Data-driven prosumer-centric energy scheduling using convolutional neural networks

Weiqi Hua[a], Jing Jiang[b], Hongjian Sun[c,*], Andrea M. Tonello[d], Meysam Qadrdan[a], Jianzhong Wu[a]

[a]*School of Engineering, Cardi University, Cardi, CF24 3AA, UK*
[b]*Department of Mathematics, Physics and Electrical Engineering, University of Northumbria, Newcastle, NE1 8ST, UK*
[c]*Department of Engineering, Durham University, DH1 3LE, Durham, UK*
[d]*Institute of Networked and Embedded Systems, University of Klagenfurt, Klagenfurt, 9020, Austria*

Abstract

The emerging role of energy prosumers (both producers and consumers) enables a more flexible and localised structure of energy markets. However, it leads to challenges for the energy scheduling of individual prosumers in terms of identifying idiosyncratic pricing patterns, cost-effectively predicting power profiles, and scheduling various scales of generation and consumption sources. To overcome these three challenges, this study proposes a novel data-driven energy scheduling model for an individual prosumer. The pricing patterns of a prosumer are represented by three types of dynamic price elasticities, i.e., the price elasticities of the generation, consumption, and carbon emissions. To improve the computational efficiency and scalability, the heuristic algorithms used to solve the optimisation problems is replaced by the convolutional neural networks which map the pricing patterns to scheduling decisions of a prosumer. The variations of uncertainties caused by the intermittency of renewable energy sources, flexible demand, and dynamic prices are predicted by the developed real-time scenarios selection approach, in which each variation is defined as a scenario. Case studies under various IEEE test distribution systems and uncertain scenarios demonstrate the effectiveness of our proposed energy scheduling model in terms of predicting scheduling decisions in microseconds with high accuracy.

Keywords: convolutional neural networks, data analytics, energy scheduling, prosumers, renewable energy, smart grids.

1. Introduction

The advances of smart grids and smart metering enable increasing number of consumers in distribution networks to produce or store energy using distributed renewable energy sources (RESs) and storage devices, which leads to a new role: energy prosumers [1]. The types of prosumers include the residential, commercial, and industrial users, who actively produce energy on-site, and strategically store energy, or shift/curtail demand in response to the dynamic electricity pricing signals. Although the engagement of prosumers enables the transition of energy markets towards more flexible and localised structures, a number of challenges have drawn attentions for the research in recent years: *1)* How dynamic electricity pricing signals affect the generation and consumption of prosumers; *2)* How to design a reliable energy scheduling model under uncertainties caused by the distributed RESs, flexible demand, and dynamic pricing signals; and *3)* How to exploit substantial volumes of the useful metering data from prosumers' promises for enhancing the energy scheduling model to be scalable for various types of prosumers.

The effects of dynamic electricity pricing signals on the generation and consumption, i.e., pricing patterns of a prosumer, have been well documented in the literature. Venizelou *et al.* [2] investigated the effects of time-of-use pricing signals on a group of prosumers through capturing the peak to off-peak elasticity of these prosumers. Le *et al.* [3] categorised the prosumers as elastic pricing patterns and non-controllable pricing patterns. Based on these two categorises, a transactive mechanism was proposed enabling the non-controllable prosumers to trade their power deviations from schedules with the elastic prosumers. In [4], the pricing patterns categorised by the reliability levels of prosumers were analysed by using the dynamic elasticity yielded from the historical data and real-time preferences of an individual prosumer. While the current works have intensively investigated the pricing patterns by analysing the dynamic elasticities from the metering data of prosumers, there are great opportunities to further investigate how the idiosyncratic pricing patterns of an individual prosumer affect its energy scheduling decisions.

Given limited budgets of the small or medium sized prosumers for their energy predictions [5], recent works have focused on cost-effectively incorporating an uncertain prediction into the energy scheduling model. The uncertainties of an energy prosumer are primarily caused by the intermittency of RESs, flexible demand, and volatility of dynamic electricity prices [6]. Using a set of scenarios is a statistical approach for predicting the variations of uncertain variables [7], by which each variation of an uncertain variable is defined as a scenario. The scenarios are generated from the probabilistic distributions of the historical data of an uncertain variable by using sam-

---
*Corresponding author.
Email addresses: huaw5@cardiff.ac.uk (Weiqi Hua), jing.jiang@northumbria.ac.uk (Jing Jiang), hongjian.sun@durham.ac.uk (Hongjian Sun), andrea.tonello@aau.at (Andrea M. Tonello), qadrdanm@cardiff.ac.uk (Meysam Qadrdan), wuj5@cardiff.ac.uk (Jianzhong Wu)

pling approaches, such as the Monte Carlo simulation and Latin hypercube sampling. Ahn *et al.* [8] examined the impacts of uncertainties caused by the solar generation and demand using both the deterministic model and Monte-Carlo based stochastic model, and found the deterministic model would underestimate the operating costs compared to the stochastic model. Santos *et al.* [9] incorporated the scenarios of the RESs and demand generated by the Monte Carlo simulation into the long-term planning problem. However, the Monte Carlo simulation would raise the computational burdens since the random sampling results in high standard deviations. This can be overcome by the Latin Hypercube sampling through using the space-filling to reduce the standard deviations. Mavromatidis *et al.* [10] implemented the Latin Hypercube sampling based uncertainty analysis into the cost minimisation problem of distributed energy systems. In [11], the uncertainties caused by the RESs, loads, and transaction prices were analysed by using scenarios generated by the Latin Hypercube sampling. These scenarios were considered into a cost minimisation problem of a multi-energy microgrid. Although the research efforts have been dedicated to improving the predicting accuracy of scenarios, the potential of enhancing the adaptability of scenarios to real-time operations of prosumers has not been well explored.

The advanced information and communication technologies of the smart grid and smart energy system enable substantial volumes of useful metering data to be produced from prosumers' promises. Using learning approaches is an efficient way for exploiting this metering data, in order to support the automatic decision making and interoperability of prosumers [12]. The learning approaches can enhance the energy scheduling model of prosumers through *1)* improving the model scalability to adapt multi-source data from various types of prosumers, *2)* reducing the computational complexity for the approaches of using heuristic algorithms to solve the energy scheduling problem, and *3)* extracting key patterns of prosumers, e.g., pricing patterns, from high-dimensional data sources consisting of various time horizons (e.g., hourly or daily) and features (e.g., generation, consumption, or carbon emissions).

For improving the model scalability, learning approaches only require historical data to generate feature representations for various scales of prosumers, without the need of predefined parameters and formulations. In [13], a data-driven urban energy simulation framework was proposed through using the residual neural networks, which is adaptable to multiple spatial and temporal scales in terms of accurately predicting the urban scale energy consumption. *For the reduction of the computational complexity*, the step of solving optimisation problems of the energy scheduling can be assisted or replaced by learning approaches. This is because the learning approaches can map the input parameters to the optimal scheduling decisions in a computation-free manner through learning from the historical data. Yang *et al.* [14] proposed an approximate model predictive control for the energy scheduling of buildings through using a dynamic feedback machine learning model. This approximate model predictive control was proved to significantly reduce the computational burdens of solving the optimisation problem. Mocanu *et al.* [15] combined the reinforcement learn-

ing with the deep learning to perform the online optimisation of the building energy scheduling. *For extracting patterns from high-dimensional data sources*, the deep approximation architectures, e.g., the deep neural networks (DNNs), convolutional neural networks (CNNs), recurrent neural networks (RNNs), and long short term memory (LSTM), can be trained to generalise key patterns through stacking multiple layers of feature representations. These approximated general patterns can be scaled to high-dimensional input spaces. It is particular for the CNNs which are capable of recognising patterns from high-dimensional array inputs, since multiple filters of each convolutional layer can capture the temporal and spatial correlations of input arrays. Claessens *et al.* [5] combined the CNNs with the reinforcement learning to extract the state-time features of residential loads from the high-dimensional data of heterogeneous demand flexibility sources. Du *et al.* [16] designed the CNNs to extract both topological patterns of the power flows and uncertain patterns of RESs from high-dimensional scenarios. Nevertheless, there is still a lack of a model which involves all the aforementioned three functions into the energy scheduling of prosumers, i.e., using the learning approaches to improve the computational efficiency and extract intrinsic features of various scales of prosumers.

This paper proposes a novel data-driven energy scheduling model by using the CNNs to improve the computational efficiency and map the pricing patterns to potential energy scheduling decisions of a prosumer, under various uncertain scenarios. This paper offers the following key contributions:

• A novel approach of pricing patterns processing is developed to analyse the local features, temporal transient features, and the correlation of dynamic price elasticities. By exploiting the pattern recognition capability of the CNNs, how these intrinsic features affect individual prosumers' scheduling decisions is investigated.

• The machine learning approaches are implemented to improve the scalability and computational efficiency of the energy scheduling model compared to solving the optimisation problem by the heuristic algorithms.

• A real-time scenarios selection approach is for the first time developed to adapt scenarios with dynamic operations of prosumers, by which each scenario provides a possible energy profile to be scheduled, and the scheduling decisions provide an update for the scenarios set.

• Case studies verify that the proposed energy scheduling model improves the accuracy of making optimal scheduling decisions with the reduced computational complexity, under various IEEE test systems and uncertain scenarios. The connection between the intrinsic features of dynamic price elasticities and scheduling results is demonstrated.

The rest of this paper is organised as follows: Section 2 introduces an overview framework for implementing the proposed energy scheduling model. The training phase of the energy scheduling model is detailed in Section 3 to describe how to use the uncertain scenarios, pricing patterns, and optimal scheduling decisions for training the neural networks. The deploying phase of the energy scheduling model is detailed in Section 4 to describe the real-time scenarios selection and energy schedul-

ing for an individual prosumer. Section 5 provides case studies under various learning approaches and IEEE test distribution systems to verify the proposed approaches. Section 6 concludes this paper and lists the future work.

2. Framework

The objective of this research is to design an effective data-driven energy scheduling model by using learning approaches to reduce computational complexity and capture pricing patterns of an individual prosumer. The type of a prosumer could be a residential, commercial, or industrial user in distribution networks. The proposed model is used for day-ahead energy scheduling to help a prosumer make optimal decisions for the next day, through analysing the metering data of this prosumer. This section introduces the overall framework of energy scheduling, consisting of a training phase and deploying phase, as presented in Fig. 1.
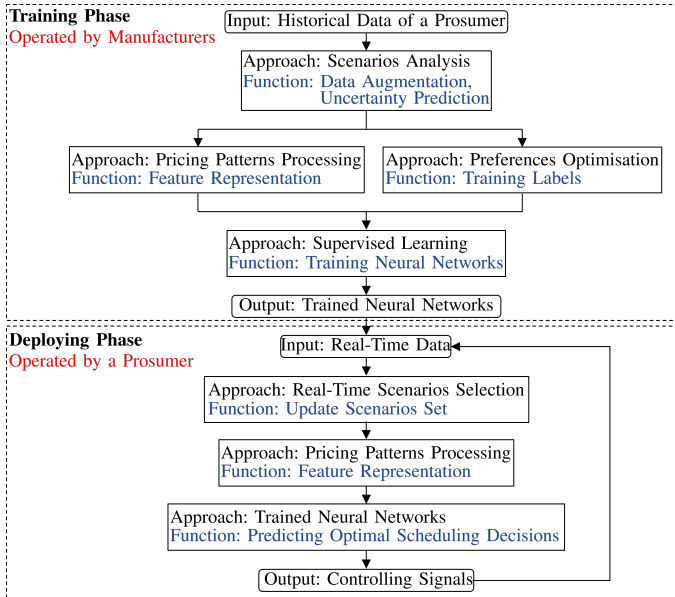

Fig. 1. Framework of the proposed energy scheduling, consisting of a training phase and deploying phase. At the training phase, manufacturers use historical data to train the designed neural networks after data pre-processing, i.e., scenarios analysis, pricing patterns processing, and preferences optimisation. At the deploying phase, a prosumer uses trained neural networks to predict optimal scheduling decisions.

