algorithm with that of the IEMS model that we propose in Chapter 3. To the best of our knowledge, none of the earlier studies investigated the partial observabilities involved in PV-powered desalination systems and made the modification we applied in the proposed method to cope with this challenge.

3 Deep Learning-Based Energy Management of the PV-RO-PRO System

3.1 Methodology

The schematic diagram of the PV-RO-PRO system is depicted in **Figure 3.1**. As it can be observed, this system is comprised of four primary modules: PV system, energy storage system, RO module, and PRO module. RO is a technology by which high salinity water can be converted into fresh, potable water. In this system, a high-pressure pump increases the pressure of the saline water, called feed solution, to overcome the osmotic pressure difference that exists across the RO membrane. According to **Figure 3.2**, the RO plant consists of five parallel pressure vessels to increase the capacity of the desalination system, and each pressure vessel contains three RO membranes connected in series to improve the recovery. In addition to potable water, retentate flow is another output of RO, which contains all solutes blocked by the membrane. Disposal of this solution causes environmental issues, which is one of the downsides of desalination processes. To diminish the impacts of brine discharge, we consider a hybrid desalination system comprised of RO and PRO. As shown in **Figure 3.1**, we use the retentate of RO as the draw solution of PRO, which becomes diluted during the process and finally depressurized by a turbine. Hence, the RO-PRO configuration can be beneficial in terms of clean energy generation and dilution of RO retentate [73]. However, the generated power by PRO is not sufficient to supply the required energy of the high-pressure pump used in the RO module. As a result, we utilize a PV system along with PRO to provide the power consumed in the RO process. Similar to most renewable energy sources, solar energy suffers from a high level of power intermittency, which can severely impact the system's performance [74]. Moreover, solar energy is not always available during the day, and this makes the efficient exploitation of this energy source more difficult. To overcome these challenges, we consider a grid-connected PV-RO-PRO system along with a battery energy storage system. The hybrid system is connected to the main grid to import power whenever the output power of PV and PRO is not adequate for water production. The battery system has a critical role in the energy management of the system as it enables scheduling the power consumption or production of devices. Among the system's modules, RO and PRO are dispatchable units, meaning that their operation can be controlled via our proposed energy management system. On the other hand, the PV system is a non-dispatchable unit as its output power depends only on meteorological conditions and cannot be controlled. The IEMS aims to minimize the imported power from the external grid to improve the hybrid desalination system's independent performance while maximizing the water production and contaminant removal efficiency. This goal can be achieved by performing an optimized power scheduling for different components, i.e., main grid, battery system, RO module, and PRO module, at each timestep. Due to the fluctuations of solar power, a DNN-based forecaster is utilized in the IEMS to predict the available solar power.

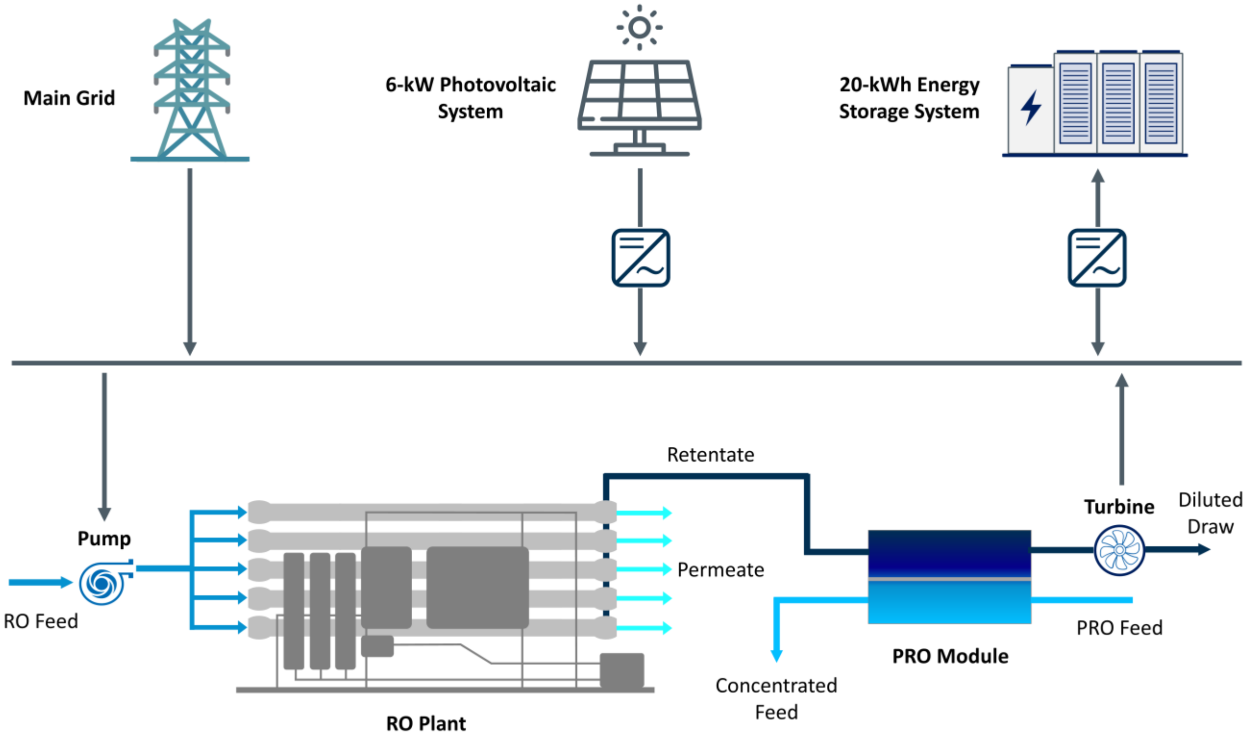

Figure 3.1. Schematic diagram of the PV-RO-PRO system

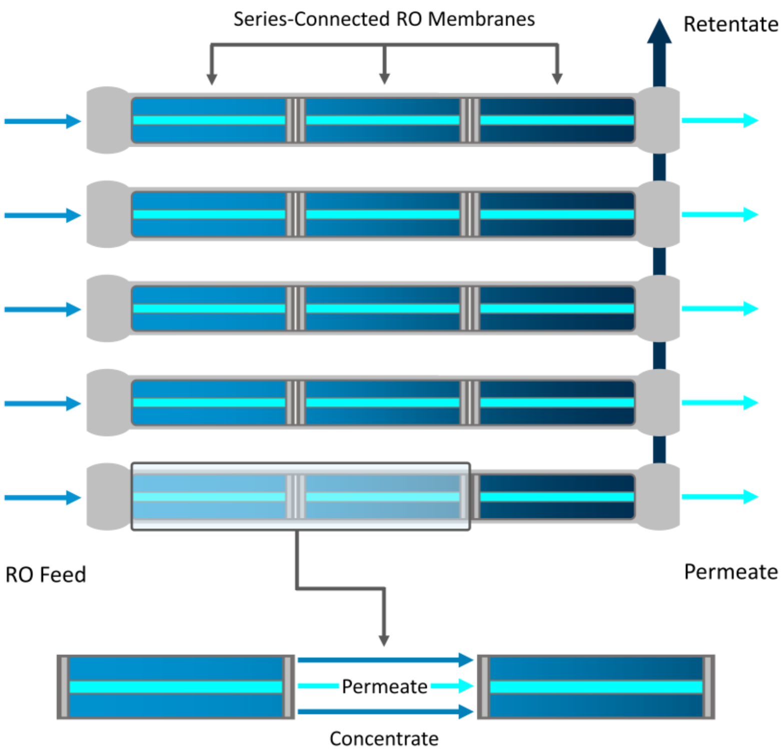

Figure 3.2. Arrangement of RO membranes. In each pressure vessel, three RO membranes are connected in series.

3.1.1 Solar Power Forecasting

To have an optimized energy management system for a PV-based process, it is of great importance to incorporate a forecasting model into the IEMS to cope with the variations of PV power. By doing so, energy can be stored in advance in the battery to be consumed later when the available solar power starts to reduce or fluctuate. In this study, we utilize deep neural networks to perform short-term PV power generation forecasting. These algorithms are trained by a set of past relevant data points to extract patterns and features of a given input sample (in this case, PV power time series) [46]. This chapter examines the performance of three deep learning forecasters for 5-hour-ahead solar power generation forecasting of a 6-kW PV system. The first model is a two-dimensional CNN that uses the PV power data as the training set. In the second model, the time series is decomposed by the VMD technique, and then it is given to a two-dimensional CNN for feature extraction. Lastly, a hybrid network consisting of the VMD technique, CNN, and LSTM layers is trained and analyzed.

The dataset of the PV power is taken from the Desert Knowledge Australia Solar Centre, Alice Springs, and is presented in **Table 3.1** [75]. From this dataset, the recorded PV power data from May 2011 to June 2016 was utilized for this study. It should be mentioned that this dataset has been created with a 5-minute resolution; however, in this study, for each hour (e.g., from 10:00 am to 10:55 am), the recorded PV power data points are summed and then divided by 12 to create a 1-hour-interval dataset [76]. Also, the PV power was not recorded from 6 pm until 7 am of the next day; hence, these values are assumed to be zero. In the following sections, the architecture of the proposed networks will be discussed.

Table 3.1. Characteristics of the PV system (Kaneka, 6.0kW, Amorphous Silicon, Fixed, 2008) [75]

Characteristic	
Array rating	6 kW
Panel rating	60 W
Number of panels	100
Array area	95.04 m^2
Array structure	Fixed: ground mount
Installation completed	11 November 2008
Array tilt/azimuth	Tilt = 20, Azi = 0 (Solar North)

3.1.1.1 Decomposition of PV Power Time Series Using VMD

By decomposing the PV power time series through the VMD technique, the signal is divided into k sub-signals that contain the seasonal and trend components of the PV power data. This can simplify the training process and convergence of neural networks [77]. To decompose the PV power time series, we utilize MATLAB, and the signal was decomposed into 5 IMFs and one residue (i.e., k = 6). The final results of the first 800 hours can be observed in **Figure 3.3**. In this figure, the original PV power time series can be seen at the top following by the decomposed components (i.e., IMF 1 through IMF 5 and the residue).

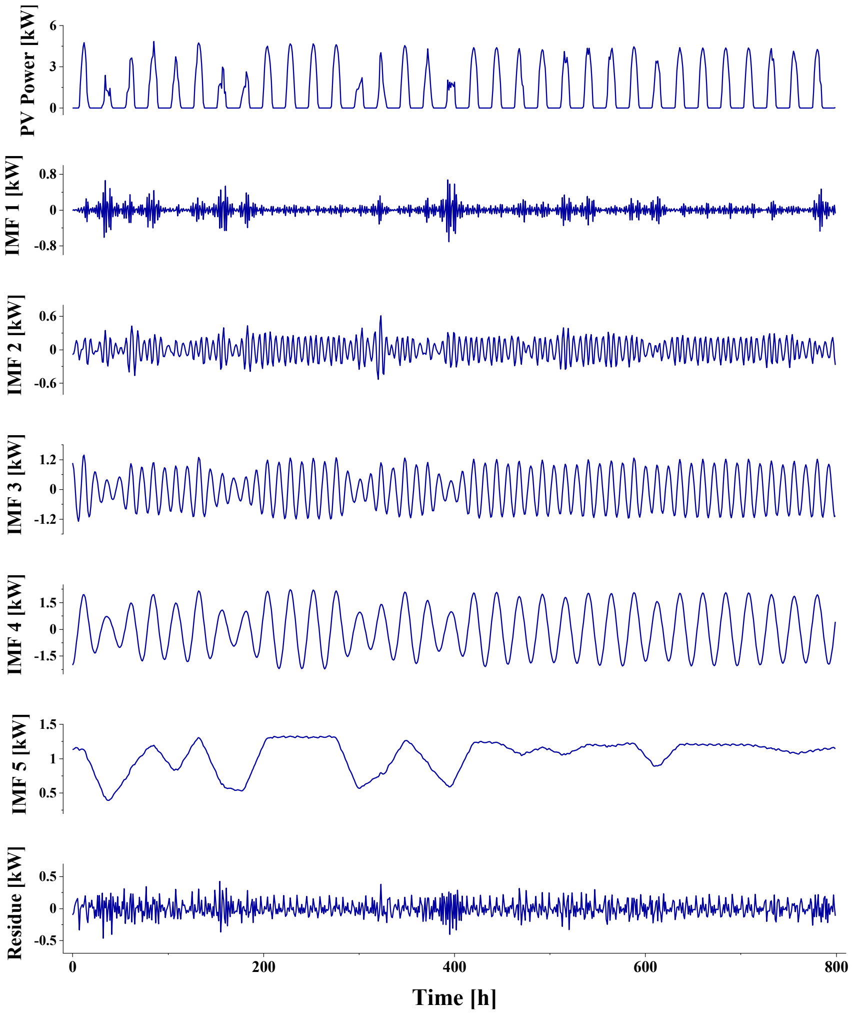

Figure 3.3. Decomposed components of the PV power time series

3.1.1.2 Data Processing

As mentioned before, in this study, three models for 5-hour-ahead solar power generation forecasting are explored: 2-D CNN, VMD-CNN, and VMD-CNN-LSTM. As a result, the primary components of the first two models are the CNN block, and in the case of the third model, the neural network is comprised of both CNN and LSTM blocks. The architecture of these models will be thoroughly analyzed in the next section. Each block requires the input data to have a particular format. Accordingly, the PV power time series must be reconstructed into other shapes to get the most out of each network.

A daily correlation exists in the obtained decomposed components by the VMD method, that may lead to improper performance of networks and inaccurate predictions. Hence, for the CNN block, we can construct two-dimensional input data with the decomposed components, and exploit CNNs to extract the spatial features [77]. The 2-D array construction for the CNN block is demonstrated in **Figure 3.4(a)**. In this process, at each timestep, we utilize $d \times h$ past data points, where d and h represent the number of days and hours (i.e., 24), respectively. Then, we reshape the one-dimensional vector into a d-by-h matrix. This procedure is followed for each mode (i.e., five IMFs and one residue). In this study, it is assumed that d is equal to 14.

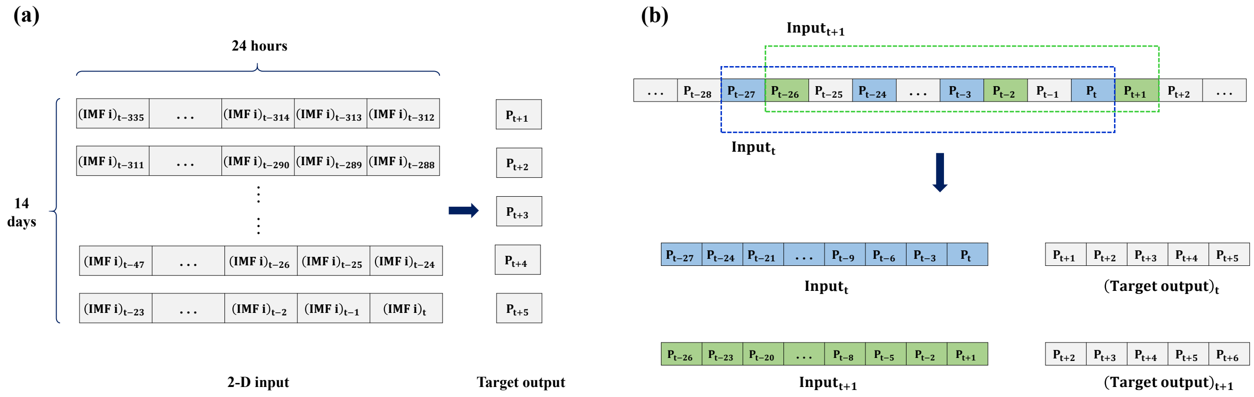
(a)
24 hours
14 days
(IMF i)t−335 ... (IMF i)t−314 (IMF i)t−313 (IMF i)t−312
(IMF i)t−311 ... (IMF i)t−290 (IMF i)t−289 (IMF i)t−288
(IMF i)t−47 ... (IMF i)t−26 (IMF i)t−25 (IMF i)t−24
(IMF i)t−23 ... (IMF i)t−2 (IMF i)t−1 (IMF i)t
Pt+1 Pt+2 Pt+3 Pt+4 Pt+5
2-D input
Target output
(b)
Inputt+1
... Pt−28 Pt−27 Pt−26 Pt−25 Pt−24 ... Pt−3 Pt−2 Pt−1 Pt Pt+1 Pt+2 ...
Inputt
Pt−27 Pt−24 Pt−21 ... Pt−9 Pt−6 Pt−3 Pt
Inputt
Pt+1 Pt+2 Pt+3 Pt+4 Pt+5
(Target output)t
Pt−26 Pt−23 Pt−20 ... Pt−8 Pt−5 Pt−2 Pt+1
Inputt+1
Pt+2 Pt+3 Pt+4 Pt+5 Pt+6
(Target output)t+1

Figure 3.4. (a) 2-D array construction for the CNN block, (b) Formation of delay embedding space

As will be discussed in the next section, the LSTM block's input is PV power data rather than the decomposed components. To construct the training sample for this network, the input time series is transformed into the delay embedding space. For this purpose, two parameters must be first determined: embedding dimension m and delay τ. Embedding dimension determines the number of previous samples that will be employed for forecasting, and delay determines the time difference between m selected samples. **Figure 3.4(b)** shows the transformation process into the delay embedding space for $m = 10$ and $\tau = 3$. As can be observed, a window with a defined size takes 10 samples with a time delay of 3 from the time series and forms the input vectors of the LSTM network. The crucial step in this process is the selection of embedding dimension and delay values, which can affect the performance of the neural network. For instance, small delays result in strongly correlated data points, while large delays lead to uncorrelated components of the sample vectors [78]. In order to select appropriate values for the embedding dimension and delay, TISEAN software was utilized. Based on the literature, for finding an appropriate value for the delay, the time-delayed mutual information should be calculated for different values of τ, and the delay at which the mutual information is at its first local minimum is selected [78]. For the embedding dimension, the false nearest neighbor method is used. In this method, the fraction of false neighbors is found for different values of m, and the value that results in a small fraction of false neighbors is a proper choice. The plots of the time-delayed mutual information and the fraction of false neighbors are provided in **Figure 3.5**. Based on this figure, $\tau = 6$ and values greater than 30 for the embedding dimension are suitable choices. It is assumed that the embedding dimension is equal to 57 so that the target outputs of the delay embedding space become consistent with those of the CNN constructed dataset. This is a vital matter for the VMD-CNN-LSTM network. As well, data normalization is used for data pre-processing so that all data points lie in the $[0, 1]$ range.

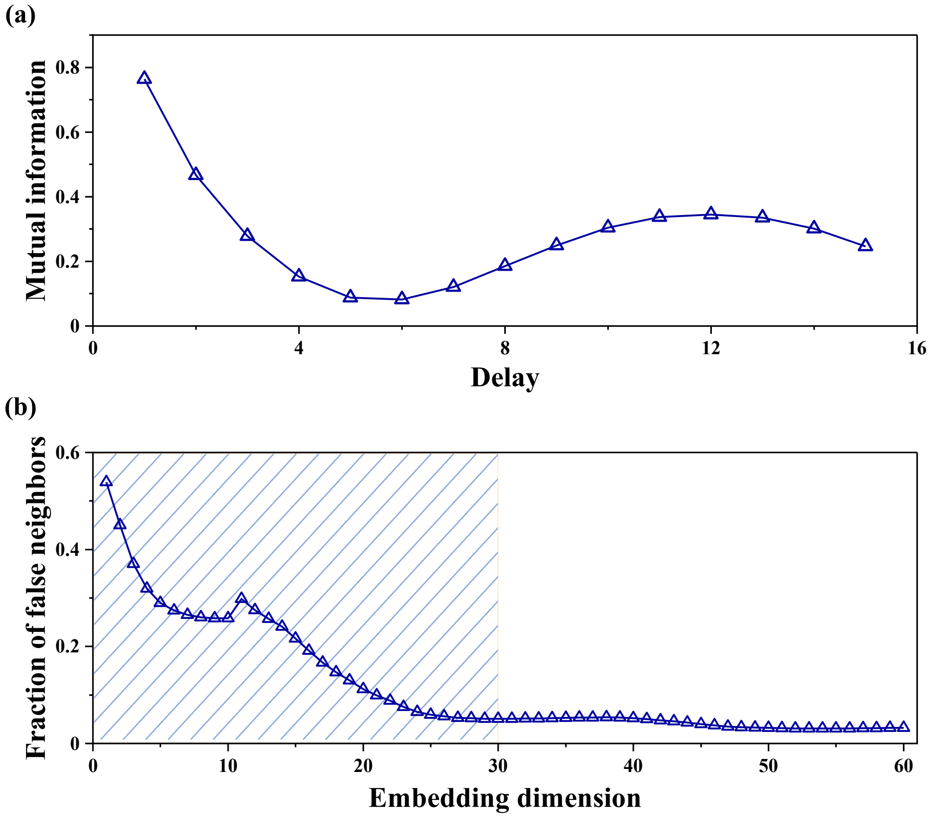

Figure 3.5. (a) Mutual information, (b) Fraction of false neighbors. The hatched region in (b) demonstrates the interval in which the fraction of false neighbors is not small enough.

3.1.1.3 Architecture of Forecasters

In this paper, three DNN-based models are trained, and the best candidate for solar power generation forecasting and, consequently, the IEMS is selected.

