CIRP Journal of Manufacturing Science and Technology 38 (2022) 844–860

Contents lists available at ScienceDirect

CIRP Journal of Manufacturing Science and Technology

journal homepage: www.elsevier.com/locate/cirpj

An integrated energy management system using double deep Q-learning and energy storage equipment to reduce energy cost in manufacturing under real-time pricing condition: A case study of scale-model factory

Li Yi[a,*], Pascal Langlotz[a], Marco Hussong[a], Moritz Glatt[a], Fábio J.P. Sousa[b], Jan C. Aurich[a]

[a] Institute for Manufacturing Technology and Production Systems, Technical University of Kaiserslautern, Germany
[b] School of Sciences and Technology, Laboratory of Instrumentation and Robotics, Federal University of Rio Grande do Norte, Brazil

ARTICLE INFO

Available online 22 July 2022

Keywords:
Energy cost
Energy management system
Reinforcement learning
Double deep Q-learning
Manufacturing system
Real-time pricing

ABSTRACT

Reducing energy costs is an emerging aspect in the research on the economic and environmental dimensions of manufacturing systems. The share of electricity cost accounts for approximately 60 % of the total energy cost of a manufacturing system, whereas the share of oil, coal, and gas accounts for the remaining 40 %. The electricity cost is dependent on the electricity price and usage. In terms of the electricity price, one of the pricing strategies widely used in the USA and Europe is called real-time pricing (RTP), which is characterised by hourly price changes. Compared to other pricing strategies, RTP yields the highest reward and the highest risk. In the RTP strategy, the electricity price is influenced by the supply and demand of the energy market. Hence, the energy cost of manufacturing cannot be determined by the manufacturing companies, implying a high level of risk. However, if manufacturing companies seize the opportunity to perform more manufacturing tasks when the energy price is low, the cost-savings will be significant, implying a high level of reward. In this study, we propose an integrated energy management system (IEMS) to reduce the energy cost of manufacturing systems. The IEMS consists of an energy storage equipment and an intelligent switch mechanism. When the electricity price is high, the manufacturing system is powered by the energy storage equipment. When the electricity price is low, the manufacturing system is powered by the public electricity grid, and the energy storage equipment is charged. The decision-making of these operations is performed by the intelligent switch mechanism based on double deep Q-learning. To validate this framework, a case study is conducted, in which an IEMS is developed to reduce the electricity cost of a scale-model factory. Based on an online test of the IEMS in different manufacturing cycles, it is concluded that the proposed IEMS approach achieves a cost reduction of approximately 57.21 %.

Introduction

In manufacturing, four general optimisation objectives are quality, time, flexibility, and costs [1]. The costs of a manufacturing system consist of staff costs, material costs, energy costs, and other relevant cost factors [2]. Since energy cost accounts for a large share of the cost in the manufacturing system, reducing energy costs has a significant impact on the cost structure of manufacturing companies. Energy costs are related to the consumption of energy from different sources, e.g., electricity, gas, oil, and coal. The most promising strategy to reduce the cost on energy expense is to reduce the electricity cost, as it accounts for approximately 60 % of the total energy cost [3].

In addition to the electricity consumption, the electricity price is a critical factor that influences the energy cost. In the USA and Europe, three main dynamic pricing strategies are widely adopted [4–6]: (1) time-of-use (TOU), in which the daily electricity price differs between peak and off-peak periods; (2) critical peak pricing (CPP), in which the electricity price is dependent on the peak power consumption of a system, which means that a company who has a higher power consumption from the public electricity grid needs to pay more cost than those companies who require a same quantity of electricity but with a lower and homogenous power consumption; (3) real-time pricing (RTP), in which the electricity price varies over time. These strategies are depicted in Fig. 1(a), (b), and (c).

Several studies on the TOU strategy have been conducted (as explained in *Energy cost of manufacturing systems*), whereas studies

* Corresponding author.
E-mail address: li.yi@mv.uni-kl.de (L. Yi).

https://doi.org/10.1016/j.cirpj.2022.07.009
1755-5817/

Nomenclature

a_t	selected action in decision period *t*.
A	set of all possible actions.
I	current.
L	loss function.
p_t	electricity price for decision period t.
p^{med}	median of electricity prices.
$p^{97,5\%}$	97.5 %-percentile of electricity prices.
$p^{2,5\%}$	2.5 %-percentile of electricity prices.
$\hat{p}_t$	electricity price ratio for decision period t.
P_t	power for decision period t.
$\hat{P}_t$	predicted power for decision period t.
$\hat{P}_{PI1}$	predicted power for power interface 1.
$\hat{P}_{PI2}$	predicted power for power interface 2.
$\hat{P}_{PI3}$	predicted power for power interface 3.
q_n	nominal charge of battery.
q_t	electric charge for decision period t.
Q	Q-value.
r_t	reward in decision period *t*.
R	reward function.
s_t	states of decision period t.
S	state space.
SOC_t	state of charge at decision period t.
t	index for decision period.
t_0	manufacturing cycle time.
t_1	prediction length.
t_2	measurement sample length.
T	transition probability.
V	value function.
w_L	weights of learning network.
w_T	weights of target network.
α	learning rate.
γ	discount factor.
ϵ	threshold for random action.
η_{est}	efficiency of battery.
κ_0	battery factor.
κ_1	battery factor.
κ_2	battery factor.
$\bar{\kappa}$	upper threshold for battery usage.
$\underline{\kappa}$	lower threshold for battery usage.
π	policy function.
τ	index for measurement samples.

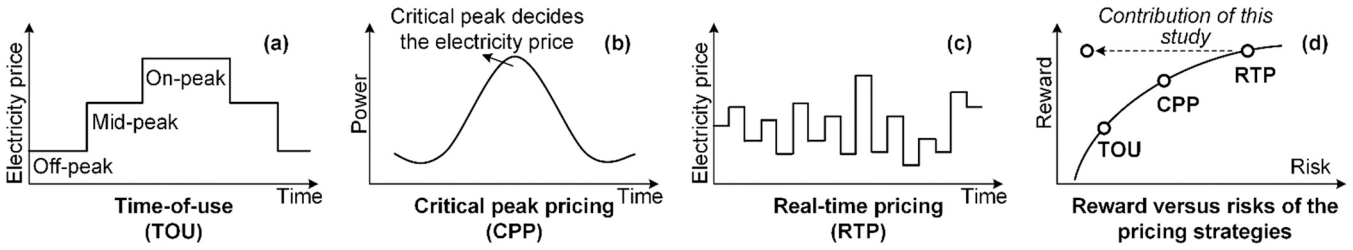

Fig. 1. The three main pricing strategies (own illustration based on [4–6]).

on RTP and CPP strategies are limited. Nevertheless, the RTP strategy is essential for manufacturing companies. Compared to the TOU and CPP strategies, the RTP strategy has the highest reward potential and the highest risk, as depicted in Fig. 1(d) [4]. In the TOU strategy, the electricity cost within a specific period is constant; therefore, the general strategy of manufacturing companies is to avoid massive manufacturing tasks during the on-peak hours [7], and it is more suitable for companies whose total energy use is relatively low [8]. In the CPP strategy, the risk is increased as manufacturing companies cannot know the exact electricity cost in advance if their power consumption is increasing rapidly [4]. However, manufacturing companies can still use appropriate scheduling strategies to avoid high peak power consumption [9]. If the critical power peak is reduced, the electricity price is reduced accordingly, indicating a higher level of reward than in the TOU strategy. RTP is the most sophisticated pricing strategy, in which the price is defined almost instantaneously and depends on the supply and demand of the market; therefore, the price varies hourly [6]. As price prediction is difficult for manufacturing companies, it can be said that RTP shows the highest level of risk [6]. However, if manufacturing companies can seize the opportunity to perform more manufacturing tasks at a lower electricity price, the cost-savings will be significant [4,5]. Therefore, owing to the successful implementation of machine learning techniques in interdisciplinary areas, reinforcement learning, which is a subcategory of machine learning, is used in this study to reduce the energy cost of manufacturing systems. As indicated in Fig. 1(d), if the objective is achieved, the contribution of this study would be that the RTP point on the reward-risk plane is moved from the right to the left, where the risk is minimised, and the high reward is maintained.

In this study, an integrated energy management system (IEMS) is proposed, and it consists of two parts: an energy storage equipment and an intelligent switch mechanism. The IEMS determines the power source according to different electricity prices. When the electricity price is high, the energy storage equipment supplies the power to the manufacturing system, as depicted in Fig. 2(a). When the electricity price is low, the energy storage equipment charges itself from the public electricity grid as needed, and the manufacturing system takes power from the public electricity grid, as depicted in Fig. 2(b). To decide whether the price is high or low, the intelligent switch mechanism analyses historical price data. Moreover, it is important to mention that our IEMS only controls the energy storage equipment and is not intended to change the activities within the manufacturing system or the configuration of the public electricity grid.

The scenarios in Fig. 2 describe only the general idea in this study. Specifically, the entire framework is based on the double deep Q-learning (DDQN) algorithm. To validate the framework, a case study is conducted, in which the energy cost of a scale-model factory is optimised. All these methods and results are explained in the following sections. The remainder of this paper is structured as follows: *Theoretical background* introduces the research background and the existing approaches related to the energy cost reduction of manufacturing systems; *Methodological formulation of the DDQN-based IEMS approach* describes the general methodological formulation of the DDQN-based IEMS for reducing energy costs using the energy storage equipment; *Development of a specific DDQN-based*

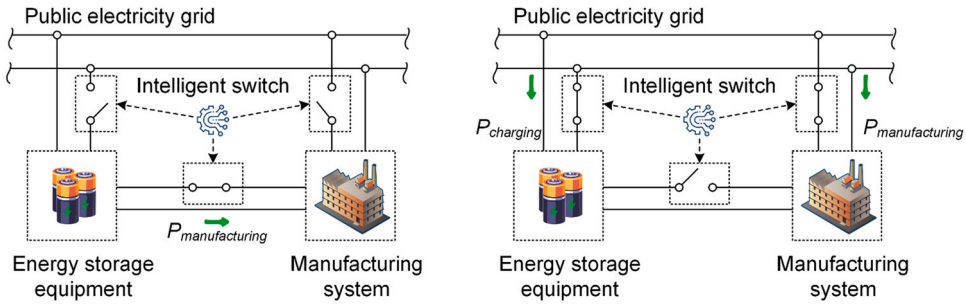

Fig. 2. Conceptual framework of the study.

IEMS for a case study of a scale-model factory explains the development of the specific DDQN-based IEMS for the case study of the scale-model factory; *Results and discussion of the case study* presents the results of the case study; *Conclusion and outlook* summarises the paper with a brief conclusion.

Theoretical background

Energy cost of manufacturing systems

In general, the electricity cost is calculated as the product of electricity use and price. As electricity pricing policies may not be decided directly by manufacturing companies, the electricity cost reduction strategies usually consider the electricity price as the boundary condition and intend to reduce electricity use. The strategies for the reduction of energy use in manufacturing systems can be classified into three main types [3]: (1) improvements in machines and processes (IMP), e.g., reuse of waste heat and optimisation of processing parameters; (2) production planning and control (PPC), e.g., rescheduling of the production tasks to avoid peak hours; (3) improvement of the supporting processes and technical building services (IST), e.g., reduction of transportation and storage or switching off the lights when no production tasks are being performed.

Based on the literature review of related studies, the approaches for energy cost reduction have been identified and summarised in Table 1. The following five findings are observed: (1) The cost reduction issues are mainly formulated as multi-objective optimisation problems and can be solved by applying meta-heuristic approaches [9–11]; (2) In most approaches, the cost reduction issues are studied under the TOU scenarios, whereas the approaches for the RTP and CPP strategies are limited [9]; (3) in terms of energy-saving strategies, most approaches are focused on the PPC, whereas the approaches aimed at the IST and IMP are limited [11,12]; (4) in the approaches related to PPC, the cost reduction is always regarded as a job scheduling problem, indicating that researchers prefer to focus on the 'design' rather than the 'execution' phase [7,9,10,12,13]; (5) in terms of the scheduling problems within the PPC, reinforcement learning has been used [14,15].