At the training phase, a prosumer's historical data is stored in the data warehouse, and used by the manufacturer of the energy scheduling model to train the neural networks under a supervised learning mode. Uncertainties caused by the intermittency of the distributed RESs, flexible demand, and dynamic prices are analysed by the developed scenarios analysis approach. The intrinsic features of an individual prosumer represented by dynamic price elasticities are processed as elasticity arrays by using the designed pricing patterns processing approach. The uncertain scenarios and elasticity arrays are used as training inputs. The training labels are optimal decision variables yielded from solving the preferences optimisation problems. These preferences are predefined by the users, such as

saving electricity bills, improving comfort levels, or reducing generating costs.

At the deploying phase, the energy scheduling model with trained neural networks is deployed to an individual prosumer's premise. With the real-time metering data from a prosumer, the scheduling model automatically makes optimal scheduling decisions, and sends these optimal decisions to the controller which controls both generators and loads.

3. Training phase of energy scheduling model

In this section, the training phase of the proposed energy scheduling model is discussed. The flowchart of the training phase is shown in Fig. 2, with specific procedures explained as follows:

Step 1 (Scenarios Analysis): The historical metering data of a prosumer is collected and processed by the approach of scenarios analysis for augmenting data samples and predicting potential variations of uncertain variables (details are described in Section 3.1).

Step 2 (Pricing Patterns Processing): The dynamic price elasticities of generation, consumption, and carbon emissions are calculated and processed as elasticity arrays by the proposed approach of pricing patterns processing, in order to represent intrinsic features of a prosumer (details are described in Section 3.2).

Step 3 (Preferences Optimisation): Each of the generated scenarios is used as inputs of the preferences optimisation problem. The problem is solved by the heuristic algorithm to yield optimal scheduling decisions (details are described in Section 3.3).

Step 4 (Training Neural Networks): The designed neural networks are trained through using the processed scenarios and elasticity arrays as inputs, and optimal scheduling decisions as training labels (the architecture of designed neural networks is described in Section 3.4).
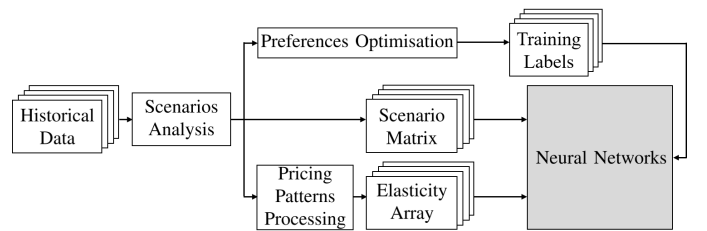

Fig. 2. Flowchart of the training phase of the energy scheduling model. Historical data is processed by the scenarios analysis and pricing patterns processing approaches as training inputs of developed neural networks. The training labels are optimal scheduling decisions yielded by solving the preferences optimisation problems.

3.1. Scenarios analysis

At the training phase, the scenarios analysis aims at: *1)* data augmentation to avoid the overfitting problem caused by limited samples; *2)* predicting potential variations of uncertain variables. Each generated scenario represents a possible variation of uncertain variables. Let $\mathcal{I}$ and $\mathcal{K}$ denote the index sets of

generators and loads of a prosumer, respectively. The power generation of the generator $i \in \mathcal{I}$ at the scheduling time t is denoted by $p_{i,t}$. The power consumption of the load $k \in \mathcal{K}$ at the scheduling time t is denoted by $p_{k,t}$. The retail electricity price at the scheduling time t is denoted by π_t. For simplicity, $p_{i,t}$, $p_{k,t}$, and π_t are represented by an uncertain variable p_t in this subsection. The scenarios are generated through the following two procedures, i.e., the kernel density estimation and Latin Hypercube sampling:

First, accurately evaluating the distributions of uncertain variables is a prerequisite for the scenarios generation. A non-parametric estimation which only relies on historical data can capture the stochastic feature of the distributed RESs, flexible demand, and dynamic prices, without the need of pre-assumed distributions and parameters as in the parametric estimation. This research therefore uses the approach of the non-parametric kernel density estimation [17] to estimate the probability density function of uncertain variables. The unknown density function of the uncertain variable p_t is then fitted from the set of historical data as

$$\tilde{f}(p_t) := \frac{1}{|\mathcal{M}| \cdot \varsigma} \sum_{m \in \mathcal{M}} f_{\text{kernel}} \left(\frac{p_t - p_{t,m}}{h} \right), \quad (1)$$

where $\tilde{f}(\cdot)$ is the estimated kernel density function of the uncertain variable p_t, $\mathcal{M}$ is the index set of the historical metering data, $|\mathcal{M}|$ is the total number of the historical metering data, $p_{t,m}$ is the data sample $m \in \mathcal{M}$ of the uncertain variable p_t, ς is the bandwidth smoothing parameter, and $f_{\text{kernel}}(\cdot)$ is the kernel function. Gaussian kernel function is used due to its high accuracy [18]. The kernel function is placed around each historical data $p_{t,m}$ to construct $\tilde{f}(p_t)$ by the sum of $|\mathcal{M}|$ kernels as shown in Fig. 3 (a).
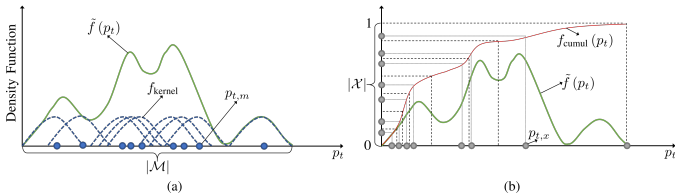

Fig. 3. Schematic illustrations of the kernel density estimation in (a) and Latin Hypercube sampling in (b). In (a), the kernel function f_{kernel} is placed around each historical data $p_{t,m}$. The estimated kernel density function $\tilde{f}(p_t)$ is constructed by the sum of $|\mathcal{M}|$ kernels. In (b), the value range of the cumulative density function $f_{\text{cumul}}(p_t)$ is equally divided into $|\mathcal{X}|$ subintervals. Through random sampling from each subinterval, the scenario $p_{t,x}$ can be obtained by the inverse function of $f_{\text{cumul}}(p_t)|_{p_t=p_{t,x}}$.

Second, with the estimated density function, the Latin Hypercube sampling [19] is implemented to produce scenarios. Let $\mathcal{X}$ denote the index set of scenarios and $|\mathcal{X}|$ denote the number of scenarios. To generate the desired $|\mathcal{X}|$ scenarios, the value range of the cumulative density function, i.e., $[0,1]$, is equally divided into $|\mathcal{X}|$ subintervals. The scenario $x \in \mathcal{X}$ of the uncertain variable p_t, denoted by $p_{t,x}$, is generated from each subinterval by using the Latin Hypercube sampling as

$$f_{\text{cumul}}(p_t)|_{p_t=p_{t,x}} := \left(\frac{1}{|\mathcal{X}|} \right) \cdot \vartheta + \frac{x-1}{|\mathcal{X}|}, \quad (2)$$

where $f_{\text{cumul}}(p_t)$ is the cumulative density function of the uncertain variable p_t calculated by the integral of $\tilde{f}(p_t)$, and $\vartheta \in [0,1]$ is a random variable following the uniform distribution. The value of a scenario can be calculated by the inverse function as

$$p_{t,x} = f_{\text{cumul}}^{-1} \left[f_{\text{cumul}}(p_t)|_{p_t=p_{t,x}} \right]. \quad (3)$$

The occurrence probability of $p_{t,x}$ can be obtained as

$$\Pr(p_{t,x}) = \frac{1}{|\mathcal{X}|}, \quad (4)$$

where $\Pr(p_{t,x})$ is the occurrence probability of the scenario $p_{t,x}$. The schematic illustration of the Latin Hypercube sampling is presented in Fig. 3 (b).

3.2. Pricing patterns processing

The effects of the real-time electricity pricing signals on the generation, consumption, and carbon emissions can be processed to corresponding three types of dynamic price elasticities. To facilitate the discussion, the following two concepts from microeconomics [20] are introduced:

• *Price Elasticity:* The price elasticity describes the relative changes in the supply/demand of a product resulting from a change in the price of this product.

• *Elastic/Inelastic Product:* In the microeconomics, when the absolute value of the elasticity of a product (e.g., electricity in the context of our research) is greater than 1, this product is elastic, which means that when the price changes, the demand/supply of this product has a dramatic change, e.g., for luxuries. By contrast, when the absolute value of the elasticity of a product is less than 1, this product is inelastic, which means that when the price changes, the demand/supply of this product does not change dramatically, e.g., for necessities. For the same product, when the price increases, this product becomes elastic, which means that this product tends to be replaced by alternatives. By contrast, when the price decreases, this product becomes inelastic, which means that the dependency on this product raises.

The pricing patterns are processed through the following two procedures, i.e., the elasticity calculation and elasticity arrays generation:

First, formulations used to calculate three types of elasticities are introduced. A prosumer's response to the change in the real-time electricity prices depends on the considered time horizon. Since our research focuses on the day-ahead energy scheduling in the half-hour time interval, the scheduling horizon would be 48 half-hour time intervals. Let $\mathcal{T}$ denote the index set of the scheduling time and Δt denote the scheduling time interval. We have $(\Delta t, |\mathcal{T}|) = (0.5, 48)$. Let $p_{i,t,x}$ and $p_{k,t,x}$ denote the power generation of the generator i and consumption of the load k of the scenario x at the scheduling time t, respectively. The aforementioned three types of elasticities can be defined as follows:

• Price elasticity of generation: When the electricity prices change from $\pi_{t_a,x}$ at the scheduling time t_a to $\pi_{t_b,x}$ at the scheduling time t_b, a prosumer's total power generation of the scenario

x correspondingly changes from $\sum_{i \in \mathcal{I}} p_{i,t_a,x}$ to $\sum_{i \in \mathcal{I}} p_{i,t_b,x}$. The price elasticity of generation between t_a and t_b is defined as

$$\xi_{p_{i,x}}(t_a, t_b) := \left(\sum_{i \in \mathcal{I}} \frac{p_{i,t_a,x} - p_{i,t_b,x}}{p_{i,t_a,x}} \right) \cdot \left(\frac{\pi_{t_a,x}}{\pi_{t_a,x} - \pi_{t_b,x}} \right), \forall t_a, t_b \in \mathcal{T}, \tag{5}$$

where $\xi_{p_{i,x}}(\cdot)$ is the function of the price elasticity of generation between any two scheduling time.

• Price elasticity of consumption: When the electricity prices change from $\pi_{t_a,x}$ at the scheduling time t_a to $\pi_{t_b,x}$ at the scheduling time t_b, a prosumer's total power consumption of the scenario x correspondingly changes from $\sum_{k \in \mathcal{K}} p_{k,t_a,x}$ to $\sum_{k \in \mathcal{K}} p_{k,t_b,x}$. The price elasticity of consumption between t_a and t_b is defined as

$$\xi_{p_{k,x}}(t_a, t_b) := \left(\sum_{k \in \mathcal{K}} \frac{p_{k,t_a,x} - p_{k,t_b,x}}{p_{k,t_a,x}} \right) \cdot \left(\frac{\pi_{t_a,x}}{\pi_{t_a,x} - \pi_{t_b,x}} \right), \forall t_a, t_b \in \mathcal{T}, \tag{6}$$

where $\xi_{p_{k,x}}(\cdot)$ is the function of the price elasticity of consumption between any two scheduling time.