The first one is a 2-D CNN, having two successive convolutional layers followed by a flatten layer. The architecture of the first network is demonstrated in **Figure 3.6(a)**. In this model, pooling layers are not utilized after the convolutional layers since they omit some of the information extracted by the convolutional layers and, thus, decrease the model accuracy [77]. The activation function of convolutional layers is the ReLU function, and padding is set to valid. After the flatten layer, two dense layers with the ReLU activation function are placed followed by a 5-neuron dense layer with the linear activation function. For the last dense layer, 5 neurons are considered as the forecast horizon is equal to 5. The ADAM algorithm is used as the optimizer, with the mean squared error (MSE) loss function to be minimized. The input data of this model is the PV power time series, which is reconstructed via the 2-D array construction method mentioned in the prior subsection.

The architecture of the second model (**Figure 3.6(b)**) is similar to the previous one, but instead of the PV power time series, the decomposed components obtained by the VMD method (i.e., 5 IMFs and one residue) are used as the input data. Hence, in this model, the input sample has 6 channels. By training this hybrid VMD-CNN model, the effectiveness of the VMD method for the decomposition of the PV power time series can be evaluated.

The final model is a hybrid VMD-CNN-LSTM neural network that can take advantage of CNN block for spatial feature extraction and LSTM block for temporal feature extraction. **Figure 3.6(c)** illustrates the architecture of the hybrid VMD-CNN-LSTM network. In this model, the decomposed components of the PV power time series are given to a CNN with two successive convolutional layers. Meanwhile, the delay embedding space samples (created with the PV power time series) are provided to a neural network with three LSTM layers. After each LSTM layer, a dropout layer is placed to randomly set the input units to zero and prevent overfitting. The outputs of the CNN and LSTM layers are concatenated and given to another neural network with four dense layers. After each first two dense layers, a dropout layer is placed as well. Similar to the previous models, the ADAM algorithm and MSE loss function are used for the training of this model.

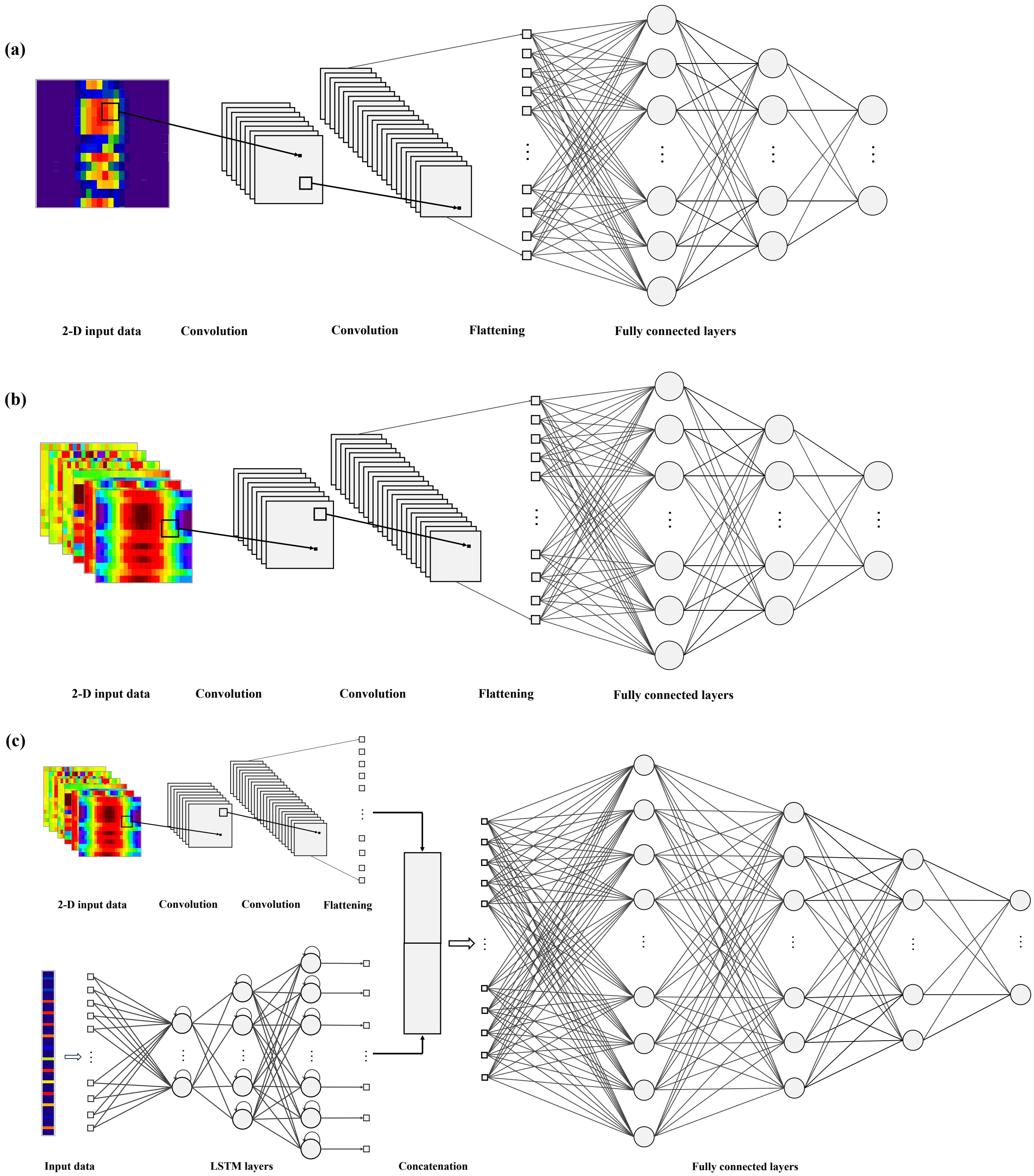

Figure 3.6. (a) 2-D CNN model, (b) VMD-CNN model, (c) VMD-CNN-LSTM model. The heatmaps of the input data are demonstrated in this figure in order to provide a better insight into the input data.

From the dataset, the recorded data from May 2011 to May 2015 is used for the training, and the rest of the dataset (roughly one year) is used for the test of the neural networks. It is worth mentioning that the grid search method is used to tune the hyperparameter of these models. It means that for each combination of hyperparameter values, the neural network is trained, and the model with the highest accuracy is selected. In order to do that, 80% of the training set is utilized for the training of each model, and the remaining 20% is reserved for the validation.

3.1.2 RO Mathematical Model

As mentioned in Chapter 2, the mathematical model proposed by Sundaramoorthy et al. [33] is used in this study to simulate the performance of the RO desalination plant. In another study, Sundaramoorthy et al. [79] conducted RO experiments using commercial thin-film composite (TFC) polyamide membranes (Ion Exchange, India) and chlorophenol as the solute. They obtained an empirical equation for the mass transfer coefficient of the chlorophenol. This equation is a function of permeate flux, velocity of the flow inside the feed channel, and solute concentration. The proposed model is validated using already-published experimental data by Sundaramoorthy et al. [79]. **Figure 3.7(a)** and **(b)** compare the modeling results and experimental data at a constant feed flow rate of 2.583×10^{-4} m^3/s, and solute concentration of 1.556 mol/m^3 and 2.335 mol/m^3, respectively. As can be observed, the mathematical model results are in good agreement with experimental data for different values of feed pressure and solute concentration.

It should be pointed out that the mass transfer coefficient obtained by Sundaramoorthy et al. [79] is specifically for chlorophenol, and it is not valid for other solutes, such as NaCl in the present study. For NaCl, the following equation, reported for spiral-wound modules, is utilized to calculate the mass transfer coefficient [80]:

$$Sh = 0.648\, Re^{0.379}\, Sc^{0.33} \tag{3.1}$$

where Sh is the Sherwood number ($\frac{k\, d_h}{D_A}$), Re is the Reynolds number ($\frac{\rho\, u_F\, d_h}{\mu}$), and Sc is the Schmidt number ($\frac{\mu}{\rho\, D_A}$). In these dimensionless numbers, d_h is the hydraulic diameter of the feed channel, D_A is the solute diffusivity, ρ is the water density, and u_F is the velocity of water inside the feed channel.

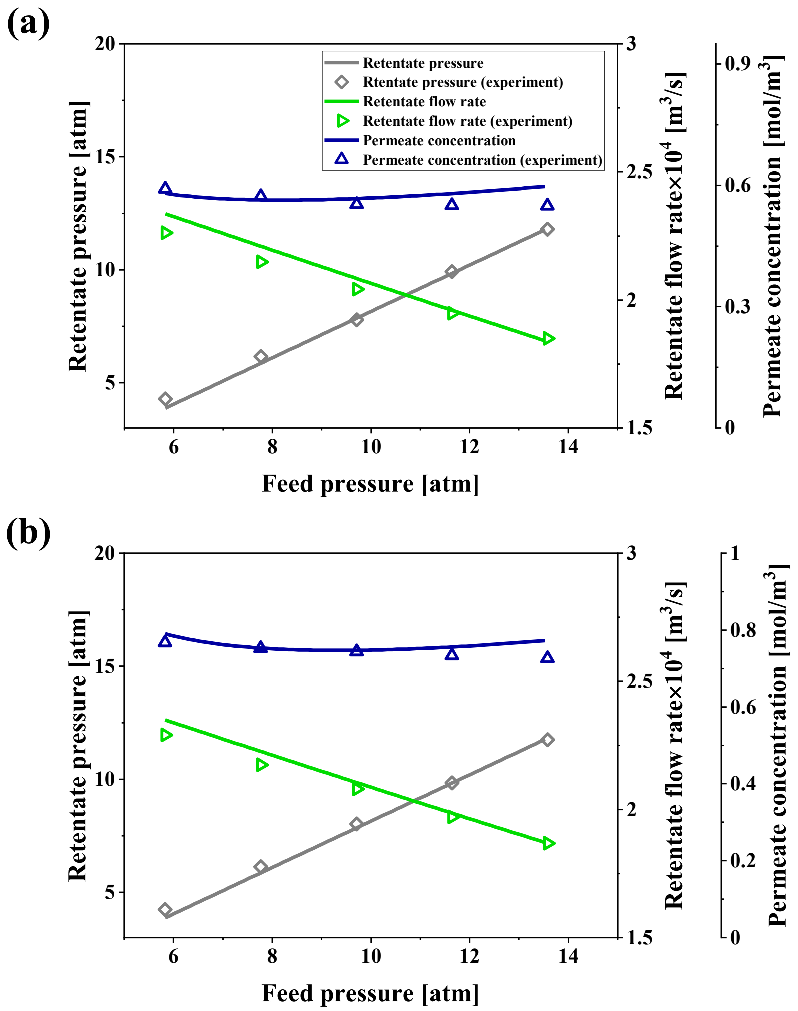

Figure 3.7. Validation of RO mathematical model with the experimental data from [79]. In (a), the flow rate and solute concentration of the feed solution is equal to 2.583×10^{-4} m^3/s and 1.556 mol/m^3, respectively, while in (b), the feed flow rate is again 2.583×10^{-4} m^3/s, and the feed solute concentration is equal to 2.335 mol/m^3.

The experimental data published by Sundaramoorthy et al. [79] contains feed pressures between 5.83 and 13.58 atm. In seawater RO, the feed pressure is higher than 14 atm; therefore, it crucial to validate the RO model in higher feed pressures since, in this study, simulations are conducted for a feed solution with seawater quality. Senthilmurugan et al. [68] provided experimental data for a seawater RO system with a Film Tech spiral-wound module (2.5” FT30). The characteristics of the RO membrane are shown in **Table 3.2**.

Table 3.2. Characteristics of the 2.5" FT30 membrane [68]

Membrane characteristics	
Membrane permeability	$4.5 \times 10^{-12} \frac{m}{Pa.s}$
Solute permeability	$3.6 \times 10^{-8} \frac{m}{s}$
Membrane length	$0.854\ m$
Membrane width	$1.10\ m$
Hight of feed channel	$7.1 \times 10^{-4}\ m$
Feed channel friction parameter	$2.5008 \times 10^{8} \frac{1}{m^2}$

The validation results of the RO model for two values of feed concentration (25 and 35 kg/m^3) and feed pressures between 50 and 80 bar are provided in **Table 3.3**. The comparison between the model and experimental data is based on the permeate flow rate and concentration values. According to **Table 3.3**, the maximum error between the model results and experimental data is 12.29%; hence, it can be concluded that the mathematical model is also valid for high values of feed pressure and concentration.

Table 3.3. Validation of the RO model with the experimental data from [68]. In this table, P_f is the feed pressure, Q_f is the feed flow rate, C_f is the feed concentration, Q_p is the permeate flow rate, and C_p is the permeate concentration. The temperature at which the experimental data was collected is 25 ˚C.

$P_f\ (bar)$	$Q_f \times 10^6\ (\frac{m^3}{s})$	$C_f\ (\frac{kg}{m^3})$	$Q_p \times 10^6\ (\frac{m^3}{s})$		Error (%)	$C_p \times 10^3\ (\frac{kg}{m^3})$		Error (%)
			Experiment	Model		Experiment	Model	
55	223.32	25	22.82	21.36	6.27	95	106.68	-12.29
60	225.8	25	25.3	24.53	3.04	89	97.28	-9.30
70	231.05	25	30.75	30.71	0.13	82	84.78	-3.39
80	235.43	25	34.95	36.73	-5.09	72	77.18	-7.19
50	213.88	35	13.38	11.83	11.58	248	235.54	5.02
55	216.68	35	16.19	14.77	8.77	207	196.81	4.92
60	218.82	35	18.32	17.69	3.44	179	171.48	4.20
70	223.32	35	22.82	23.42	-2.63	141	140.6	0.28
80	227.27	35	26.77	28.99	-8.29	129	122.97	4.67

3.1.3 PRO Mathematical Model

For the validation of the PRO mathematical model, the experimental data of two already-published studies are used. In the first study (**Figure 3.8(a)**), the experiment was conducted with 3 PRO hollow fiber membrane with a surface area of 2.71 cm^2. Also, feed and draw solutions of 0.011 M and 0.81 M NaCl are utilized, and the flow rate of the feed and draw solutions are set to 0.2 L/min [38]. In the second study (**Figure 3.8(b)**), the membrane surface area is 14.43 cm^2. Moreover, 1 M salt solution and deionized (DI) water are used as the draw and feed solutions, and the flow rate of both solutions is kept constant at 0.1 L/min [81]. As can be observed in these figures, the PRO mathematical model results align well with the experimental data for different hydraulic pressure difference values.

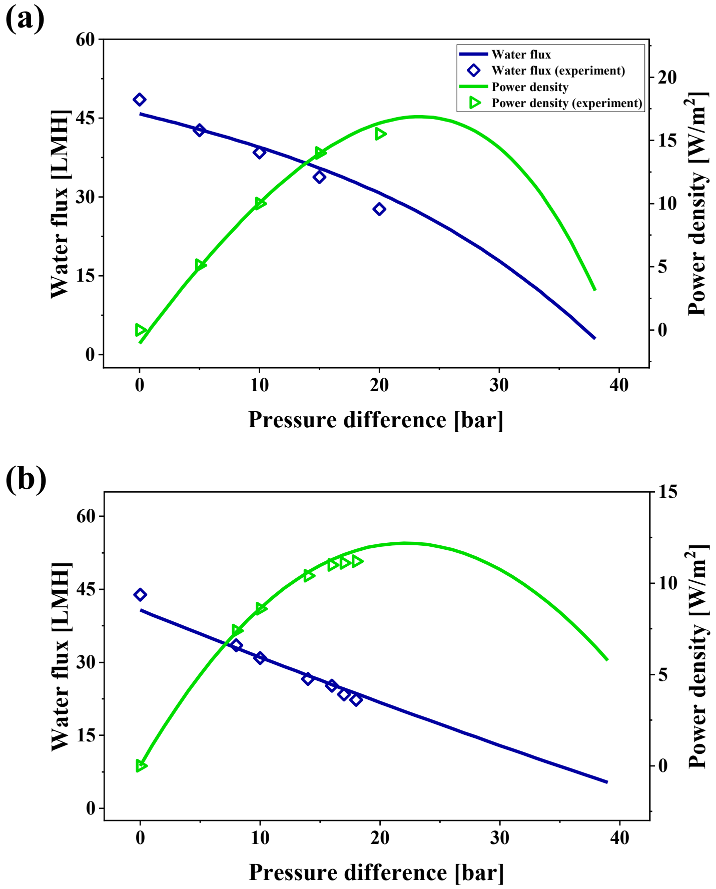

Figure 3.8. Validation of PRO mathematical model. In (a), the PRO model results are compared with the reported experimental data in [38]. The flowrate of the draw and feed solutions are kept constant at 0.2 L/min, while the solute concentration of the draw and feed solutions are equal to 0.81 M and 0.011 M NaCl, respectively. In (b), the PRO model is validated with the experimental data from [81]. In this experiment, DI water with a flow rate of 0.1 L/min is used as the feed solution, while the flow rate and solute concentration of the draw solution are set to 0.1 L/min and 1 M NaCl, respectively.

3.1.4 Intelligent Energy Management System (IEMS)

The performance of the hybrid PV-RO-PRO process is highly dependent on operating conditions. Therefore, it is critically important to control different parameters so that the hybrid system operates in an optimized manner. There are four decision variables whose optimum values need to be achieved at each timestep: main grid power, battery power, and RO system feed flow rate and pressure. By employing the IEMS in this study, the aim is to minimize the main grid's imported power while maximizing the water production rate and salt rejection percentage or improving the water quality. Taking a closer look at the objectives of the IEMS, it is found that a multi-objective optimization problem should be formulated and solved as the objectives are in conflict with each other. For instance, the higher the water production rate, the higher the amount of main grid power.

To overcome this challenge, the cost function is formulated through the global criterion method that is based on the relevant L_p metrics. This method makes the cost function a measure of closeness to an ideal condition [82]. The cost function in this method is defined as follows:

$$L_p = \left(\sum_{i=1}^{k} \left| \frac{O_i(\vec{\chi}) - O_i^*}{O_i^*} \right|^p \right)^{\frac{1}{p}} \tag{3.2}$$

where k is the total number of objectives, $O_i(\vec{\chi})$ is the value of i^{th} objective function, $\vec{\chi}$ is the decision variable vector, and O_i^* is the ideal value of the i^{th} objective function. p is assumed to be 2 in this study. In order to formulate the optimization problem in the aforementioned form, two objectives are considered: (1) the ratio of main grid power to water production rate and (2) the permeate solute concentration. The minimization of the first objective function increases water production and decreases grid power. Minimizing the second objective function leads to higher solute rejection or better water quality. The ideal values for the first and second objective functions are 0.1 kWh/m^3 and 3.6×10^{-3} kmol/m^3, respectively. To obtain the ideal value of the first objective function, the optimization is carried out for different values of this parameter, and the value with the better performance is selected. Also, the ideal value of the second objective function is chosen based on the specifications of the RO membrane. Hence, the cost function CF that must be minimized at each timestep is as follows:

$$CF = \left(\left| \frac{O_1 - 0.1}{0.1} \right|^2 + \left| \frac{C_P - 3.6 \times 10^{-3}}{3.6 \times 10^{-3}} \right|^2 \right)^{\frac{1}{2}} \tag{3.3}$$

where O_1 is the ratio of the grid power to the permeate flow rate. Moreover, during the optimization process, the optimization algorithm must consider the constraints of the problem, such as equality and inequality constraints. The cost function is subjected to one equality constraint in the present work, which is the power balance constraint, and five bound constraints for the decision variables and battery SOC. The following equations represent the constraints of the optimization problem:

$$P_{Grid} = P_{RO} - P_B - P_{PV} - P_{PRO} \tag{3.4}$$

$$0 \leq P_{Grid} \tag{3.5}$$

$$P_B^{min} \leq P_B \leq P_B^{max} \tag{3.6}$$

$$Pr_{RO}^{min} \leq Pr_{RO} \leq Pr_{RO}^{max} \tag{3.7}$$

$$Q_{f,RO}^{min} \leq Q_{f,RO} \leq Q_{f,RO}^{max} \tag{3.8}$$

$$0.2 \leq SOC \leq 0.8 \tag{3.9}$$

where P_{Grid} is the main grid power, P_{RO} is the power consumption of the RO system, P_{PV} is the available solar power, P_{PRO} is the generated power by the PRO system, and Pr_{RO} and $Q_{f,RO}$ are the feed pressure and flow rate in the RO system. The consumed power by the RO system and produced power in the PRO system can be calculated via the following equations:

$$P_{RO} = \frac{Q_{f,RO}\,(Pr_{RO} - P_0)}{\eta_{pump}\,\eta_{motor}} \tag{3.10}$$

$$P_{PRO} = Q_{d,out}\left(Pr_{d,out} - P_0\right)\eta_{turbine}\,\eta_{gen} \tag{3.11}$$

where P_0 is the atmospheric pressure, $Q_{d,out}$ is the flow rate of the diluted draw solution, $Pr_{d,out}$ is the pressure of the diluted draw solution, and η_{pump}, η_{motor}, $\eta_{turbine}$, and η_{gen} represent the efficiency of the pump, electrical motor, turbine, and generator, respectively. Also, it is assumed that the consumed power by the low-pressure pump on the feed side is negligible.