Compared to other studies, our study demonstrates the following differences that can be regarded as innovative: (1) in terms of problem solving approach, we have chosen reinforcement learning, whereas most other approaches have adopted meta-heuristics; (2) in terms of electricity pricing strategies, our approach is focused on RTP, whereas other approaches mainly focus on TOU; (3) in terms of energy-saving strategies, our approach is aimed at ISP, whereas other approaches mainly aim at PPC; (4) our approach is implemented when the manufacturing tasks are performed (execution phase), whereas other approaches are employed during the task scheduling phase (design phase). The reasons for these differences are based on the following assumptions:

- We chose reinforcement learning instead of meta-heuristics because meta-heuristic is not appropriate for the RTP scenario. Meta-heuristic approaches demonstrate a significant high dependence on domain expertise to build exact mathematical models [14], which is especially challenging for stochastically volatile problems, as one of them is the RTP problem in our work. In other words, a manufacturing system is impossible or only possible with a significant number of efforts to precisely predict energy prices in their meta-heuristics. Additionally, meta-heuristics are not suitable for online application due to the time-consuming iterative solving strategy, and the possible failure of convergence of the algorithm under certain parameter combinations.
- Integrated Energy Systems (IES) refer to energy systems that include multiple subsystems associated with the generation, distribution, and storage of multiple energy forms [17]. It is evident that IES have been successfully implemented in multidisciplinary areas [18]. We regard our approach, called 'Integrated Energy Management System (IEMS)', as a subcategory of IES, with the term 'management' emphasise the controlling of the energy distribution under different price conditions.
- Energy storage equipment has been widely used in interdisciplinary areas related to IES as well as IEMS and has the potential for energy cost reduction, as explained in *Energy storage equipment for energy saving*.
- RTP is the most challenging pricing strategy, with the greatest risks and rewards. As mentioned in *Introduction*, if the risks of RTP can be minimised using reinforcement learning, the remaining rewards will be the highest compared to TOU and CPP.
- The DDQN uses two artificial neural networks (ANNs) to minimise the problem of overoptimistic value estimation, which can occur with deep Q-learning due to a continuously changing target [19]. Considering the high uncertainty of prices under the RTP, the DDQN should have a higher level of prediction stability.
- The energy reduction using PPC and IMP is associated with changes in manufacturing activities (e.g. changes in processing parameters and rescheduling of production jobs), which may require the adaptation of the manufacturing processes and systems, e.g., new manufacturing control plans and new machine layouts. These changes will lead to more time and cost investment in operations. In ISP, only peripheral processes are adapted, and production activities remain constant.
- For the RTP strategy, the execution phase is more important than the design phase. The reason for this is that the price within the RTP scenario is defined instantaneously (as described in *Introduction*) and can neither be assumed in the job scheduling to be fixed for different periods, as in the TOU strategy, nor be

Table 1
Existing approaches related to the energy cost reduction of manufacturing systems.

References	Research problem	Energy-saving strategy	Pricing strategies	Problem-solving approaches
[10]	Reschedule deteriorating jobs to minimise energy cost	PPC	TOU	Variable neighbourhood search algorithm
[12]	Schedule microgrids energy management system to reduce energy cost and carbon emission	IST	Not specified	Lightning search algorithms
[11]	Multi-objective optimisation of the processing parameters of a CNC machine to reduce energy use and cost	IMP	Assumed as a constant	Genetic algorithms and particle swarm optimisation
[16]	Use matrix-based tools to investigate, classify and reduce energy loss and cost	PPC and IST	Not specified	Matrix-based evaluation tools
[9]	Schedule production jobs of a grinding process to reduce energy cost	PPC	TOU, CPP, and RTP	Genetic algorithms
[13]	Schedule production jobs to reduce the energy cost of three collaborative factories	PPC	TOU	GAMS/CPLEX solver
[7]	Schedule production jobs with eight processes to reduce energy cost and emission	PPC	TOU	Time indexed integer programming
[15]	Machine control for production tasks in response to dynamic electricity price changes to minimise energy cost in lithium-ion battery assembly system	PPC	Not specified	Multi-agent deep deterministic policy gradient
[14]	Scheduling of energy storage system to minimise the energy cost in micro-grid system.	PPC	Not specified	Double deep Q-learning
Our study	Use reinforcement learning and an energy storage-integrated energy management system to enable the intelligent switch of the energy supply for a factory to reduce energy cost	IST	RTP	Double deep Q-learning

described using analytical or empirical models of power consumption as in the CPP strategy.

In summary, our study objective is to use DDQN and an energy storage equipment to develop an IEMS for energy cost reduction of manufacturing systems under the RTP scenario, as outlined in the last row of Table 1.

Energy storage equipment for energy saving

Energy storage is a key component of IEMS and is defined as an energy technology facility for storing energy in the form of internal, potential, or kinetic energy using energy storage equipment [20]. In general, energy storage equipment should be able to perform at least three operations: charging (loading energy), storing (holding energy), and discharging (unloading energy) [20]. Energy storage equipment can be categorised into electrical, chemical, mechanical, thermal, and electrochemical types based on different physical principles [20,21]: (1) electrical storage equipment is used to store electricity in electrostatic fields or magnetic fields, e.g., bi-layer capacitors, superconducting coils, and permanent magnets; (2) chemical storage equipment is implemented for storing primary or secondary energy converted into energy carriers, e.g., hydrogen, natural gas, and biogas; (3) mechanical storage equipment stores energy in gaseous, liquid, or solid media associated with a specific position (potential), speed (kinetic), or thermodynamic state (pressure), e.g., flywheels and springs; (4) thermal storage equipment stores heat using, for instance, heat capacity or thermo-reversible chemical reactions, e.g., water tanks and reversible reactors; (5) in electrochemical storage equipment, energy is stored in chemical form in the electrode materials, e.g., batteries and accumulators. Due to the different physical principles used, the technologies exhibit varying energy densities. Energy density describes the storage capability of energy storage equipment [21]. Table 2 shows examples of the energy storage equipment and their energy densities. In addition to the energy density, an important criterion for evaluating energy storage equipment is storage time. In general, electrical storage equipment can only hold electricity for a very short time (short-term), the storage time of mechanical and thermal storage equipment is moderate (medium-term), and chemical and electrochemical storage equipment can hold energy for a relatively long time (long-term) [20].

Energy storage equipment has been applied in many areas, such as power supply, logistics, and manufacturing engineering. In terms of manufacturing engineering, the application of energy storage equipment is mainly from an environmental perspective, e.g., improving efficiency to reduce heat waste, reducing fossil fuel use, and increasing the power-to-heat ratio [20,22]. Nevertheless, from an economic perspective, the application of energy storage equipment has high potential. A paper by McKinsey Sustainability pointed out the four most important applications: demand-charge management, grid-scale renewable power, small-scale solar-plus-storage, and frequency regulation [23]. For demand-charge management, it is evident that charging during off-peak times and discharging during peak times helps customers in the USA to reduce the cost per use from $9/kW to $4–5/kW [23]. Therefore, inspired by this background, we focus on the economic potential of energy storage and consider the energy storage equipment as a core element of the DDQN-based IEMS.

Reinforcement learning

Reinforcement learning is a subfield of machine learning. Compared to supervised and unsupervised learning, which focus on learning patterns or clusters from large datasets [24], reinforcement learning aims to acquire knowledge of a strategy for solving

Table 2
Examples of different energy storage systems and their energy densities [21].

Type	Technology	Energy density in kWh/mm³
Mechanical	Potential energy (e.g., pumped-storage hydroelectricity with 360 m) (electrical energy)	1
	Kinetic energy (e.g., flywheel) (electrical energy)	10
Electrical	Electrostatic field (capacitor) (electrical energy)	10
	Electromagnetic fields (coils) (electrical energy)	10
Electrochemical	Lead-acid battery (electrical energy)	100
	Lithium-ion battery (electrical energy)	500
Thermal	Sensible heat (e.g. water with ΔT = 100 K) (thermal energy)	116
	Phase transition (e.g. from water to steam) (thermal energy)	636
Chemical	Liquid hydrogen (thermal energy)	2400
	Petrol (thermal energy)	8500

optimisation problems in the form of the Markov Decision Process (MDP), which consists of four core elements: states, actions, transition probabilities, and rewards [25]. In the MDP, an *agent* receives a representation of a *state* s_t from an *environment* and performs an *action* a_t at a time step t, e. g. a point in time, when an action has to be performed dependent on the current state. The state describes the specific condition of the agent in that environment at time t, and the set of the states is called *state space S*. The action is defined as a process to change the state of the agent, and the set of actions is called *action space A*. If an action is performed, a *reward* r_t is allocated to the agent and the environment changes to another state with a transition probability T. The reward represents the recognition of the achievement for a state change by an action, and the function for mapping the relationship of rewards to the state changes by actions is called *reward function R*. Therefore, the MDP can be described as a tuple (S, A, T, R). The *function* of choosing an action in a certain state is called the *policy*. Furthermore, the *value function V* is defined as the discounted sum of expected rewards of the states [26]. The goal of the agent is to maximise the expected rewards of different states. Based on this principle, several approaches for reinforcement learning have been developed. We adopt the DDQN approach, which is an extension of Q-learning using two artificial neural networks (ANNs) for the prediction of the Q-values.

In other engineering areas like hybrid electric vehicles or smart electricity grid managements, energy management systems based on energy storage equipment and the Q-learning as well as DDQN approaches are applied. For example, in the work [27], Q-learning is applied to optimise the energy efficiency of a hybrid electric vehicle, while in the work of [14], the DDQN is applied to control the electricity gird according to the uncertainties in the environment. In the context of manufacturing, different reinforcement learning methods including Q-learning are utilised to solve engineering problems. Examples of different applications are optimal preventive maintenance in a manufacturing inventory system [28], job-shop scheduling of various tasks to validate the payloads placed in a cargo bay [29], and an agent-based dynamic job-shop scheduling system [30,31]. In terms of the implementation of reinforcement learning to improve the energy performance in manufacturing, some works are focused on the energy price, e.g. [14,15], while some other works are focused on the energy demand and efficiency, e.g. [32,33]. Nevertheless, to the best of our knowledge, reinforcement learning as well as Q-learning has not been used to reduce the energy costs in manufacturing under the RTP scenario. Therefore, this work focuses on the RTP problem and tries to solve it with the DDQN approach. Moreover, the reason for choosing DDQN approach over the deep Q-learning is that the RTP scenario generates price stochastically and demonstrates a high level of uncertainty. Since DDQN uses two ANN to avoid the problem of overoptimistic value estimation that the deep Q-learning may have due to a continuously changing target [19], the prediction stability of the DDQN approach should be better than that of deep Q-learning.

Methodological formulation of the DDQN-based IEMS approach

In the proposed DDQN-based IEMS approach, an agent is assumed to take over the decision of the power source for a manufacturing system. Thus, action a_t for time step t is selected from action space A, where the action space contains all operation states of the underlying IEMS. For example, in our case which will be described in *Development of a specific DDQN-based IEMS for a case study of a scale-model factory*, the IEMS is based on a rechargeable battery, and the actions that this battery can perform include the 'charging', 'discharging', and 'idle'. For each action, the state of the agent changes due to the environmental dynamics. For the IEMS, state space S can be expressed as follows.

$$S = (SOC_t,\ \hat{P}_t,\ \hat{p}_t)^T \tag{1}$$

In Eq. 1, SOC_t represents the state of charge of the energy storage equipment, $\hat{P}_t$ represents the predicted power requirements of the manufacturing system, and $\hat{p}_t$ represents the electricity price ratio.