• Price elasticity of carbon emissions: There are two portions of incurred carbon emissions when a prosumer uses electricity: *1)* When a prosumer consumes electricity from the utility grid, carbon emissions will be caused by the generation from traditional fossil-fuel based power plants, e.g., coal and gas. *2)* When a prosumer uses distributed biomass or diesel generators, carbon emissions will be caused by its on-site generation. These two portions of carbon emissions can be described as the first and second terms of the following equation.

$$r_{t,x} = p_{\text{utility},t,x} \cdot \rho_{\text{utility}} + \sum_{i \in \mathcal{I}} p_{i,t,x} \cdot \rho_i, \forall t \in \mathcal{T}, x \in \mathcal{X}, \tag{7}$$

where $r_{t,x}$ is the carbon emissions rate of a prosumer's scenario x at the scheduling time t in the unit of kg/h, $p_{\text{utility},t,x}$ is the power imported from the utility grid by a prosumer's scenario x at the scheduling time t, ρ_{utility} is the average carbon intensity of all generation sources from the utility grid in the unit of kg/kWh, and ρ_i is the carbon intensity of a prosumer's own generator i. The first term of Eq. (7) only holds when $p_{\text{utility},t,x} > 0$.

As a part of the generating costs, the carbon cost affects the retail electricity prices. To consider this impact, we define the elasticity of carbon emissions in electricity price. When the electricity prices change from $\pi_{t_a,x}$ at the scheduling time t_a to $\pi_{t_b,x}$ at the scheduling time t_b, a prosumer's total carbon emissions rate of the scenario x correspondingly changes from $r_{t_a,x}$ to $r_{t_b,x}$. The price elasticity of carbon emissions between t_a and t_b is defined as

$$\xi_{r_x}(t_a, t_b) := \left(\frac{r_{t_a,x} - r_{t_b,x}}{r_{t_a,x}} \right) \cdot \left(\frac{\pi_{t_a,x}}{\pi_{t_a,x} - \pi_{t_b,x}} \right), \forall t_a, t_b \in \mathcal{T}, \tag{8}$$

where $\xi_{r_x}(\cdot)$ is the function of the price elasticity of carbon emissions between any two scheduling time.

Next, how to process the calculated three types of elasticities as elasticity arrays is introduced. Each type of these three price elasticities includes both the self-elasticity and cross-elasticity.

The self-elasticity relates the generation, consumption, or carbon emissions during a half-hour to the electricity price during the same half-hour. The cross-elasticity relates the generation, consumption, or carbon emissions during a half-hour to the electricity price during another half-hour. In the context of our research, the energy scheduling implies that a prosumer reduces its generation and consumption during some half-hours and increases them during others, which corresponds to the cross-elasticity as indicated in Eqs. (5), (6), and (8). For this reason, the self-elasticity is set as zero. If this scheduling is within the scheduling horizon $|\mathcal{T}|$, i.e., 48 half-hour time intervals. Each type of self-elasticity and cross-elasticity can be arranged into a 48×48 matrix, and three types of elasticities can be arranged into a 48×48×3 array as shown in Fig. 4, denoted as $\mathbf{\Phi}_x(t_a, t_b, \Xi)$, $\forall t_a, t_b \in \mathcal{T}$, where $\Xi = \left[\xi_{p_{i,x}}, \xi_{p_{k,x}}, \xi_{r_x} \right]$ indicates three types of price elasticities.
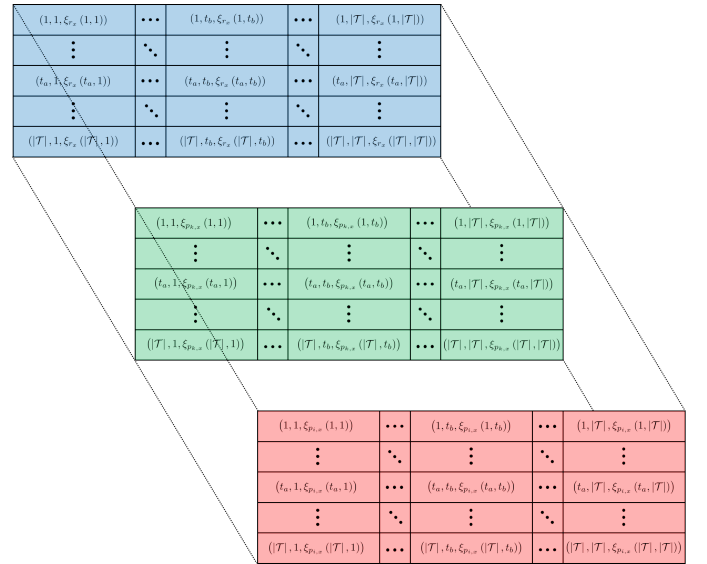

Fig. 4. Schematic illustration of a 48×48×3 array $\mathbf{\Phi}_x(t_a, t_b, \Xi)$, $\forall t_a, t_b \in \mathcal{T}$, $\Xi = \left[\xi_{p_{i,x}}, \xi_{p_{k,x}}, \xi_{r_x} \right]$.

In each 48×48 matrix, the diagonal elements represent the self-elasticities and off-diagonal elements represent the cross-elasticities. Each row t_a (or column t_b) indicates how a change in the electricity price during one half-hour time interval impacts the generation, consumption, or carbon emissions during every half-hour time interval within the scheduling horizon. Since a price elasticity from the scheduling time t_a to the scheduling time t_b does not equal to that from the scheduling time t_b to the scheduling time t_a according to Eqs. (5) (6) and (8), each matrix is asymmetric with respect to its diagonal.

The role and importance of elasticity arrays include *1)* representing intrinsic features of an individual prosumer in responding to dynamic electricity prices, defined as the pricing patterns, *2)* enabling our proposed model to map the intrinsic features of an individual prosumer to potential scheduling decisions, *3)* enabling the designed CNN to extract the local feature, temporal transient feature, and correlation of elasticities from the elasticity arrays through exploiting the pattern recognition capability of the CNN, *4)* normalising the scale difference between the

power (in the unit of kW) and carbon emissions (in the unit of kg), since three types of elasticities have the same value range, and *5)* helping improve the accuracy of designed learning model.

Remark 1: To help readers visualise the elasticity arrays. The 48×48×3 elasticity array is presented as an image, by which the '48×48' corresponds to the 'length×width' and '3' corresponds to the 'three colour channels (R G B)' by proportionally scaling up to the value range of the colour channel, i.e., [0,255]. Using the scheduling time t_a and t_b, we can locate the position of a pixel at an image. In such an image, a brighter pixel indicates a higher price elasticity, which means that given a certain amount of the price change (increase/decrease) as an incentive, the amount of generation, consumption, or carbon emissions between the scheduling intervals indicated by t_a and t_b would change dramatically through scheduling. By contrast, a darker pixel indicates a lower price elasticity, which means that given a certain amount of the price change as an incentive, the amount of generation, consumption, or carbon emissions between the scheduling intervals indicated by t_a and t_b would not change dramatically through scheduling. Additionally, the correlation of $\xi_{p_{i,x}}, \xi_{p_{k,x}}$, and ξ_{r_x} is reflected by combining three colour channels, i.e., the colour overlay.

3.3. Preferences optimisation

The problem of preferences optimisation is only solved at the training phase by manufacturers of the energy scheduling model, based on the historical metering data and pre-defined preferences of an individual prosumer. At the deploying phase, the trained neural networks can replace the step of solving optimisation problems so as to help each prosumer automatically make predicted optimal scheduling decisions. In our research, the preferences of individual prosumers are instantiated as minimising the costs and carbon emissions of using electricity. The following assumptions have been made when we consider a prosumer's energy scheduling:

Assumption 1: Since the generators and loads in the distribution networks are nearby, and the generation from prosumers is small relative to the total generation in power systems, the transmission losses in the distribution networks are neglected.

Assumption 2: When a prosumer sells the electricity to the utility grid, i.e., $p_{\text{utility},t,x} < 0$, the selling price equals to the electricity price at that scheduling time. The pricing design of the electricity exchange between prosumers and the utility grid is beyond the scope of our research.

Assumption 3: The energy storage devices are equipped for individual prosumers to help them schedule the non-dispatchable distributed RESs. Zero charging/discharging losses are assumed for energy storage devices. For the prosumers without the energy storage devices, the maximum storage capacity and maximum charging/discharging rate can be set as zero.

Firstly, from the economic perspective, a prosumer aims to minimise the costs of using electricity by strategically modifying the power exchange with the utility grid, generation, consumption, and storage charging/discharging of each scenario.

The objective function of the costs of using electricity consists of the cost of importing electricity from the utility grid, cost of on-site generation, and cost of running storage devices, which can be described as

$$f_c\left(\Delta p_{\text{utility},t,x}, \Delta p_{i,t,x}, \Delta p_{k,t,x}\right) := \sum_{t \in \mathcal{T}} \left[\left(p_{\text{utility},t,x} - \Delta p_{\text{utility},t,x}\right) \cdot \pi_{t,x} \right.$$
$$+ \sum_{i \in \mathcal{I}} \left(p_{i,t,x} - \Delta p_{i,t,x}\right) \cdot \delta_i$$
$$\left. + \left|\Delta p_{s,t,x}\right| \cdot \delta_s \right] \cdot \Delta t,$$
(9)

where $f_c(\cdot)$ is the objective function of the costs of using electricity by a prosumer, $\Delta p_{\text{utility},t,x}$ is the amount of modifying the power exchange from the utility grid of a prosumer's scenario x at the scheduling time t, $\Delta p_{i,t,x}$ is the amount of modifying the generation by the generator i of a prosumer's scenario x at the scheduling time t, $\Delta p_{k,t,x}$ is the amount of modifying the consumption by the load k of a prosumer's scenario x at the scheduling time t, δ_i is the coefficient of the operating cost of the generator i, $\Delta p_{s,t,x}$ is the power charging/discharging rate of a prosumer's energy storage device of the scenario x at the scheduling time t, and δ_s is the cost coefficient of the energy storage devise. The operating costs include the costs of the operation, maintenance, fuel, and carbon emissions (excluding the costs of the pre-development, construction, decommissioning, and waste). The coefficients of operating costs are evaluated by the levelised cost of electricity generation (LCoE) [21]. The storage costs include the costs of the operation, maintenance, and charging/discharging (excluding the initial installed cost). The coefficient of storage costs is evaluated by the levelised cost of storage (LCoS) [22] which is a discounted cost per unit of charged/discharged electrical energy.

The power dynamics of a prosumer's energy storage device can be described as

$$\Delta p_{s,t,x} = \left(p_{\text{utility},t-1,x} - \Delta p_{\text{utility},t-1,x}\right) - \sum_{k \in \mathcal{K}} \left(p_{k,t-1,x} - \Delta p_{k,t-1,x}\right)$$
$$+ \sum_{i \in \mathcal{I}} \left(p_{i,t-1,x} - \Delta p_{i,t-1,x}\right),$$
(10)

where $\Delta p_{s,t,x} \cdot \Delta t = s_{t,x} - s_{t-1,x}$ indicates the power charging/discharging when the value of $\Delta p_{s,t,x}$ is positive/negative, respectively, and $s_{t,x}$ is the stored energy of a prosumer's energy storage device of the scenario x at the scheduling time t. There are two constraints for an energy storage device as

• *Storage capacity constraint*: The maximum storage capacity is determined by the medium of storage devices [23]. The stored energy should be restricted to certain limits as

$$0 \leq s_{t,x} \leq s^{\max}, \tag{11}$$

where $s^{\max}$ is the maximum storage capacity.

