Article

Reinforcement Learning for Energy-Storage Systems in Grid-Connected Microgrids: An Investigation of Online vs. Offline Implementation

Khawaja Haider Ali [1,2,*], Marvin Sigalo [1], Saptarshi Das [1], Enrico Anderlini [3], Asif Ali Tahir [1] and Mohammad Abusara [1,*]

[1] Penryn Campus, College of Engineering, Mathematics and Physical Sciences, University of Exeter, Cornwall TR10 9FE, UK; ms924@exeter.ac.uk (M.S.); saptarshi.das@ieee.org (S.D.); a.tahir@exeter.ac.uk (A.A.T.)
[2] Department of Electrical Engineering, Sukkur IBA University, Sukkur 65200, Pakistan
[3] Department of Mechanical Engineering, Roberts Building, University College London, London WC1E 7JE, UK; e.anderlini@ucl.ac.uk
[*] Correspondence: ka400@exeter.ac.uk (K.H.A.); m.abusara@exeter.ac.uk (M.A.)

Abstract: Grid-connected microgrids consisting of renewable energy sources, battery storage, and load require an appropriate energy management system that controls the battery operation. Traditionally, the operation of the battery is optimised using 24 h of forecasted data of load demand and renewable energy sources (RES) generation using offline optimisation techniques, where the battery actions (charge/discharge/idle) are determined before the start of the day. Reinforcement Learning (RL) has recently been suggested as an alternative to these traditional techniques due to its ability to learn optimal policy online using real data. Two approaches of RL have been suggested in the literature viz. offline and online. In offline RL, the agent learns the optimum policy using predicted generation and load data. Once convergence is achieved, battery commands are dispatched in real time. This method is similar to traditional methods because it relies on forecasted data. In online RL, on the other hand, the agent learns the optimum policy by interacting with the system in real time using real data. This paper investigates the effectiveness of both the approaches. White Gaussian noise with different standard deviations was added to real data to create synthetic predicted data to validate the method. In the first approach, the predicted data were used by an offline RL algorithm. In the second approach, the online RL algorithm interacted with real streaming data in real time, and the agent was trained using real data. When the energy costs of the two approaches were compared, it was found that the online RL provides better results than the offline approach if the difference between real and predicted data is greater than 1.6%.

Keywords: reinforcement learning (RL); microgrid; battery management; offline and online RL; optimisation

check for updates

Citation: Ali, K.H.; Sigalo, M.; Das, S.; Anderlini, E.; Tahir, A.A.; Abusara, M. Reinforcement Learning for Energy-Storage Systems in Grid-Connected Microgrids: An Investigation of Online vs. Offline Implementation. *Energies* **2021**, *14*, 5688. https://doi.org/10.3390/en14185688

Academic Editor: Seon-Ju Ahn

Received: 7 August 2021
Accepted: 7 September 2021
Published: 9 September 2021

1. Introduction

Grid-connected microgrids are becoming the main building blocks of smart grids. They facilitate the vast deployment and better utilisation of RES, reduce stress on the existing power grid, and provide consumers with uninterrupted power supply. The main aim for any Energy Management System (EMS) for grid-connected microgrids is to reduce operational costs by reducing the cost of power imported from the grid. This is achieved by controlling the Battery Energy Storage System (BESS) to store power when RES generation is higher than load demand and release power when it is less than the load demand. However, BESS capacity is finite and hence, depending on the battery size, imported power from the grid is likely to be used. Therefore, grid tariffs, which can vary during the day, must be taken into account when deciding charging and discharging commands. In addition, the feed-in tariff can also be available to enable consumers or prosumers to sell their excess power to the grid [1]. These factors demonstrate the need for an intelligent optimisation method. However, due to the lack of information regarding future generation and load profiles, this presents a challenge for EMS.

Inspired by the classical economic dispatch of power systems, various studies have suggested different optimisation techniques to plan, schedule, and control the BESS in grid-connected microgrids. These studies have formulated the operation of the BESS as a dispatch optimisation problem and solved it using Linear Programming (LP) [2,3], Mixed Integer Programming (MIP) [4], Mixed-Integer Linear Programming (MILP) [5–7], and Mixed-Integer Nonlinear Programming (MINLP) [8,9]. In Chen et al. [2], a general algebraic modelling system (GAMS) was developed and solved by CPLEX solver and then tested in a physical system based in Taiwan. The study compared two models to determine the impact of energy storage on optimal scheduling. The first model consisted of thermal and electrical loads and a CHP unit. The second model used additional thermal and electrical storage. Both models used in this work relied on the prediction of load profiles. In Luna et al. [5], the model reflects a deterministic problem that promotes self-consumption based on 24 h look-ahead forecast data. The microgrid consists of a supervisory control stage that compensates for any mismatch between the offline scheduling process and the real-time microgrid operation. In Li et al. [7], the microgrid optimisation problem is formalised using a general algebraic modelling system (GAMS) via a discretised step transformation (DST) approach and finally solved using the CPLEX solver. This paper proposes a new optimal scheduling mode by modelling the uncertainty of spinning reserves provided by energy storage with probabilistic constraints. These achievements are highly dependent on the proper estimation of spinning reserves, which is a big challenge while working on a real system. In Mosa and Ali [9], the MINLP algorithm was used to reduce the operational cost of a DC microgrid consisting of a photovoltaic (PV), fuel cell (FC), micro turbine (MT), diesel generator (DE), and battery/BESS. This study uses Egyptian grid load profiles over four seasons of the year based on the prediction.