In general, SOC_t is defined as the ratio of the energy storage capacity at time step q_t to the nominal capacity q_n. The nominal capacity describes the maximum amount of energy that can be stored by the energy storage equipment. Therefore, SOC_t is defined as follows.

$$SOC_t = \frac{q_t}{q_n} \cdot 100 \quad \% \tag{2}$$

The remaining capacity in the energy storage equipment q_t cannot be measured directly; therefore, various mathematical methods are used to estimate the remaining capacity. If the state of charge is known at time step $t = 0$, the Coulomb counting method can be used to calculate the change in the state of charge of the battery between two successive time steps [34]. Hence, SOC_t is expressed using the following equation.

$$SOC_t = SOC_{t-1} + \frac{\eta_{est}}{q_n} \int_{t-1}^{t} I \quad dt \cdot 100 \quad \% \tag{3}$$

In Eq. 3, η_{est} is the efficiency of the energy storage equipment. To measure the state of charge with high accuracy, the current I is measured frequently between the time steps in which the agent's decision is made. Therefore, a new time variable τ is defined to describe the time intervals at which a measurement is made (denoted as sampling episode τ). Because of the measurements of current I at different sampling episodes, τ, the integral must be discretised. This is achieved by applying the trapezoidal rule; therefore, Eq. 3 can be rewritten as follows.

$$SOC_\tau = SOC_{\tau-1} + \frac{100\% \cdot \eta_{est}}{q_n} \cdot \frac{I_\tau + I_{\tau-1}}{2} \Delta\tau \tag{4}$$

Note that variable τ is only necessary for the implementation of the trained DDQN model because the frequency of power

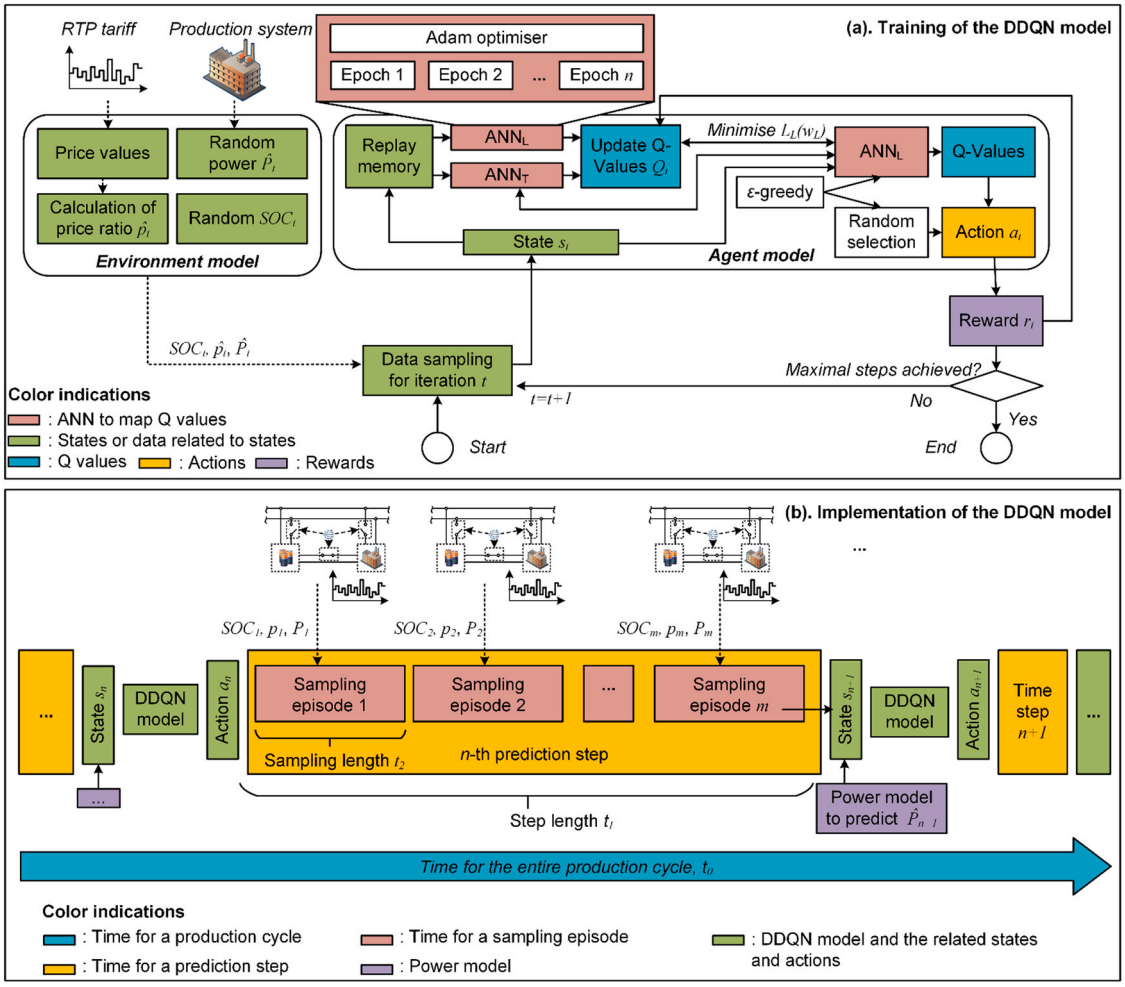

Fig. 3. Training and implementation of the DDQN model.

measurement and the frequency of decisions by the agent are different. For the training process of the DDQN model, variable τ is not required because the SOC values during the training can be set as random values varying from 0 % to 100 %. To predict the power $\hat{P}_t^i$, regression methods or other machine learning methods can be used. For example, in the case study used for this work, the polynomial regression method was applied (see *Development of a specific DDQN-based IEMS for a case study of a scale-model factory*). The price ratio, $\hat{p}_t$, is a statistical variable defined by the following equation.

$$\hat{p}_t = \frac{p_t - p^{med}}{0,5 \cdot (p^{97.5\%} - p^{2.5\%})} \tag{5}$$

In Eq. 5, p_t is the electricity price, and p^{med} describes the median of the past electricity prices. The difference of $p^{97.5\%} - p^{2.5\%}$ is the 95 % percentile distance of the past electricity prices. The price ratio, $\hat{p}_t$, is defined to normalise the values of the electricity prices between − 1 and 1 at times where the electricity price is not extraordinarily high. Therefore, the DDQN model can be trained faster. Furthermore, during the training process, a sufficient amount of extraordinarily high electricity pricing scenarios, out of the range of − 1 and 1 are presented to the model, so that it would be able to learn these cases as well.

Given state s_t and action a_t, the environment transitions to the next state, s_{t+1}, as a reaction of the agent's behaviour with a transition probability. Furthermore, the agent receives reward r_t according to the environment dynamics or reward function R, which is described as follows.

$$r_t = R(s_t, a_t) \tag{6}$$

For different energy storage equipment, A and R should be defined differently. For example, the case study in this work requires a rechargeable battery as the storage equipment; therefore, three actions are defined to describe the different modes of the battery: battery idle, discharging, and charging. Due to the dependency of the reward and the action, the reward function is defined dependent on the possible actions (see *Development of a specific DDQN-based IEMS for a case study of a scale-model factory*).

In Q-learning, the following rule is applied to update the Q value.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)) \tag{7}$$

In a deep Q-learning network (DQN), an ANN is used as a nonlinear function approximator mapping the Q-values to a state-action-pair. The ANN with weights w is referred to as a Q-network. The ANN is trained by minimising the loss function $L(w)$, as expressed by the following equation.

$$L(w) = \mathbb{E}[(r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}, w) - Q(s_t, a_t, w))^2] \tag{8}$$

In DQN, the 'max' operator uses the same weights to select and evaluate actions. This causes the value of the target $r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}, w)$ to change with every training step, since the weights are adjusted during the training of the neural network, resulting in an overoptimistic value estimation. To prevent the estimation of overoptimistic values, the selection and evaluation of actions should be decoupled [19]. The decoupling of selection and

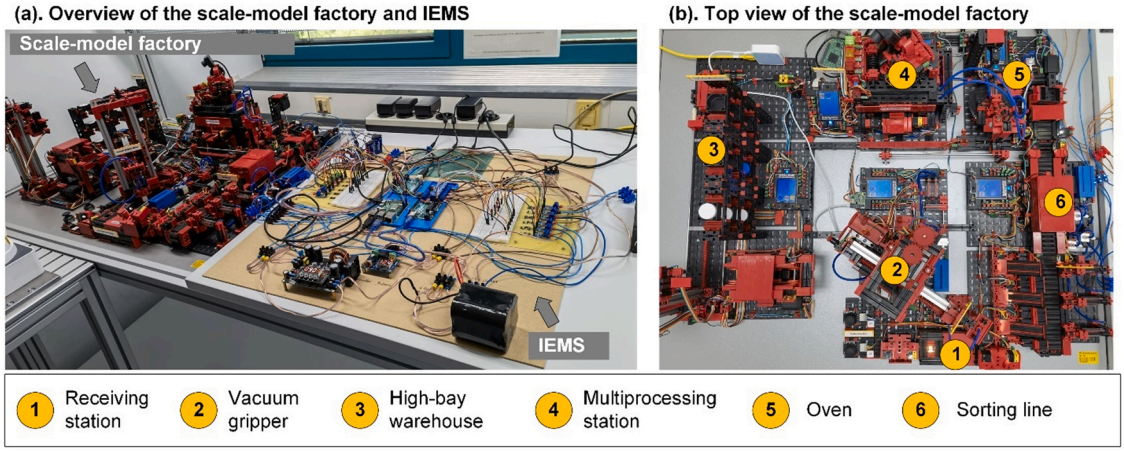

Fig. 4. Scale-model factory.

evaluation can be performed by adding a second ANN. The primary ANN chooses the action and is referred to as the learning network. The second ANN evaluates the target Q-value and is therefore referred to as the target network. Hence, Eq. 8 is rewritten into the following equation, where the index L refers to the learning network and accordingly index T refers to the target network.

$$L_L(w_L) = \mathbb{E}\left[\left(r_t + \gamma Q\left(s_{t+1}, \underset{a_{t+1}}{\operatorname{argmax}}\, Q\left(s_{t+1}, a_{t+1}, w_L\right), w_T\right) - Q\left(s_t, a_t, w_L\right)\right)^2\right] \tag{9}$$

The DDQN training process based on the previous equations is shown in Fig. 3(a). For a better understanding, we have marked the blocks for different functions or data objects with different colours, see the description of the colour indications in the figure. The entire framework consists of an environment model and an agent model. The environment model is responsible for calculating the states s_t. For iteration t, the values of the states s_t are sampled from the environment. The received states s_t are used by the agent for deciding the action a_t, and both variables are saved in replay memory for training. Furthermore, the value of the reward r_t and the following states s_{t+1} for the next iteration $t+1$ are saved in the replay memory, creating a tuple of (s_t, a_t, r_t, s_{t+1}). If a training step is initialised, random tuples are sampled from the replay memory to create a batch. Each tuple is used to calculate the Q-values using the learning network and the target network. Then, the created batch is used in a learning epoch to train the network by updating the corresponding weights of the learning network w_L to minimise the loss function $L_L(w_L)$ (modelled by Eq. 9). The learning network is updated for each epoch using the Adam optimiser, while the target network is updated every 50 epochs by transferring the weights of the learning network w_L to the target network.

To manage the agent's conflict between exploring new strategies and exploiting already learned strategies, an ϵ-greedy approach is implemented during the training process. Following the ϵ-greedy-policy, an agent selects a random action with probability ϵ for the current state. With a probability of $1-\epsilon$ the action is chosen following the Q-learning approach in which the action is selected based on the highest Q-value for the current state and actions $(a_t = \underset{a_t}{\operatorname{argmax}}\, Q(s_t, a_t))$. In the beginning of the training process an increased exploration of the environment is desired, therefore the probability for choosing a random action ϵ is set to 1. The value of ϵ is declined linearly and continuously until it reaches 0.05 during training, so with the advance of the training process the agent is able to exploit it's learned strategies. An approach with a linear and continuous decline of ϵ was chosen because superior results can be achieved compared with other existing approaches, for example a constant ϵ, due to a higher accumulated reward after a defined maximum of training steps.

The model is trained iteratively, as expressed by the gateway before 'End' node in Fig. 3(a). This iterative training process is performed until the maximal training steps are achieved. After the training is completed, the DDQN model can be implemented, as illustrated in Fig. 3(b), in which functions and data objects are marked with different colours. For a given manufacturing cycle of the scale-model factory (which indicates the total operation time for a specific manufacturing task), with time t_0 and prediction length t_1, the number of prediction steps is calculated by t_0/t_1. Within n-th prediction step, the length of a sampling episode is defined as t_2. Thus, the number of sampling episodes (m) within a prediction step is calculated as t_1/t_2. For n-th prediction step, the trained DDQN model will generate an action a_n for n-th prediction step based on current SOC_n, P_n, and p_n values. Within the n-th prediction step, the reward is updated for each sampling episode based on the measured data. For the new prediction step $n+1$, the SOC_m and p_m values of the last sampled measurement episode m in the n-th prediction step are used as the SOC_{n+1} and p_{n+1} values for calculating state s_{n+1}. The predicted power within state s_{n+1} is based on a power forecasting model. For example, the case study adopts a polynomial regression model for power forecasting. The prediction steps are repeated until the manufacturing cycle is finished, or the intelligent switch is cancelled. To validate the feasibility of the proposed DDQN-based IEMS framework, a case study is conducted, as explained in *Development of a specific DDQN-based IEMS for a case study of a scale-model factory*.