The IEMS incorporates the prediction results obtained from the best DNN into the optimization process to enhance the performance of the PV-RO-PRO system. As mentioned in previous sections, the proposed DNNs perform 5-hour-ahead PV power forecasting. Hence, the optimization algorithm must utilize the prediction results to determine the optimum values of decision variables at each timestep. For this purpose, at each timestep, the performance of the hybrid system is simulated from the present time up to 5 hours later by using the present actual value and predicted values of the PV power. To elaborate further on this process, suppose we are at timestep t, and we wish to find the optimum values of the decision variables (i.e., $P_{Grid}^{(t)}$, $P_B^{(t)}$, $Pr_{RO}^{(t)}$, and $Q_{f,RO}^{(t)}$) at this time. The determined operating conditions at time t influence the state of the next timesteps; therefore, if we intend to optimize the system performance, the optimal values of the decision variables must be found based on the current state and possible future states. To account for possible future states, the optimization algorithm simulates the system performance for six hours of operation by considering the actual PV power $P_{PV}^{(t)}$ and predicted values $\hat{P}_{PV}^{(t+1)}$ to $\hat{P}_{PV}^{(t+5)}$ to optimize the system so that the cost function of this six-hour operation becomes minimum. Since the optimization is performed for six hours, the decision variable vector $\vec{\chi}^{(t)}$ is comprised of 24 parameters:

$$\vec{\chi}^{(t)} = \left[\vec{X}^{(t)}, \vec{\hat{X}}^{(t+1)}, \vec{\hat{X}}^{(t+2)}, \dots, \vec{\hat{X}}^{(t+5)}\right] \tag{3.12}$$

where:

$$\vec{X}^{(t)} = \left[P_{Grid}^{(t)}, P_B^{(t)}, Pr_{RO}^{(t)}, Q_{f,RO}^{(t)}\right] \tag{3.13}$$

$$\vec{\hat{X}}^{(t+i)} = \left[\hat{P}_{Grid}^{(t+i)}, \hat{P}_B^{(t+i)}, \hat{Pr}_{RO}^{(t+i)}, \hat{Q}_{f,RO}^{(t+i)}\right], \quad i = 1, \dots, 5. \tag{3.14}$$

Once the optimization is completed, the first four values of the solution are selected as the optimum values of the decision variables of timestep t . Then the optimization of the next timestep will begin. It is worth noting that P_{RO} and P_{PRO} in Equation 3.4 are related to Pr_{RO}, and $Q_{f,RO}$, and therefore P_{Grid} is a function of P_B, Pr_{RO}, and $Q_{f,RO}$. Given that, in order to decrease the number of decision variables from 24 to 18 and simplify the optimization problem, P_{Grid} is eliminated from the decision variables and Equation 3.4 is used to calculate the main grid power.

The PSO algorithm is used to solve the optimization problem mentioned above [49]. In the present work, it is assumed that the inertia weight decreases linearly from 0.9 in the first iteration to 0.2 in the last iteration. The PSO algorithm randomly places all particles in the search space in the first iteration and updates their positions throughout the iteration. However, in this study, this procedure was followed only in the first timestep (i.e., placing all particles randomly in the first iteration). After the first timestep, the obtained solution in each timestep is saved and is utilized as the initial position of 30% of the particles in the next timestep. In other words, in the optimization of all timesteps, except for the first one, 30% of particles will not be distributed randomly and are placed in the previous timestep solution. By doing so, the optimization algorithm can be guided so that smaller values for the number of particles and iterations are considered. In this study, the acceleration coefficients are assumed to be 2. Also, the number of particles and iterations for the first timestep is set to 700 and 600, respectively, while for the next timesteps, the same number of particles and iterations equal to 350 is considered.

3.2 Results and Discussion

The IEMS was employed to optimize the performance of the hybrid PV-RO-PRO system. The specifications of each module are provided in **Table 3.4**. The optimization and simulation of the hybrid system are implemented in MATLAB, and deep learning models are implemented using Keras version 2.4.3 on a personal computer with an NVIDIA GeForce RTX 2080 Ti GPU and 32 GB RAM.

3.2.1 Results of Deep Learning Models

As mentioned earlier, the prediction of the available solar power is crucial for the optimization of the hybrid process. In the present study, different models (2-D CNN, VMD-CNN, and VMD-CNN-LSTM) are used to forecast solar power generation. Before using the prediction results in the simulation of the PV-RO-PRO system, it is essential to evaluate and compare the performance of each model. In order to do that, the results of the three proposed models are compared in different seasons, i.e., fall, winter, spring, and summer. **Table 3.5** presents the architecture and values of the hyperparameters of each model.

Table 3.4. Specifications of the battery system, RO module, PRO module, and optimization constrains

Module	Parameter	
Battery system	η_C	0.95
	η_D	0.95
	Initial SOC	30%
	$C_{Battery}$	$20\ kWh$
RO system	Feed spacer thickness [79]	$0.8\ mm$
	Channel length [79]	$0.934\ m$
	Membrane width [79]	$8.4\ m$
	b [79]	$8529.45\ \frac{atm.s}{m^4}$
	A_w [19]	$4.17 \times 10^{-7} \frac{m}{atm.s}$
	B [19]	$2.9 \times 10^{-8} \frac{m}{s}$
	C_{f_i}	$32 \frac{g}{L}$
	η_{pump}	0.8
	η_{motor}	0.98
PRO system	d_i [38]	$575\ \mu m$
	L	$1.5\ m$
	A [38]	$3.5\ \frac{LMH}{bar}$
	B [38]	$0.32\ LMH$
	S [38]	$450\ \mu m$
	Feed flow rate	$15 \frac{L}{min}$
	Feed concentration	$0.011\ \frac{mol}{L}$
	Number of fibers	100
	$\eta_{turbine}$	0.8
	η_{gen}	0.98
Optimization	P_B^{min}	$-5\ kW$
	P_B^{max}	$5\ kW$
	Pr_{RO}^{min}	$35\ atm$
	Pr_{RO}^{max}	$70\ atm$
	$Q_{f,RO}^{min}$	$15\ \frac{L}{min}$
	$Q_{f,RO}^{max}$	$300\ \frac{L}{min}$

Table 3.5. The architecture of deep learning models. The activation function of all convolutional and dense layers (except for the last dense layer) is ReLU.

Model	Layer	Specifications	Learning rate	Epoch	Batch size
2-D CNN	Convolution 2D	$(7,7) \times 128$	0.005	40	128
	Convolution 2D	$(7,7) \times 256$			
	Flatten				
	Dense	64 neurons			
	Dense	32 neurons			
	Dense	5 neurons			
VMD-CNN	Convolution 2D	$(7,7) \times 128$	0.0005	80	64
	Convolution 2D	$(7,7) \times 256$			
	Flatten				
	Dense	80 neurons			
	Dense	40 neurons			
	Dense	5 neurons			
VMD-CNN-LSTM	Convolution 2D	$(3,3) \times 256$		40	64
	Convolution 2D	$(3,3) \times 512$			
	Flatten				
	LSTM	128 cells, Dropout: 0.2			
	LSTM	256 cells, Dropout: 0.2			
	LSTM	384 cells, Dropout: 0.2			
	Dense	512 neurons, Dropout: 0.1			
	Dense	256 neurons, Dropout: 0.1			
	Dense	128 neurons			
	Dense	5 neurons			

In the field of time series forecasting, there are a variety of error metrics that can be used to assess the performance of models. In this study, three evaluation indices are used: mean absolute error (MAE), root mean square error (RMSE), and integral normalized mean square error (inRSE). The mathematical equation of these error functions are as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \tag{3.15}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|^2} \tag{3.16}$$

$$inRSE = \sqrt{\frac{\sum_{i=1}^{N}|\hat{y}_i - y_i|^2}{\sum_{i=1}^{N}|y_i - \bar{y}|^2}} \tag{3.17}$$

where y_i, $\hat{y}_i$, and $\bar{y}$ are the i^{th} measured value, i^{th} predicted value, and the average of measured values (i.e., the normalized PV powers in the forecast horizon).

The results of the 2-D CNN, VMD-CNN, and VMD-CNN-LSTM models for 5-hour-ahead forecasting and five consecutive days in different seasons are illustrated in **Figure 3.9**. The larger the forecast horizon value, the lower the accuracy of predictions. The output power of the PV system typically has different trends and peak values in different seasons due to the variation of weather conditions such as temperature and solar irradiation. Hence, it is essential to assess the models' outcomes in all seasons to ensure the predicted solar power is close to the actual value in all weather conditions. It can be observed that the 2-D CNN model cannot accurately predict the actual values on overcast days, and even on sunny days, it does not provide accurate results. On the other hand, VMD-CNN and VMD-CNN-LSTM models' predictions are close to the actual values on overcast and sunny days. For example, as shown in the second panel of **Figure 3.9**, the fluctuations of PV power on the second day have been perfectly predicted by these two models. However, to quantitatively compare the predictive ability of these models, the error metrics must be calculated.

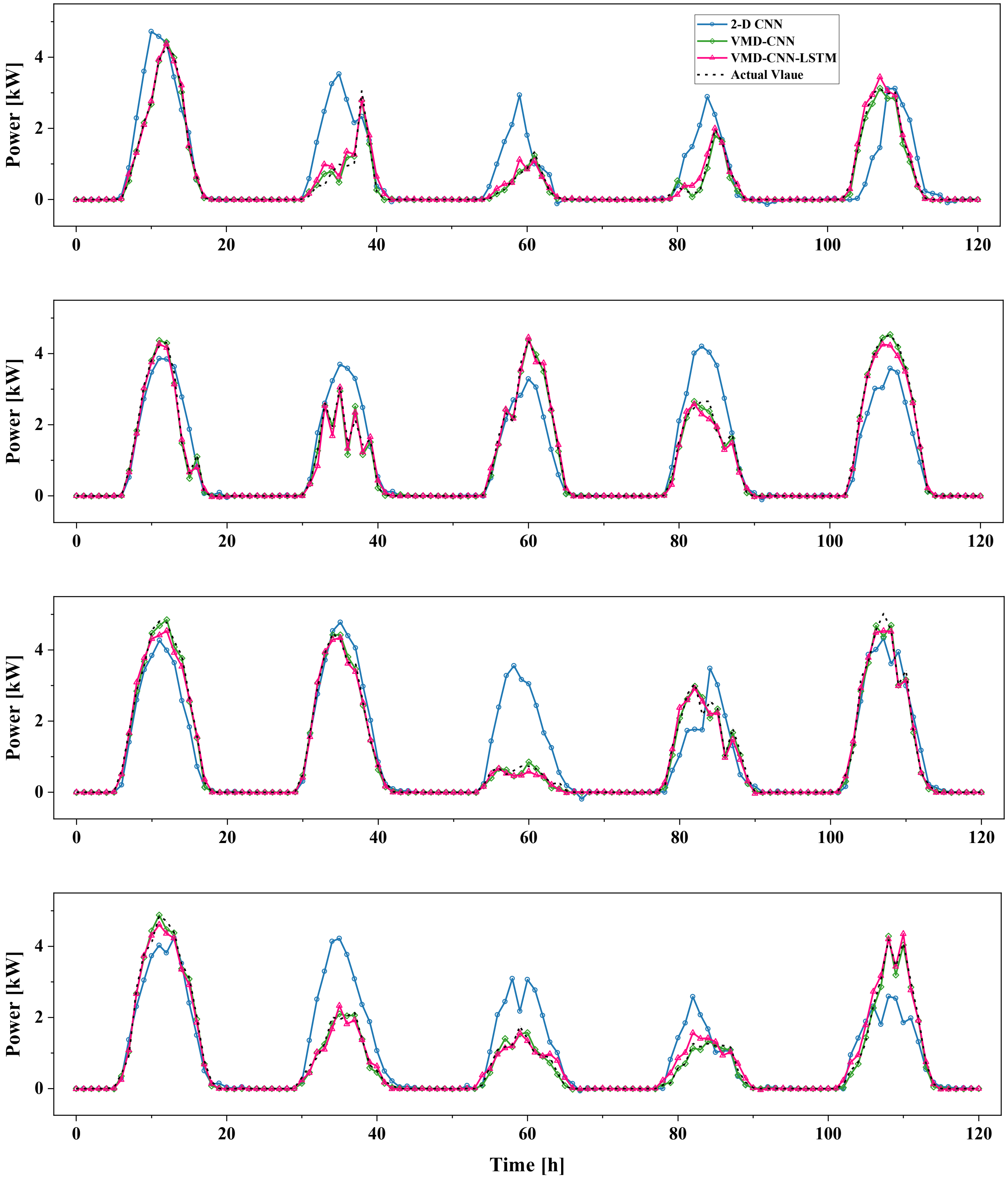

Figure 3.9. 5-hour-ahead forecasting results of PV power: (a) Spring, (b) Summer, (c) Fall, (d) Winter

The obtained values for the error metrics are presented in **Table 3.6**. This table contains the error values of each model for all seasons as well as the overall error. According to this table, in most cases, the obtained errors in summer are smaller than those of the other seasons. The worst accuracy is achieved in winter due to more fluctuations of power and variation of weather conditions. For instance, the MAE, RMSE, and inRSE of the VMD-CNN model for winter are increased 68%, 67.8%, and 44.5%, respectively, compared to summer. Also, it can be realized that the VMD-CNN model outperforms other models in all seasons as it provides much smaller errors than those of 2-D CNN and VMD-CNN-LSTM. Hence, this model's overall error is also smaller than the other models as evident in **Table 3.6**. The overall MAE, RMSE, and inRSE of the VMD-CNN network are 6.05, 6.32, and 6.32 times smaller than those of 2-D CNN, and 2.04, 1.95, and 1.95 times smaller than those of VMD-CNN-LSTM. Accordingly, the predictions of the VMD-CNN model are utilized for the optimization of the PV-RO-PRO system.

Table 3.6. Performance of the deep learning models

Season	Model	MAE	RMSE	inRSE
Spring	2-D CNN	0.03773	0.07199	0.24367
	VMD-CNN	0.00583	0.01082	0.03661
	VMD-CNN-LSTM	0.01262	0.02235	0.07566
Summer	2-D CNN	0.02881	0.06061	0.22904
	VMD-CNN	0.00467	0.00863	0.03261
	VMD-CNN-LSTM	0.00722	0.01294	0.04891
Fall	2-D CNN	0.03025	0.05601	0.17833
	VMD-CNN	0.00548	0.00968	0.03081
	VMD-CNN-LSTM	0.01295	0.02205	0.07022
Winter	2-D CNN	0.04646	0.08701	0.28309
	VMD-CNN	0.00785	0.01448	0.04712
	VMD-CNN-LSTM	0.01612	0.02748	0.08942
Overall	2-D CNN	0.03591	0.07019	0.23811
	VMD-CNN	0.00593	0.01111	0.03767
	VMD-CNN-LSTM	0.01207	0.02164	0.07339

3.2.2 Results of IEMS

In this section, the obtained results from the PSO algorithm are reviewed and analyzed. It is important to define appropriate performance indices (PIs) by which the overall performance of the hybrid system can be assessed in terms of energy efficiency, water production, and final concentration. In order to take all these response variables into account, four PIs are defined as follows:

$$PI_1 = \frac{\int_0^T P_{Grid} dt}{\int_0^T Q_P dt} \tag{3.18}$$

$$PI_2 = \frac{\int_0^T (P_{RO} - P_{PRO}) dt}{\int_0^T Q_P dt} \tag{3.19}$$

$$PI_3 = \frac{\int_0^T |P_B| dt}{\int_0^T P_{RO} dt} \tag{3.20}$$

$$PI_4 = 1 - \frac{1}{C_F} \frac{\int_0^T (Q_P C_P) dt}{\int_0^T Q_P dt} \tag{3.21}$$

where T is the time of the simulation and Q_p is the permeate flow rate. The first PI shows the amount of power that has been purchased from the main grid for producing one cubic meter of water. The lower the PI_1, the more independent the system is from the main grid. The second PI indicates the efficiency of the RO-PRO system in terms of energy consumption as it measures the amount of consumed energy in the RO-PRO system to produce one cubic meter of water. PI_3 provides information regarding battery utilization during the simulation. Finally, PI_4 shows the total solute rejection or permeate water quality.

In order to validate the effectiveness of the IEMS, two benchmark methods are introduced. The first benchmark method is similar to the proposed method but it optimizes the system at each timestep regardless of the next hours' conditions. In other words, it does not incorporate the prediction results into the PSO algorithm. Contrary to the first benchmark, the second benchmark is not an energy management system. In this case, the PI values of the IEMS and first benchmark are compared to their best values that can be achieved. For instance, the optimum values of PI_1, PI_2, and PI_4 are 0, 2.2303 kWh/m^3, and 0.993, respectively. It should be mentioned that these three values cannot be achieved at the same time as the operating conditions to obtain the optimum value for each of them are different.

The results of the proposed method for 120 hours of operation are presented in **Figure 3.10** and **Figure 3.11**. **Figure 3.10(a)** shows the power of each component determined by the optimization algorithm. As can be observed, when the produced power by the PV system starts to increase, the PSO algorithm decides to consume more power in the RO system and reduce the grid power as the required power for water production can be supplied by the PV system. The power of batteries determines the amount of energy that must be charged or discharged, and their level of charge is shown in **Figure 3.10(b)**. During the day, the battery power is negative, which shows that a portion of the PV, PRO, and grid power is dedicated to the energy storage system. This is evident in **Figure 3.10(b)** as the battery SOC increases during the day. The storage of energy during the day is the result of the exploitation of the solar power generation forecasting algorithm. By predicting the available solar power, the PSO algorithm realizes that the output power of the PV system is zero during the night. Hence, to keep the main grid power as low as possible, energy must be stored in the batteries in advance to be utilized later. It is apparent in **Figure 3.10(a)** that the stored energy during the day is used during the night, and until several hours after the PV power becomes zero, the supplied power from the main grid does not increase. This performance concurs that the IEMS is capable of planning for the future and performing power scheduling more intelligently. The operating conditions of the RO system (i.e., feed flow rate and pressure) are shown in **Figure 3.11**. As can be seen, the feed flow rate and transmembrane pressure increase during the hours in which the PV power is high since the available power for the RO system increases. The higher pressure and feed flow rate result in higher solute rejection and permeate flow rate, as shown in **Figure 3.11**. The same analysis was done for the first benchmark method, and the results are depicted in **Figure 3.12** and **Figure 3.13**. When the output power of the PV system is large, the main grid power is small, and all generated power by the PV and PRO system is utilized by the RO system for clean water production. As shown in **Figure 3.13**, the feed flow rate and transmembrane pressure increase during the day. However, based on **Figure 3.12(a)**, in this method, except for the first hour, the power of the battery is zero, and the energy storage system is not utilized during the operation. As a consequence, during the hours in which the PV power is zero, the lack of energy cannot be supplied by the energy storage system, and the main grid power must increase. Also, since the increase of the main grid power increases the cost function, the PSO algorithm does not keep the feed flow rate and pressure as high as high-PV-power hours. As a result, the water production rate and rejection percentage reduce.

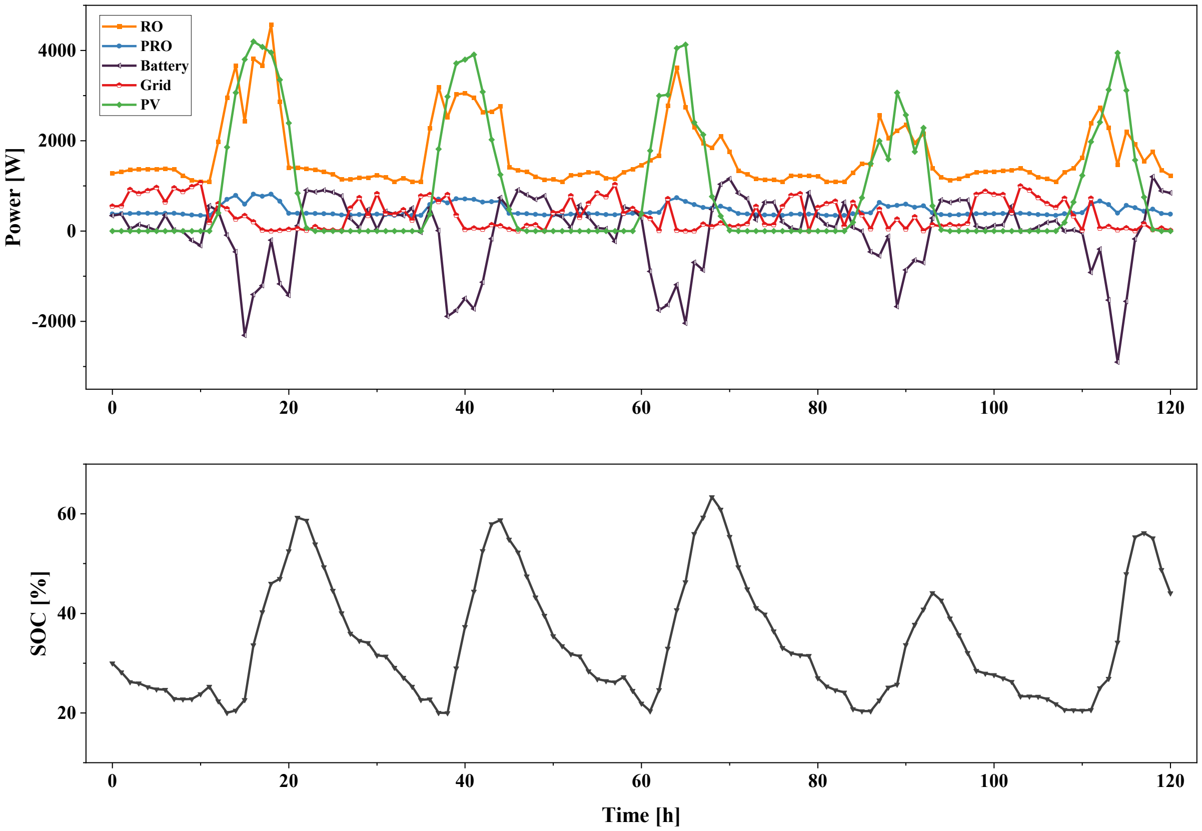

Figure 3.10. Scheduling results of the IEMS: (a) Power of different components and (b) Battery SOC