• *Charging/Discharging rate constraint*: The charging/discharging rate should be restricted below the maximum limit as

$$\left|\Delta p_{s,t,x}\right| \leq \Delta p_s^{\max}, \tag{12}$$

where $\Delta p_s^{\max}$ is the maximum charging/discharging rate. For the prosumers without the energy storage devices, we have $s^{\max} = 0$ and $\Delta p_{s,t,x} = 0$.

Secondly, from the environmental perspective, a prosumer aims to minimise its total carbon emissions. Recall that ρ_{utility} is the average carbon intensity of all generation sources from the utility grid, and ρ_i is the carbon intensity of a prosumer's own generator i. The objective function of total carbon emissions can be modelled as

$$f_e\left(\Delta p_{\text{utility},t,x}, \Delta p_{i,t,x}, \Delta p_{k,t,x}\right) := \sum_{t \in \mathcal{T}} \left[\left(p_{\text{utility},t,x} - \Delta p_{\text{utility},t,x}\right) \cdot \rho_{\text{utility}} \right.$$
$$\left. + \sum_{i \in \mathcal{I}} \left(p_{i,t,x} - \Delta p_{i,t,x}\right) \cdot \rho_i \right] \cdot \Delta t, \tag{13}$$

where $f_e(\cdot)$ is the objective function of total carbon emissions incurred by a prosumer. The first term of Eq. (13) holds when $\left(p_{\text{utility},t,x} - \Delta p_{\text{utility},t,x}\right) > 0$.

The power exchange, generation and consumption should be restricted to certain limits considering the capacities of power grids, generators and loads as

$$p_{\text{utility}}^{\min} \leq p_{\text{utility},t,x} - \Delta p_{\text{utility},t,x} \leq p_{\text{utility}}^{\max}, \tag{14}$$

$$p_i^{\min} \leq p_{i,t,x} - \Delta p_{i,t,x} \leq p_i^{\max}, \tag{15}$$

$$p_k^{\min} \leq p_{k,t,x} - \Delta p_{k,t,x} \leq p_k^{\max}, \tag{16}$$

where $p_{\text{utility}}^{\min}$ and $p_{\text{utility}}^{\max}$ are the minimum and maximum power exchange levels of power grids, respectively, $p_i^{\min}$ and $p_i^{\max}$ are the minimum and maximum power generation levels of the generator i, respectively, and $p_k^{\min}$ and $p_k^{\max}$ are the minimum and maximum power consumption levels of the load k, respectively.

Therefore, the objectives of a prosumer are to minimise the costs and carbon emissions of using electricity, with the decision variables of modifying the power exchange, generation, and consumption of each scenario, which leads to a multi-objective optimisation problem (MOP) as

$$\min_{\Delta p_{\text{utility},t,x}, \Delta p_{i,t,x}, \Delta p_{k,t,x}} : \{f_c, f_e\}, \tag{17}$$

s.t.: Eqs. (10) - (12) and (14) - (16).

A multi-objective immune algorithm [24] is used to solve the preferences optimisation problem. The detailed concepts and solution algorithms are provided in the Appendix A. The optimal decision variables are subsequently used as training labels to train the neural networks. The labels indicate how far the predicted optimal scheduling decisions from the theoretical optimal ones.

3.4. Neural networks architecture

The designed neural networks consist of convolutional layers to extract intrinsic features from an elasticity array $\mathbf{\Phi}_x$, and fully-connected layers to import the scenario information from
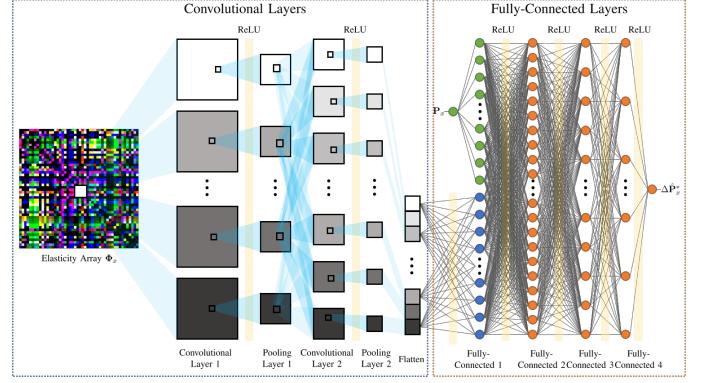

Fig. 5. Architecture of designed neural networks. The elasticity array is imported to the convolutional layers. The scenario matrix is imported to the fully-connected layers. The training output is the predicted optimal scheduling decisions of a scenario.

a scenario matrix as

$$\mathbf{P}_x = \begin{pmatrix} p_{\text{utility},1,x} & \cdots & p_{\text{utility},t,x} & \cdots & p_{\text{utility},|\mathcal{T}|,x} \\ p_{1,1,x} & \cdots & p_{1,t,x} & \cdots & p_{1,|\mathcal{T}|,x} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ p_{i,1,x} & \cdots & p_{i,t,x} & \cdots & p_{i,|\mathcal{T}|,x} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ p_{|\mathcal{I}|,1,x} & \cdots & p_{|\mathcal{I}|,t,x} & \cdots & p_{|\mathcal{I}|,|\mathcal{T}|,x} \\ p_{1,1,x} & \cdots & p_{1,t,x} & \cdots & p_{1,|\mathcal{T}|,x} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ p_{k,1,x} & \cdots & p_{k,t,x} & \cdots & p_{k,|\mathcal{T}|,x} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ p_{|\mathcal{K}|,1,x} & \cdots & p_{|\mathcal{K}|,t,x} & \cdots & p_{|\mathcal{K}|,|\mathcal{T}|,x} \end{pmatrix}. \tag{18}$$

The processed inputs of the elasticity array and scenario matrix are further merged by following fully-connected layers. The training output is the matrix of the predicted optimal scheduling decisions $\Delta \hat{\mathbf{P}}_x^*$ which is in the same format as Eq. (18). The architecture of the designed neural networks is presented in Fig. 5. This structure of paralleled neural networks has been validated as an efficient approach for extracting information from both the matrix and array in [5]. The relationship between the training input and output can be described as

$$\Delta \hat{\mathbf{P}}_x^* = f_{\text{nn}}\left(\mathbf{P}_x, \mathbf{\Phi}_x\right), \tag{19}$$

where $f_{\text{nn}}(\cdot)$ is the relationship function parametrised by tuning the neural networks.

The convolutional layers convolve an elasticity array with multiple filters to extract the following three features as shown in Fig. 6:

• *Local Feature*: The local feature of each type of price elasticity within the filter size can be detected when the filter is on a patch of the array. For instance, if the filter size is 5×5, the local feature within every 5 consecutive scheduling intervals is extracted.

• *Temporal Transient Feature*: The temporal transient feature of each type of price elasticity can be detected when the filter
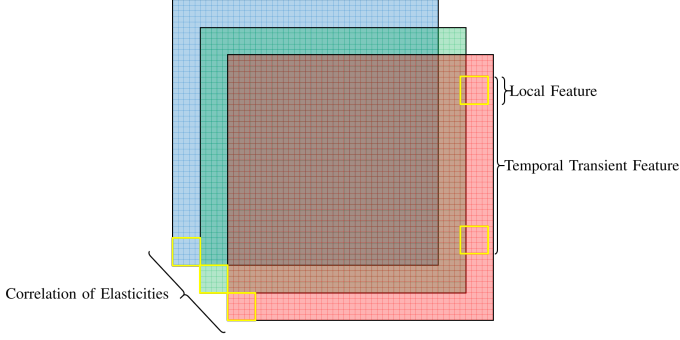
Fig. 6. Schematic illustration of the local feature, temporal transient feature, and correlation of elasticities. The filter size is indicated by the yellow box.

slides through the array by strides. For instance, if the stride is (24,24), the temporal transient feature over every 24 inconsecutive scheduling intervals, e.g., between the day and night, is extracted.

• *Correlation of Elasticities*: The correlation of three types of price elasticities can be detected when multiple filters simultaneously convolve three dimensions.

All these three extracted features are stacked as a feature map and processed by further layers. The feature map of all filters can be described as

$$\mathbf{\Phi}_{\mathrm{map}} = f_{\mathrm{ReLU}}\left(\mathbf{W} \cdot \mathbf{\Phi}_x + \mathbf{b}\right), \tag{20}$$

where $\mathbf{\Phi}_{\mathrm{map}}$ is the array of the feature map, $f_{\mathrm{ReLU}}\left(\cdot\right)$ is the activation function, $\mathbf{W}$ is the weight array, and $\mathbf{b}$ is the bias vector. The rectified linear unit (ReLU) is used as the activation function.

As shown in Fig. 5, the function of pooling layers is to progressively reduce the spatial size of the feature representations, so as to reduce the parameters and computational burden of neural networks. Each convolutional layer is followed by a pooling layer to downsample the feature map through using the max pooling. In order to control the spatial size of convolutional outputs, the same padding is used to pad the input of each convolutional layer with zeros around the border. Over multiple convolutional layers and pooling layers, a global feature map is formed by integrating the feature representations from every layer. The global feature map is subsequently converted to a vector by a flatten layer and processed by fully-connected layers. The function of fully-connected layers is to further extract feature representations from merged feature representations of both the elasticity array and scenario matrix.

The overall algorithm of the training phase is shown in **Algorithm 1**.

4. Deploying phase of energy scheduling model

This section introduces the deploying phase of the proposed energy scheduling model. The flowchart of the deploying phase is shown in Fig. 7. The detailed steps of the deploying phase are given as follows:

Algorithm 1 Training phase of the energy scheduling model

input: historical data $p_{t,m}$
1: generate $|\mathcal{X}|$ scenarios, each scenario $p_{t,x}$ with occurrence probability $\mathrm{Pr}(p_{t,x}) = 1/|\mathcal{X}|$
2: **for** $x = 1, ..., |\mathcal{X}|$ **do**
3:     process the elasticity array $\mathbf{\Phi}_x\left(t_a, t_b, \Xi\right)$
4:     solve the preferences optimisation problem to obtain optimal decision variables $\Delta\mathbf{p}_x^*$
5:     train the neural networks as Eq. (19)
6: **end for**
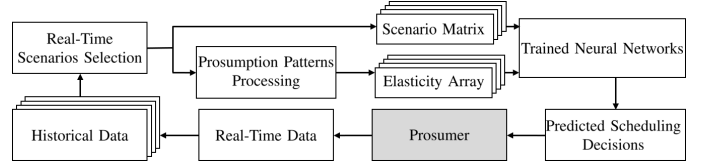**output:** trained neural networks f_{nn}

Fig. 7. Flowchart of the deploying phase of the energy scheduling model. Historical data of a prosumer is dynamically updated by the real-time metering data. The scenarios are selected by the real-time scenarios selection approach. The processed scenario matrix and elasticity array are fed into trained neural networks to yield the predicted optimal scheduling decisions.

Step 1 (Real-Time Scenarios Selection): The historical data of a prosumer is dynamically updated by the real-time metering data. To provide a prosumer with an accurate prediction that considers the uncertainties of the distributed RESs, flexible demand, and dynamic prices for the following day, a real-time scenarios selection approach is developed. This approach aims to dynamically select the scenarios with current characteristics of uncertainties and discard dated scenarios. Let $p_{t,x}$ and $p'_{t,x}$ denote the scenarios generated from the historical data and updated data, respectively. The occurrence probability of $p'_{t,x}$ is updated as [25] by

$$\mathrm{Pr}\left(p'_{t,x}\right) = \mathrm{Pr}\left(p_{t,x}\right) + \frac{1}{|\mathcal{X}| + 1}\left[\vartheta_{\mathrm{Pr}} - \mathrm{Pr}\left(p_{t,x}\right)\right], \tag{21}$$

where $\vartheta_{\mathrm{Pr}} \in \{0, 1\}$ is a binary value determined by

$$p_{t,x}^{'*} = \arg\min_{p'_{t,x}} l_{x,x'}, \tag{22}$$

where $l_{x,x'} = p_{t,x} - p'_{t,x}$ is the distance between $p_{t,x}$ and $p'_{t,x}$. If $p'_{t,x} = p_{t,x}^{'*}$, $\vartheta_{\mathrm{Pr}} = 1$ and the predicted error term $\left[\vartheta_{\mathrm{Pr}} - \mathrm{Pr}\left(p_{t,x}\right)\right]$ becomes positive to reinforce the previous probability; If $p'_{t,x} \neq p_{t,x}^{'*}$, $\vartheta_{\mathrm{Pr}} = 0$ and the predicted error term becomes negative to weaken the previous probability.