Development of a specific DDQN-based IEMS for a case study of a scale-model factory

Description of the scale-model factory

To validate the proposed DDQN-based IEMS in *Methodological formulation of the DDQN-based IEMS approach*, a scale-model factory was used in the case study, as shown in Fig. 4(a), together with the developed IEMS. The scale-model factory is used for educational purposes and is powered by three power interfaces (PI1, PI2, and PI3, each supplying 9 V DC). The scale-model factory consisted of six stations, as depicted in Fig. 4(b). When the factory was in operation, three modes were observed, in which different stations executed different tasks.

- Mode 1: Storage, which simulates the receiving of raw material and the transportation of the raw material to a high-bay

warehouse. As shown in Fig. 4(b), the product (it is a cylindrical object in this work) can be placed in receiving station (①). Thereafter, the vacuum gripper (②) picks up the product and put it to the high-bay warehouse (③). The cycle time of this mode is approximately 120 s
- Mode 2: Production, which simulates the manufacturing process of a factory. As shown in Fig. 4(b), the product is first transported by the vacuum gripper (②) to the oven (④), where a heat treatment is simulated. Afterwards, the product is transferred to a multiprocessing station (⑤) and then to the sorting line (⑥). Finally, the product will be picked up by the vacuum gripper (②) and transferred to the end station (①). The cycle time for this mode is approximately 160 s
- Mode 3: Standby, which simulates the standby of the machines in a factory. In this mode, all stations are standby and waiting for the command of an operator, and there is no pre-set cycle time for this mode.

In addition, the scale-model factory was controlled by six microcontrollers, which communicate with message queuing telemetry transport (MQTT). In this context, one microcontroller is the host, which processes the information sent by the other microcontrollers. The scale-model factory consisted of six stations, where each station is driven by a single microcontroller.

Hardware and software of the IEMS for the scale-model factory

The system architecture of the IEMS are depicted in Fig. 5 and the circuit plan is shown in Fig. 6. The hardware consists of one constant-current-constant-voltage (CCCV) charger, two Arduino UNO units,[1] one Raspberry Pi 4 unit, one relay module, six pairs of voltage and current sensors, one rechargeable 12 V battery as the energy storage equipment, one DC converter (12–9 V), three AC-DC converters (230 V, 60 Hz AC to 9 V DC), three 9 V batteries to avoid power interruptions during switching, and cables for data transfer and electricity supply. First, the CCCV charger is directly connected to the electricity grid (230 V, 60 Hz AC). Its function is to convert 230 V AC power to 12 V DC power which is required to charge the battery. The three AC-DC converters are connected to the electricity grid to convert the power to 9 V DC which is required for the scale-model factory. The Raspberry Pi 4 (4 GB) is a single-board computer with a LINUX operation system that is used for data processing, decision-making for the intelligent switch, signal transfer, and data visualisation on a web-based graphical user interface (GUI). The trained DDQN model was installed and implemented on the Raspberry Pi 4 for the intelligent switch. The two Arduino UNO units are used as microcontrollers for the current and voltage sensors, respectively. Six pairs of current and voltage sensors are used to measure the power data from the CCCV charger, the 12 V-battery, the power interfaces 1, 2, and 3 of the scale-model factory, and the 9 V battery blocks. The relay module is used as a switch for the power supply. The operation of the relay module followed the signal produced by the trained DDQN model. The relay module consisted of four mechanical relay units whose switching operation led to a very short power interruption that turned off the microcontrollers of the scale-model factory (but not the controller of the IEMS). To solve this problem, three 9 V batteries are used to bridge the power interruption. Since they are not used to drive the scale-model factory and their operation time is very short, their energy was neglected in calculating the energy transfer from the IEMS to the scale-model factory. The rechargeable 12 V battery is used as the energy storage equipment, and a DC converter (12–9 V) was used to transform the 12 V of the battery into the 9 V supply required by the scale-model factory. All these components are connected via electric or data cables.

The software of the IEMS consisted of the trained DDQN model, the data interfaces for the sensors and microcontroller, and a web-based GUI. The development and training of the DDQN model are explained in the next subsection, and the data interfaces were provided by the hardware manufacturers. The web-based GUI was programmed using HTML and installed on the Raspberry Pi 4. The web-based GUI consisted of a dashboard to show power and electrical information and a control panel for the sensors, battery, and intelligent switch function, as depicted in Fig. 5.

Development of the environment model

RTP model

As described in *Theoretical background*, the environment model calculates the values for the state of charge, price ratio, and predicted power use (indicted by SOC, $\hat{p}$, and $\hat{P}$, respectively). During training, SOC values were generated in random, varying from 0 % to 100 %. The price model adopted the spot market data [35]. The data were sampled for 10 days from 1 to 12 February (note that the 6th and 7th were the weekend). In this RTP scenario, the pricing frequency was one price per hour; therefore, one day contained 24 data points. However, in this study, as the manufacturing system scenario was scaled down to a scale-model factory, which is not appropriate for running for 10 days, the pricing frequency was modified to one price value per minute, and the times were rearranged into 10 h. As depicted in Fig. 7, the x-axis is revised from an array with dates (on the top of the plot) into an array with time points from 0 to14,400 min (at the bottom of the plot). This dataset was used for training the DDQN model. Note that this study is performed in Germany; therefore, the currency in this study is €.

Power model to predict power use

In this case study, the real power data of the scale-model factory and their polynomial regression models were used as the power model. For the training process, as depicted in Fig. 3(a), the random power values are sampled from the measured power data to train the DDQN model, whereas for the implementation of the DDQN model, as depicted in Fig. 3(b), the polynomial regression models are used to predict the power consumption.

To measure the power data of the scale-model factory, nine trials were performed for each manufacturing mode, in which the power use at the three power interfaces was collected with a sampling time of 1 s. Afterwards, polynomial regression was applied to fit the power data. The coefficients of the regression models are summarised in Table 3, and the regression curves are depicted in Fig. 8. Consequently, predicting the power use of the scale-model factory at time step t can be expressed by the following equation, in which $\hat{P}_{PI1}$, $\hat{P}_{PI2}$, *and* $\hat{P}_{PI3}$ indicate the regression models for power interfaces 1, 2, and 3, respectively.

$$\hat{P}(t) = \hat{P}_{PI1}(t) + \hat{P}_{PI2}(t) + \hat{P}_{PI3}(t) \tag{10}$$

In addition, it is important to mention that although the scale-model factory in this work offers only three different manufacturing modes, this does not mean that the real factory has only these three modes. There may be other modes of operation in the real factory, such as maintenance of machines or modification of factory layout. For these modes of operation, the DDQN model has to be retrained by the use of other power consumption data.

[1] Naming this and other companies in this article is intended to provide complete information about the details of this study and does not necessarily imply an endorsement of the named companies, nor does it imply that these products are necessarily the best. The sources where the products are bought are provided in the Appendix.

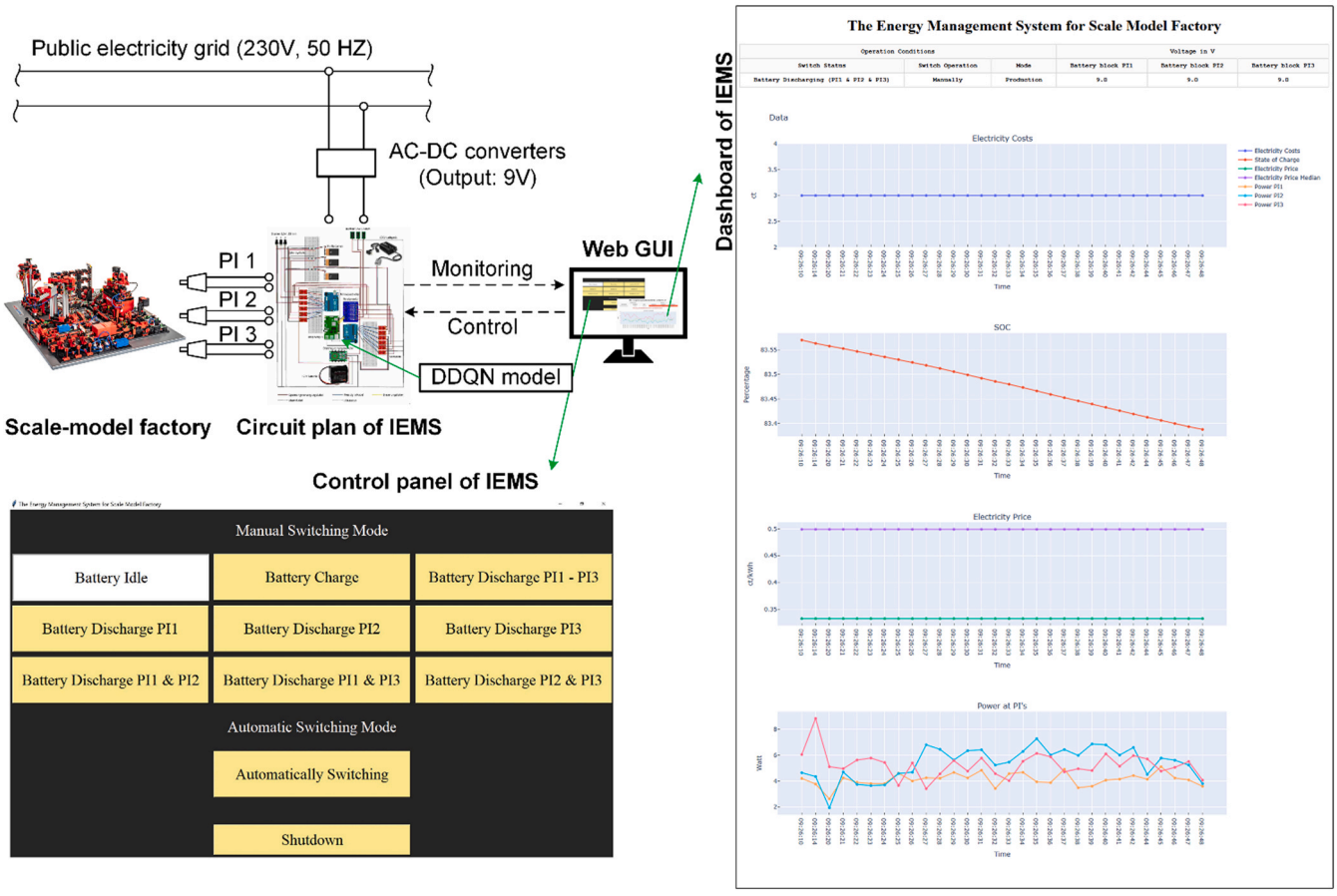

Fig. 5. System architecture of the IEMS.

Development of the agent model

Definition of the action space and reward function

As outlined in the methodological framework described in *Methodological formulation of the DDQN-based IEMS approach*, action space A and reward function R should be defined according to the specific case. In this case study, a rechargeable battery was used. Thus, action space A consisted of 'battery charge', 'battery discharge', and 'battery idle'. In the case of 'battery idle' the scale-model factory is powered by the electricity grid whereas in the case of 'battery charge' the scale-model factory and the charging of the battery is powered by the electricity grid. The case of 'battery charge' refers to the action, where the scale-model factory is powered by the battery. Therefore, the action space A can be expressed by the following equation.

$$A = \{0 \ : = Battery \quad idle; 1 \ : = Battery \quad charging; 2 \ : = Battery \quad discharging\} \tag{11}$$

Because three actions are defined, reward function R should consider the three situations by defining rewards. The reward function, R, is defined as follows.

$$R = \begin{Bmatrix} P_t \cdot \left(\frac{p^{med} - p_t}{0.5 \cdot (p^{97.5\%} - p^{2.5\%})} + \kappa_2 \left(\frac{SOC_t}{100} - \kappa_0 \right) \right) & for \quad a_t = 0 \\ P_t \cdot \left(\frac{p^{med} - p_t}{0.5 \cdot (p^{97.5\%} - p^{2.5\%})} + \kappa_1 \left(\kappa_0 - \frac{SOC_t}{100} \right) \right) for & a_t = 1 \\ P_t \cdot \left(\frac{p_t - p^{med}}{0.5 \cdot (p^{97.5\%} - p^{2.5\%})} + \kappa_2 \left(\frac{SOC_t}{100} - \kappa_0 \right) \right) for & a_t = 2 \end{Bmatrix} \tag{12}$$

The reward function R rates the required power of the scale-model factory P_t with two multipliers (electricity price multiplier and the state of charge multiplier), which are both depended on the chosen action. For the electricity price multiplier, it can be observed that when action 'battery idle' ($a_t = 0$) or 'battery charging' ($a_t = 1$) is chosen, the agent receives a positive reward, if the electricity price p_t is lower then the median p^{med} calculated on historical electricity prices. Whereas when the action 'battery discharging' ($a_t = 2$) is chosen, the reward can only be positive, if the current electricity price p_t is higher than the median p^{med}. The difference of $p^{med} - p_t$ is scaled with the factor $0.5 \cdot (p^{97.5\%} - p^{2.5\%})$ so that the values of that electricity price multiplier are in the same range as the values state of charge multiplier. Therefore, both multipliers are weighted equally. Furthermore, note that $p^{97.5\%} - p^{2.5\%}$ is always positive. For the state of charge multiplier, it can be observed that the agent receives a positive reward, if the state of charge $\frac{SOC_t}{100}$ is larger than the threshold κ_0 for the cases 'battery idle' ($a_t = 0$) and 'battery discharging' ($a_t = 2$). Vice versa, it can be seen that for 'battery charging' ($a_t = 1$), the reward can only be positive with a low state of charge.