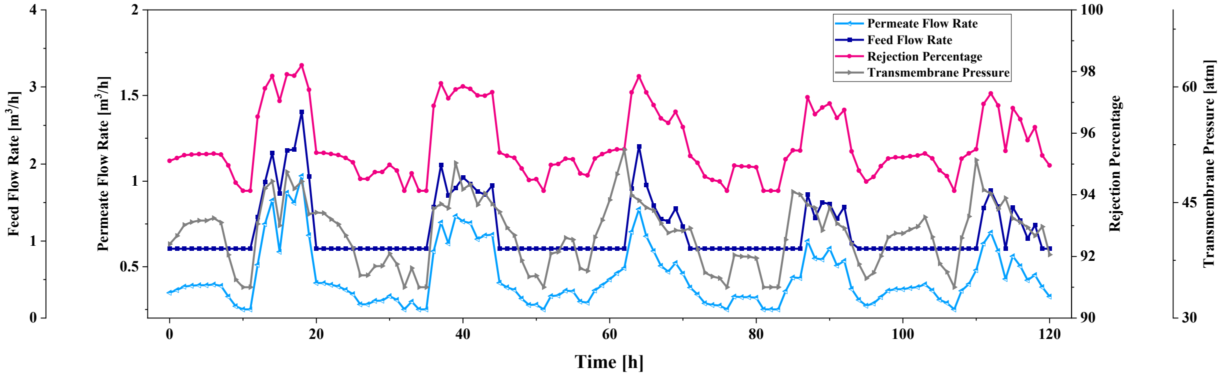

Figure 3.11. Operating conditions determined by the IEMS

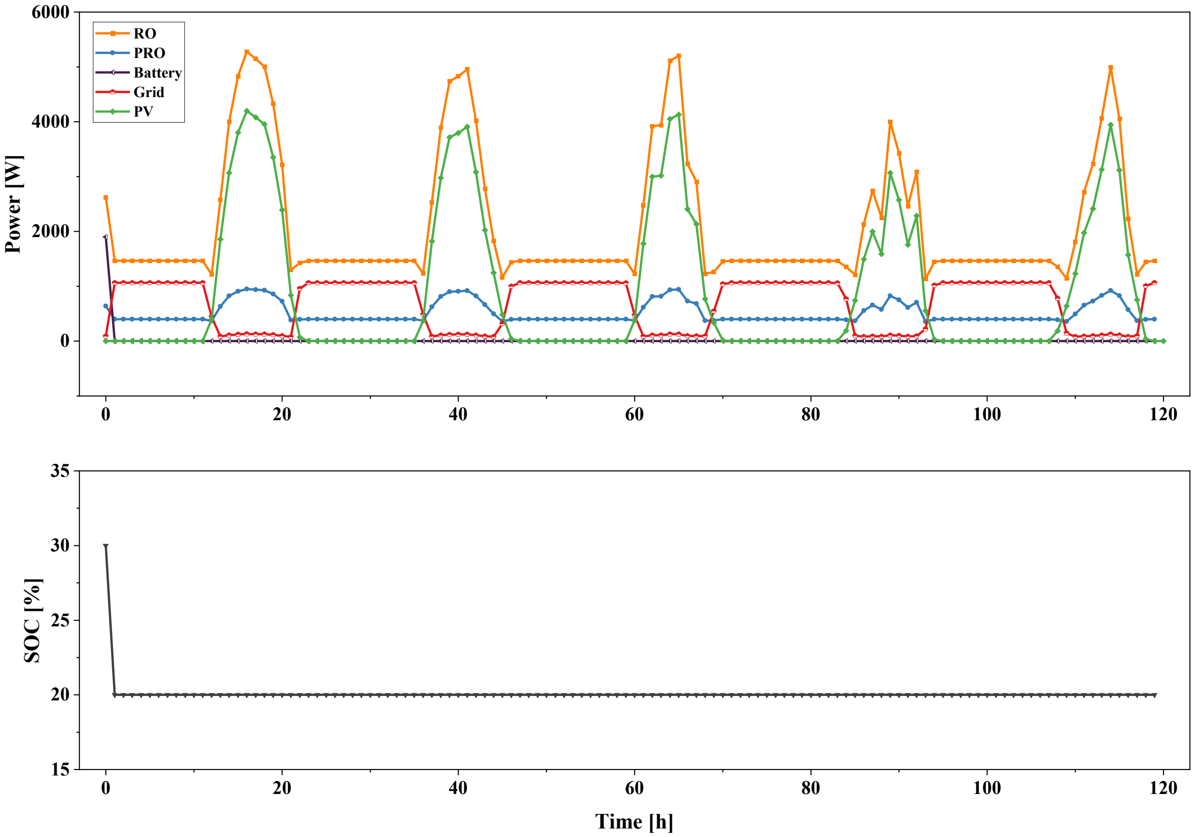

Figure 3.12. Scheduling results of the first benchmark: (a) Power of different components and (b) Battery SOC

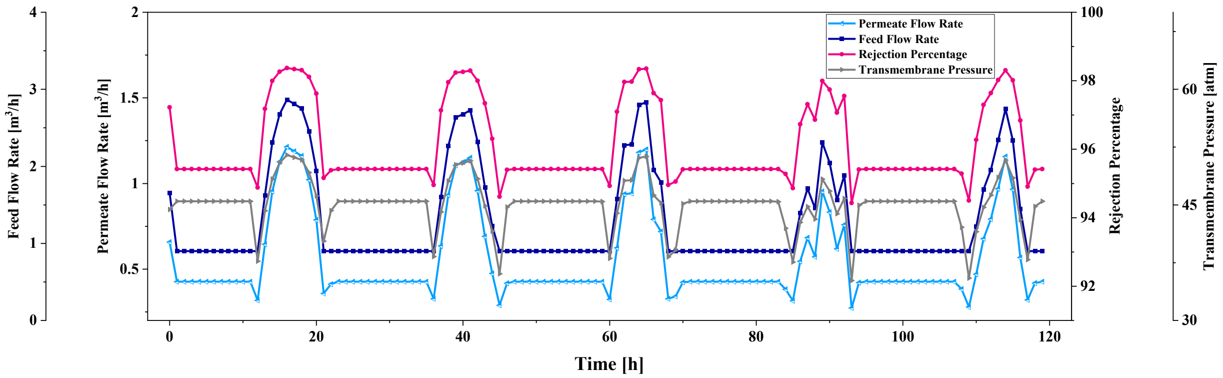

Figure 3.13. Operating conditions determined by the first benchmark

The results shown in the previous four figures are just obtained for five consecutive days, regardless of weather conditions. Here, the performance of the proposed method in different weather conditions is investigated to make sure that acceptable results can be achieved in different scenarios. In **Table 3.7**, the results of two successive sunny days, two successive cloudy days, and ten successive days are presented. In the case of two sunny days, the water production of the proposed method is close to that of the first benchmark, while, its power supply from the main grid is much lower than the first benchmark. As a result, the PI_1 of the proposed method is 43.6% lower than that of the first benchmark. Also, the utilization of the proposed method has led to a smaller PI_2 than the first benchmark. This result shows that IEMS is more efficient in terms of energy consumption. Moreover, based on the third PI, the proposed method leverages the energy storage system, while the first benchmark does not depend on the battery system. The total solute rejection of these two models is almost the same, and they are 2% lower than the maximum rejection percentage. In the case of two successive cloudy days, again, the proposed method provides a smaller PI_1 compared to the first benchmark but a slightly larger PI_2. Also, in this scenario, the second PI of both models is close to the minimum value (i.e., 2.2303 kWh/m^3). Finally, for the simulation of ten successive days, the proposed method achieved 32.4% and 1.8% lower PI_1 and PI_2, respectively, than the first benchmark, and almost the same total rejection as the first benchmark. Moreover, since the second PI value obtained by using the IEMS is close to its minimum possible value, it can be concluded that the IEMS can cope with the complex structure of the RO-PRO system. As a result, the proposed method outperforms the first benchmark in all three scenarios due to the exploitation of the solar power generation forecasting algorithm and the energy storage system.

Table 3.7. Performance of the methods

Weather	Method	$P_{Grid}\ [kWh]$	$Q_P\ [m^3]$	$PI_1\ [\frac{kWh}{m^3}]$	$PI_2\ [\frac{kWh}{m^3}]$	PI_3	PI_4
Sunny	IEMS	14.1126	27.4719	0.5137	3.1325	0.3147	0.9692
	First benchmark	29.7902	32.7280	0.9102	3.1500	0.0072	0.9718
	Second benchmark	-	-	0	2.2303	-	0.993
Cloudy	IEMS	31.1664	17.1401	1.8183	2.5866	0.1136	0.9513
	First benchmark	38.3672	20.0779	1.9109	2.5532	0.0182	0.9549
	Second benchmark	-	-	0	2.2303	-	0.993
10 days	IEMS	88.8643	106.4353	0.8349	2.8387	0.3257	0.9608
	First benchmark	164.3552	132.9860	1.2359	2.8896	0.0020	0.9657
	Second benchmark	-	-	0	2.2303	-	0.993

The presented results so far are obtained by a 5-hour-ahead forecast horizon (FH); that is, the next 5 hours' predicted results are utilized in the optimization algorithm at each timestep. To investigate FH impact, the optimization is performed with different values of FH, and the obtained results are provided in **Table 3.8**. It is found that, in all scenarios, by increasing the FH, the first PI improves, but the second PI shows inconsistent behavior. For instance, in the case of two successive cloudy days, by increasing the forecast horizon from 2 to 5, the first performance index decreases by 1.3%, but the second PI first increases and then decreases. Moreover, increasing the FH enhances the utilization of the energy storage system as the third PI increases. The increase in PI_3 is due to the fact that higher FH provides more information regarding the future solar power values, and IEMS optimizes the system's performance better by using the energy storage system. In all scenarios, the effect of FH on PI_4 is insignificant. Overall, better performance can be achieved by considering higher values for the FH. It should be pointed out that although higher FH results in more optimum operation, by increasing the FH, the accuracy of solar power forecasting models will reduce, which can severely impact the optimization results. Therefore, if we want to incorporate the prediction results of a long-term forecasting model into the optimization process, it is of great significance to ensure that impact of forecast errors on the direction of the optimization process is not significant.

Table 3.8. Performance of the proposed method under different values of FH

Weather	Forecast horizon	$P_{Grid}\ [kWh]$	$Q_P\ [m^3]$	$PI_1\ [\frac{kWh}{m^3}]$	$PI_2\ [\frac{kWh}{m^3}]$	PI_3	PI_4
Sunny	2	21.3231	30.0709	0.7091	3.1371	0.1519	0.9710
	3	18.4181	29.5499	0.6233	3.0711	0.2296	0.9706
	4	14.9987	28.4980	0.5263	3.0370	0.3427	0.9694
	5	14.1126	27.4719	0.5137	3.1325	0.3147	0.9692
Cloudy	2	34.4011	18.6793	1.8417	2.5612	0.0515	0.9534
	3	32.5546	17.6905	1.8402	2.6004	0.0828	0.9522
	4	30.9656	17.0050	1.8210	2.6067	0.1088	0.9512
	5	31.1664	17.1401	1.8183	2.5866	0.1136	0.9513
10 days	2	137.8615	123.5261	1.1161	2.8789	0.1028	0.9645
	3	117.7742	116.7077	1.0091	2.8618	0.1739	0.9632
	4	103.2575	112.1997	0.9203	2.8330	0.2483	0.9621
	5	88.8643	106.4353	0.8349	2.8387	0.3257	0.9608

3.2.3 Robustness of IEMS

In this section, the robustness of the proposed method is evaluated. First, in order to explore the effects of forecast uncertainties on power scheduling, all the above simulations are performed with the actual PV power values rather than the predicted values. This analysis shows if the proposed deep VMD-CNN network can still provide accurate results given the forecast uncertainties. The results of the optimization with the predicted and actual values are provided in **Table 3.9**. According to this table, in the case of two consecutive sunny days, the PIs obtained using the predicted data deviated from the ones achieved based on the actual data by 0.0303 kWh/m^3, 0.1154 kWh/m^3, 0.0672, and 0.0002 for PI$_1$, PI$_2$, PI$_3$, and PI$_4$, respectively. In the second scenario, the changes in PI$_1$, PI$_2$, PI$_3$, and PI$_4$ are found to be 0.0106 kWh/m^3, 0.0112 kWh/m^3, 0.0049, and 0.0004, respectively. In the last scenario, again, the results of actual and predicted data are roughly similar. Although the results of the actual data are slightly better than those of the predicted one, the errors of the network are found to have a minor impact on the final results of the optimization. Hence, the predictions are accurate enough to be utilized in the optimization algorithm.

Table 3.9. Results of IEMS with the predicted and actual PV data

Weather	PV data	$P_{Grid}\ [kWh]$	$Q_P\ [m^3]$	$PI_1\ [\frac{kWh}{m^3}]$	$PI_2\ [\frac{kWh}{m^3}]$	PI_3	PI_4
Sunny	Predicted	14.1126	27.4719	0.5137	3.1325	0.3147	0.9692
	Actual	13.5894	28.1141	0.4834	3.0171	0.3819	0.9690
Cloudy	Predicted	31.1664	17.1401	1.8183	2.5866	0.1136	0.9513
	Actual	31.5270	17.4405	1.8077	2.5754	0.1087	0.9517
10 days	Predicted	88.8643	106.4353	0.8349	2.8387	0.3257	0.9608
	Actual	88.0430	106.6146	0.8258	2.8207	0.3412	0.9606

Second, the performance of the PSO algorithm is evaluated since, in the field of optimization, no algorithm is capable of finding proper solutions for all optimization problems. Hence, the results of the PSO algorithm are compared with two other stochastic optimization algorithms: GA which is an evolutionary algorithm, and GWO, which is a swarm-based algorithm. The optimization is performed for ten days of operation using GA and GWO algorithms, and the obtained results are compared with those obtained by PSO. The parameters of these algorithms and their corresponding values are provided in **Table 3.10**.

Table 3.10. Parameters of GA and GWO

Algorithm	Description	Values of the first timestep	Values of next timesteps
GWO	Number of wolves	700	350
	Number of iterations	600	350
GA	Number of chromosomes	300	100
	Number of generations	600	300
	Probability of crossover	0.95	0.95
	Probability of mutation	0.01	0.01
	Elitism ratio	0.3	0.3

The results of different optimization algorithms are presented in **Table 3.11**. As can be observed, the power scheduling performed by GA does not provide suitable results as the first and second PIs are increased by 0.8318 kWh/m^3 and 0.4615 kWh/m^3, compared to PSO, even though the energy storage system is utilized more. These results demonstrate that GA is not a proper algorithm for solving this optimization problem. Using the GWO algorithm, better results are obtained in comparison with GA, but not PSO. In this case, PI$_1$ and PI$_2$ have increased by 0.1023 kWh/m^3 and 0.0378 kWh/m^3 compared to those of PSO. The total solute rejection of all these algorithms is almost the same. Accordingly, the PSO algorithm outperforms GA and GWO, and this algorithm is suitable for solving the optimization problem of the hybrid PV-RO-PRO system.

Table 3.11. Results of different optimization algorithms

Algorithm	$P_{Grid}\ [kWh]$	$Q_P\ [m^3]$	$PI_1\ [\frac{kWh}{m^3}]$	$PI_2\ [\frac{kWh}{m^3}]$	PI_3	PI_4
PSO	88.8643	106.4353	0.8349	2.8387	0.3257	0.9608
GA	208.6327	125.1775	1.6667	3.3002	0.5449	0.9675
GWO	104.2096	111.1914	0.9372	2.8765	0.2193	0.9627

3.3 Conclusion

In this study, an energy management system was designed for a hybrid PV-RO-PRO desalination system to maximize the total water production and rejection percentage and minimize the main grid power at the same time. The proposed IEMS exploited the PV power prediction results to enhance its effectiveness. To perform solar power forecasting, three DNNs were designed: 2-D CNN, VMD-CNN, and VMD-CNN-LSTM. The hyperparameters of these models were found via grid search, and the best design of each network was evaluated based on its performance for all seasons. The error metrics indicated that the VMD-CNN model outperforms other models in all cases, and thus, it was selected for solar power forecasting. In order to examine the effectiveness of the IEMS, its performance was studied in three scenarios and compared with two benchmark methods. The evaluation of the IEMS and the first benchmark was based on four performance indices. The first PI demonstrates the amount of supplied power from the main grid for producing one cubic meter of water. The second PI is a measure of the energy efficiency of the RO-PRO system. The third PI shows the battery utilization during the operation, and the fourth PI is the total solute rejection. In all scenarios, the simulation results revealed a significant reduction in the value of PI_1, when the proposed technique was employed. In addition, the effects of the forecast horizon on the optimization results were investigated, and it was observed that the utilization of a higher forecast horizon resulted in a smaller PI_1. Furthermore, the impact of VMD-CNN forecast uncertainties on the optimization results was studied. It was found that the proposed network in this study possesses adequate accuracy as the difference between the results of simulations with actual and predicted data was minor. Lastly, the optimization was performed by GWO and GA for ten days of operation, and it was observed that PSO outperforms these algorithms.

4 Novel Data-Driven Energy Management of the PV-RO-PRO System Using Deep Reinforcement Learning

In this chapter, a novel deep reinforcement learning-accelerated energy management system is proposed for the PV-RO-PRO desalination plant described in Chapter 3. Therefore, the same characteristics and constraints are considered for the hybrid system's modules. Also, we use the recorded PV data from 01 June 2015 to 30 May 2016 to train and evaluate the proposed CNN-SAC algorithm.

4.1 Methodology

4.1.1 Formulation of the Energy Management Problem as a Reinforcement Learning Problem

The energy management problem investigated in this study is a sequential decision-making process attributed to the existing time-coupling property that creates a connection between the possible future decisions and currently taken actions. For instance, at each timestep, the current battery SOC limits the maximum amount of energy that can be charged to or discharged from the energy storage system. Also, the current SOC is determined by the previous decisions made by the energy management system. Hence, due to the temporally coupled constraints, the control signals sent by the energy management system influence the available decisions it can make in the future timesteps. To solve this sequential decision-making problem, we convert the optimal control problem of the hybrid desalination system into a reinforcement learning task and solve it by exploiting a deep reinforcement learning-based agent. In this subsection, the formulation of the problem is discussed, and in the next subsection, the developed CNN-SAC algorithm is explained.

In the present study, the environment is the PV-RO-PRO system, whose structure is demonstrated in **Figure 3.1**. At each timestep, the environment provides the agent with the system's state through which the action must be determined. In Markovian environments, the information that the system's state provides is adequate for optimal control. However, in many real-world control problems, the environment can be partially observed, meaning that the agent does not have access to complete information about the environment [83]. Partial observability emerges from various sources, such as the need to remember temporarily available information, limitations of sensors, and noisy information [84]. Partially observed environments can be modeled as partially observable Markov decision processes (POMDPs) in which the agent, instead of receiving the state s_t at timestep t, is provided with observation o_t of the system. In these cases, the agent may need to have access to history $h_t = (o_1, a_1, o_2, a_2, \ldots, a_{t-1}, o_t)$, which consists of all observations and actions from the first timestep to timestep t to describe the state [71]. The energy management problem of the PV-RO-PRO system is a POMDP due to uncertainties and unknown information about the generation of solar power, and the results provided in the next section clearly show that. The history h_t we define for this problem includes only the PV power data of the previous 48 hours since we wish to provide more information about the state of the PV system and alleviate uncertainties about solar power generation. Also, in most cases, it is not feasible to utilize the entire sequence of observations, and consequently, other methods must be exploited to summarize the whole history data [84]. An additional point that should be mentioned is that as a result of the uncertainties of PV power generation, at the beginning of each timestep, the agent cannot observe the current output power of the PV system [67]. Therefore, based on the explanations mentioned above and modeling of the PV-RO-PRO system, at each timestep, the agent receives the historical PV power data $h_t = \left(P_{PV}^{(t-48)}, P_{PV}^{(t-47)}, P_{PV}^{(t-46)}, \ldots, P_{PV}^{(t-1)}\right)$ and battery SOC to determine the action.

The actions that are selected based on the policy control devices of the hybrid system. According to the mathematical model of RO, the transmembrane pressure and RO feed flow rate are two decision variables associated with the RO plant through which the agent can control the permeate flow rate, rejection percentage, and power consumption of the desalination unit. For the PRO model, the pressure, flow rate, and concentration of draw and PRO feed solutions are required to estimate the power that can be generated. In this study, we assume that the properties and flow rate of the PRO feed solution are constant and known. Moreover, as discussed in Chapter 3, the RO retentate is used as the draw solution of PRO; hence, the agent can control the output power of PRO via the RO feed flow rate and transmembrane pressure as well. The power of the energy storage system is another decision variable by which charging and discharging of batteries is performed, and the agent can manage the operation of the energy storage system using this variable. The imported power from the external grid is the last variable that should be determined during operation and can be found using the power balance equation. As discussed in Chapter 3, by determining the transmembrane pressure, RO feed flow rate, and battery power, the main grid power can be calculated using Equation 3.4, which eliminates the necessity to consider it as one of the decision variables. Therefore, the action can be defined as follows:

$$a_t = \left[Q_{f,RO}^{(t)}, \Delta P_{RO}^{(t)}, P_B^{(t)}\right] \tag{4.1}$$

where a_t is the action selected by the agent at timestep t. Additionally, due to the constraints of the hybrid system's devices, the inequality constraints mentioned in Equation 3.5 through Equation 3.9 are considered.