Step 2 (Pricing Patterns Processing): Elasticity array $\mathbf{\Phi}_x$ is processed by the pricing patterns processing approach.

Step 3 (Predicting Scheduling Decisions): The processed scenario matrix and elasticity array are subsequently fed into trained neural networks to yield the predicted optimal scheduling decisions. The predicted optimal scheduling decisions are subsequently examined by the constraints of the preference optimisation problem as Eqs. (10) - (12) and (14) - (16). Only those predicted optimal scheduling decisions which satisfy the

constraints are remained. The trained neural networks keep predicting optimal scheduling decisions until all the predicted optimal scheduling decisions are feasible.

Step 4 (Control): The controlling signals for the next day are yielded by aggregating the predicted optimal scheduling decisions of $|\mathcal{X}|$ scenarios, weighted by the occurrence probabilities as

$$p_{\text{utility},t} = \sum_{x \in \mathcal{X}} \left(p_{\text{utility},t,x} - \Delta \hat{p}^*_{\text{utility},t,x} \right) \cdot \text{Pr}\left(p_{\text{utility},t,x} \right), \quad (23)$$

$$p_{i,t} = \sum_{x \in \mathcal{X}} \left(p_{i,t,x} - \Delta \hat{p}^*_{i,t,x} \right) \cdot \text{Pr}\left(p_{i,t,x} \right), \quad (24)$$

$$p_{k,t} = \sum_{x \in \mathcal{X}} \left(p_{k,t,x} - \Delta \hat{p}^*_{k,t,x} \right) \cdot \text{Pr}\left(p_{k,t,x} \right). \quad (25)$$

The scheduling decisions are performed by a prosumer's controller and serve as the real-time data for the next day. This real-time data is processed by the *Step 1* cyclically.

The overall algorithm of the deploying phase is shown in **Algorithm 2**

Algorithm 2 Deploying phase of the energy scheduling model

---

input: real-time data $p_{\text{utility},t}$, $p_{i,t}$, $p_{k,t}$, and π_t
1: use scenarios analysis approach to generate scenarios from the updated data
2: **for** $x = 1, ..., |\mathcal{X}|$ **do**
3:     update occurrence probabilities using Eqs. (21) and (22)
4:     process the elasticity array $\mathbf{\Phi}_x\left(t_a, t_b, \Xi\right)$
5:     **while** constraints in Eqs. (10) - (12) and (14) - (16) do not hold **do**
6:         use trained neural networks to obtain predicted optimal scheduling decisions $\Delta \hat{\mathbf{P}}^*_x$
7:     **end while**
8: **end for**
9: aggregate the predicted optimal scheduling decisions of $|\mathcal{X}|$ scenarios as Eqs. (23), (24), and (25), and perform the scheduling decisions
output: controlling signals $p_{\text{utility},t}$, $p_{i,t}$, $p_{k,t}$

---

5. Case studies

This section introduces case studies to evaluate the proposed model by comparing different learning approaches under various IEEE test distribution systems and uncertain scenarios.

5.1. Simulation Setup

The simulations are performed by using a machine with the Intel$^\text{R}$ Core$^\text{TM}$ i9-9900K CPU at 3.60 GHz and a NVIDIA GeForce RTX 2080 GPU. The proposed model is written in the Python language by using the PyTorch. To improve the computational efficiency, the training process is performed on the GPU, and 8-core parallel computing is used during the scenarios analysis, pricing patterns processing, and preferences optimisation. The simulations are repeated 10 times to eliminate the randomness and outliers.

Since the scale of a prosumer varies from households, communities, industries, to entire cities, to evaluate the scalability of our proposed model, the modified IEEE 69-bus, 33-bus, and 18-bus distribution systems are adopted by our research, with details provided in the Appendix B. Each prosumer possesses multiple generation sources and loads, indicated by the dished blue box in Fig. B.13 - Fig. B.15. Rather than focusing on topological structures of these distribution networks, our research uses these distribution networks to reflect the scale change of prosumers based on practical distribution networks, i.e., various sources and loads in three standard IEEE distribution networks. The initial stored power is set as zero. The lithium energy storage model and cost coefficient studied in [26] is used. The maximum charging/discharging rate is set as 300 kW and maximum storage capacity is set as 2000 kWh according to the model of distribution-scale energy storage device in [26]. The half-hourly retail electricity prices obtained from the GB energy market [27] are used. The coefficients of operating costs [21] and carbon intensities [28] are presented in Table 1.

Table 1 Coefficients of operating costs and carbon intensities

Source	Diesel	Wind	Biomass	Solar	Storage
δ_i/δ_s (£/kWh)	0.123	0.015	0.080	0.009	0.135
ρ_i (kg/kWh)	1.69	0	0.05	0	0

The inputs of neural networks are separated into 70% of the training set and 30% of the validation set with randomly sampling. The data is preprocessed by the z-score normalisation [29]. The Adam [30] is used as an optimiser to train the neural networks for 50 epochs, with 4-size of the mini-batch, 1×10^{-4} of the initial learning rate, and 1×10^{-2} of the weight decay. The learning rate will be reduced if no improvement of the accuracy is seen for 5 epochs. To avoid the overfitting problem caused by parameters of the deep structure, 0.1 of the dropout [31] is used for each layer to randomly drop units. The mean squared error (MSE) [32] is used as a performance metric to indicate the learning losses. The training labels are used as a benchmark to examine the training accuracy. The architecture of the designed neural networks is shown in Table 2. These parameters and structures are determined by the tradeoff between the accuracy and computational time. The impact of tuning parameters and structures of the proposed model on the validation accuracy is shown in Fig. 8.

5.2. Evaluation of the training phase

To test the convergence performance and learning accuracy, our designed neural networks are compared with the following deep approximation architectures, with other training parameters remaining unchanged:

• *RNNs*: The convolutional layers of our designed neural networks are replaced by the RNNs with a 1024-node hidden layer. Different from the CNNs to simultaneously convolve the 3 dimensions of an array, the RNNs can only import a matrix as an input. Hence, the elasticity array is reshaped as a 48×(48 × 3) matrix as shown in Fig. 9 (a). At each time step, the RNNs import each row of this matrix and return a hidden state and an

Table 2 Architecture of the designed neural networks

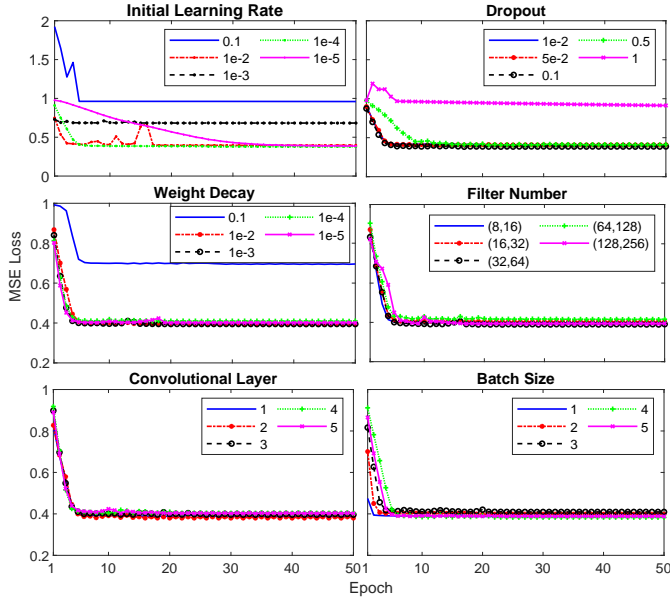| Layer	Input size	Output size	Filter number	Filter size	Stride	Padding
Convolution 1	48,48,3	48,48,32	32	5×5	(1,1)	2
Pooling 1	48,48,32	24,24,32	1	2×2	(2,2)	0
Convolution 2	24,24,32	24,24,64	64	3×3	(1,1)	1
Pooling 2	24,24,64	12,12,64	1	2×2	(2,2)	0
Flatten	12,12,64	12×12×64	-	-	-	-
Fully-connected 1	$12 \times 12 \times 64 + (1 + |\mathcal{I}| + |\mathcal{K}|) \times |\mathcal{T}|$	2048	-	-	-	-
Fully-connected 2	2048	1024	-	-	-	-
Fully-connected 3	1024	512	-	-	-	-
Fully-connected 4	512	256	-	-	-	-
Output	256	$(1 + |\mathcal{I}| + |\mathcal{K}|) \times |\mathcal{T}|$	-	-	-	-

Fig. 8. Impact of tuning parameters and structures of the proposed model on the validation accuracy. The *x* axes indicate epochs, and *y* axes indicate the MSE losses.

Fig. 9. Reshaping an elasticity array to be a matrix as the inputs of the RNNs in (a) and LSTM-RNNs in (b). At each time step indicated by the yellow box, each row vector is imported and a hidden state and an output are returned. For the RNNs, the hidden state is used by the next time step. For the LSTM-RNNs, the hidden state for the entire image is stored and selected by a memory cell.

output. The hidden state is used by the next time step to analyse the temporal transient feature of price elasticities. When the RNNs process an entire matrix, i.e., 48 time steps, the outputs are stacked as a 48×1024 vector and processed by further fully-connected layers.

• *LSTM-RNNs*: The convolutional layers of our designed neural networks are replaced by the LSTM-RNNs with a 1024-node hidden layer. The input matrix of the LSTM-RNNs is the same as that of the RNNs. However, the LSTM-RNNs only return an output of a 1024 vector at the last time step as shown in Fig. 9 (b), since it has the memory of the entire matrix.

• *DNNs*: The convolutional layers of our designed neural networks are excluded. Only the fully-connected layers are remained. The neural networks become the DNNs without importing the information from elasticity arrays.

5.2.1. Training and validation performances

The learning process is considered to be converged when the learning rate drops below 1×10^{-7} and no improvement of accuracy is seen for 5 epochs. Prosumers of the modified IEEE 69-bus distribution network with 5000 scenarios are used as
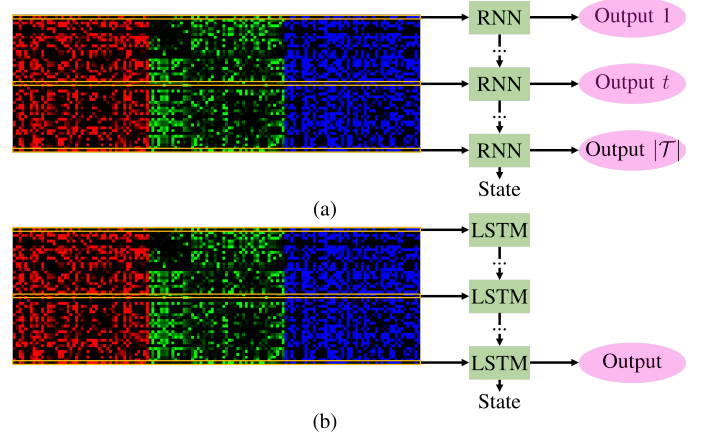
samples to demonstrate the performances of the training and validation. As shown in Fig. 10, both the training and validation of these four learning approaches converge within around 30 epochs. With the additional information of pricing patterns extracted from elasticities arrays, the training losses of the proposed neural networks, RNNs, and LSTM-RNNs are lower than the training losses of the DNNs. The validation losses of our proposed neural networks are the lowest for all prosumers. By contract, the validation losses of the DNNs are the highest for the prosumer 1, prosumer 3, and prosumer 5, and the validation losses of the RNNs are the highest for the prosumer 2 and prosumer 4. One potential reason is that the DNNs cannot extract the information of pricing patterns from the elasticity arrays, and the RNNs are incapable of extracting the global features from an entire image without the memory cell of the LSTM-RNNs. Therefore, the information of pricing patterns is a crucial factor for improving the learning accuracy.