In Eq.12, parameter κ_0 is the threshold for loading the battery, and κ_1 and κ_2 are battery factors depending on the range in which the battery is used. These are defined by the following equations.

$$\kappa_1 = \frac{1}{\kappa_0 - \underline{\kappa}} \tag{13}$$

$$\kappa_2 = \frac{1}{\bar{\kappa} - \kappa_0} \tag{14}$$

In Eqs. 13 and 14, parameter $\underline{\kappa}$ is the lower threshold of the battery state of charge, and parameter $\bar{\kappa}$ is the upper threshold of the

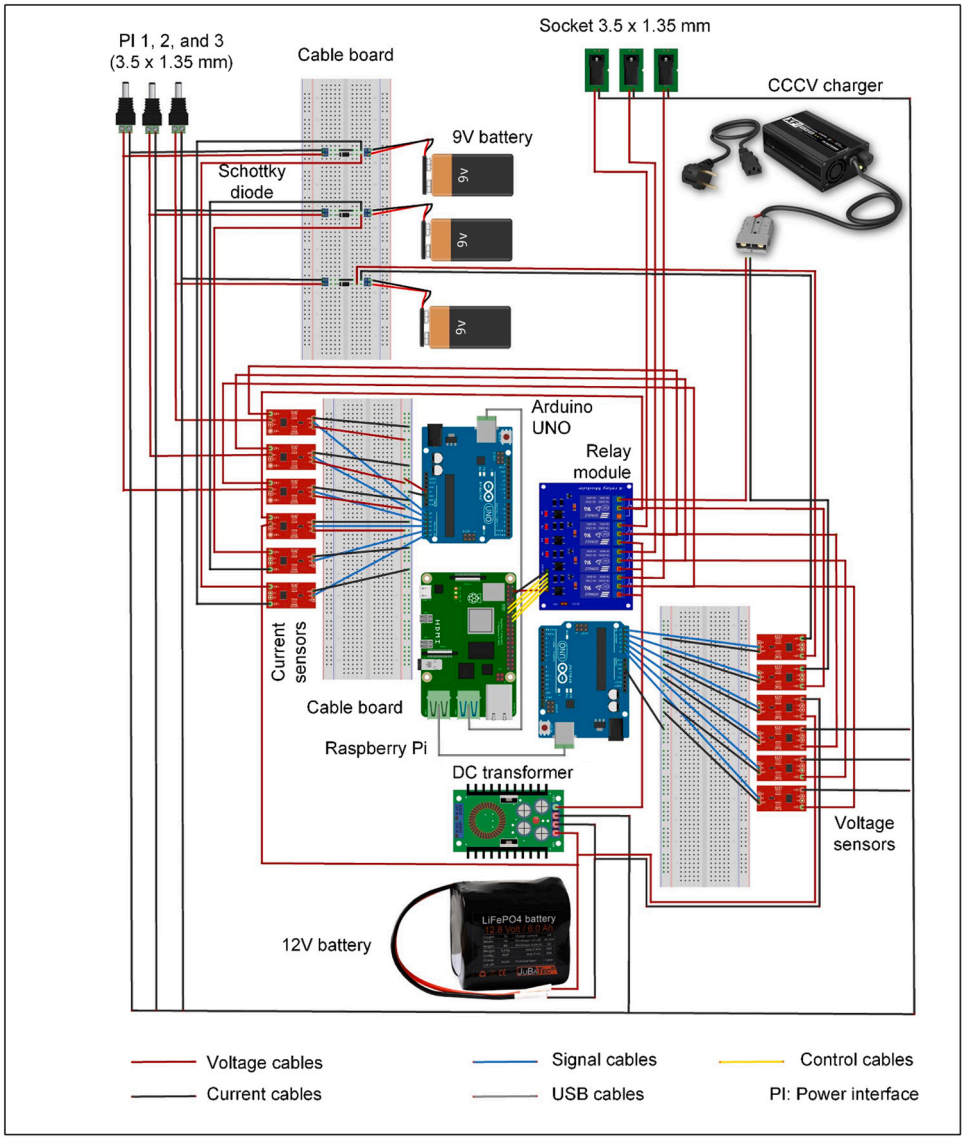

Fig. 6. Circuit plan of the IEMS.

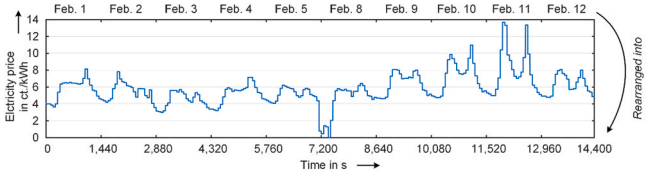

Fig. 7. Electricity prices within the RTP scenario for the training.

battery state of charge. The reward functions differ based on the sign used in the scaling factor. Therefore, the reward functions evaluate actions differently depending on the electricity price and the state of charge. For example, when low electricity prices and a low state of charge of the battery are observed by the agent, selecting the action to charge the battery is evaluated with the highest reward. In contrast, when high electricity prices and a high state of charge are observed, selecting the action to discharge the battery results in the

Table 3
Polynomial regression models as the power forecasting model.

Regression model: $Y=C_1+C_2X+C_3X^2+C_4X^3+C_5X^4+C_6X^5+C_7X^6$, in which Y indicates power, and X indicates time									
	Mode: Standby			Mode: Storage			Mode: Production		
Coefficient	PI 1	PI 2	PI 3	PI 1	PI 2	PI 3	PI 1	PI 2	P I3
C_1	3.881	3.564	3.602	3.949	3.387	3.735	3.705	7.197	5.546
C_2	-0.091e-2	0.846e-3	0.626e-3	-0.297e-2	0.379	0.051	0.061	-0.546	-0.008
C_3	–	–	–	–	-0.023	-0.004e-1	-0.005	0.028	–
C_4	–	–	–	–	0.651e-3	–	0.000142	-0.006e-1	–
C_5	–	–	–	–	-0.085e-4	–	-0.016e-4	0.583e-5	–
C_6	–	–	–	–	0.5.04e-7	–	0.782e-8	-0.026e-6	–
C_7	–	–	–	–	-0.011e-8	–	-0.014e-9	0.445e-10	–

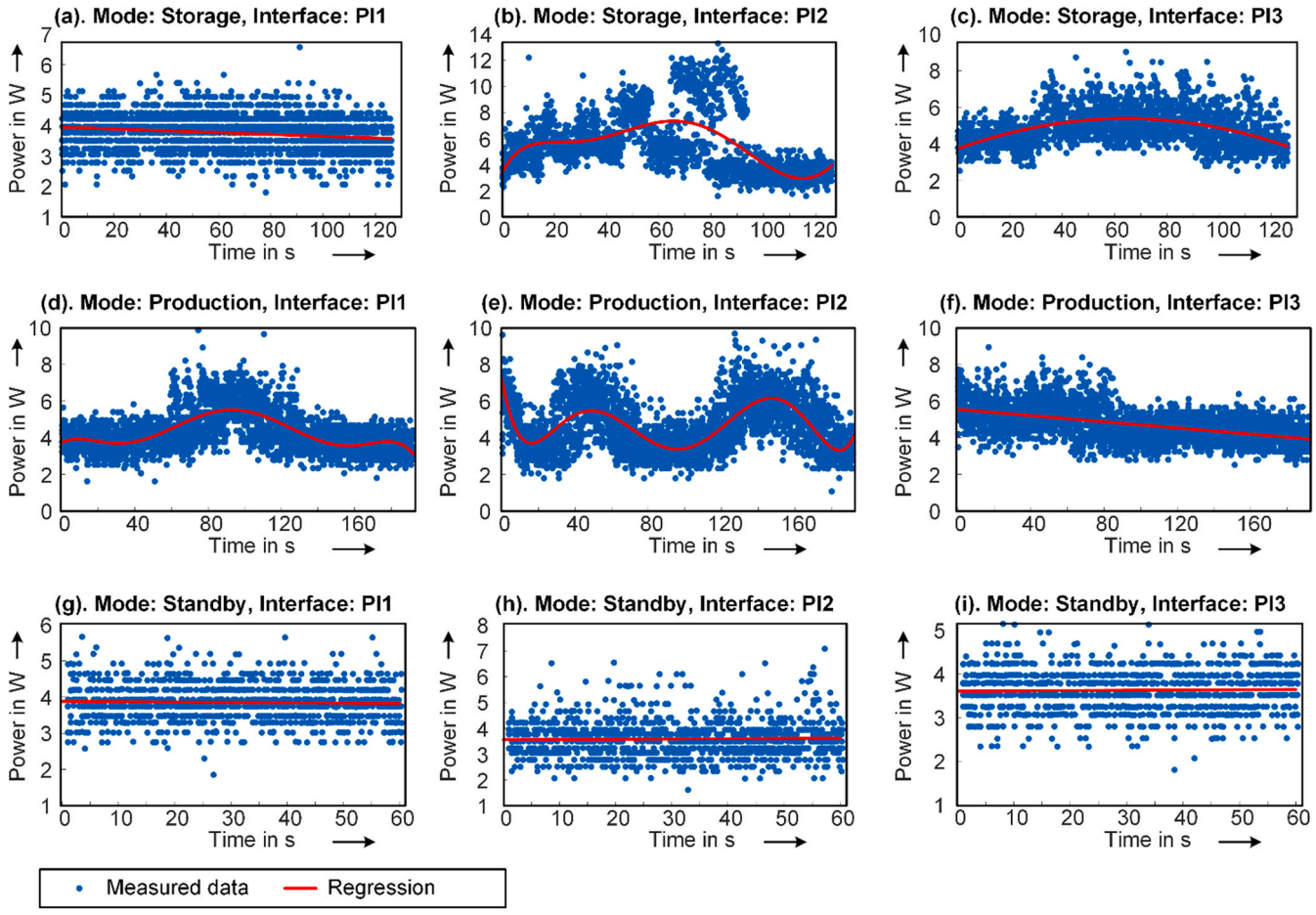

Fig. 8. Regression of the collected power data.

highest reward. Moreover, it is important to mention that the reward function in this section is only suitable for our case. In other cases, if other types of energy storage equipment are applied or other energy forms are involved with the control scope, the reward function should be adapted according to specific situations.

Settings for training the DDQN model

The training of the DDQN model adopted the parameter values listed in Table 4. For each decision (iteration), the values of SOC_t, $\hat{p}_t$, $\hat{P}_t$ were sampled from the environment model. Within one training session, 4000 learning epochs were performed, in which 100 batches were learned within one epoch. The batches were formed by memory experiences. A batch consists of a dataset for training and a dataset to validate the training step, split in a ratio of 4:1. During the training process, the Adam Optimiser was used with a learning rate of 0.001 and exponential decay rates of 0.9 and 0.999. For the training policy, the ϵ-greedy-policy was adopted in which $\epsilon \in (0,1)$. Following the ϵ-greedy-policy, an agent selects a random action with probability ϵ for the current state. The ϵ-greedy-policy was used during training, starting with $\epsilon = 1$ and continuously decaying the value of ϵ up to the learning threshold to 0.1. This leads to an increased exploration of the environment in the early stages of learning. As learning continues, the agents' choice exploits the environment with a higher probability. Finally, the coding of the DDQN model was performed using Python and TensorFlow and Keras libraries. The training of the DDQN model was performed using a computer with an Intel Xeon E5–1620 v2 processor and 32 GB RAM, and the total training time is approximately 10 h. *Results and discussion* of the case study presents the results of this study.