The reward function plays a crucial role in the training of reinforcement learning algorithms. This function can be regarded as an evaluation metric for assessing the performance of the learned policy [52]. Hence, it is of great significance to define the reward function so that we convey the objectives of the optimization and constraints of the problem to the agent. As mentioned in previous sections, we intend to design an energy management system to minimize the supplied power from the external grid while maximizing the permeate flow rate and contaminant removal efficiency. In addition, the control scheme that the agent learns should violate none of the mentioned constraints. Accordingly, to take all of these points into consideration, we define a reward function of the following form:

$$r_t = w_1 r_1 + w_2 r_2 + w_3 r_3 + w_4 r_4 + w_5 r_5 \tag{4.2}$$

As can be observed in Equation 4.2, the reward signal emitted by the environment at each timestep is the weighted sum of five sub-rewards, each examining the agent's performance in different aspects. The weights represent the contribution of each sub-reward function to the total reward r_t. To assess the performance of the agent in terms of water production and interactions with the external grid, we utilize the ratio of supplied power from the main grid to the permeate flow rate to define the r_1 function as follows:

$$r_1 = \frac{-(r_{max} - r_{min})}{r_1^*} \frac{\left|P_{Grid}^{(t)}\right|}{Q_p^{(t)}} + r_{max} \tag{4.3}$$

where $Q_p^{(t)}$ is the permeate flow rate. The function r_1 maps the ratio of $\frac{\left|P_{Grid}^{(t)}\right|}{Q_p^{(t)}}$ to a number between r_{max} and r_{min} by comparing this ratio with the reference value r_1^*. It is evident that values of r_1 can exceed r_{max} or become less than r_{min}; however, the value of r_1^* is selected in a way that r_1 lies in the range of r_{min} to r_{max} most of the time. The optimum value of the $\frac{\left|P_{Grid}^{(t)}\right|}{Q_p^{(t)}}$ is zero since our aim is to minimize $P_{Grid}^{(t)}$ and maximize $Q_p^{(t)}$. Therefore, the actions that result in small values of $\frac{\left|P_{Grid}^{(t)}\right|}{Q_p^{(t)}}$ will receive the highest score (i.e., r_{max}), while corresponding scores of actions with high values of $\frac{\left|P_{Grid}^{(t)}\right|}{Q_p^{(t)}}$ (close to r_1^*) would be around r_{min}. It is worth mentioning that using the $\frac{P_{Grid}^{(t)}}{Q_p^{(t)}}$ ratio to calculate the function r_1 would not be wise since, in that case, negative values of P_{Grid} will lead to scores higher than r_{max}, signaling to the agent that the lower the P_{Grid}, the better the performance. However, negative values of P_{Grid} indicate that the agent is not using the available solar energy efficiently since the summation of the net power consumption of the RO-PRO system and battery power becomes less than P_{PV} (see Equation 3.4), suggesting that instead of utilizing the free solar energy to increase the permeate flow rate or storing energy in batteries, the agent decides to neglect the excessive PV power. Hence, to maintain the balance between power consumption and generation and fully exploit the PV power, the absolute value of $\frac{P_{Grid}^{(t)}}{Q_p^{(t)}}$ is used to calculate the function r_1. Moreover, to further emphasize the importance of keeping $P_{Grid}^{(t)}$ close to zero, the r_2 and r_3 functions are defined as follows:

$$r_2 = \begin{cases} r_{max}, & \left|P_{Grid}^{(t)}\right| < r_2^* \\ 0, & \left|P_{Grid}^{(t)}\right| \geq r_2^* \end{cases} \tag{4.4}$$

$$r_3 = \begin{cases} 0, & \left|P_{Grid}^{(t)}\right| < P_{Grid}^* \\ \frac{-r_{max} - r_{min}}{r_3^*}\left(\left|P_{Grid}^{(t)}\right| - P_{Grid}^*\right) + r_{min}, & \left|P_{Grid}^{(t)}\right| \geq P_{Grid}^* \end{cases} \tag{4.5}$$

By using the r_2 function, the agent achieves the score r_{max} when the main grid power is between $-r_2^*$ and r_2^*, and by defining the r_3 function, the agent realizes that if the absolute value of the imported power from the main grid exceeds P_{Grid}^{*}, lower rewards will be obtained. To evaluate the agent's performance in terms of contaminant removal efficiency, we define the r_4 function:

$$r_4 = \frac{r_{max} - r_{min}}{r_4^*}(R - R^*) + r_{min} \tag{4.6}$$

where R is the rejection percentage that is calculated using the following equation:

$$R = \left(1 - \frac{C_p^{(t)}}{C_f}\right)100 \tag{4.7}$$

Equation 4.7, similar to the function r_1, maps the obtained rejection percentage to a reward in the range of r_{min} to r_{max}. r_4^* and R^* are two reference parameters whose values are selected according to the characteristics of RO membranes and solute rejections we expect from the RO desalination plant. Finally, to take the constraint of the battery SOC into account, the following definition is considered for the function r_5:

$$r_5 = \begin{cases} -\left(SOC_{min} - SOC^{(t+1)}\right), & SOC^{(t+1)} < SOC_{min} \\ -\left(SOC^{(t+1)} - SOC_{max}\right), & SOC^{(t+1)} > SOC_{max} \\ 0, & Otherwise \end{cases} \tag{4.8}$$

According to Equation 4.8, the total rewards received by the agent decrease if the taken actions lead to a SOC higher than SOC_{max} or lower than SOC_{min}. As can be observed, the penalty for violating this constraint is equal to the difference between the new SOC and the maximum (minimum) value considered for the battery SOC. Also, in these cases where the obtained power for the energy storage system results in SOC outside the defined interval, we bound the battery power to keep the SOC at the upper or lower limits. The reference values and weights of the sub-reward functions are carefully selected based on a trial-and-error approach and are tabulated in **Table 4.1**.

In each episode, the agent starts the simulation at 12 am and controls the devices for 7 days, meaning that each episode comprises 168 timesteps. However, it is assumed that if, after 23 hours, the ratio of the total supplied energy from the external grid to the total produced water is higher than 10 kWh/m^3, then the episode is terminated. The environment is designed so that every four episodes, one random day in each season is selected as the starting point (the order of seasons is also random).

Table 4.1. Parameters of the reward function

Parameter	Value
r_{min}	0
r_{max}	1
r_1^*	4 kWh/m^3
r_2^*	10 W
r_3^*	800 W
P_{Grid}^*	200 W
r_4^*	4.34
R^*	95
SOC_{min}	20
SOC_{max}	80
w_1	9
w_2	8
w_3	8
w_4	5
w_5	1

4.1.2 CNN-SAC Algorithm

The agent interacts with the environment to perceive the system's structure and the purpose of the training to carry out the optimization in the predefined direction. According to the formulation discussed in the previous section, the state and action spaces are continuous in this study. The classical reinforcement learning algorithms do not apply to problems with high-dimensional continuous state and action spaces [63]. The combination of reinforcement learning with deep neural networks, known as deep reinforcement learning, boosts the learning process and allows us to tackle problems with high-dimensional continuous state and action spaces through automatic pattern extraction [63], [85]. In this study, we utilize the soft actor-critic (SAC) algorithm [69] as the base model of the proposed energy management system and modify the architecture of its neural networks to address the partial observability involved in the system. SAC can be categorized as a model-free deep reinforcement learning algorithm as the state transition probability and reward function are not a requisite for the training of this model. As its name implies, this algorithm employs the actor-critic architecture where the actor maps the states to actions, and the critic evaluates the state or action values of the actor policy and helps the actor network with the improvement of the policy. One of the distinctive features of the SAC algorithm is that, in contrast to standard reinforcement learning models, it maximizes both the expected entropy of the policy and the expected return to optimize the policies. Specifically, based on the maximum entropy formulation, the agent learns a policy that maximizes the summation of the expected sum of rewards and $\alpha\mathcal{H}(\pi)$, where $\mathcal{H}$ is the entropy and α is the temperature parameter through which the relative importance of the entropy against the reward can be controlled. This formulation significantly enhances the exploration and robustness of the algorithm [69].

In the SAC algorithm, the Q-function (critic) Q_θ, state value function V_ψ, and stochastic policy π_ϕ are parameterized by means of deep neural networks whose parameters are θ, ψ, and ϕ, respectively. Training a separate function approximator for the state value function brings more stability to the learning process. This network is trained simultaneously with other neural networks by minimizing the following loss function:

$$J_V = \mathbb{E}\left[\frac{1}{2}\left(V_\psi(s_t) - \mathbb{E}\left[Q_\theta(s_t, a_t) - \log \pi_\phi(a_t|s_t)\right]\right)^2\right] \tag{4.9}$$

The update of the Q-function parameters is carried out by minimizing the soft Bellman residual:

$$J_Q = \mathbb{E}\left[\frac{1}{2}\left(Q_\theta(s_t, a_t) - \hat{Q}(s_t, a_t)\right)^2\right] \tag{4.10}$$

where $\hat{Q}(s_t, a_t)$ is defined as follows:

$$\hat{Q}(s_t, a_t) = r(s_t, a_t) + \gamma\mathbb{E}\left[V_{\bar{\psi}}(s_{t+1})\right] \tag{4.11}$$

In Equation 4.11, $V_{\bar{\psi}}$ is the target value network whose weights can be updated by gradually tracking the weights of the state value network V_ψ via soft update:

$$\bar{\psi} \leftarrow \tau\psi + (1 - \tau)\bar{\psi} \tag{4.12}$$

where τ is the soft update parameter. In this study, the policy is modeled as a Gaussian whose mean and standard deviation are given by the policy network. The parameters of the policy network are updated by minimizing the following loss function:

$$J_\pi = \mathbb{E}\left[\log \pi_\phi(a_t|s_t) - Q_\theta(s_t, a_t)\right] \tag{4.13}$$

Value-based reinforcement learning models are prone to overestimation bias in which high values are estimated for bad states. This phenomenon degrades the performance of actor-critic algorithms, resulting in suboptimal policy updates. To alleviate this problem, the SAC algorithm employs two Q-functions with parameters θ_i and trains them separately. Then, the minimum of the Q-functions is taken to calculate the loss functions of the policy and value networks. Moreover, the SAC algorithm makes use of a replay buffer to utilize off-policy data for the training of the value network, policy network, and Q-function networks. Specifically, at each timestep, the state, action, reward, and new state are stored in a replay buffer, and, in training steps, random batches of data are sampled from the replay buffer to update the neural networks.

As discussed in the prior subsection, we provide the agent with a history of PV data to create a more accurate depiction of the system's state. However, it is of great significance to utilize an effective method to extract necessary features from the historical data to make the interpretation of the extracted information less challenging. To this end, we introduce 1-dimensional convolutional neural networks (1-D CNNs) to the function approximators of the SAC algorithm. 1-D CNNs have almost a similar structure to 2-D CNNs that are widely used for computer vision problems [86]. In the case of sequential data, we can treat the time as a spatial dimension, similar to dimensions of 2-D images. In the convolutional layer of 1-D CNNs, a kernel is convolved with the sequential data and calculates the weighted sum of the data points it observes. The components of the kernel are the weights that will be updated by the backpropagation process. Also, several kernels can be used in a convolutional layer to improve the performance in terms of recognizing the local patterns hidden in the historical PV data. The structure of the Q-function networks, value network, and actor network are illustrated in **Figure 4.1**. As can be observed in **Figure 4.1(a)**, the historical PV data is fed into a 1-D CNN with two convolutional layers. The extracted features by the convolutional layers are then concatenated with the output of a dense layer whose inputs are the battery SOC and actions. Subsequently, the concatenated tensor is given to a neural network with three dense layers to estimate the action-value function. The value network has almost the same architecture; however, the actions are not fed into the input layer of this network as they are not needed. The actor network comprises the same layers considered for the value network except for the output layer, where two separate dense layers are considered for outputting the mean and standard deviation of the Gaussian distribution associated with each action variable. During the training of the algorithm, we take samples from the Gaussian distributions to determine the action. However, for evaluating the trained model, we make the final policy deterministic and use the mean of the distributions. Also, the hyperbolic tangent function is used to bound the Gaussian samples (or means) between -1 and 1.

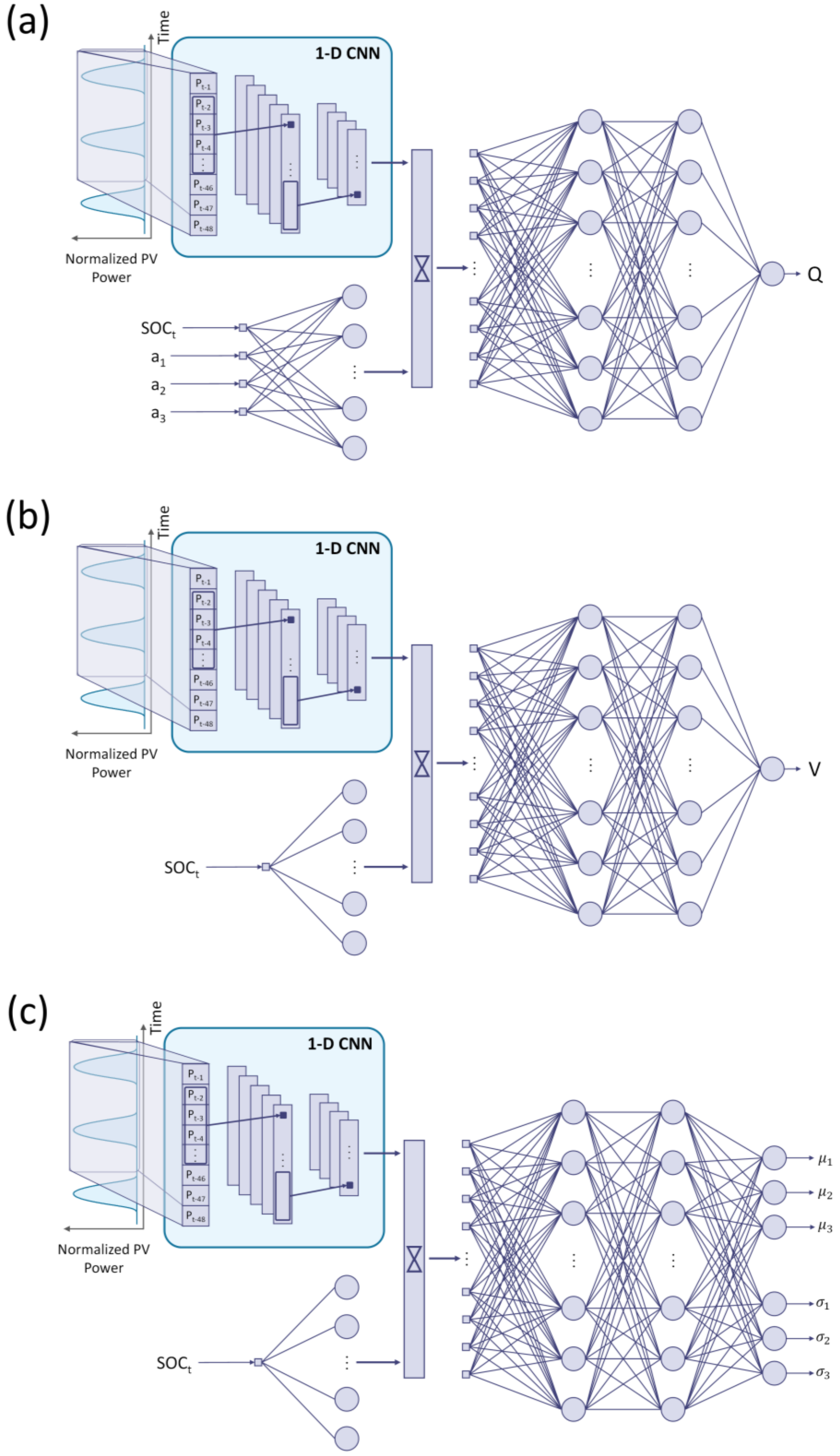

Figure 4.1. (a) Structure of the Q-function (critic) networks, (b) Structure of the value network, (c) Structure of the actor network. In the first panel, a_1, a_2, and a_3 denote the action variables. In the last panel, μ and σ represent the mean and standard deviation of Gaussian distributions, respectively.

We call this new model, whose neural networks are modified to extract information favorable to value, critic, and actor networks, the CNN-SAC algorithm. **Figure 4.2** demonstrates the training process of the proposed method. The actor network determines the action according to the provided historical PV data by the environment and current battery SOC. The action is executed in the environment, and the obtained reward, the performed action as well as previous and new historical data and battery SOC are stored in the replay buffer. Then, a random batch of data is sampled from the memory for the training of the neural networks. The critic values are calculated based on the previously sampled states and actions (according to the current policy), and the minimum value is taken to find the gradients necessary to update of the value network. After that, the critic networks are updated separately using the same target values calculated based on the target value network. Next, the loss function of the actor network is calculated according to the minimum of critic values and logarithm of the probabilities of the actions sampled from the current policy. Lastly, the target value network is updated via soft update. Once the networks are updated, the actor network delivers the action of the new observation, and this process continues until the end of the episode. **Table 4.2** and **Table 4.3** detail the hyperparameters of the CNN-SAC algorithm and architecture of the neural networks, respectively.

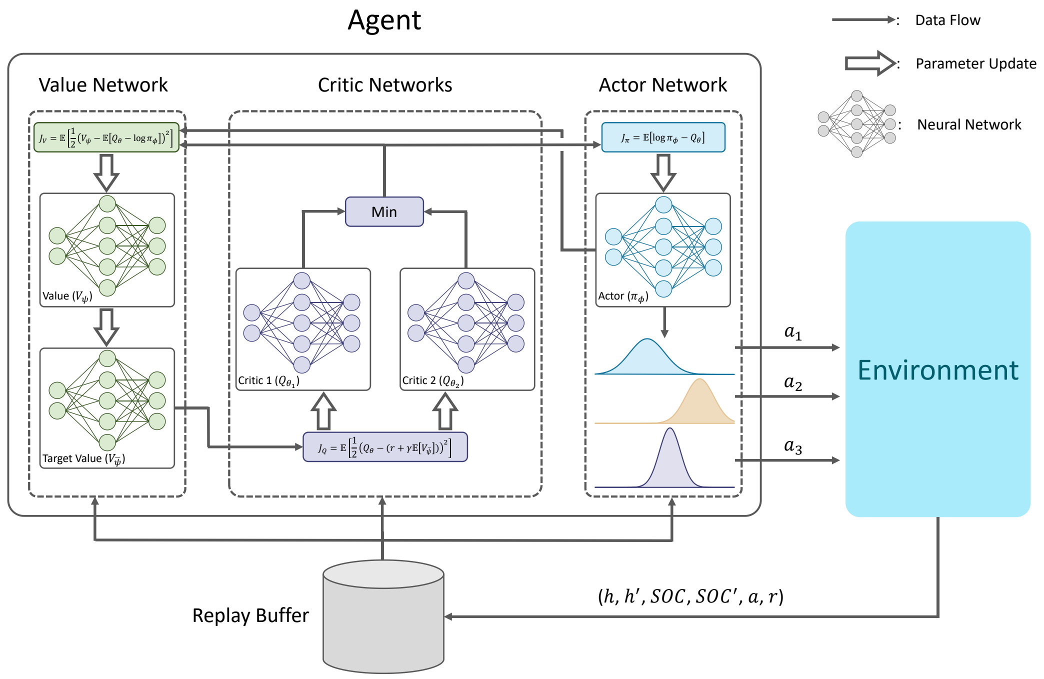
Agent
Data Flow
Parameter Update
Neural Network
Value Network
Critic Networks
Actor Network
$J_V = \mathbb{E}\left[\frac{1}{2}(V_\psi - \mathbb{E}[Q_\theta - \log\pi_\phi])^2\right]$
$J_\pi = \mathbb{E}[\log\pi_\phi - Q_\theta]$
Min
Value (V_ψ)
Actor (π_ϕ)
Critic 1 (Q_{θ_1})
Critic 2 (Q_{θ_2})
Target Value ($V_{\bar{\psi}}$)
$J_Q = \mathbb{E}\left[\frac{1}{2}(Q_\theta - (r + \gamma\mathbb{E}[V_{\bar{\psi}}]))^2\right]$
a_1
a_2
a_3
Environment
(h, h', SOC, SOC', a, r)
Replay Buffer

Figure 4.2. The training process of CNN-SAC algorithm

Table 4.2. Hyperparameters of the CNN-SAC algorithm

Hyperparameter	Value
Discount factor (γ)	0.99
Temperature parameter (α)	0.1
Target networks update rate (τ)	0.005
Batch size	256
Replay buffer size	1000000
Optimizer	Adam
Nonlinearity	ReLU
Actor network learning rate	0.0003
Critic network learning rate	0.0003
Value network learning rate	0.0003

Table 4.3. The architecture of the CNN-SAC algorithm neural networks

Network	Layers					
	Convolution 1D	Convolution 1D	Flatten	Dense	Dense	Dense
Actor	64 (kernel size=12, stride=2)	32 (kernel size=12, stride=1)		128	256	256
Critic	64 (kernel size=12, stride=2)	32 (kernel size=12, stride=1)		128	256	256
Value	64 (kernel size=12, stride=2)	32 (kernel size=12, stride=1)		128	256	256

4.2 Results and Discussion

We utilize the CNN-SAC algorithm to solve the reinforcement learning problem defined for the hybrid PV-RO-PRO system. The CNN-SAC algorithm is implemented using the PyTorch deep learning framework on a personal computer with an NVIDIA GeForce RTX 2080 Ti graphics card and 32GB RAM.

4.2.1 Evaluation of CNN-SAC

In this section, the proposed CNN-SAC algorithm is evaluated by analyzing the accumulated rewards obtained over the course of training. To benchmark the CNN-SAC algorithm, we compare its performance against four state-of-the-art deep reinforcement learning algorithms: Deep Deterministic Policy Gradient (DDPG), Proximal Policy Optimization (PPO), Twin Delayed Deep Deterministic Policy Gradient (TD3), and vanilla SAC. At each timestep t, the observation o_t of these algorithms comprises the output power of the PV system at the previous timestep as well as the current battery SOC. Additionally, we benchmark CNN-SAC against the CNN-TD3 algorithm in which a 1-D CNN (with the same architecture as the 1-D CNN of CNN-SAC) is added to the critic and actor networks of the TD3 algorithm. An in-depth comparison between CNN-SAC and vanilla SAC is made in Section 4.2.3, where we investigate the importance of introducing the 1-D CNN to the neural networks of SAC. The hyperparameters and neural networks architecture of the benchmark algorithms are provided in **Table 4.4** and **Table 4.5**, respectively.