5.2.2. Testing performance

The testing data is used to evaluate the learning accuracy affected by the number of scenarios. The testing data of the load 1 in the modified IEEE 69-bus distribution network is sampled to examine the testing losses with various number of scenarios as shown in Fig. 11. With the increasing number of scenarios, the testing losses rise for all four architectures, because a more
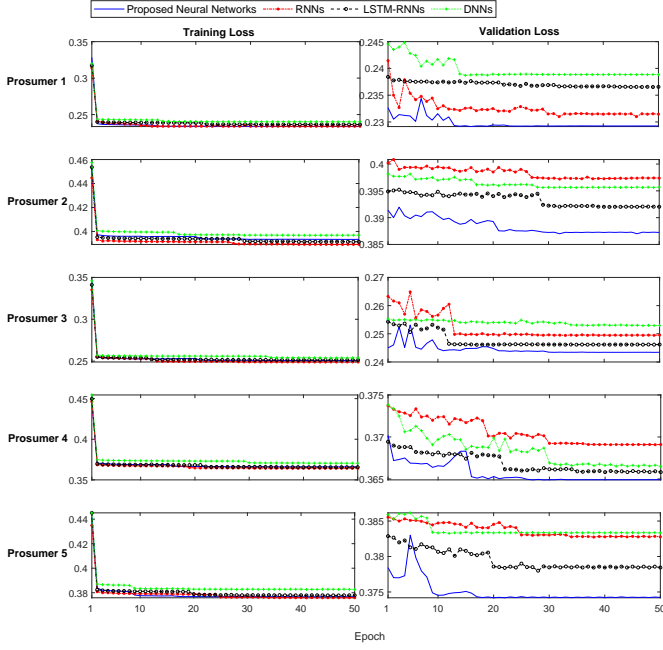
11

Fig. 10. Comparison for the training and validation performances under the proposed neural networks, RNNs, LSTM-RNNs, and DNNs. The *x* axes indicate epochs, and *y* axes indicate the MSE losses.

diverse set of scenarios would cause a larger bias for the training inputs. However, the proposed neural networks outperform other architectures under any selected number of scenarios. To cover diverse uncertainties, 5000 scenarios are used in the simulation of the deploying phase in our research. In the practical implementation, an appropriate number of scenarios can be adjusted according to a prosumer's computational power.

5.3. Evaluation of the deploying phase

5.3.1. Mapping pricing patterns to scheduling decisions

To demonstrate the connection between the intrinsic features of pricing patterns and potential scheduling decisions of a prosumer, an electricity array and scheduling results from the same scenario are sampled as shown in Fig. 12. To help readers visualise the electricity array, it is presented in the form of an image (see the left-top figure) and corresponding decomposition of three colour channels (see following figures on the left hand side), i.e., 'R' for the elasticity of generation, 'G' for the elasticity of consumption, and 'B' for the elasticity of carbon emissions. The brighter region in these images means that given a certain amount of price change as an incentive, the generation, consumption, or carbon emissions between the scheduling intervals indicated by the *x* axes and *y* axes of these images would change dramatically through scheduling, due to a higher price elasticity. By contrast, the darker region means that given a certain amount of price change as an incentive, the generation, consumption, or carbon emissions between the scheduling intervals indicated by the *x* axes and *y* axes of these images would not change dramatically through scheduling, due to a lower price elasticity. The sub-figures on the right hand side of Fig.
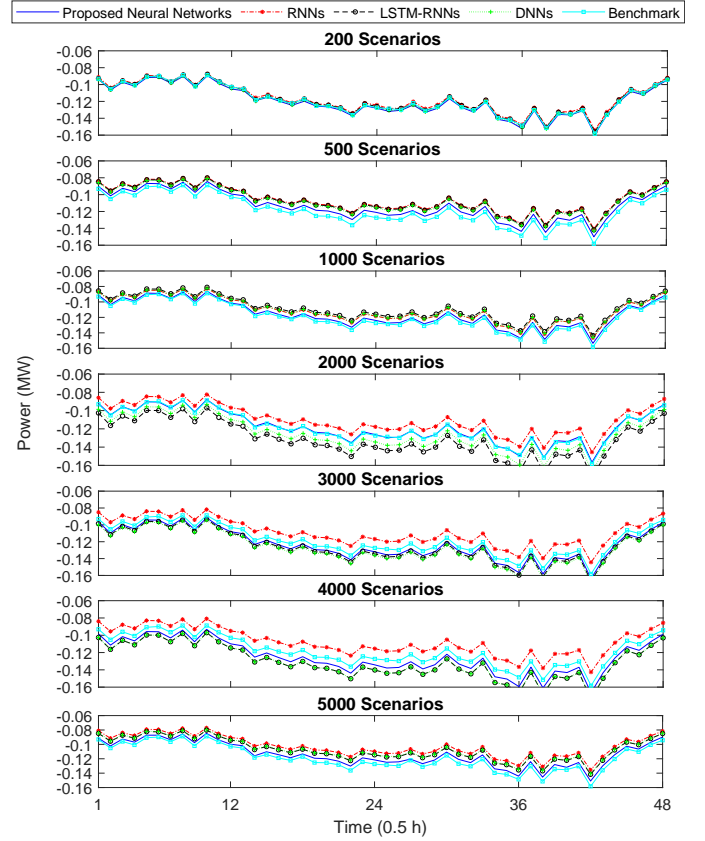

Fig. 11. Comparison for the testing accuracy of various number of scenarios under the proposed neural networks, RNNs, LSTM-RNNs, and DNNs. The *x* axes indicate the scheduling time of the day, and the *y* axes indicate the power.

12 are the information of the electricity price, costs of using electricity, power generation, charging/discharging from storage devices, power consumption, and carbon emissions. The original results come from the selected scenario through using the approach of scenarios analysis,, whereas the scheduling results come from the predicted optimal scheduling decisions through using the proposed learning approach. The difference between the original and scheduling results shows the change of generation, consumption, and carbon emissions through the energy scheduling. The positive value of the second sub-figure on the right hand side means the costs of using electricity, and the negative value of the second sub-figure on the right hand side means the revenue of exporting electricity to the utility grid. The positive/negative value of the fourth sub-figure on the right hand side means charging/discharging from the storage devices.

For the electricity generation, when the electricity price at one scheduling time is higher than that at another scheduling time, a prosumer would reduce the power import from the utility grid, and increase the on-site generation or discharge from the storage devices, so that this prosumer can complement to its demand or export extra power to the utility grid for earning revenues. It can be seen from the third and fourth sub-figures on the right hand side of the Fig. 12, when the electricity price
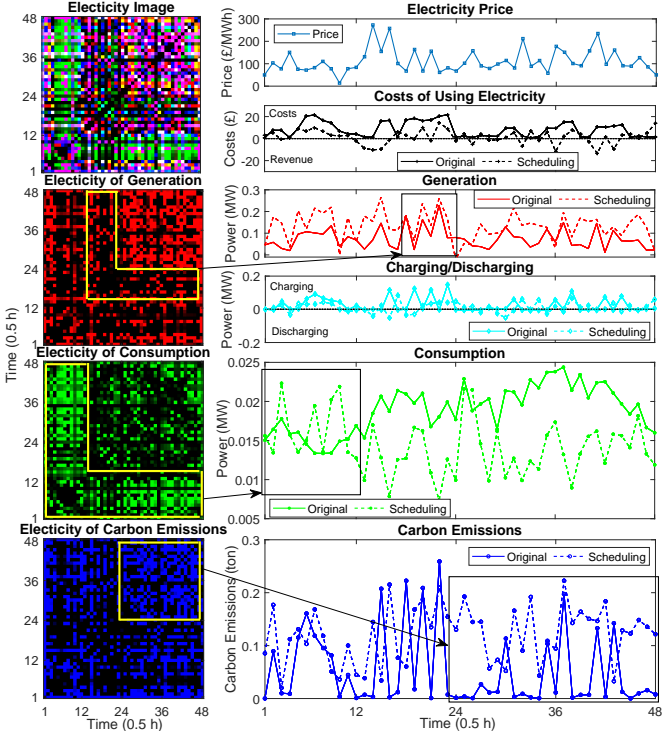
Fig. 12. Schematic illustration of the connection between the pricing patterns and scheduling results. The left four images show the visualised electricity array and decomposition of 3 dimensions corresponding to 3 colour channels. The right six figures show the comparison between the original scenario and scheduled results.

increases, the scheduled generation increases correspondingly with remaining the storage discharge unchanged. For the period from the eighteenth scheduling time to the twenty-forth scheduling time, the price elasticity of generation is relatively low (as indicated by the yellow box on the image of the elasticity of generation). Hence, the scheduled generation keeps almost the same as the original generation.

For the electricity consumption, when the electricity price at one scheduling time is higher than that at another scheduling time, the consumption is shifted away or curtailed. By contrast, when the electricity price at one scheduling time is lower than that at another scheduling time, the consumption is shifted to this scheduling time with the lower price. It can be seen from the fifth sub-figure on the right hand side of the Fig. 12, for the period from the first scheduling time to the twelfth scheduling time, the price elasticity of consumption is relatively high (as indicated by the yellow box on the image of the elasticity of consumption) and the electricity price is relatively low. Hence, the scheduling drives the demand to be shifted from the rest of scheduling time intervals to this time interval.

For the carbon emissions, a higher price elasticity of carbon emissions indicates that the prosumer is more willing to increase or decrease carbon emissions given a certain electricity price, as indicated by the yellow box on the image of the elasticity of carbon emissions. Additionally, since the overall price elasticity of carbon emissions is lower than the elasticities of generation and consumption during the entire scheduling

horizon (indicated by the elasticity image which is dominated by the red colour and green colour), the prosumer inherently prefers to save the costs of using electricity, disregarding the increase of carbon emissions.

5.3.2. Uncertainty prediction

To test the accuracy for the proposed approach of the real-time scenarios selection, the data of the first 300 days is taken as the historical data. From day 301 to day 364, the proposed approach of the real-time scenarios selection is used to update the scenarios set and occurrence probabilities. The data on day 365 is used as a benchmark to examine the accuracy of the selected scenarios in terms of predicting uncertainties caused by the distributed RESs, flexible demand and dynamic prices. Our proposed approach of the real-time scenarios selection is compared with the approach of the Gaussian mixture model (GMM) [33]. Both approaches firstly generate 5000 scenarios, and then select the high probable scenarios as a prediction for the current day. The mean absolute error (MAE) and root mean square error (RMSE) [34] are used as matrices to evaluate the accuracy of these two approaches for the uncertainty prediction. Three prosumers in the modified IEEE 33-bus distribution network are sampled as shown in Table 3. The proposed approach of the real-time scenarios selection is able to more accurately capture the current features of uncertainties and thus yield more precise predictions, compared to the GMM. This is because the proposed approach of the real-time scenarios selection can take advantage of the information from all dynamically generated scenarios by continuously updating the scenarios set and occurrence probabilities with the prosumer's real-time data. By contrast, the GMM directly drops the scenarios with low occurrence probabilities, which causes the selected scenarios to be dominated by certain scenarios.