Table 4
Setting of hyper parameters.

Category	Parameter	Parameter value
ANN 1 (ANN 2)	Hidden layers	1
	Number of neurons	128
	Activation function	Leaky ReLU
	Slope for the activation function	0.001
	Use of biases	Yes
Adam Optimiser	Learning rate	0.001
	Decay rates	0.9, 0.999
	Small scaler	1e-8
Learning process	Epochs	4000
	Batch size	100
	Data splitting (learning: validation)	4:1
DDQN algorithms	Replay memory size	500
	Learning threshold	5000
	Transition threshold	0.001
	Discount factor γ	0.7

Results and discussion of the case study

Results of the training

After the DDQN training process, 13,336 decisions were made by the agent, as shown in Fig. 9(a). Note that each decision implies an iteration t for the sampling data from the environment model in Fig. 3(a). After approximately 8000 decisions, the agent approaches the maximal mean reward. The loss of the DDQN model versus epochs is plotted in Fig. 9(b). Note that an epoch implies one

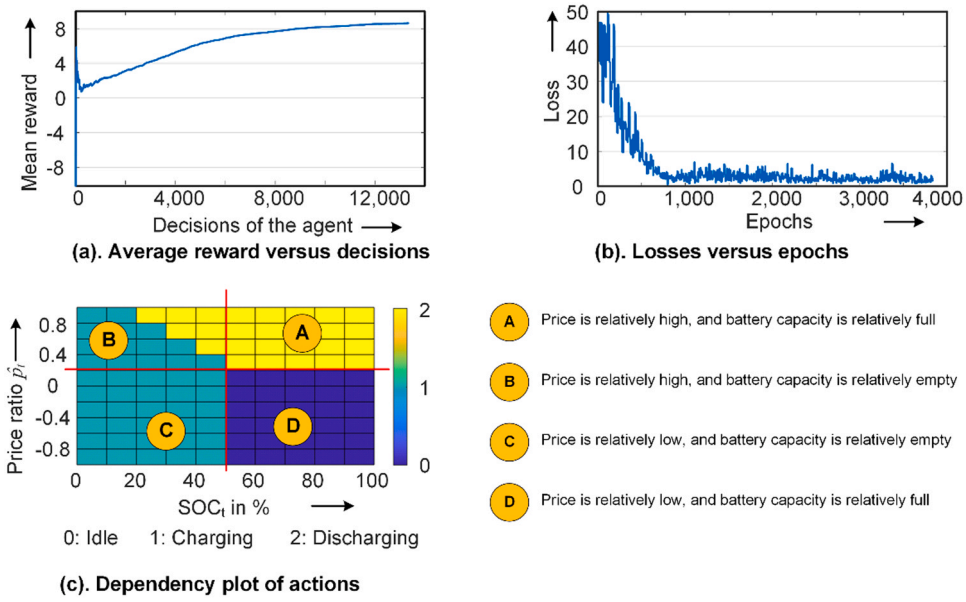

Fig. 9. Training performance evaluation of the DDQN model.

learning step of the ANN_L using Adam Optimiser and the 100 batches in Fig. 3(a), and the losses plotted here are the losses during the validation steps. After approximately 1000 epochs, the loss converges to approximately 1–2, which indicates a good training performance. Finally, the dependency of the actions on SOC_t and price ratio $\hat{p}_t$ is shown in Fig. 9(c). The plot can be divided into four areas, marked by A, B, C, and D. Based on the evaluation of this plot, the dependency of the actions can be described as follows:

- When the price is relatively high ($\hat{p}_t > 0.2$) and the battery capacity is relatively full ($SOC_t > 50$ %), the agent prefers discharging the battery to supply the power of the scale-model factory.
- When the price is relatively high ($\hat{p}_t > 0.2$) and the battery capacity is relatively low ($SOC_t < 50$ %), both battery discharging or charging are possible.
- When the price is relatively low ($\hat{p}_t < 0.2$) and the battery capacity is relatively full ($SOC_t > 50$ %), the agent prefers to idle the battery, so that the scale-model factory takes power from the electricity grid.
- When the price is relatively low ($\hat{p}_t < 0.2$) and the battery capacity is relatively low ($SOC_t < 50$ %), the agent prefers to charge the battery, and the scale-model factory takes power from the electricity grid.

Based on the performance evaluation, it can be concluded that the trained DDQN model is feasible for implementation. In the implementation, an online test is performed, in which manufacturing cycles with and without the intelligent switch of the trained DDQN model are performed, and their costs are compared, as explained in *Implementation in an online test*.

Implementation in an online test

Settings of the RTP price model and different manufacturing cycles for the test

In the online test, a new RTP scenario was used, as depicted in Fig. 10(a). Note that the data of the RTP scenario for the test were not used during the training. Fig. 10(b) shows the box plot of the price data within the RTP scenario for the training and test, with the price changing from minute to minute. Note that the currency in this case study is €.

Six manufacturing cycles were executed in the online test. In the first manufacturing cycle (denoted as the reference cycle), the developed DDQN-based IEMs were not implemented, whereas in the next four manufacturing cycles (denoted as DDQN cycles 1, 2, 3, and 4, respectively), the IEMs were used online to perform intelligent switching of the power supply for the scale-model factory. The design of the trials was based on the Taguchi method, in which an orthogonal array for three factors and two levels is applied. The three factors were the initial SOC value of the battery, charging rate, and prediction step length, as listed in Table 5. To mention is that the prediction step length also implies the frequency of the intelligent switch. Therefore, the action of the agent is dependent on a constant timestep and consequently not triggered by an event, for example if a price change is induced. For example, if the prediction step length is 5 s, it means that the DDQN model makes a decision for every 5 s. The prediction steps of the IEMS in the case study were chosen according to the technical feasibility of the battery. If the frequency is too high, it may danger the battery lifetime. We have decided to investigate 5 s and 10 s according to our own experience with this battery product. Last but not least, in order to validate the benefit of our IEMS framework over the conventional energy management control strategies, we have developed another approach called 'rule-based intelligent energy management (RUBIE)' strategy and implemented it in the last manufacturing cycle. In the RUBIE strategy, the supply switch is made according to four pre-set rules: (1). When SOC value is larger than 50 % and price ratio is smaller than 0, the battery is idled, and the factory takes power from the electricity grid; (2). When SOC value is smaller than 50 % and price ratio is smaller than 0, the battery is charged, and the factory is powered by the electricity grid; (3). When SOC value is larger than 50 % and price ratio is larger than 0, the battery supplies the factory; (4). When SOC value is smaller than 50 % and price ratio is larger than 0, the battery is idled, and the factory is powered by the electricity grid. The initial SOC, battery charging rate, and prediction step length for the RUBIE cycle are the same as those in DDQN cycle 4.

The power data of the manufacturing cycles were collected, and the electricity costs were calculated based on the same RTP scenario. Fig. 11 shows the power curves. For the reference cycle, there is only one power curve, whereas for each DDQN cycle as well as the RUBIE cycle, two power curves are plotted, of which the orange power curve describes the power consumed by the scale-model factory (i.e.

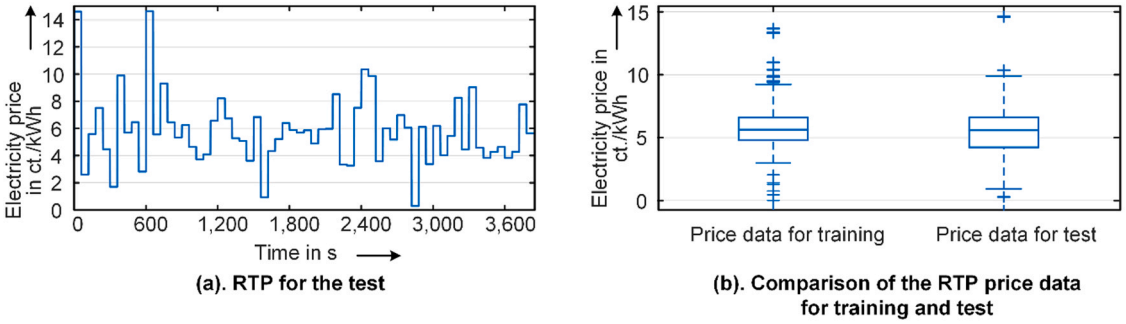

Fig. 10. Results for testing the developed IEMS for the scale-model factory.

Table 5
Settings for the manufacturing cycles in the online test.

Manufacturing cycle	Initial SOC value (%)	Battery charging rate (A)	Prediction step length T_1 (s)
Reference cycle	–	–	–
DDQN cycle 1	50	1	5
DDQN cycle 2	50	2	10
DDQN cycle 3	100	1	10
DDQN cycle 4	100	2	5
RUBIE cycle	100	2	5

the sum of the power at interfaces PI 1, 2, and 3 in Fig. 5), and the blue power curve indicates the total power obtained from the electricity grid. In comparing the power curves, it is seen that the total power curve and the power curve of the scale-model factory in the DDQN cycles and the RUBIE cycle sometimes overlap, as shown by the example in Fig. 11(b). This implies that at these points in time, the scale-model factory takes power directly from the grid. When the two power curves do not overlap, it is observed that the total power curve drops to 0 W, which implies that the scale-model factory is taking power from the battery, and therefore, there is no power taken from the grid.

Comparison and evaluation of the cost reduction of the DDQN cycles

The total electricity costs of the manufacturing cycles in the test are depicted in Fig. 12(a). An evaluation of the curves shows that the cost curve of the reference cycle has a linear relationship with time, whereas the curves for the DDQN cycles exhibit significant 'steps', as presented in Fig. 12(a). This is because of the switching of the power supply; when the scale-model factory takes power from the battery, the cost curves tend to be flat, and when the scale-model factory takes power from the electricity grid, the cost curves tend to increase. As the total electricity use is different in the different cycles of the test, the total cost cannot be directly compared. Thus, we consider the cost per use to compare the DDQN cycles with the reference cycle. The cost per use describes the ratio between the total electricity cost and the total electricity use in a period. The values of the cost per use for all manufacturing cycles in the test are summarised in Table 6 and plotted in Fig. 12(b). A comparison of the curves for the cost per use clearly shows that the curves for the DDQN cycles are significantly lower than the curve for the reference cycle. After approximately 1 h of manufacturing time, the cost per use for the reference cycle is calculated as 5.86 ct./kWh, whereas the values for the four DDQN cycles and the RUBIE cycle are 3.54, 2.94, 1.77, 1.78, and 2.33, respectively. As indicated by the comparison of the values of cost per use, the percentage cost reduction in the four DDQN cycles varies from 39.59 % to 69.79 %, with a mean of 57.21 %.

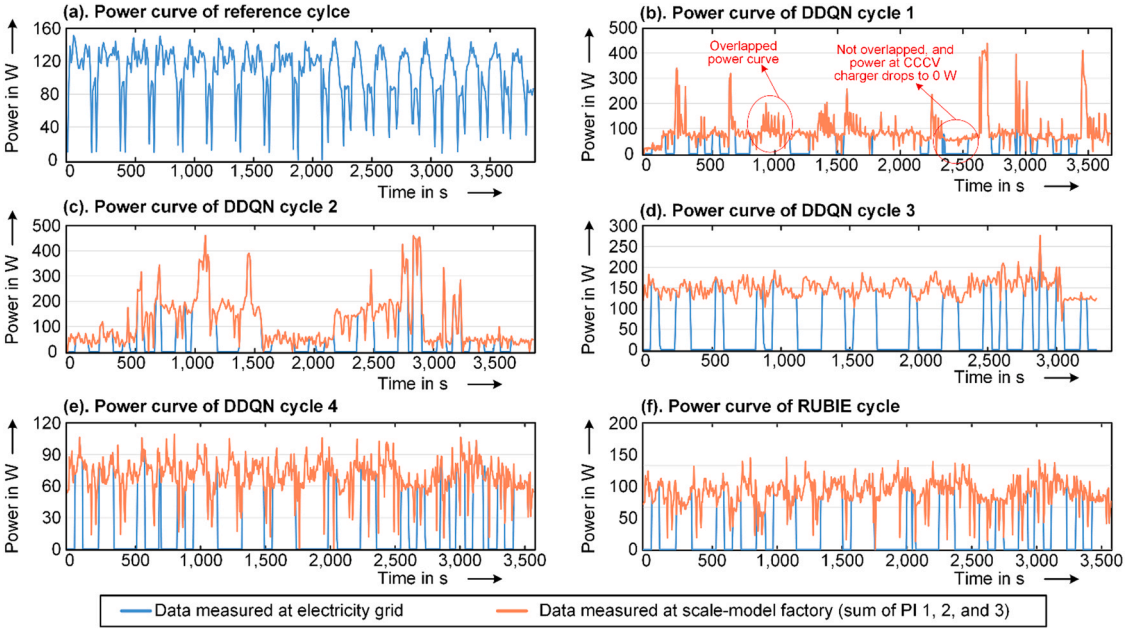

Fig. 11. Power curves in the test.