In order to examine the CNN-SAC and baseline algorithms in terms of learning performance, we train four different instances with four different random seeds for each algorithm. Each instance is run for 6,000 episodes (more than 1 million timesteps) and is evaluated in 8 episodes every 8,000 timesteps. We design the evaluation process in a way that the performance of the agent is tested twice in each season. The learning curve of the previously mentioned algorithms is illustrated in **Figure 4.3**. A separate panel is considered for the PPO algorithm to have a better visualization of the learning curves. In this figure, the solid curves indicate the mean value of the cumulative rewards over all test episodes and random seeds, and the shaded region represents the standard deviation of the average accumulated rewards (obtained in each evaluation step) over the four trials. As can be observed in **Figure 4.3(a)**, the total rewards obtained by the CNN-SAC algorithm are lower than other benchmark methods (except for PPO) during the first stages of training. However, after roughly 40,000 timesteps, the cumulative scores start to increase monotonically and reach a plateau around the 0.7 millionth timestep with the highest cumulative reward compared to all baseline methods. After the CNN-SAC algorithm, the CNN-TD3 and TD3 exhibit better performance than the rest of the models, respectively.

Table 4.4. Hyperparameters of benchmark algorithms. In this table, $N(\mu, \sigma)$ represents a Gaussian distribution with a mean value of μ and a standard deviation of σ.

Hyperparameter	Algorithm				
	CNN-TD3	TD3	SAC	PPO	DDPG
Discount factor (γ)	0.99	0.99	0.99	0.99	0.99
Temperature parameter (α)	-	-	0.1	-	-
Target networks update rate (τ)	0.005	0.005	0.005	-	0.001
Batch size	100	100	256	512	64
Replay buffer size	1000000	1000000	1000000	-	1000000
Optimizer	Adam	Adam	Adam	Adam	Adam
Nonlinearity	ReLU	ReLU	ReLU	ReLU	ReLU
Actor network learning rate	0.001	0.001	0.0003	0.0001	0.0001
Critic network learning rate	0.001	0.001	0.0003	-	0.001
Value network learning rate	-	-	0.0003	0.0001	-
Actor network exploration noise	$N(0, 0.05)$	$N(0, 0.05)$	-	-	$N(0, 0.1)$
Target actor network noise	$N(0, 0.2)$	$N(0, 0.2)$	-	-	-
Target actor network noise clip boundary	0.5	0.5	-	-	-
Target networks update delay	2	2	-	-	-
Actor network update delay	2	2	-	-	-
Generalized advantage estimation parameter	-	-	-	0.95	-
Number of epochs	-	-	-	2	-
Horizon (T)	-	-	-	4096	-
Epsilon (ϵ)	-	-	-	0.2	-

Table 4.5. Architecture of the benchmark algorithms neural networks. The kernel size of convolutional layers is 12.

Algorithm	Network	Layers					
		Convolution 1D	Convolution 1D	Flatten	Dense	Dense	Dense
CNN-TD3	Actor	64 (stride=2)	32 (stride=1)		128	400	300
	Critic	64 (stride=2)	32 (stride=1)		128	400	300
TD3	Actor	-	-	-	-	400	300
	Critic	-	-	-	-	400	300
SAC	Actor	-	-	-	-	256	256
	Critic	-	-	-	-	256	256
	Value	-	-	-	-	256	256
PPO	Actor	-	-	-	-	512	512
	Value	-	-	-	-	512	512
DDPG	Actor	-	-	-	-	512	512
	Critic	-	-	-	-	512	512

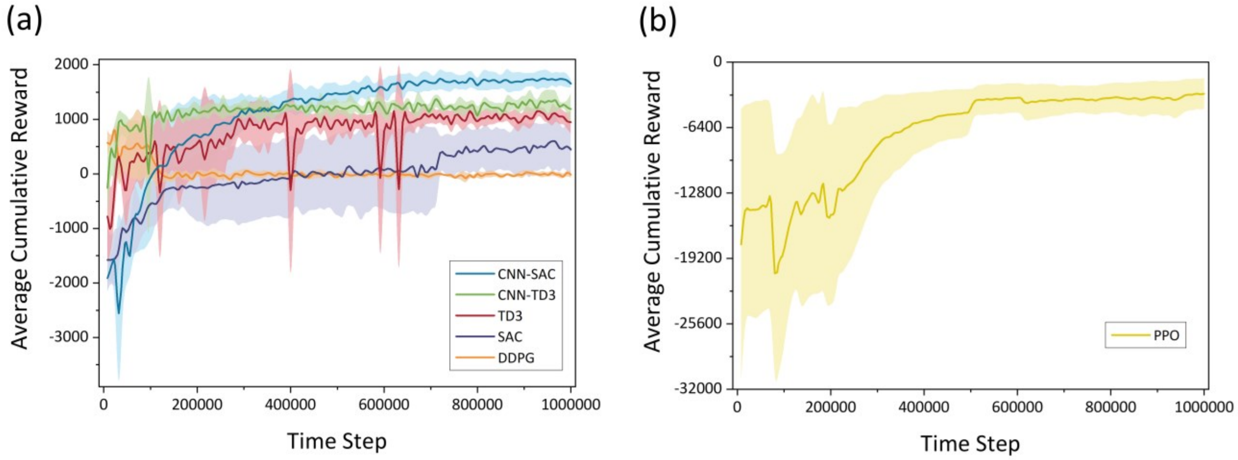

Figure 4.3. The learning curve of the CNN-SAC and baseline algorithms

An essential point worth mentioning is the effects of the added 1-D CNN to SAC and TD3. We can observe that the convolutional neural network enhances the learning speed, final performance, and stability across the random seeds, especially when it comes to the SAC algorithm, as there is a huge gap between the final scores of CNN-SAC and vanilla SAC. Also, by comparing the learning curves of CNN-TD3 and TD3, we can notice that the stability has improved significantly since the CNN-TD3 algorithm performs more consistently than TD3 throughout the learning process. These results clearly demonstrate the critical role the 1-D CNN plays in the training process. The DDPG algorithm has the highest stability, especially after the 0.2 millionth timestep. However, its final performance is not comparable to previous models. Lastly, PPO has the poorest performance in terms of both stability and obtained rewards among all methods, as shown in **Figure 4.3(b)**.

In addition to investigating the learning curves, we carry out a quantitative analysis of the last 10 evaluations of each method. Specifically, the overall average accumulated reward (over the last 10 evaluations), standard deviation (over the last 10 evaluations), best instance accumulated reward, as well as maximum and minimum average accumulated rewards, come under scrutiny to compare the stability and final performances of the algorithms. If we assume that $R_n^{(i)}$ is the average accumulated reward of the i$^{\text{th}}$ instance at the n$^{\text{th}}$ evaluation step, then the mentioned parameters can be calculated via the following equations:

$$\text{Overall average accumulated reward} = \frac{1}{10I}\sum_{N-9}^{N}\sum_{i=1}^{I} R_n^{(i)} \tag{4.14}$$

$$\text{Standard deviation} = \sqrt{\frac{1}{10I-1}\sum_{N-9}^{N}\sum_{i=1}^{I}\left(R_n^{(i)} - \mu\right)^2} \tag{4.15}$$

$$\text{Maximum average accumulated reward} = \max_n \sum_{i=1}^{I}\frac{1}{I}R_n^{(i)} \tag{4.16}$$

$$\text{Minimum average accumulated reward} = \min_n \sum_{i=1}^{I}\frac{1}{I}R_n^{(i)} \tag{4.17}$$

$$\text{Best instance accumulated reward} = \max_n \max_i R_n^{(i)} \tag{4.18}$$

where I is the total number of instances, N is the total number of evaluations, and μ is the overall average accumulated reward calculated using Equation 4.14. The evaluation results of CNN-SAC and benchmark methods based on the evaluation metrics defined in Equations 4.14-4.18 are presented in **Table 4.6**. The CNN-SAC algorithm achieves the highest overall and maximum average accumulated rewards with values of 1722.58 and 1752.66, respectively. The difference between the overall average accumulated rewards of CNN-SAC and CNN-TD3 (the next-best method) is roughly 35%, indicating the superior performance of CNN-SAC to baseline algorithms. Furthermore, the best instance accumulated reward obtained by this algorithm is higher than that of other methods. The CNN-SAC algorithm provided the lowest standard deviation after DDPG; however, this model distinguishes itself from the DDPG algorithm by a large margin in terms of overall average accumulated reward. By comparing the results of CNN-SAC and vanilla SAC, the substantial enhancement made by the 1-D CNN becomes crystal clear as the overall average accumulated reward and standard deviation are improved by 237% and 68%, respectively. This improvement reveals that the introduced CNN not only boosts the final performance of SAC but also stabilizes this algorithm since less variability is observed across the evaluations. Although the minimum average accumulated reward of CNN-SAC is lower than that of all benchmark methods (except for PPO), this algorithm manages to promote its policy during the training process and accomplish the best final performance in the end. As the evaluation metrics indicate, PPO shows the worst performance since it has the lowest overall average accumulated reward and highest standard deviation among all methods. According to the results presented in **Figure 4.3** and **Table 4.6**, it can be concluded that CNN-SAC outperforms the benchmark algorithms in almost every aspect.

Table 4.6. Comparing CNN-SAC and benchmark methods. The best value obtained for each evaluation metric is highlighted in bold text.

Evaluation metric	CNN-SAC	CNN-TD3	TD3	SAC	DDPG	PPO
Overall average accumulated reward	**1722.58**	1278.06	1058.04	511.39	-7.87	-3310.40
Standard deviation	126.83	166.19	146.13	396.85	**47.33**	1438.59
Maximum average accumulated reward	**1752.66**	1373.88	1153.97	598.68	795.17	-3078.33
Minimum average accumulated reward	-2555.38	-257.68	-961.06	-1578.80	**-93.33**	-20665.25
Best instance accumulated reward	**2011.62**	1761.83	1431.27	1401.49	1176.06	-1114.89

4.2.2 Simulation Results of CNN-SAC

In this section, the simulation results obtained using the CNN-SAC algorithm are reviewed, and the effectiveness of the proposed method in terms of power scheduling, utilization of available solar power, and water production are analyzed. In order to provide a better insight into the CNN-SAC algorithm performance and evaluate the simulation results in different aspects, four performance indicators (PI) are defined as follows:

$$PI_1 = \frac{\int_0^T \max(P_{Grid}, 0)\, dt}{\int_0^T Q_p dt} \tag{4.19}$$

$$PI_2 = 100\left(1 - \frac{1}{C_f}\frac{\int_0^T (Q_p C_p) dt}{\int_0^T Q_p dt}\right) \tag{4.20}$$

$$PI_3 = \frac{\int_0^T (P_{RO} - P_{PRO}) dt}{\int_0^T Q_p dt} \tag{4.21}$$

$$PI_4 = \int_0^T \min(P_{Grid}, 0) dt \tag{4.22}$$

where T is the total time of the simulation. The first PI measures how much energy on average has been supplied from the main grid for producing 1 m^3 of potable water. Small values of PI$_1$ show that the agent is capable of relying only on the PV system to provide the required power of the RO process. The second PI is the obtained total rejection percentage and represents the quality of the produced water. PI$_3$ is the net specific energy consumption of the RO-PRO system and determines how much energy on average has been consumed in the RO-PRO system during the simulation to produce 1 m^3 of water. The last PI measures the total amount of solar energy that has not been utilized during the simulation and indicates to what degree decisions made by the agent have resulted in the waste of available, free energy provided by the PV system.

In order to demonstrate the performance of the proposed CNN-SAC algorithm in different weather conditions and scenarios, the simulation results of the agent in two episodes with various levels of solar energy availability and fluctuations are provided in **Figure 4.4** and **Figure 4.5**. In these figures, to better demonstrate the simulation results, only the performance of the CNN-SAC method in the first 73 timesteps (hours) of the episodes are depicted. It should be pointed out that the positive values of the grid power are shown in these figures. As can be observed in **Figure**

4.4(a), at the beginning of the episode, the batteries are discharged immediately to impede the increase of the main grid power; however, since the initial SOC is 30%, the main grid power is increased inevitably after two hours. After that batteries' SOC reaches their minimum value (20%), the power of this device decreases, which shows that the CNN-SAC algorithm observes the constraints considered for the energy storage system and does not allow any further discharging. Once the PV power increases, we can notice an increase in the power consumption of RO and SOC of batteries and a sudden reduction in the main grid power. According to **Figure 4.4(c)** and **(d)**, the increase of the RO power consumption can be attributed to the higher feed flow rate and transmembrane pressure, which results in a higher permeate flow rate and contaminant removal efficiency, as evidenced by **Figure 4.4(e)** and **(f)**. This power allocation scheme concurs with the defined reward function since the agent increases the water production and rejection percentage while minimizing the supplied power from the external grid. The energy allocation scheduling of the energy storage system demonstrates that a portion of solar energy is stored in batteries during high-PV-power hours and discharged almost uniformly during the night. This control scheme prevents the CNN-SAC algorithm from increasing the main grid power when the PV power becomes zero, in contrast to the first timesteps where enough energy is not stored in the batteries, and grid power increases after two hours. This result indicates the essential role of the energy storage system in improving the system's independence from the external grid. An additional point that can be discerned from **Figure 4.4(a)** is the capability of the CNN-SAC algorithm in predicting the distribution of solar power. As discussed in the previous section, the observation that the CNN-SAC method has access to does not include the current output of the PV system. However, with solar power fluctuations, the CNN-SAC algorithm quickly adapts itself to the new conditions (that does not know of), suggesting that this model predicts the PV power time series without exploiting any forecasting model. In **Figure 4.5**, the results of the proposed method are illustrated for a low-PV-power scenario. The first point we can notice is that the power consumption of RO is lower than the previous scenario due to the absence of sufficient solar energy. In this case, the agent follows almost the same procedure for the energy storage system to prevent the increase of main grid power during zero-PV-power hours. As shown in **Figure 4.5(a)**, the storage of energy during the day allows the agent to keep the power consumption of RO almost constant and impede a significant reduction in permeate flow rate. This figure also manifests that the CNN-SAC algorithm predicts the pattern of solar power generation since the feed flow rate and transmembrane pressure are adjusted according to the current output of the PV system to which the algorithm does not have access.

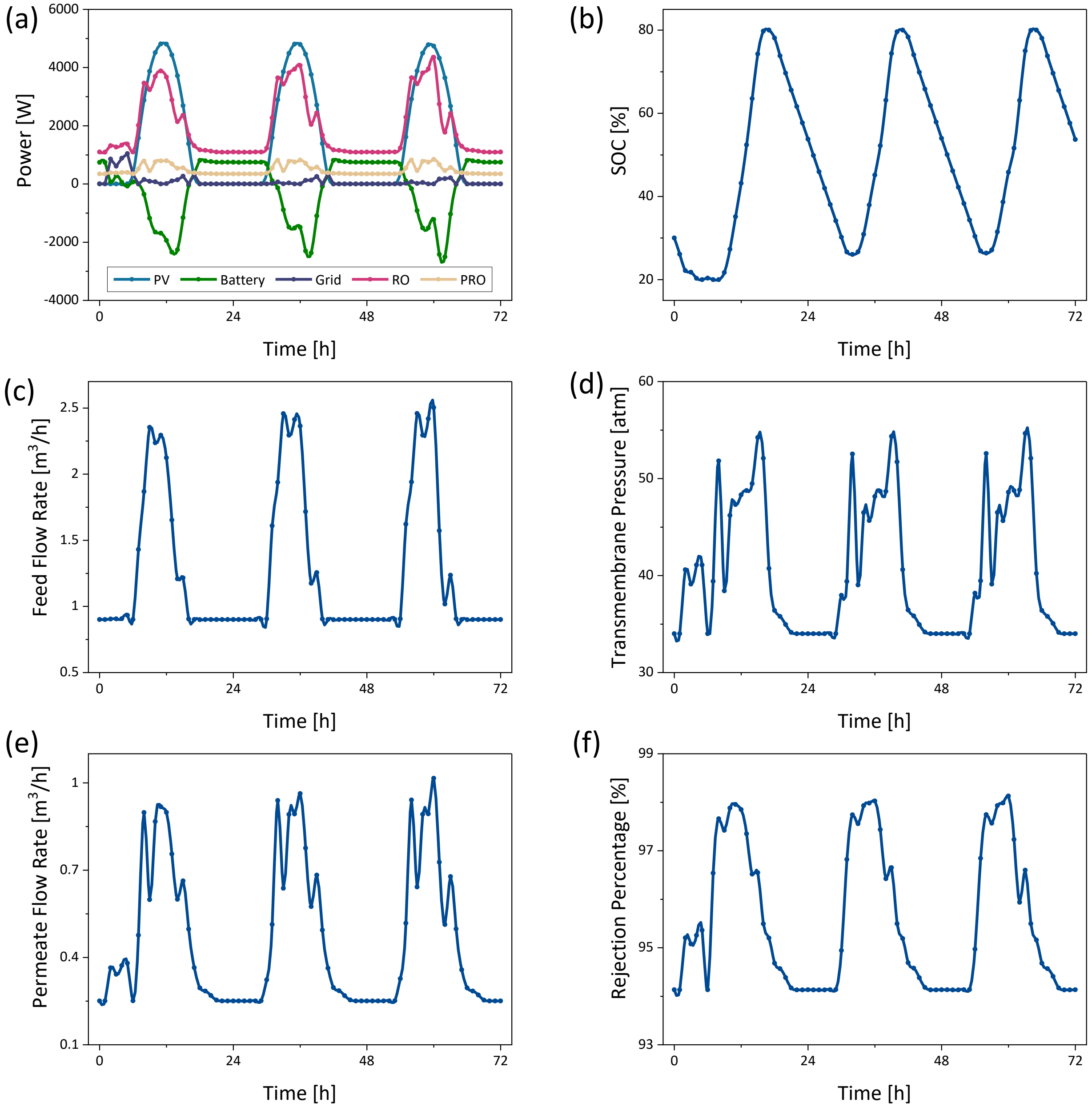

Figure 4.4. Simulation results of the CNN-SAC algorithm for the high-PV-power scenario: (a) Power of different devices, (b) Battery SOC, (c) Feed flow rate, (d) Transmembrane pressure, (e) Water production rate, (f) Rejection percentage

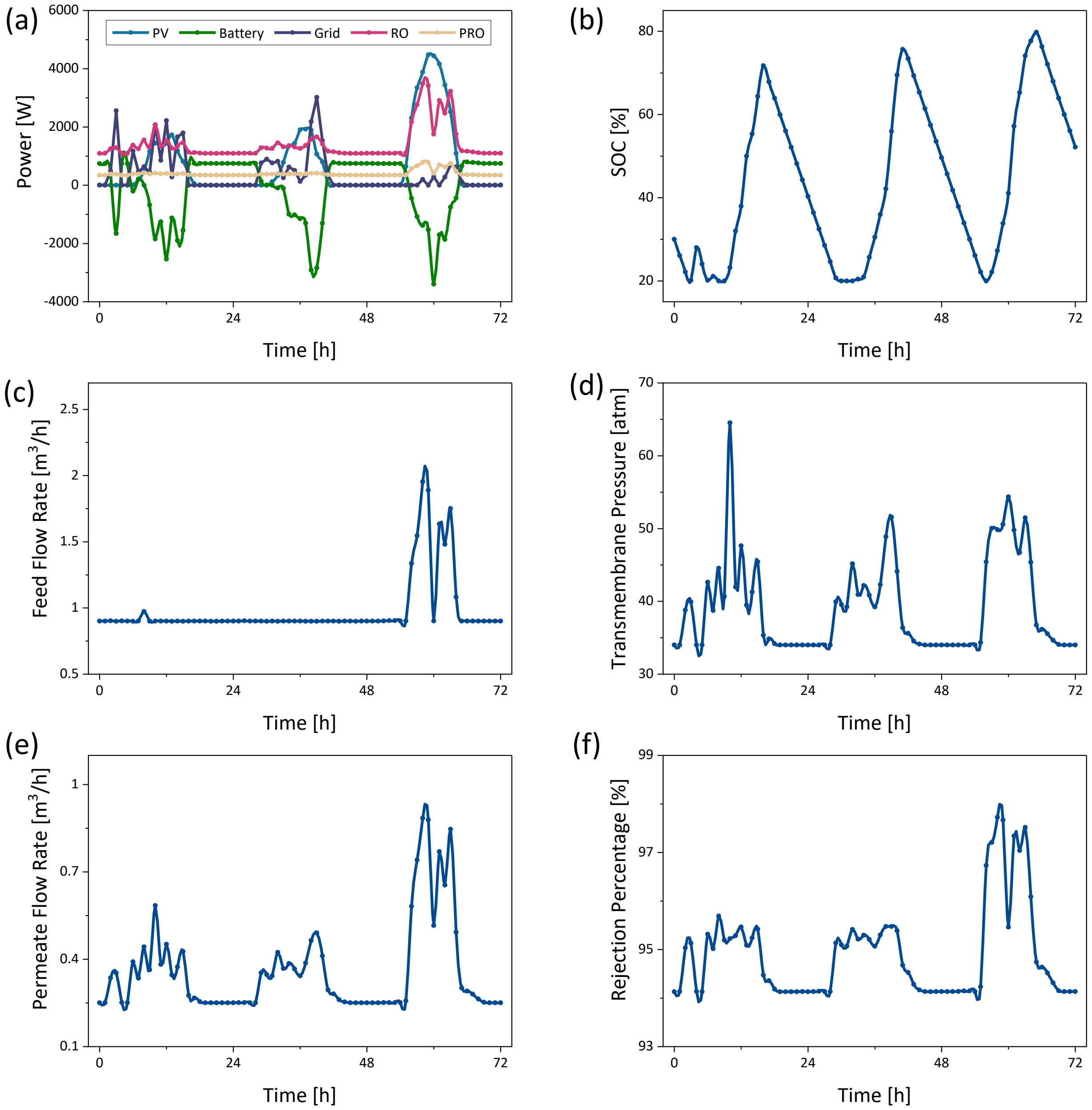

Figure 4.5. Simulation results of the CNN-SAC algorithm for the low-PV-power scenario: (a) Power of different devices, (b) Battery SOC, (c) Feed flow rate, (d) Transmembrane pressure, (e) Water production rate, (f) Rejection percentage

In addition to the comparative evaluation conducted in the prior subsection, we validate the CNN-SAC algorithm against the benchmark methods by taking the PI values obtained in simulations into account. To make a comprehensive comparison between these models, we investigate three case studies based on the PV dataset: an episode with high-PV-power days, an episode with low-

PV-power days, and an episode containing both high- and low-PV-power days. The performance indicators obtained using the proposed algorithm and baseline methods for the case studies are tabulated in **Table 4.7**. It should be pointed out that the best instance of each algorithm is used to carry out these simulations. The CNN-SAC model achieves the lowest PI_1 among all benchmark methods with 0.156 kWh/m^3, 1.203 kWh/m^3, and 0.45 kWh/m^3 for the first, second, and third case studies, respectively. This performance indicates that CNN-SAC is the most capable model compared to the baseline methods in terms of minimum interaction with the external grid and utilization of energy storage system for efficient power scheduling. Moreover, the proposed model has the best performance in fully exploiting the available solar energy since the lowest PI_4 belongs to this algorithm in all case studies.