Table 3 Accuracy comparison for the approaches of the uncertainty prediction

Error (MW)	Prosumer 1		Prosumer 2		Prosumer 3	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
Proposed Approach	0.26	0.32	0.24	0.41	0.28	0.40
GMM	0.39	0.47	0.34	0.54	0.36	0.49

5.3.3. Scalability evaluation

To test the scalability of our proposed energy scheduling model, the artificial immune algorithm is compared with the learning approach on the computational time per scenario of the daily energy scheduling under various IEEE test distribution systems. For the proposed energy scheduling tool, the training phase is operated by the manufacturer, and the computational time for the training phase is the time used for the data pre-processing (i.e., scenarios analysis, pricing patterns processing, and solving optimisation by the heuristic algorithm to produce the training labels) and training of neural networks. By contrast, the deploying phase is operated by a prosumer, and the computational time for the deploying phase is the time used for predicting scheduling decisions using trained neural networks. For the optimisation based scheduling tool using the heuristic algorithm, there is no training phase, and the computational

time for the deploying phase is the time used for scenarios analysis, pricing patterns processing, and solving optimisation by the heuristic algorithm. Since one of the goals of our designed energy scheduling tool is to help a prosumer improve the computational efficiency at the deploying phase through replacing the heuristic algorithm with the designed neural networks, the time used by the manufacturer at the deploying phase does not affect the efficacy for prosumers' energy scheduling. As shown in Table 4, with the increase of the system scale including the numbers of generators and loads, the computational time of the heuristic algorithm dramatically increases. On the contrary, once the proposed neural networks are trained, it only requires microseconds to predict the optimal scheduling decisions, irrespective the increase of system scales. This is because the neural networks can generalise high-complexity problems to extract feature representations.

Table 4 Evaluation of scalability and computational time under various IEEE test distribution systems

IEEE Test System	Average Computational Time Per Prosumer (s)			
	Training Phase		Deploying Phase	
	Proposed Model	Heuristic Algorithm	Proposed Model	Heuristic Algorithm
18-bus	1006.98	-	0.00043	984.79
33-bus	2200.17	-	0.0007	2177.93
69-bus	3475.806	-	0.00038	3462.32

6. Conclusions

This paper proposes a data-driven prosumer-centric energy scheduling model through using the machine learning approach to improve the computational efficiency and scalability. By generating scenarios to analyse variations of uncertainties and using the CNNs to extract the pricing patterns, the optimal scheduling decisions are automatically made by the trained neural networks with the strategy of minimising costs and carbon emissions. Case studies based on various IEEE test distribution systems demonstrate that the designed neural networks outperform other learning approaches in terms of the testing accuracy. The information of pricing patterns from elasticity arrays is a crucial factor for improving the learning accuracy. The connection between the pricing patterns and potential scheduling decisions has been testified. The proposed real-time scenarios selection approach is able to accurately capture the current features of uncertainties by using the information from a prosumer's real-time data and generated scenarios. For future works, once the pricing patterns of prosumers are captured by the market operator, it would be possible to formulate a corresponding decentralised pricing scheme targeting on each individual prosumer.

Appendix A. Solution algorithm for the preferences optimisation

According to Eq. (10), the decision variable $\Delta p_{s,t,x}$ can be substituted by $\Delta p_{\text{utility},t,x}$, $\Delta p_{i,t,x}$, and $\Delta p_{k,t,x}$. The inequality constraints in Eqs. (11) and (12) can be substituted by an additional function as

$$
\begin{aligned}
f_a\left(\Delta p_{\text{utility},t,x}, \Delta p_{i,t,x}, \Delta p_{k,t,x}\right) := & \max\left\{-s_{t,x}, 0\right\} \\
& + \max\left\{s_{t,x} - s^{\max}, 0\right\} \\
& + \max\left\{\left|\Delta p_{s,t,x}\right| - \Delta p_s^{\max}, 0\right\},
\end{aligned}
\tag{A.1}
$$

where $f_a(\cdot)$ is the function substituting the inequality constraints in Eqs. (11) and (12). A solution satisfies the inequality constraints in Eqs. (11) and (12) if and only if $f_a = 0$.

The vector-valued decision variables and objective functions are defined as Eqs. (A.2) and (A.3), respectively, for facilitating the discussion of the solution.

$$
\Delta\mathbf{p}_x = \left[\Delta p_{\text{utility},t,x}, \Delta p_{i,t,x}, \Delta p_{k,t,x} | i \in \mathcal{I}, k \in \mathcal{K}, t \in \mathcal{T}\right], \tag{A.2}
$$

$$
\mathbf{f}_x = [f_c(\Delta\mathbf{p}_x), f_e(\Delta\mathbf{p}_x), f_a(\Delta\mathbf{p}_x)], \tag{A.3}
$$

where $\Delta\mathbf{p}_x$ is a row vector to denote the decision variables of the scenario x, and $\mathbf{f}_x$ is a row vector to denote the objective functions and constraints of the scenario x. Additionally, the lower bounds and upper bounds of the decision variables as described in Eqs. (14), (15), and (16) are denoted by vectors $\underline{\mathbf{p}}$ and $\overline{\mathbf{p}}$, respectively.

The definitions with respect to the artificial immune system and Pareto optimality are introduced as:

• *Definition 1 (Antigen-antibody)*: A random vector $\mathbf{p}$ in the decision variable space $\left[\underline{\mathbf{p}}, \overline{\mathbf{p}}\right]$ is termed as an antigen. The corresponding objective function $\mathbf{f}(\mathbf{p})$ is termed as an antibody. All vectors generated from the decision variable space form an antigen population as

$$
\mathcal{A} = \left\{\mathbf{p}_1, ..., \mathbf{p}_{|\mathcal{A}|}\right\}, \tag{A.4}
$$

where $\mathcal{A}$ is the set of the antigen population, and $|\mathcal{A}|$ is the number of antigens in this population.

• *Definition 2 (Clone and mutation)*: The clonal process enables more antigens to be reproduced over the decision variable space $\left[\underline{\mathbf{p}}, \overline{\mathbf{p}}\right]$. Through preserving the diversity of antigens, the entire feasible space of decision variables can be searched to ensure the global optimal solution. The amount of reproduced antigens can be described by the clonal rate as

$$
r_c := \left\lfloor \frac{|\mathcal{A}^{\max}|}{|\mathcal{A}|} \right\rfloor, \tag{A.5}
$$

where r_c is the clonal rate, $|\mathcal{A}^{\max}|$ is the maximum number of antigens in the population, and $\lfloor \cdot \rfloor$ is the floor function. Hence, each original antigen in (A.4) is cloned by $(r_c - 1)$ antigens through the mutation process to form the set of the clonal antigen population as

$$
\mathcal{A}_c = \left\{\mathbf{p}_1^1, ..., \mathbf{p}_1^{r_c-1}, ..., \mathbf{p}_{|\mathcal{A}|}^1, ..., \mathbf{p}_{|\mathcal{A}|}^{r_c-1}\right\}, \tag{A.6}
$$

where $\mathscr{A}_c$ is the set of the clonal antigen population, in which each mutant can be calculated as: $\vartheta \cdot \mathbf{p} + (1 - \vartheta) \cdot \mathbf{p}'$, where $\vartheta \in [0, 1]$ is a random number, and $\mathbf{p}'$ is a random vector in the decision variable space $\left[\underline{\mathbf{p}}, \overline{\mathbf{p}}\right]$. Through the clone and mutation process, the antigen population becomes $\mathscr{A}^{\max} = \mathscr{A} \cup \mathscr{A}_c$.

- *Definition 3 (Pareto dominance)*: A vector of the objective function $\mathbf{f}(\mathbf{p}_a)$ dominates another vector of the objective function $\mathbf{f}(\mathbf{p}_b)$ in the decision variable space $\mathbf{p}_a, \mathbf{p}_b \in \left[\underline{\mathbf{p}}, \overline{\mathbf{p}}\right]$, denoted as $\mathbf{f}(\mathbf{p}_a) \preceq \mathbf{f}(\mathbf{p}_b)$, if $f(p_a) \leq f(p_b)$, $\forall f(p_a) \in \mathbf{f}(\mathbf{p}_a)$, $f(p_b) \in \mathbf{f}(\mathbf{p}_b)$ holds true and at least one inequality is strict. The vector $\mathbf{f}(\mathbf{p}_b)$ is termed as the dominated antibody.

- *Definition 4 (Pareto optimal solution)*: A vector of the decision variable $\mathbf{p}^* \in \left[\underline{\mathbf{p}}, \overline{\mathbf{p}}\right]$ is a Pareto optimal solution, if its objective function $\mathbf{f}(\mathbf{p})$ dominates all objective functions of any other feasible decision variables in $\left[\underline{\mathbf{p}}, \overline{\mathbf{p}}\right]$.

- *Definition 5 (Pareto optimal set and Pareto frontier)*: The set of all Pareto optimal solutions is termed as the Pareto optimal set, denoted as $\mathcal{P} = \{\mathbf{p}^*\}$. The graphical presentation of the objective functions of the Pareto optimal solutions in the Pareto optimal set is termed as the Pareto frontier.

The proposed algorithm is performed over the entire scheduling horizon $|\mathcal{T}|$ for the following day. During the operation of the artificial immune algorithm, the antigens are randomly generated and cloned to explore the entire decision variable space. In each iteration, the dominated antigen-antibody pairs are removed to keep the non-dominated antigen-antibody pairs. Until the iteration ends, the antigens of all non-dominated antibodies form the optimal solution. The results serve as a set of Pareto optimal solution in the Pareto frontier that achieves a trade off between the cost saving and carbon emissions reduction.

For comparing the results of the MOP, a criteria in [24] is used to select a representative solution from the Pareto frontier. An optimal solution that maximizes the minimum improvement (after normalisation) of all objective functions is selected as the representative solution as

$$\mathbf{f}_{\text{rep}} = \max_{\mathbf{p} \in \mathscr{A}} \min_{f \in \mathbf{f}} \frac{\overline{f} - f(\mathbf{p})}{\overline{f} - \underline{f}}, \tag{A.7}$$

where $\mathbf{f}_{\text{rep}}$ is the vector of representative objective functions form the Pareto frontier, $\overline{f}$ and $\underline{f}$ are the minimum and maximum values of each objective function.

Let ι_{IA} and $\iota_{\text{IA}}^{\max}$ denote the nominal and maximum numbers of iterations of the proposed algorithm, respectively. The pseudocode code of the algorithm for solving the preferences optimisation problem is shown in **Algorithm 3**.