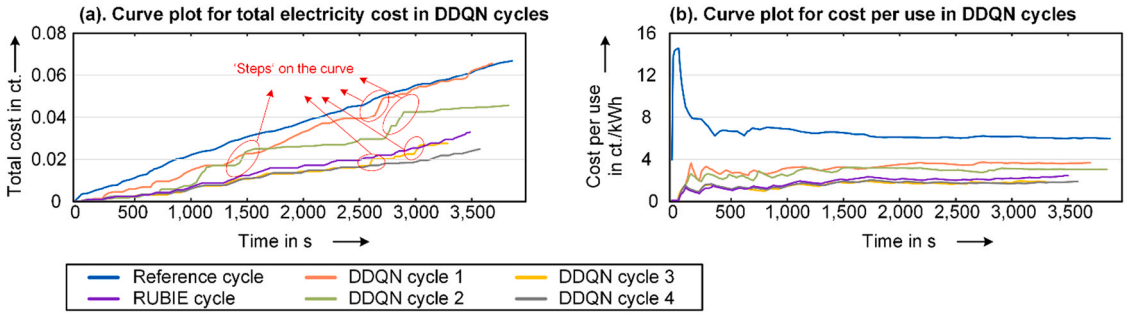

Fig. 12. Total cost and cost per use in the test.

This result clearly confirms the advantage of the proposed DDQN-based IEMS in terms of cost reduction in the scale-model factory. Moreover, as the average reductions of DDQN cycles 3 and 4 are more significant than those of DDQN cycles 1 and 2 and their initial SOC values are 100 %. This suggests that a higher initial SOC leads to a higher level of cost reduction. However, a higher initial SOC also implies a higher investment cost, since the battery is pre-charged before the IEMS is used. Therefore, the overall cost reduction benefit of 100 % SOC values (cycles 3 and 4) may be overestimated considering that the energy pre-charged in the battery still needs to be paid for. Nevertheless, if the user can ensure that the battery is pre-charged at a very low price that can be negligible (e.g. charging during the night), the cost reduction advantage of 100 % SOC value can be ensured. Moreover, in comparing the DDQN cycles with the RUBIE cycle, it is observed the reduction rates for DDQN cycles 3 and 4 (69.79 % and 69.62 % reductions) are approximately 9 % higher than that of RUBIE cycle (60.24 % reduction), whereas the reduction rates for DDQN cycles 1 and 2 (39.59 % and 49.83 % reductions) are lower than that of the RUBIE cycle. This observation leads to the conclusion that the parameter set for our DDQN-based IEMS framework has a significant impact on the cost reduction, and the benefit of the DDQN-based IEMS framework over conventional RUBIE strategy is only valid when the parameters are set appropriately.

For a more intuitive understanding of the decision mechanism of the trained DDQN model, the price data and actions of the DDQN model are plotted in Fig. 13. It is observed that the DDQN model tends to follow action 2 when the prices start to rise, which implies the battery discharging to supply power to the scale-model factory. For example, in Fig. 13(a), three areas outlined with red circles clearly show that the battery discharging depicts a synchronised behaviour with the price increase. On the contrary, when the price starts to lower, the DDQN model tends to follow actions 0 and 1, which imply battery idle and charging, respectively. A further observation is that in DDQN cycles 3 and 4, as shown in Fig. 13(c) and (d), the DDQN model tends to take actions 0 and 2, whereas in DDQN cycles 1 and 2, as shown in Fig. 13(a) and (b), actions 0, 1, and 2 are all adopted by the DDQN model. The only exception is outlined in Fig. 13(c), which indicates that after approximately 2500 s, the DDQN model starts to adopt action 1, whereas in the first 2500 s, only actions 0 and 2 are used. This is because in DDQN cycles 3 and 4, the initial SOC value was set to 100 %; therefore, the DDQN model considers the battery capacity as 'relatively full' and aims to idle the battery when the price is 'relatively low', as indicated by area D in

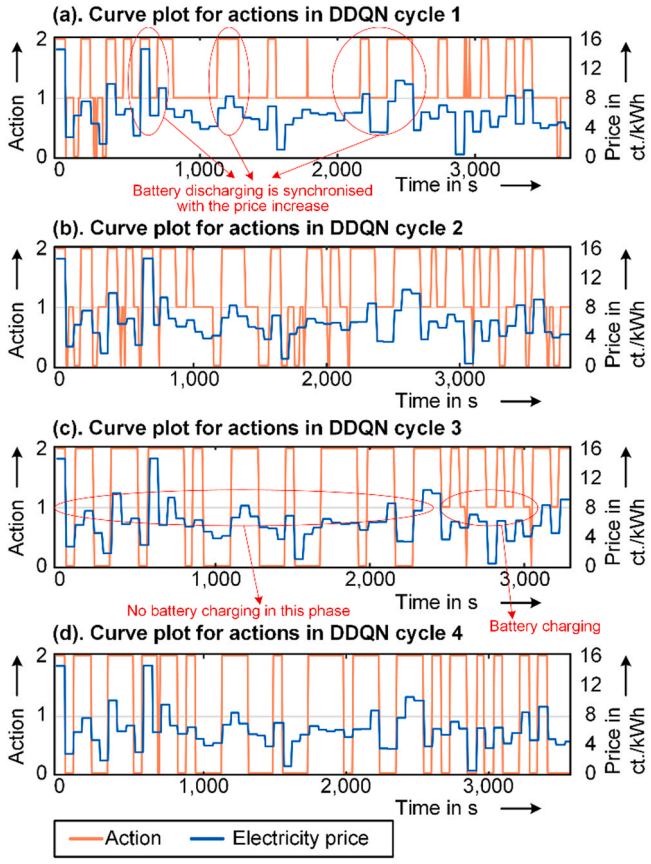

Fig. 13. Actions in the DDQN cycles.

Fig. 9(c). In DDQN cycles 1 and 2, the initial SOC value was set to 50 %; therefore, the DDQN model considers the battery capacity as 'relatively low' and tends to charge the battery when the price is 'relatively low'.

The curve plot for SOC values over time in Fig. 14(a) clearly shows the decision-making of the DDQN model. It is observed that the curves are increasing, flat, and decreasing when the battery is charging, idling, and discharging, respectively, as outlined by the red circles in Fig. 14(a). Moreover, as outlined in Fig. 14(a), a 'turbulence' area, in which the SOC curves increase and decrease frequently, is

Table 6
Cost per use of the manufacturing cycles in the test.

Manufacturing cycles	Reference cycle	DDQN cycle 1	DDQN cycle 2	DDQN cycle 3	DDQN cycle 4	RUBIE cycle
Cost per use (ct./kWh)	5.86	3.54	2.94	1.77	1.78	2.33
Difference to the reference (Δ in ct./kWh)	–	-2.32	-2.92	-4.09	-4.08	-3.53
Percentage reduction (|Δ|/reference)	–	39.59 %	49.83 %	69.79 %	69.62 %	60.24 %

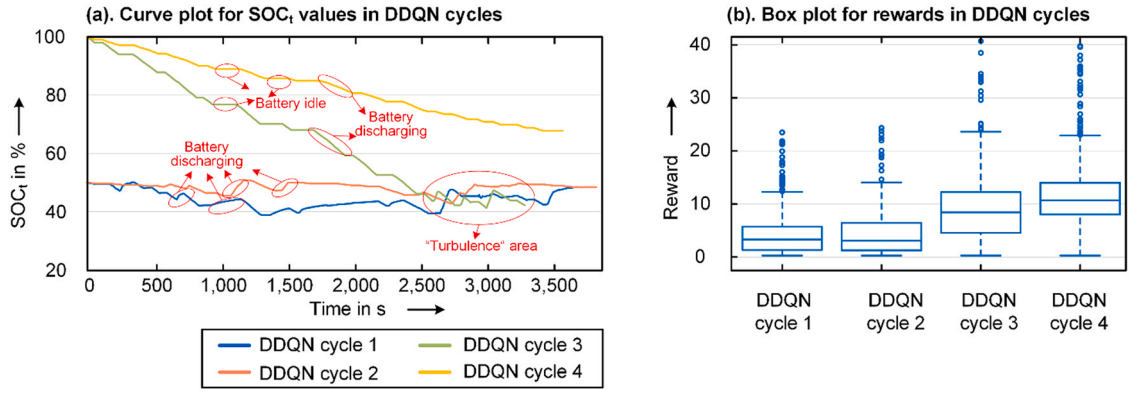

Fig. 14. SOC and rewards in the DDQN cycles.

observed after a time of approximately 2500 s. This is because the SOC value is approaching 50 %, and actions 1 and 2 are both possible, as implied by area B in Fig. 9(c). Considering that frequent charging and discharging may decrease the service life of the battery, the battery should be charged to 100 % before the DDQN-based IEMS is applied.

The distribution of rewards obtained by the DDQN model during the DDQN cycles is illustrated in Fig. 14 and Fig. 13(b) and summarised in Table 7. In the four DDQN cycles, the average rewards are 3.78, 4.27, 9.22, and 11.72, respectively. It is observed that the rewards in DDQN cycles 3 and 4 are relatively higher than those in DDQN cycles 1 and 2. This implies that when the initial SOC value is set to 100 %, the rewards of the DDQN model are higher. Moreover, it is observed that the reward of DDQN cycle 2 is higher than that of DDQN cycle 1, and the reward of DDQN cycle 4 is higher than that of DDQN cycle 3. As the charging rates of DDQN cycles 2 and 4 are set to 2 A, it can be concluded that a higher charging rate will lead to a higher reward during the manufacturing cycle. In terms of the prediction step length, no significant impact on the reward was observed.

Based on the evaluation of the performance of the DDQN model during the online test, it can be concluded that the proposed DDQN-based IEMS is feasible for realising a cost reduction in the scale-model factory, with an average reduction of 57.21 %. Moreover, the most appropriate setting of the online implementation of the proposed DDQN-based IEMS is that the initial SOC value and charging rate of the battery should be set to 100 % and 2 A, respectively.

Comparison of the cost reduction in this study with those in the existing literature

Compared with the approaches in the literature mentioned in *Theoretical background*, in which the pricing strategy and percentage cost reduction are specified, our approach shows a relatively significant cost reduction. Only the approach proposed by Gong et al. achieved a similar level of a 69 % cost reduction for a CPP scenario [9]. However, for the RTP scenario in their approach, a cost reduction of 22 % was observed [9]. Thus, to the best of our knowledge, it can be said that the percentage energy cost reduction in our study is the highest found for an RTP scenario. Moreover, to the best of our knowledge, we are first authors who apply the intelligent energy control during an online test in the RTP scenario for manufacturing systems. In meta-heuristics-based solutions for RTP scenarios, if the price is changed, the scheduling solutions based on other days will be no longer feasible. In the RTP scenario, the price can change for each hour, but it seems infeasible for real factories that the production plan has to be rescheduled every hour. Therefore, the scheduling-based energy saving strategy as proposed in Gong et al. is not suitable for solving the RTP problem during the implementation stage. However, our DDQN model is not sensitive to the price change and is able to give out the best strategy for different prices under a RTP scenario. For other CPP and TOU scenarios, a direct comparison of the cost reduction may not be appropriate because the comparison baseline is different. The reason why our approach achieves a higher cost reduction than other approaches may be that other approaches focusing on PPC and IMP are based on multi-optimisation problems formulated by parameters such as feed rate and machine capacities. For specific cases, these parameters are considered either as constants or as a range of values. Thus, in such approaches, these parameters could be the boundary conditions within which finding the optimum is constrained. However, our approach is aimed at ISP, in which the infrastructure to support the manufacturing processes rather than the processes themselves are modified. This means that any changes in the parameters in the manufacturing processes have no influence on the switch by the IEMS; therefore, it can be said that the control of the IEMS is independent of the manufacturing. The absence of these influences also implies the absence of the boundary conditions derived thereby. Because finding the optimum is not subject to these boundary conditions, the proposed DDQN model has more chances to achieve a higher level of cost reduction than the approaches in which finding the optimum is constrained by the boundary conditions of the manufacturing parameters.