Table 4.7. Simulation results of CNN-SAC and baseline algorithms. The unit of P_{Grid}, Q_p, PI_1, PI_3, PI_4 are kWh, m^3, kWh/m^3, kWh/m^3, and kWh, respectively.

Case study	Algorithm	P_{Grid}	Q_p	PI_1	PI_2	PI_3	PI_4
High-PV-power days	CNN-SAC	12.187	77.937	0.156	96.336	3.099	3.053
	CNN-TD3	13.894	86.673	0.160	96.503	2.807	3.669
	TD3	41.826	95.158	0.440	96.772	2.910	3.953
	SAC	41.319	92.176	0.448	96.438	2.899	7.636
	DDPG	102.186	103.388	0.988	97.099	3.188	15.745
	PPO	176.710	120.2	1.470	97.200	2.880	74.729
Low-PV-power days	CNN-SAC	67.131	55.784	1.203	95.099	2.782	2.559
	CNN-TD3	80.049	62.754	1.276	95.465	2.695	7.146
	TD3	102.332	71.320	1.435	95.903	2.742	2.617
	SAC	126.842	78.913	1.607	96.145	2.792	2.961
	DDPG	131.113	77.604	1.690	96.270	2.885	6.233
	PPO	258.237	117.840	2.191	97.246	2.913	14.421
High- and low-PV-power days	CNN-SAC	28.810	64.027	0.450	95.598	2.828	2.236
	CNN-TD3	45.118	71.752	0.629	95.897	2.717	7.806
	TD3	67.913	79.986	0.849	96.293	2.820	5.034
	SAC	88.425	84.586	1.045	96.260	2.847	6.940
	DDPG	119.017	90.167	1.320	96.712	2.996	16.257
	PPO	215.941	119.453	1.808	97.236	2.899	36.860

The CNN-TD3 method achieves the lowest net specific energy consumption (PI_3) by a slight margin compared to the rest of the models; however, this model struggles to efficiently use the solar energy in the second and third case studies as PI_4 of this model is roughly three times higher than that of CNN-SAC. Regarding contaminant removal efficiency, the PPO algorithm has the best performance, which can be attributed to the fact that this model mostly focuses on improving the rejection percentage to maximize its rewards. The PI_1 and PI_4 obtained by the PPO model indicate that it struggles to improve its performance in those aspects. The maximum difference between the rejection percentage of CNN-SAC and that of PPO is only 2.2%, showing that the performance of the proposed method in this regard is slightly worse than the best-achieved value. All in all, we can conclude that the CNN-SAC algorithm outperforms the benchmark methods in all case studies.

Finally, to compare the effectiveness of CNN-SAC with non-reinforcement learning-based models, we benchmark this algorithm against the IEMS proposed in Chapter 3. We compare the performance of these methods for the same weather conditions as the ones reported in our previous investigation. The results of these scenarios are presented in **Table 4.8**. The CNN-SAC algorithm outperforms IEMS in the case of two consecutive sunny days as it achieves a 24% lower PI_1 while obtaining a similar rejection percentage and net specific energy consumption. In the case of ten consecutive days, the CNN-SAC model decreases the first performance indicator from 0.835 kWh/m^3 to 0.616 kWh/m^3, accomplishing 26% improvement compared to IEMS. However, in the second scenario, IEMS achieves a slightly lower (roughly 10%) PI_1 than CNN-SAC, which can be attributed to the lower net specific energy consumption obtained by this benchmark method. It is worth mentioning that IEMS is not a simple baseline model but rather an extremely powerful energy management system that effectively exploits forecasted PV power data to optimize the hybrid system's performance. Hence, it can be concluded that the CNN-SAC algorithm can outperform IEMS in most cases and achieve comparable performance in low-PV-power scenarios.

Table 4.8. Comparison of simulation results between CNN-SAC and IEMS

Weather	Method	$PI_1\ \left[\frac{kWh}{m^3}\right]$	PI_2	$PI_3\ \left[\frac{kWh}{m^3}\right]$
Sunny	CNN-SAC	0.389	96.408	3.158
	IEMS	0.514	96.920	3.133
Cloudy	CNN-SAC	1.999	94.659	2.767
	IEMS	1.818	95.130	2.587
10 days	CNN-SAC	0.616	95.450	2.805
	IEMS	0.835	96.080	2.839

Additionally, to examine the consistency and robustness of the CNN-SAC algorithm in all seasons, we run a separate simulation for each season entirely. **Table 4.9** details the results of these simulations. The proposed method achieved the lowest PI_1 and highest contaminant removal efficiency in spring. In this season, the supplied power from the external grid is lower than in other seasons, while the highest total produced water belongs to this case. This performance has led to a quite noticeable difference between the first performance indicator acquired in spring and other seasons. A likely explanation is the different levels of solar power intermittency in spring compared to other seasons. The fluctuations of solar power in spring are lower than in other cases, making the exploitation of solar power much easier for the CNN-SAC algorithm. The first, second, and third performance indicators obtained in summer, fall, and winter demonstrates that the proposed method has almost identical performance in these cases. The maximum difference between PI_1 of the last three seasons is 20% that takes place between PI_1 of fall and winter, which can be justified by the different meteorological conditions of these seasons. An additional point we can notice in **Table 4.9** is the considerable difference between PI_4 obtained in summer and other seasons. This performance can be attributed to the fact that summer had the highest solar power fluctuations among the rest of the seasons, which has resulted in the worst performance in terms of the utilization of solar power.

Table 4.9. Performance of CNN-SAC in different seasons

Season	$P_{Grid}\ [kWh]$	$Q_p\ [m^3]$	$PI_1\ \left[\frac{kWh}{m^3}\right]$	PI_2	$PI_3\ \left[\frac{kWh}{m^3}\right]$	$PI_4\ [kWh]$
Spring	190.920	986.614	0.194	96.245	3.057	69.744
Summer	299.947	985.723	0.304	96.211	3.014	111.676
Fall	310.708	922.032	0.337	95.980	2.978	60.811
Winter	237.411	843.575	0.281	95.588	2.840	40.293

4.2.3 Ablation Study

To analyze the contribution of the 1-D CNN introduced to the SAC algorithm and the importance of providing a history of PV data, we further examine the learning curve of CNN-SAC and vanilla SAC. Moreover, we train the vanilla SAC algorithm with the historical PV data utilized for training the CNN-SAC model in order to observe the essential role that the 1-D CNN plays in extracting features from the PV history data. The learning curve of these methods is depicted in **Figure 4.6(a)**. As can be observed, the total rewards of the SAC algorithm, whose input is the historical PV data, have a sudden increase during the first stages of training and reach a plateau after approximately 200,000 timesteps. The overall average accumulated reward of this model is roughly 1044, showing a 104% improvement with respect to the vanilla SAC method. This finding reveals that by providing a history of PV data, the performance of the SAC algorithm is enhanced substantially, indicating that our reinforcement learning problem is a POMDP as the SAC algorithm utilizes the knowledge of previous observations to create a more accurate depiction of the system's true state. Therefore, it is necessary to provide a history of PV data to tackle the partial observability dilemma. Despite the considerable improvement obtained by using the historical PV data, the final scores of the CNN-SAC algorithm are incomparable to this model, which can be attributed to the added 1-D CNN. The convolutional layers of the CNN-SAC neural networks have a crucial role in extracting essential information from the PV power time series that utterly distinguishes this algorithm from other models. Hence, it is of great significance to exploit an appropriate feature extractor to take advantage of the information hidden in the PV power time series.

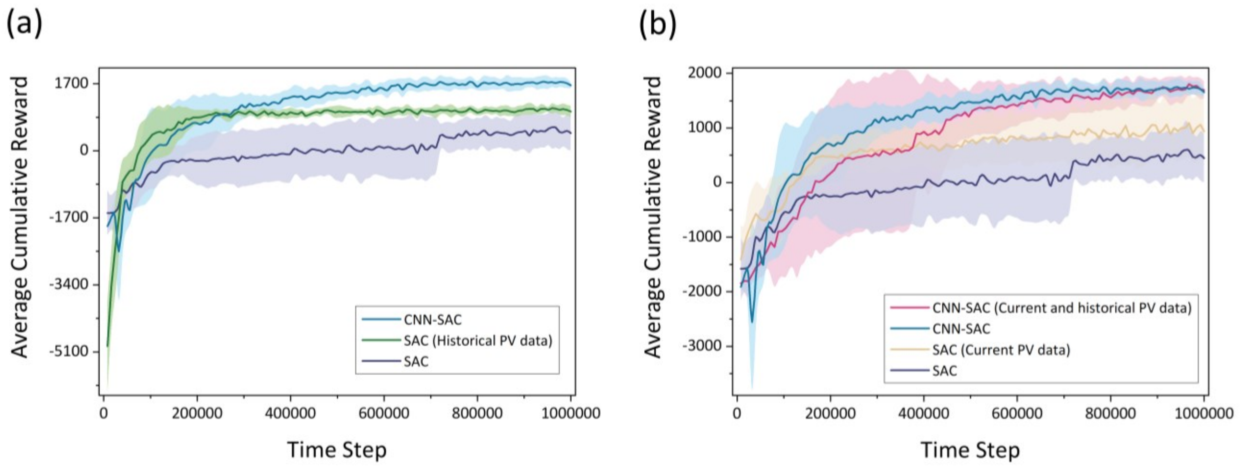

Figure 4.6. Ablation study: (a) Comparing learning curve of CNN-SAC, vanilla SAC, and vanilla SAC with historical PV data, (b) Comparing CNN-SAC and vanilla SAC with the case in which current PV power is provided

Furthermore, we explore the effects of not providing the current output power of the PV system. Specifically, we train the CNN-SAC and vanilla SAC algorithms in conditions where these models have access to the current PV power and compare their learning curve with that of regular CNN-SAC and vanilla SAC. The learning curve of these methods is illustrated in **Figure 4.6(b)**. The CNN-SAC method, whose input is the current and historical PV data, initially has lower scores than CNN-SAC. However, during the last stages of training, its learning curve lies on that of CNN-SAC, achieving an almost identical overall average accumulated reward to the one obtained by CNN-SAC. This result demonstrates that the 1-D CNN used in the function approximators of the CNN-SAC algorithm is perfectly capable of compensating for the lack of information about the current output of the PV system since the final performance of CNN-SAC with and without current PV data are identical. By comparing the learning curve of the SAC methods, the importance of current PV data can be studied. As shown in **Figure 4.6(b)**, the average cumulative reward of the SAC algorithm that has access to the current PV data is always higher than that of regular SAC throughout the training process. Under these conditions, the overall average accumulated reward is improved by 92%, indicating that the current output of the PV system is rather influential in the final performance and learning speed of the SAC algorithm. Moreover, since the average cumulative reward of CNN-SAC is much higher than that of SAC with current PV data, it can be deduced that even by having access to the current output of the PV system, the SAC model cannot compete with CNN-SAC that does not have access to that information. This again reveals that this problem is a POMDP and requires a history of previous observations to improve its performance.

4.3 Conclusion

This study investigated the energy management of the hybrid PV-RO-PRO system using a modified version of the SAC algorithm. We formulated the energy management problem as a POMDP and introduced 1-D CNNs to the function approximators of this model to cope with the partial observability involved in the problem. To examine the effectiveness of the proposed CNN-SAC algorithm, we benchmarked the learning performance of this model against four actor-critic algorithms: DDPG, PPO, vanilla SAC, and TD3. Also, the same 1-D CNN used for CNN-SAC was added to the actor and critic networks of the TD3 algorithm to design a more powerful baseline method. The evaluations made in the final stages of training demonstrated that the CNN-SAC algorithm had the best performance in terms of overall average accumulated reward. Moreover, the simulation results of the proposed model obtained in three different case studies were compared with those of the benchmark methods. The CNN-SAC algorithm exhibited the best performance in terms of exploitation of solar power, minimization of supplied power from the external grid, and managing the operation of the energy storage system and outperformed the baseline methods in almost all aspects. A comparison between the CNN-SAC model and the energy management system proposed in Chapter 3 was made. The results demonstrated that CNN-SAC outperformed IEMS in most cases and achieved comparable performance in low-PV-power conditions. The ablation study revealed that the introduced 1-D CNN could extract essential information from the PV power time series to tackle the partial observability, compensate for the lack of information about the current output power of the PV system, and ultimately enhance the SAC algorithm performance. It was also substantiated that even by providing the SAC algorithm with the current output power of the PV system, the SAC algorithm still required access to a history of PV data to improve its performance, indicating that this problem was a POMDP. This study shows the high promise of reinforcement learning in the energy management of water desalination plants since the proposed model could comprehend the complex structure of the RO-PRO system and determine the optimum operating conditions almost instantly.

5 Conclusions and Future Work

5.1 Summary of Contributions and Results

In this thesis, we explored the optimum energy management of a hybrid grid-connected desalination plant comprised of PV, RO, and PRO. Although the energy management of energy distribution systems and microgrids have been investigated extensively, energy management of PV-powered RO desalination systems has received less attention, especially when it comes to designing reinforcement learning-based energy management systems. We developed two AI-powered methods to solve the PV-RO-PRO multi-objective optimization problem consisting of three objectives: minimization of supplied power from the external grid, maximization of potable water production, and maximization of total rejection percentage.

In Chapter 3, an intelligent energy management system based on deep learning techniques and the PSO algorithm was developed. Three DNNs using the VMD technique, CNNs, and LSTM networks were designed to perform 5-step-ahead PV power forecasting. The defined error metrics indicated that the VMD-CNN neural network has the best accuracy among the designed models. The proposed IEMS incorporated the predicted PV power for the next five hours into the optimization process performed by the PSO algorithm to find the optimum operating conditions of the hybrid system at each timestep. The simulation results revealed that the incorporation of forecasted PV power data into the optimization process could improve the ratio of the total supplied power from the main grid to the total produced water by 43%, showing the importance of knowledge of future PV power data for optimum power scheduling. The effects of forecast errors of the VMD-CNN model on the optimization process were also studied, and it was found that the errors had a minor impact on the direction of the optimization, indicating that the VMD-CNN network was a suitable choice for solar power generation forecasting. Moreover, the impact of forecast horizon on the effectiveness of IEMS was investigated. The results demonstrated that by increasing the forecast horizon, better performance could be achieved. However, there is a caveat to this conclusion that a higher forecast horizon deteriorates the forecasting accuracy inevitably, which can severely degrade the optimization performance.

In Chapter 4, we developed a model-free deep reinforcement learning-accelerated energy management system to optimize the performance of the PV-RO-PRO system. Model-free algorithms are a subset of reinforcement learning algorithms that do not require any model of the system, meaning that they are capable of perceiving the structure of the system without utilizing any model or prior knowledge about the uncertain parameters. As a result, these algorithms eliminate the necessity for a forecasting model to predict solar power generation. Moreover, after the training process, these models can determine the optimum or near-optimum actions within several milliseconds, making them an efficient and powerful technique for real-time control problems. The SAC algorithm was used as the core of the energy management system, and the architecture of its function approximators was modified to cope with the partial observability dilemma caused by the PV system. Specifically, we introduced 1-D CNNs to the actor, critic, and value networks of the SAC algorithm to simplify the feature extraction process of PV power time series performed by these networks and provide them with a more accurate depiction of the system's true state. The proposed CNN-SAC algorithm was benchmarked against five deep reinforcement learning models: DDPG, PPO, vanilla SAC, vanilla TD3, and CNN-TD3. The training results demonstrated that by providing the vanilla SAC algorithm with historical PV data and utilizing 1-D CNNs to analyze the time series, the overall average accumulated reward and standard deviation of the last ten evaluation scores improved by 237% and 68%, respectively. The simulation results revealed that although CNN-SAC did not have access to information regarding the current output power of the PV system, it could forecast the distribution of solar power and determine the optimum operating conditions accordingly. By studying three case studies, it was concluded CNN-SAC has the best performance in terms of efficient exploitation of solar energy and power scheduling compared to the baseline methods. The comparison between the simulation results of CNN-SAC and IEMS demonstrated that the CNN-SAC algorithm could outperform IEMS in most scenarios and achieve similar performance in low-PV-power conditions. The ablation study that we carried out revealed that the energy management problem of the PV-RO-PRO system was a POMDP, and it was necessary to provide the algorithm with historical PV data and utilize advanced methods to analyze that information. Also, it was found that even by having access to information about the current output power of the PV system, the SAC algorithm still needed the historical PV data to improve its performance.

5.2 Future Research Directions

One of the hyperparameters of the proposed CNN-SAC algorithm is the history length considered for the PV power time series. This parameter controls the information received by the CNN-SAC model regarding the previous output power values of the PV system. Consequently, the value of this hyperparameter can significantly affect the learning performance of the proposed algorithm. Hence, further investigation on the effects of the history length is recommended to be carried out in order to make sure that information sent to the CNN-SAC algorithm is neither limited nor too much that complicates the interpretation of historical PV data and degrades the model's performance.

Moreover, the reinforcement learning algorithm that we utilized in Chapter 4 is a centralized algorithm, meaning that a single agent controls devices of the hybrid system. This control architecture can become problematic if the communication line between the devices and the agent fails. To solve this problem, multi-agent deep reinforcement learning algorithms can be exploited. In these algorithms, a group of agents, which share a common environment, interact with each other and the environment to improve their own policy. The multiple agents can interact with each other either in cooperative or competitive settings. Since the agents' policy changes during the training process, the environment becomes non-stationary from the perspective of each individual agent, causing learning stability challenges. Consequently, further investigation is required to effectively implement these algorithms for solving the energy management problem.

References

[1] P. Karami, B. Khorshidi, L. Shamaei, E. Beaulieu, J. B. P. Soares, and M. Sadrzadeh, "Nanodiamond-Enabled Thin-Film Nanocomposite Polyamide Membranes for High-Temperature Water Treatment," *ACS Appl. Mater. Interfaces*, vol. 0, no. 0, p. null, doi: 10.1021/acsami.0c15194.

[2] M. A. Abdelkareem, M. El Haj Assad, E. T. Sayed, and B. Soudan, "Recent progress in the use of renewable energy sources to power water desalination plants," *Desalination*, vol. 435, pp. 97–113, 2018, doi: https://doi.org/10.1016/j.desal.2017.11.018.

[3] L. Shamaei, B. Khorshidi, M. A. Islam, and M. Sadrzadeh, "Development of antifouling membranes using agro-industrial waste lignin for the treatment of Canada's oil sands produced water," *J. Memb. Sci.*, vol. 611, p. 118326, 2020, doi: https://doi.org/10.1016/j.memsci.2020.118326.

[4] Y. Liu *et al.*, "Global water use associated with energy supply, demand and international trade of China," *Appl. Energy*, vol. 257, p. 113992, 2020, doi: https://doi.org/10.1016/j.apenergy.2019.113992.

[5] E. Jones, M. Qadir, M. T. H. van Vliet, V. Smakhtin, and S. Kang, "The state of desalination and brine production: A global outlook," *Sci. Total Environ.*, vol. 657, pp. 1343–1356, 2019, doi: https://doi.org/10.1016/j.scitotenv.2018.12.076.

[6] M. W. Shahzad, M. Burhan, L. Ang, and K. C. Ng, "Energy-water-environment nexus underpinning future desalination sustainability," *Desalination*, vol. 413, pp. 52–64, 2017, doi: https://doi.org/10.1016/j.desal.2017.03.009.

[7] M. Qasim, M. Badrelzaman, N. N. Darwish, N. A. Darwish, and N. Hilal, "Reverse osmosis desalination: A state-of-the-art review," *Desalination*, vol. 459, pp. 59–104, 2019, doi: https://doi.org/10.1016/j.desal.2019.02.008.