Appendix B. IEEE test distribution systems

To modify the static default data of generation and consumption from these IEEE test distribution systems as dynamic data, the real-time states of the GB power generation and consumption in 2019 from the GridWatch [36] are used. The ratio of peak real-time consumption from the GB power systems to the peak static consumption from these IEEE test distribution systems is used to scale down the GB real-time consumption and

---

Algorithm 3 Solution of the preferences optimisation problem
---
input: scenario set $p_{i,t,x}$, $p_{k,t,x}$, and $\pi_{t,x}$, $\forall x \in \mathcal{X}$, nominal and maximum number of antigens in the population of decision variables $|\mathscr{A}|$ and $|\mathscr{A}^{\max}|$, respectively
1: **for** $x = 1, ..., |\mathcal{X}|$ **do**
2:     randomly initialise the antigen population within the decision variable space $\left[\underline{\mathbf{p}}, \overline{\mathbf{p}}\right]$ as $\mathscr{A}(0) = \Delta\mathbf{p}_{x,1}, ..., \Delta\mathbf{p}_{x,|\mathscr{A}|}$
3:     **while** $\iota_{\text{IA}} \leq \iota_{\text{IA}}^{\max}$ **do**
4:         implement the clone and mutation operation according to Eq. (A.6), and the number of current antigens $|\mathscr{A}(\iota_{\text{IA}})|$ increases to $|\mathscr{A}^{\max}|$
5:         remove dominated antibodies and corresponding antigens from $\mathscr{A}(\iota_{\text{IA}})$
6:         **while** $|\mathscr{A}(\iota_{\text{IA}})| > |\mathscr{A}|$, or $f_a(\Delta\mathbf{p}_x) \neq 0$ **do**
7:             remove the antigen-antibody pairs with the highest positive value of $f_a(\Delta\mathbf{p}_x)$
8:             remove the antigen-antibody pairs with small avidities according to [35], i.e., remove the vectors of objective function in a crowded region
9:         **end while**
10:         $\mathscr{A}(\iota_{\text{IA}} + 1) = \mathscr{A}(\iota_{\text{IA}})$, $\iota_{\text{IA}} = \iota_{\text{IA}} + 1$
11:     **end while**
12:     remove the antigen-antibody pairs that yield $f_a(\Delta\mathbf{p}_x) > 0$
13:     select a representative solution according to Eq. (A.7)
14: **end for**
output: optimal solution $\Delta\mathbf{p}_x^*$

---

generation of diesel, solar, wind, and biomass. The percentages of consumption of each load in the IEEE test distribution systems are used to allocate the total dynamic consumption to each load. The details of the modified IEEE test distribution systems are provided as follows:

- *Case 1 (Modified IEEE 69-bus distribution network):* The schematic illustration of the modified IEEE 69-bus distribution network is presented in Fig. B.13. The network is partitioned into 5 prosumers. 15 solar photovoltaics, 6 diesel generators, 4 wind turbines, and 3 biomass generators are arbitrarily assigned to each prosumer, and 69 loads are assigned to each bus.

- *Case 2 (Modified IEEE 33-bus distribution network):* The schematic illustration of the modified IEEE 33-bus distribution network is presented in Fig. B.14. The network is partitioned into 3 prosumers. 12 solar photovoltaics, 3 diesel generators, 3 wind turbines, and 1 biomass generator are arbitrarily assigned to each prosumer, and 33 loads are assigned to each bus.

- *Case 3 (Modified IEEE 18-bus distribution network):* The schematic illustration of the modified IEEE 18-bus distribution network is presented in Fig. B.15. The network is partitioned into 3 prosumers. 7 solar photovoltaics, 3 diesel generators, 2 wind turbines, and 1 biomass generator are arbitrarily assigned to each prosumer, and 18 loads are assigned to each bus.

References

[1] Y. Parag, B. K. Sovacool, Electricity market design for the prosumer era, Nat Energy 1 (4) (2016) 1–6.
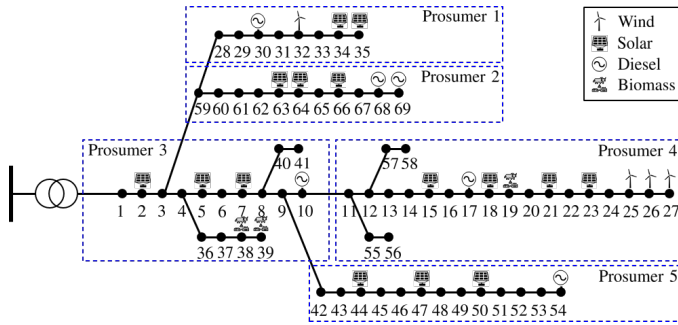
Fig. B.13. Modified IEEE 69-bus distribution network. The network is partitioned into 5 prosumers. 15 solar photovoltaics, 6 diesel generators, 4 wind turbines, and 3 biomass generators are arbitrarily assigned to each prosumer, and 69 loads are assigned to each bus.
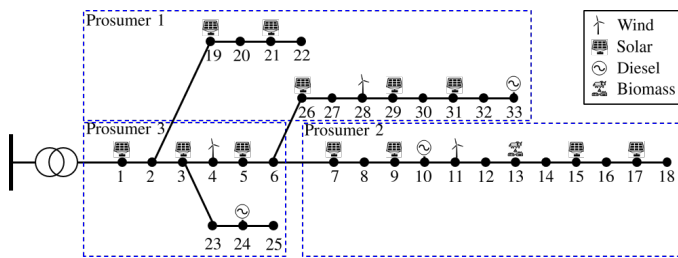

Fig. B.14. Modified IEEE 33-bus distribution network. The network is partitioned into 3 prosumers. 12 solar photovoltaics, 3 diesel generators, 3 wind turbines, and 1 biomass generator are arbitrarily assigned to each prosumer, and 33 loads are assigned to each bus.

Fig. B.15. Modified IEEE 18-bus distribution network. The network is partitioned into 3 prosumers. 7 solar photovoltaics, 3 diesel generators, 2 wind turbines, and 1 biomass generator are arbitrarily assigned to each prosumer, and 18 loads are assigned to each bus.

[2] V. Venizelou, N. Philippou, M. Hadjipanayi, G. Makrides, V. Efthymiou, G. E. Georghiou, Development of a novel time-of-use tariff algorithm for residential prosumer price-based demand side management, Energy 142 (2018) 633–646.

[3] X. Le, S. Chen, Z. Yan, J. Ping, X. Zhao, N. Zhang, J. Xi, Enabling a transactive distribution system via real-time distributed optimization, IEEE Trans Smart Grid 10 (5) (2019) 4907–4917.

[4] X. Chen, B. Liu, J. Qiu, W. Shen, L. Reedman, Z. Y. Dong, A new trading mechanism for prosumers based on flexible reliability preferences in active distribution network, Appl Energy 283 (2021) 116272.

[5] B. J. Claessens, P. Vrancx, F. Ruelens, Convolutional neural networks for automatic state-time feature extraction in reinforcement learning applied to residential load control, IEEE Trans Smart Grid 9 (4) (2018) 3259–3269.

[6] J. M. Zepter, A. Lüth, P. C. Del Granado, R. Egging, Prosumer integration in wholesale electricity markets: Synergies of peer-to-peer trade and residential storage, Energy Buildings 184 (2019) 163–176.

[7] J. Iria, F. Soares, M. Matos, Optimal bidding strategy for an aggregator of prosumers in energy and secondary reserve markets, Appl Energy 238 (2019) 1361–1372.

[8] H. Ahn, D. Rim, G. S. Pavlak, J. D. Freihaut, Uncertainty analysis of energy and economic performances of hybrid solar photovoltaic and combined cooling, heating, and power (cchp+ pv) systems using a montecarlo method, Appl Energy 255 (2019) 113753.

[9] M. J. Santos, P. Ferreira, M. Araújo, A methodology to incorporate risk and uncertainty in electricity power planning, Energy 115 (2016) 1400–1411.

[10] G. Mavromatidis, K. Orehounig, J. Carmeliet, Design of distributed energy systems under uncertainty: A two-stage stochastic programming approach, Appl Energy 222 (2018) 932–950.

[11] Z. Li, Y. Xu, Temporally-coordinated optimal operation of a multi-energy microgrid under diverse uncertainties, Appl Energy 240 (2019) 719–729.

[12] K. Mahmud, B. Khan, J. Ravishankar, A. Ahmadi, P. Siano, An internet of energy framework with distributed energy resources, prosumers and small-scale virtual power plants: An overview, Renew Sust Energ Rev 127 (2020) 109840.
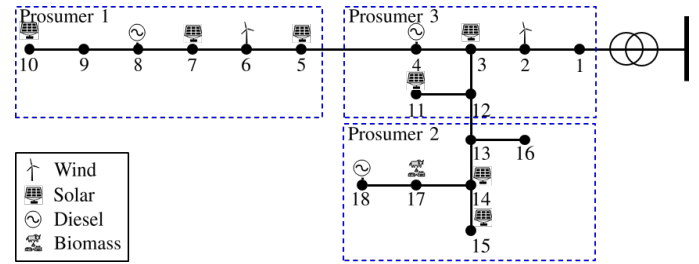
[13] A. Nutkiewicz, Z. Yang, R. K. Jain, Data-driven urban energy simulation (due-s): A framework for integrating engineering simulation and machine learning methods in a multi-scale urban energy modeling workflow, Appl Energy 225 (2018) 1176–1189.

[14] S. Yang, M. P. Wan, W. Chen, B. F. Ng, S. Dubey, Experiment study of machine-learning-based approximate model predictive control for energy-efficient building control, Appl Energy 288 (2021) 116648.

[15] E. Mocanu, D. C. Mocanu, P. H. Nguyen, A. Liotta, M. E. Webber, M. Gibescu, J. G. Slootweg, On-line building energy optimization using deep reinforcement learning, IEEE Trans Smart Grid 10 (4) (2019) 3698–3708.

[16] Y. Du, F. Li, J. Li, T. Zheng, Achieving 100x acceleration for n-1 contingency screening with uncertain scenarios using deep convolutional neural network, IEEE Trans Power Syst 34 (4) (2019) 3303–3305.

[17] B. W. Silverman, Density estimation for statistics and data analysis (2018).

[18] V. A. Epanechnikov, Non-parametric estimation of a multivariate probability density, Theory Probab Its Appl 14 (1) (1969) 153–158.

[19] M. Stein, Large sample properties of simulations using latin hypercube sampling, Technometrics 29 (2) (1987) 143–151.

[20] R. M. Ayers, R. A. Collinge, Microeconomics: Explore and Apply and Companion Website PLUS, Prentice Hall, 2003.

[21] Electricity generation costs, Tech. rep., Department for Business, Energy and Industrial Strategy (Nov. 2016).

[22] O. Schmidt, S. Melchior, A. Hawkes, I. Staffell, Projecting the future levelized cost of electricity storage technologies, Joule 3 (1) (2019) 81 – 100.

[23] A. Castillo, D. F. Gayme, Grid-scale energy storage applications in renewable energy integration: A survey, Energy Conversion and Management 87 (2014) 885–894.

[24] W. Chiu, H. Sun, H. Vincent Poor, A multiobjective approach to multimicrogrid system design, IEEE Trans Smart Grid 6 (5) (2015) 2263–2272.

[25] F. Meng, X. Zeng, A profit maximization approach to demand response management with customers behavior learning in smart grid, IEEE Trans Smart Grid 7 (3) (2016) 1516–1529.

[26] L. Lazard, Lazards levelized cost of storage analysis, version 1.0 (2015).

[27] https://www.group.rwe (Mar. 2020).

[28] A. D. Hawkes, Estimating marginal CO_2 emissions rates for national electricity systems, Energy Policy 38 (10) (2010) 5977–5987.

[29] A. Jain, K. Nandakumar, A. Ross, Score normalization in multimodal biometric systems, Pattern Recog 38 (12) (2005) 2270–2285.

[30] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[31] G. E. Hinton, A. Krizhevsky, I. Sutskever, N. Srivastva, System and method for addressing overfitting in a neural network, US Patent 9,406,017 (Aug. 2 2016).

[32] E. L. Lehmann, G. Casella, Theory of point estimation, Springer Science & Business Media, 2006.

[33] D. A. Reynolds, T. F. Quatieri, R. B. Dunn, Speaker verification using adapted gaussian mixture models, Digit Signal Process 10 (1-3) (2000) 19–41.

[34] C. J. Willmott, K. Matsuura, Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance, Clim Res 30 (1) (2005) 79–82.

[35] R. Shang, L. Jiao, F. Liu, W. Ma, A novel immune clonal algorithm for mo problems, IEEE Trans Evolut Comput 16 (1) (2012) 35–50.

[36] `https://www.gridwatch.templar.co.uk/download.php` (Mar. 2020).