The IEMS proposed in this work has been successful in achieving the cost reduction, but it is still important to mention that this work is only focused on the scale-model factory level and cannot be regarded as a mature solution for practical applications. If the IEMS concept is implemented for a real factory, at least three issues should be considered. First, the power use of machines in a real factory can be as high as tens or hundreds of kilowatts, and aggressive charging and discharging of batteries at such scale will reduce their lifetime. Thus, the cost reduction of energy savings and the cost increase of battery maintenance or replacement will be a trade-off question. Second, frequent power switching at large power consumption scales may cause damage to the public electricity grid and pose a safety risk for human workers. Last but not least, it has been confirmed in this work that the prediction step length has impact on the cost reduction. In applying the IEMS approach for real factories, at least one prediction must be made when the price is changed. Therefore, further research could be the development of prediction models for electricity prices. With an adequate prediction in

Table 7
Distribution of rewards obtained by the DDQN model in the DDQN cycles.

Rewards	DDQN cycle 1	DDQN cycle 2	DDQN cycle 3	DDQN cycle
3rd quartile	5.46	6.18	11.98	13.73
Median	3.05	2.81	8.16	10.44
1st quartile	1.04	1	4.31	7.76
Mean	3.78	4.27	9.22	11.72

electricity prices, the prediction step length can be increased further. Moreover, another solution would be to not define a specific prediction step length, but rather to use a dynamic step length. For example, a second intelligent algorithm can be developed to decide how long the prediction step length should be, such as anomaly detection to trigger a prediction based on the agent. Nevertheless, it is still to note that the SOC of batteries must be considered to prevent harmful over(dis-)charge.

Conclusion and outlook

In this study, a methodological framework for developing an IEMS based on DDQN and an energy storage equipment was proposed and validated in a case study of a scale-model factory. In the test, different manufacturing cycles with and without the intelligent switch of the DDQN model were performed and compared. The following four conclusions are observed: (1) based on a case study in which an average cost reduction of 57.21 % was achieved, the capability of the proposed DDQN-based IEMS for energy cost reduction of manufacturing systems is confirmed; (2) In the case study, the best cost reduction rate of our DDQN-based IEMS framework is approximately 9 % higher than the reduction rate of the conventional RUBIE strategy. This suggests our method is better than conventional methods. However, considering that the parameter set has a significant impact on the cost reduction rate, the advantage of our approach over conventional approach is valid only when the parameters are set appropriately; (3) ISP-based energy savings are not constrained by the manufacturing processes in which the boundary conditions may have an impact on the optimisation of the energy cost; (4) although previous studies pointed out that the RTP strategy bears both the highest risk and reward for manufacturing companies, this study clearly proves that this risk can be minimised by using reinforcement learning techniques. In terms of future investigations, the following five issues are recommended for the research community: (1) the CPP strategy deserves more research attention as the TOU strategy has been extensively studied in the past and the RTP strategy was investigated in this study; (2) exploration of a range of algorithms reinforcement learning and other optimisation techniques to minimise the energy cost of manufacturing systems; (3) up-scaling of the proposed approach to a real factory level with consideration of the problems related to the trade-off between the increased maintenance cost and reduced energy cost, risk to damage public electric grid, as well as the work safety of humans. (4) extending the IEMS approach by a prediction mode for electricity prices. (5) using an intelligent system to trigger a prediction of the trained agent by using (for example) anomaly detection.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was funded by the German Research Foundation within the project “International Research Training Group 2057” (DFG funding number 252408385 – IRTG 2057). Furthermore, the research was also partly funded by the German-Brazilian collaborative program “Manufacturing System Models for Industry 4.0 based on highly heterogeneous and unstructured data sets” within the framework of the Collaborative Research Initiative – PIPC 8881.473092/2019-1 (DFG funding number AU 185/72).

References

[1] Hayes, R.H., Wheelwright, S.C., 1984, Restoring our Competitive Edge: Competing through Manufacturing. Wiley, New York, N.Y.
[2] Ostwald, P.F., McLaren, T.S., 2004, Cost Analysis and Estimating for Engineering and Management. 1st ed. Pearson Education, Upper Saddle River, NJ.
[3] Thiede, S., 2011, Energy Efficiency in Manufacturing Systems. Springer, Berlin: 2012. Zugl.: Braunschweig, Techn. Univ., Diss..
[4] Hu, Z., Kim, J., Wang, J., Byrne, J., 2015, Review of Dynamic Pricing Programs in the U.S. and Europe: Status Quo and Policy Recommendations. Renewable and Sustainable Energy Reviews, 42:743–751.
[5] Kathan D., Daly C., Gadani J., Gruenke D., Icart E., Irwin R. et al. Assessment of demand response and advanced metering. [June 10, 2021]; Available from: ⟨https://cms.ferc.gov/sites/default/files/2020–04/sep-09-demand-response.pdf⟩.
[6] Faruqui A., Palmer J. Dynamic pricing and its discontents: empirical data show dynamic pricing of electricity would benefit consumers, including the poor. [June 10, 2021]; Available from: ⟨https://www.cato.org/sites/cato.org/files/serials/files/regulation/2011/9/regv34n3–5.pdf⟩.
[7] Zhang, H., Zhao, F., Fang, K., Sutherland, J.W., 2014, Energy-conscious Flow Shop Scheduling under Time-of-use Electricity Tariffs. CIRP Annals, 63/1: 37–40.
[8] Stromback J., Dromaque C., Yassin M. The potential of smart meter enealbed programs to increase energy and systems efficiency: a mass pilot comparison, Vaasaett- European Smart Grid Industry Group. [June 10, 2021]; Available from: ⟨https://esmig.eu/sites/default/files/2011.10.12_empower_demand_report_final.pdf⟩.
[9] Gong, X., Pessemier, T., de, Joseph, W., Martens, L., 2015, An Energy-Cost-Aware Scheduling Methodology for Sustainable Manufacturing. Procedia CIRP, 29:185–190.
[10] Kong, M., Xu, J., Zhang, T., Lu, S., Fang, C., Mladenovic, N., 2021, Energy-efficient Rescheduling with Time-of-use Energy Cost: Application of Variable Neighborhood Search Algorithm. Computers & Industrial Engineering, 156:107286.
[11] Xiao, Y., Jiang, Z., Gu, Q., Yan, W., Wang, R., 2021, A Novel Approach to CNC Machining Center Processing Parameters Optimization Considering Energy-saving and Low-cost. Journal of Manufacturing Systems, 59:535–548.
[12] Roslan, M.F., Hannan, M.A., Jern Ker, P., Begum, R.A., Indra Mahlia, T.M., Dong, Z.Y., 2021, Scheduling Controller for Microgrids Energy Management System using Optimization Algorithm in Achieving Cost Saving and Emission Reduction. Applied Energy, 292:116883.
[13] Zhang, H., Zhao, F., Sutherland, J.W., 2015, Manufacturing Scheduling of Collaborative Factories for Energy Cost Reduction. Procedia Manufacturing, 1:122–133.
[14] Bui, Y.-H., Hussain, A., Kim, H.-M., 2020, Double Deep Q -Learning-Based Distributed Operation of Battery Energy Storage System Considering Uncertainties. IEEE Transactions on Smart Grid, 11/1: 457–469.
[15] Lu, R., Li, Y.-C., Li, Y., Jiang, J., Ding, Y., 2020, Multi-agent Deep Reinforcement Learning Based Demand Response for Discrete Manufacturing Systems Energy Management. Applied Energy, 276:115473.
[16] Braglia, M., Castellano, D., Gabbrielli, R., Marrazzini, L., 2020, Energy Cost Deployment (ECD): A Novel Lean Approach to Tackling Energy Losses. Journal of Cleaner Production, 246:119056.
[17] Li, G., Bie, Z., Kou, Y., Jiang, J., Bettinelli, M., 2016, Reliability Evaluation of Integrated Energy Systems Based on Smart Agent Communication. Applied Energy, 167:397–406.
[18] Wu, J., Yan, J., Jia, H., Hatziargyriou, N., Djilali, N., Sun, H., 2016, Integrated Energy Systems. Applied Energy, 167:155–157.
[19] van Hasselt, H., 2010, Double Q-learning. Advances in Neural Information Processing Systems, 23.
[20] Sterner, M., Stadler, I., 2019, Handbook of Energy Storage. Springer Berlin Heidelberg, Berlin, Heidelberg.
[21] Sauer D.U., Fuchs G., Lunz B., Leuthold M. Technology Overview on Electricity Storage - Overview on the potential and on the deployment perspectives of electricity storage technologies: Smart Energy for Europe Platform GmbH (SEFEP), report prepared by ISEA / RWTH Aachen University; 2012.
[22] Mourtzis, D., Boli, N., Alexopoulos, K., Różycki, D., 2018, A Framework of Energy Services: From Traditional Contracts to Product-Service System (PSS). Procedia CIRP, 69:746–751.
[23] D'Aprile P., Newman J., Pinner D.. The new economics of energy storage: Energy storage can make money right now. Finding the opportunities requires digging into real-world data. [June 16, 2021]; Available from: ⟨https://www.mckinsey.com/business-functions/sustainability/our-insights/the-new-economics-of-energy-storage⟩.
[24] Marsland, S., 2015, Machine Learning - An Algorithmic Perpective. CRC Press, Boca Raton, Fl.
[25] Otterlo, Mv, Wiering, M., 2012, Reinforcement Learning and Markov Decision Processes. Wiering M, van Otterlo M, (Eds.) Reinforcement learning - State-of-the-art. Springer, Berlin, Heidelberg u.a: 3–44.
[26] Sutton, R.S., Barto, A.G., 1998, Reinforcement Learning: An introduction. MIT Press, Cambridge Mass..
[27] Zhou, Q., Li, J., Shuai, B., Williams, H., He, Y., Li, Z., et al., 2019, Multi-step Reinforcement Learning for Model-free Predictive Energy Management of an Electrified Off-highway Vehicle. Applied Energy, 255:113755.
[28] Mahadevan S., Marchalleck N., Das T.K., Gosavi A. Self-improving factory simulation using continous-time average-reward reinforcment learning. Proceedings of the Fourth International Machine Learning Conference 1997:202–10.

[29] Zhang W., Dietterich T.G. A reinforcement learning approach to job-shop scheduling. Proceedings of the 14th International Joint Conference on Artificial Intelligence 1995:1114–1120.
[30] Aydin, M.E., Oztemel, E., 2000, Dynamic Job-shop Scheduling using Reinforcement Learning Agents. Robotics and Autonomous Systems, 33:169–178.
[31] Waschneck, B., Reichstaller, A., Belzner, L., Altenmüller, T., Bauernhansl, T., Knapp, A., et al., 2018, Optimization of Global Production Scheduling with Deep Reinforcement Learning. Procedia CIRP, 72:1264–1269.
[32] Schwung D., Schwung A., Ding S.X. On-line Energy Optimization of Hybrid Production Systems Using Actor-Critic Reinforcement Learning. In: 2018 International Conference on Intelligent Systems (IS): IEEE; 2018, p. 147–154.
[33] Dayani A.B., Fazlollahtabar H., Ahmadiahangar R., Rosin A., Naderi M.S., Bagheri M. Applying Reinforcement Learning Method for Real-time Energy Management. In: 2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe): IEEE; 2019, p. 1–5.
[34] Ng, K.S., Moo, C.-S., Chen, Y.-P., Hsieh, Y.-C., 2009, Enhanced Coulomb Counting Method for Estimating State-of-charge and State-of-health of Lithium-ion Batteries. Applied Energy, 86/9: 1506–1511.
[35] EEX. Spot market data. [June 15, 2021]; Available from: ⟨https://www.powernext.com/spot-market-data⟩.