[8] B. Gu, D. Y. Kim, J. H. Kim, and D. R. Yang, "Mathematical model of flat sheet membrane modules for FO process: Plate-and-frame module and spiral-wound module," *J. Memb. Sci.*,

vol. 379, no. 1, pp. 403–415, 2011, doi: https://doi.org/10.1016/j.memsci.2011.06.012.

[9] A. Al-Karaghouli and L. L. Kazmerski, “Energy consumption and water production cost of conventional and renewable-energy-powered desalination processes,” *Renew. Sustain. Energy Rev.*, vol. 24, pp. 343–356, 2013, doi: https://doi.org/10.1016/j.rser.2012.12.064.

[10] J. Kim, K. Park, D. R. Yang, and S. Hong, “A comprehensive review of energy consumption of seawater reverse osmosis desalination plants,” *Appl. Energy*, vol. 254, p. 113652, 2019, doi: https://doi.org/10.1016/j.apenergy.2019.113652.

[11] A. Altaee, G. J. Millar, and G. Zaragoza, “Integration and optimization of pressure retarded osmosis with reverse osmosis for power generation and high efficiency desalination,” *Energy*, vol. 103, pp. 110–118, 2016, doi: https://doi.org/10.1016/j.energy.2016.02.116.

[12] W. K. Biswas and P. Yek, “Improving the carbon footprint of water treatment with renewable energy: a Western Australian case study,” *Renewables Wind. Water, Sol.*, vol. 3, no. 1, 2016, doi: 10.1186/s40807-016-0036-2.

[13] P. K. Cornejo, M. V. E. Santana, D. R. Hokanson, J. R. Mihelcic, and Q. Zhang, “Carbon footprint of water reuse and desalination: A review of greenhouse gas emissions and estimation tools,” *J. Water Reuse Desalin.*, vol. 4, no. 4, pp. 238–252, 2014, doi: 10.2166/wrd.2014.058.

[14] D. P. Clarke, Y. M. Al-Abdeli, and G. Kothapalli, “Multi-objective optimisation of renewable hybrid energy systems with desalination,” *Energy*, vol. 88, pp. 457–468, 2015, doi: https://doi.org/10.1016/j.energy.2015.05.065.

[15] B. Wu, A. Maleki, F. Pourfayaz, and M. A. Rosen, “Optimal design of stand-alone reverse osmosis desalination driven by a photovoltaic and diesel generator hybrid system,” *Sol. Energy*, vol. 163, pp. 91–103, 2018, doi: https://doi.org/10.1016/j.solener.2018.01.016.

[16] H. Cherif and J. Belhadj, “Large-scale time evaluation for energy estimation of stand-alone hybrid photovoltaic–wind system feeding a reverse osmosis desalination unit,” *Energy*, vol. 36, no. 10, pp. 6058–6067, 2011, doi: https://doi.org/10.1016/j.energy.2011.08.010.

[17] S. Lee, S. Myung, J. Hong, and D. Har, “Reverse osmosis desalination process optimized

for maximum permeate production with renewable energy," *Desalination*, vol. 398, pp. 133–143, 2016, doi: https://doi.org/10.1016/j.desal.2016.07.018.

[18] N. Ahmad, A. K. Sheikh, P. Gandhidasan, and M. Elshafie, "Modeling, simulation and performance evaluation of a community scale PVRO water desalination system operated by fixed and tracking PV panels: A case study for Dhahran city, Saudi Arabia," *Renew. Energy*, vol. 75, pp. 433–447, 2015, doi: https://doi.org/10.1016/j.renene.2014.10.023.

[19] M. Monnot, G. D. M. Carvajal, S. Laborie, C. Cabassud, and R. Lebrun, "Integrated approach in eco-design strategy for small RO desalination plants powered by photovoltaic energy," *Desalination*, vol. 435, pp. 246–258, 2018, doi: https://doi.org/10.1016/j.desal.2017.05.015.

[20] A. T. D. Perera and P. Kamalaruban, "Applications of reinforcement learning in energy systems," *Renew. Sustain. Energy Rev.*, vol. 137, p. 110618, 2021, doi: https://doi.org/10.1016/j.rser.2020.110618.

[21] K. Hussey and J. Pittock, "The Energy–Water Nexus: Managing the Links between Energy and Water for a Sustainable Future," *Ecol. Soc.*, vol. 17, no. 1, 2012.

[22] P. D'Odorico *et al.*, "The Global Food-Energy-Water Nexus," *Rev. Geophys.*, vol. 56, no. 3, pp. 456–531, 2018, doi: https://doi.org/10.1029/2017RG000591.

[23] P. Zeng, H. Li, H. He, and S. Li, "Dynamic Energy Management of a Microgrid Using Approximate Dynamic Programming and Deep Recurrent Neural Network Learning," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 4435–4445, 2019, doi: 10.1109/TSG.2018.2859821.

[24] M. Elsied, A. Oukaour, T. Youssef, H. Gualous, and O. Mohammed, "An advanced real time energy management system for microgrids," *Energy*, vol. 114, pp. 742–752, 2016, doi: https://doi.org/10.1016/j.energy.2016.08.048.

[25] G. K. Venayagamoorthy, R. K. Sharma, P. K. Gautam, and A. Ahmadi, "Dynamic Energy Management System for a Smart Microgrid," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 27, no. 8, pp. 1643–1656, 2016, doi: 10.1109/TNNLS.2016.2514358.

[26] M. S. Alam and S. A. Arefifar, "Energy Management in Power Distribution Systems:

Review, Classification, Limitations and Challenges," *IEEE Access*, vol. 7, pp. 92979–93001, 2019, doi: 10.1109/ACCESS.2019.2927303.

[27] M. F. Roslan, M. A. Hannan, P. J. Ker, and M. N. Uddin, "Microgrid control methods toward achieving sustainable energy management," *Appl. Energy*, vol. 240, no. October 2018, pp. 583–607, 2019, doi: 10.1016/j.apenergy.2019.02.070.

[28] M. F. Zia, E. Elbouchikhi, and M. Benbouzid, "Microgrids energy management systems: A critical review on methods, solutions, and prospects," *Appl. Energy*, vol. 222, pp. 1033–1055, 2018, doi: https://doi.org/10.1016/j.apenergy.2018.04.103.

[29] A. Chaouachi, R. M. Kamel, R. Andoulsi, and K. Nagasaka, "Multiobjective Intelligent Energy Management for a Microgrid," *IEEE Trans. Ind. Electron.*, vol. 60, no. 4, pp. 1688–1699, 2013, doi: 10.1109/TIE.2012.2188873.

[30] B. Blankert, Y. Kim, H. Vrouwenvelder, and N. Ghaffour, "Facultative hybrid RO-PRO concept to improve economic performance of PRO: Feasibility and maximizing efficiency," *Desalination*, vol. 478, p. 114268, 2020, doi: https://doi.org/10.1016/j.desal.2019.114268.

[31] W. He, Y. Wang, and M. H. Shaheed, "Stand-alone seawater RO (reverse osmosis) desalination powered by PV (photovoltaic) and PRO (pressure retarded osmosis)," *Energy*, vol. 86, pp. 423–435, 2015, doi: https://doi.org/10.1016/j.energy.2015.04.046.

[32] D. Attarde, M. Jain, and S. K. Gupta, "Modeling of a forward osmosis and a pressure-retarded osmosis spiral wound module using the Spiegler-Kedem model and experimental validation," *Sep. Purif. Technol.*, vol. 164, pp. 182–197, 2016, doi: https://doi.org/10.1016/j.seppur.2016.03.039.

[33] S. Sundaramoorthy, G. Srinivasan, and D. V. R. Murthy, "An analytical model for spiral wound reverse osmosis membrane modules: Part I — Model development and parameter estimation," *Desalination*, vol. 280, no. 1, pp. 403–411, 2011, doi: https://doi.org/10.1016/j.desal.2011.03.047.

[34] M. A. Al Mamun, S. Bhattacharjee, D. Pernitsky, and M. Sadrzadeh, "Colloidal fouling of nanofiltration membranes: Development of a standard operating procedure," *Membranes (Basel).*, vol. 7, no. 1, 2017, doi: 10.3390/membranes7010004.

[35] S. Sarp, Z. Li, and J. Saththasivam, “Pressure Retarded Osmosis (PRO): Past experiences, current developments, and future prospects,” *Desalination*, vol. 389, pp. 2–14, 2016, doi: https://doi.org/10.1016/j.desal.2015.12.008.

[36] Z. L. Cheng and T.-S. Chung, “Mass transport of various membrane configurations in pressure retarded osmosis (PRO),” *J. Memb. Sci.*, vol. 537, pp. 160–176, 2017, doi: https://doi.org/10.1016/j.memsci.2017.05.008.

[37] J. Y. Xiong, D. J. Cai, Q. Y. Chong, S. H. Lee, and T. S. Chung, “Osmotic power generation by inner selective hollow fiber membranes: An investigation of thermodynamics, mass transfer, and module scale modelling,” *J. Memb. Sci.*, vol. 526, no. November 2016, pp. 417–428, 2017, doi: 10.1016/j.memsci.2016.12.056.

[38] C. F. Wan and T.-S. Chung, “Osmotic power generation by pressure retarded osmosis using seawater brine as the draw solution and wastewater retentate as the feed,” *J. Memb. Sci.*, vol. 479, pp. 148–158, 2015, doi: https://doi.org/10.1016/j.memsci.2014.12.036.

[39] Z. L. Cheng, X. Li, Y. Feng, C. F. Wan, and T.-S. Chung, “Tuning water content in polymer dopes to boost the performance of outer-selective thin-film composite (TFC) hollow fiber membranes for osmotic power generation,” *J. Memb. Sci.*, vol. 524, pp. 97–107, 2017, doi: https://doi.org/10.1016/j.memsci.2016.11.009.

[40] J. Jeong and H. Kim, “DeepComp: Deep reinforcement learning based renewable energy error compensable forecasting,” *Appl. Energy*, vol. 294, p. 116970, 2021, doi: https://doi.org/10.1016/j.apenergy.2021.116970.

[41] C. Chen, S. Duan, T. Cai, B. Liu, and G. Hu, “Smart energy management system for optimal microgrid economic operation,” *IET Renew. Power Gener.*, vol. 5, no. 3, pp. 258–267, May 2011, doi: 10.1049/iet-rpg.2010.0052.

[42] K. Dragomiretskiy and D. Zosso, “Variational mode decomposition,” *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 531–544, 2014, doi: 10.1109/TSP.2013.2288675.

[43] H. Zang, L. Cheng, T. Ding, K. W. Cheung, Z. Wei, and G. Sun, “Day-ahead photovoltaic power forecasting approach based on deep convolutional neural networks and meta learning,” *Int. J. Electr. Power Energy Syst.*, vol. 118, no. December 2019, 2020, doi:

10.1016/j.ijepes.2019.105790.

[44] N. Aloysius and M. Geetha, “A review on deep convolutional neural networks,” *Proc. 2017 IEEE Int. Conf. Commun. Signal Process. ICCSP 2017*, vol. 2018-Janua, pp. 588–592, 2018, doi: 10.1109/ICCSP.2017.8286426.

[45] K. Wang, X. Qi, and H. Liu, “Photovoltaic power forecasting based LSTM-Convolutional Network,” *Energy*, vol. 189, 2019, doi: 10.1016/j.energy.2019.116225.

[46] K. Wang, X. Qi, and H. Liu, “A comparison of day-ahead photovoltaic power forecasting models based on deep learning neural network,” *Appl. Energy*, vol. 251, no. November 2018, p. 113315, 2019, doi: 10.1016/j.apenergy.2019.113315.

[47] H. Zhou, Y. Zhang, L. Yang, Q. Liu, K. Yan, and Y. Du, “Short-Term photovoltaic power forecasting based on long short term memory neural network and attention mechanism,” *IEEE Access*, vol. 7, pp. 78063–78074, 2019, doi: 10.1109/ACCESS.2019.2923006.

[48] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[49] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, 1995, vol. 4, pp. 1942–1948 vol.4, doi: 10.1109/ICNN.1995.488968.

[50] F. van den Bergh and A. P. Engelbrecht, “A study of particle swarm optimization particle trajectories,” *Inf. Sci. (Ny).*, vol. 176, no. 8, pp. 937–971, 2006, doi: https://doi.org/10.1016/j.ins.2005.02.003.

[51] Eberhart and Yuhui Shi, “Particle swarm optimization: developments, applications and resources,” in *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, 2001, vol. 1, pp. 81–86 vol. 1, doi: 10.1109/CEC.2001.934374.

[52] G. Pinto, M. S. Piscitelli, J. R. Vázquez-Canteli, Z. Nagy, and A. Capozzoli, “Coordinated energy management for a cluster of buildings through deep reinforcement learning,” *Energy*, vol. 229, p. 120725, 2021, doi: https://doi.org/10.1016/j.energy.2021.120725.

[53] L. Yu *et al.*, “Deep Reinforcement Learning for Smart Home Energy Management,” *IEEE*

Internet Things J., vol. 7, no. 4, pp. 2751–2762, 2020, doi: 10.1109/JIOT.2019.2957289.

[54] O. Dogru, R. Chiplunkar, and B. Huang, "Reinforcement Learning with Constrained Uncertain Reward Function Through Particle Filtering," *IEEE Trans. Ind. Electron.*, p. 1, 2021, doi: 10.1109/TIE.2021.3099234.

[55] I. Ben Ali, M. Turki, J. Belhadj, and X. Roboam, "Optimized fuzzy rule-based energy management for a battery-less PV/wind-BWRO desalination system," *Energy*, vol. 159, pp. 216–228, 2018, doi: https://doi.org/10.1016/j.energy.2018.06.110.

[56] R. Xavier, S. Bruno, N. D. Trung, and B. Jamel, "Optimal System Management of a Water Pumping and Desalination Process Supplied with Intermittent Renewable Sources," *IFAC Proc. Vol.*, vol. 45, no. 21, pp. 369–374, 2012, doi: https://doi.org/10.3182/20120902-4-FR-2032.00066.

[57] G. Kyriakarakos, A. I. Dounis, K. G. Arvanitis, and G. Papadakis, "A fuzzy logic energy management system for polygeneration microgrids," *Renew. Energy*, vol. 41, pp. 315–327, 2012, doi: https://doi.org/10.1016/j.renene.2011.11.019.

[58] G. Kyriakarakos, A. I. Dounis, K. G. Arvanitis, and G. Papadakis, "Design of a Fuzzy Cognitive Maps variable-load energy management system for autonomous PV-reverse osmosis desalination systems: A simulation survey," *Appl. Energy*, vol. 187, pp. 575–584, 2017, doi: https://doi.org/10.1016/j.apenergy.2016.11.077.

[59] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014, doi: https://doi.org/10.1016/j.advengsoft.2013.12.007.

[60] D. Whitley, "A genetic algorithm tutorial," *Stat. Comput.*, vol. 4, no. 2, pp. 65–85, Jun. 1994, doi: 10.1007/BF00175354.

[61] D. Zhang, X. Han, and C. Deng, "Review on the research and practice of deep learning and reinforcement learning in smart grids," *CSEE J. Power Energy Syst.*, vol. 4, no. 3, pp. 362–370, 2018, doi: 10.17775/CSEEJPES.2018.00520.

[62] T. Yang, L. Zhao, W. Li, and A. Y. Zomaya, "Reinforcement learning in sustainable energy and electric systems: a survey," *Annu. Rev. Control*, vol. 49, pp. 145–163, 2020, doi:

https://doi.org/10.1016/j.arcontrol.2020.03.001.

[63] Y. Ye, D. Qiu, H. Wang, Y. Tang, and G. Strbac, “Real-Time Autonomous Residential Demand Response Management Based on Twin Delayed Deep Deterministic Policy Gradient Learning,” *Energies*, vol. 14, no. 3, 2021, doi: 10.3390/en14030531.

[64] D. J. B. Harrold, J. Cao, and Z. Fan, “Data-driven battery operation for energy arbitrage using rainbow deep reinforcement learning,” *Energy*, vol. 238, p. 121958, 2022, doi: https://doi.org/10.1016/j.energy.2021.121958.

[65] P. Kofinas, A. I. Dounis, and G. A. Vouros, “Fuzzy Q-Learning for multi-agent decentralized energy management in microgrids,” *Appl. Energy*, vol. 219, pp. 53–67, 2018, doi: https://doi.org/10.1016/j.apenergy.2018.03.017.

[66] G. Zhang *et al.*, “Data-driven optimal energy management for a wind-solar-diesel-battery-reverse osmosis hybrid energy system using a deep reinforcement learning approach,” *Energy Convers. Manag.*, vol. 227, p. 113608, 2021, doi: https://doi.org/10.1016/j.enconman.2020.113608.

[67] L. Lei, Y. Tan, G. Dahlenburg, W. Xiang, and K. Zheng, “Dynamic Energy Dispatch Based on Deep Reinforcement Learning in IoT-Driven Smart Isolated Microgrids,” *IEEE Internet Things J.*, vol. 8, no. 10, pp. 7938–7953, 2021, doi: 10.1109/JIOT.2020.3042007.

[68] S. Senthilmurugan, A. Ahluwalia, and S. K. Gupta, “Modeling of a spiral-wound module and estimation of model parameters using numerical techniques,” *Desalination*, vol. 173, no. 3, pp. 269–286, 2005, doi: https://doi.org/10.1016/j.desal.2004.08.034.

[69] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor,” in *Proceedings of the 35th International Conference on Machine Learning*, 2018, vol. 80, pp. 1861–1870.

[70] S. Fujimoto, H. van Hoof, and D. Meger, “Addressing Function Approximation Error in Actor-Critic Methods,” in *Proceedings of the 35th International Conference on Machine Learning*, 2018, vol. 80, pp. 1587–1596.

[71] T. P. Lillicrap *et al.*, “Continuous control with deep reinforcement learning,” *4th Int. Conf.*

Learn. Represent. ICLR 2016 - Conf. Track Proc., 2016.

[72] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal Policy Optimization Algorithms.” 2017.

[73] J. L. Prante, J. A. Ruskowitz, A. E. Childress, and A. Achilli, “RO-PRO desalination: An integrated low-energy approach to seawater desalination,” *Appl. Energy*, vol. 120, pp. 104–114, 2014, doi: https://doi.org/10.1016/j.apenergy.2014.01.013.

[74] U. K. Das *et al.*, “Forecasting of photovoltaic power generation and model optimization: A review,” *Renew. Sustain. Energy Rev.*, vol. 81, pp. 912–928, 2018, doi: https://doi.org/10.1016/j.rser.2017.08.017.

[75] “Desert Knowledge Australia Centre. Download Data: Kaneka, 6.0kW, Amorphous Silicon, Fixed, 2008. Alice Springs.” [Online]. Available: http://dkasolarcentre.com.au/historical-data/download, date accessed: 01/10/2020.

[76] I. Koprinska, D. Wu, and Z. Wang, “Convolutional Neural Networks for Energy Time Series Forecasting,” *Proc. Int. Jt. Conf. Neural Networks*, vol. 2018-July, 2018, doi: 10.1109/IJCNN.2018.8489399.

[77] H. Zang *et al.*, “Hybrid method for short-term photovoltaic power forecasting based on deep convolutional neural network,” *IET Gener. Transm. Distrib.*, vol. 12, no. 20, pp. 4557–4567, 2018, doi: 10.1049/iet-gtd.2018.5847.

[78] R. Hegger, H. Kantz, and T. Schreiber, “Practical implementation of nonlinear time series methods: The TISEAN package,” *Chaos*, vol. 9, no. 2, pp. 413–435, 1999, doi: 10.1063/1.166424.

[79] S. Sundaramoorthy, G. Srinivasan, and D. V. R. Murthy, “Reprint of: ‘An analytical model for spiral wound reverse osmosis membrane modules: Part II — Experimental validation,’” *Desalination*, vol. 280, no. 1, pp. 432–439, 2011, doi: https://doi.org/10.1016/j.desal.2011.08.008.

[80] M. L. Crowder and C. H. Gooding, “Spiral wound, hollow fiber membrane modules: A new approach to higher mass transfer efficiency,” *J. Memb. Sci.*, vol. 137, no. 1, pp. 17–29, 1997,

doi: https://doi.org/10.1016/S0376-7388(97)00174-9.

[81] S. C. Chen, C. F. Wan, and T.-S. Chung, "Enhanced fouling by inorganic and organic foulants on pressure retarded osmosis (PRO) hollow fiber membranes under high pressures," *J. Memb. Sci.*, vol. 479, pp. 190–203, 2015, doi: https://doi.org/10.1016/j.memsci.2015.01.037.

[82] G. Chiandussi, M. Codegone, S. Ferrero, and F. E. Varesio, "Comparison of multi-objective optimization methodologies for engineering applications," *Comput. Math. with Appl.*, vol. 63, no. 5, pp. 912–942, 2012, doi: https://doi.org/10.1016/j.camwa.2011.11.057.

[83] L. Lei, Y. Tan, K. Zheng, S. Liu, K. Zhang, and X. Shen, "Deep Reinforcement Learning for Autonomous Internet of Things: Model, Applications and Challenges," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 3, pp. 1722–1760, 2020, doi: 10.1109/COMST.2020.2988367.

[84] N. Heess, J. J. Hunt, T. P. Lillicrap, and D. Silver, "Memory-based control with recurrent neural networks," *CoRR*, vol. abs/1512.0, 2015.

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

[86] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-To-end encrypted traffic classification with one-dimensional convolution neural networks," *2017 IEEE Int. Conf. Intell. Secur. Informatics Secur. Big Data, ISI 2017*, pp. 43–48, 2017, doi: 10.1109/ISI.2017.8004872.