The above traditional approaches require a detailed and accurate mathematical model of the system, while some of them require the linearisation of the system. In addition, previous knowledge of future RES generation and load demand over a period is required as an input to the optimisation problem. The accuracy of the prediction may affect the accuracy of the BESS operation. Therefore, different forecasting algorithms that can handle the stochastic nature of load demand and RES have been suggested in the literature. These algorithms are designed to forecast short (daily), medium (seasonal), and long (yearly) load demand and availability of RES. Most advanced forecasting algorithms include Artificial Neural Networks (ANN), dynamic programming (DP)-based optimisation, and fuzzy logic by considering the weather conditions. Although forecasting techniques vary within the vast amount of existing literature [10–18], the most common objective of these techniques is to decrease the forecasting error by better modelling the uncertainties in real time. Although forecasting algorithms have improved in recent years, it is still challenging to predict the future load demand and availability of RES with minimum error, especially if the decision making is implemented in real time.

Recent studies [19–26] have introduced reinforcement learning as a potential solution for the optimal operation of BESS due to its ability to develop an optimal policy online. In RL, an agent interacts with the surrounding environment and develops an optimal policy for taking the right action after exploring its state. The agent takes the action to maximise a future accumulative reward. The main advantage for RL over traditional methods is that it does not need any model of the environment and it can learn the optimum policy in real time. Yoldas et al. [27] used the MINLP technique guided by a Q-learning algorithm to decrease the daily energy cost and emission of harmful gases simultaneously. Performance comparisons were made using only conventional Q-learning. The result showed an approximately 1.2% reduction of the daily operational costs associated with the proposed technique over conventional Q-learning approaches.

There are two main types of online RL algorithms: off-policy and on-policy. In off-policy methods (e.g., Q-learning), the action-valued function is approximated independently of the policy being followed. Conversely, in on-policy approaches, e.g., in state–action–reward–state–action (SARSA), the action-valued function is continuously updated according to the developed policy, which makes it harder to converge [28]. Whereas with off-policy RL, the agent does not need to follow any specific policy and in fact could even act randomly, on-policy schemes rely on the policy that is being established. Despite the possibly random behaviour, off-policy methods, including the Q-learning algorithm, can converge onto the optimal policy independent on the policy employed during exploration [29]. On the other hand, offline, or data-driven RL develops the policy on pre-collected data. Once the policy is developed, it is deployed to control the system. The policy is not updated by interacting with the system in real time. Offline learning, such as batch RL and other conventional techniques such as MILP, MINLP, and LP algorithms, work with data in bulk. Therefore, the uncertainty of some unknown variables such as load demand and PV profiles make these offline methods more challenging because the training is done on forecasted and not real data. Conventionally, offline learning algorithms need to be re-run from scratch in order to learn from modified or new data.

In Mbuwir et al. [19], the authors proposed batch reinforcement learning, offline RL, to solve the optimisation of the microgrid problem in order to achieve a cost-efficient solution. The goal was to find or statistically learn the pattern of the best control policy from the training data (previous year's load and PV profiles) in the form of several smaller batches (sets) and then use this policy on the current environment in real time. When the batch RL was compared to the MILP approach, it was shown to be 19% less efficient than MILP. Kuznetsova et al. [30] developed a two-step-ahead RL algorithm to cut down utility bills by learning the stochastic behaviour of the environment using RL and then scheduling the battery two hours ahead from the current time. RL trains the agent and produces the optimal actions of the battery using forecasted wind and load demand power profiles. Liu et al. [20] proposed a cooperative RL algorithm for distributed economic dispatch in microgrids. However, the challenge of using forecasted PV and load data, which can affect function approximation, is not addressed in this paper. Jiang and Fei [21], suggested a Q-learning based, economical smart microgrid with two-level hierarchical agents with flexible demand response and distributed energy resource management. The authors claim that the suggested scheme is very effective while satisfying the user's preference. However, the suggested work requires load demand from the user before scheduling its distributed units and batteries. This can affect the cost optimisation adversely if the user's demand request changes during real-time operations.

In [22,25,31–43], shallow and deep neural networks have been suggested to approximate the Q-value function to achieve better optimisation results with shorter convergence time. Low convergence time is also desirable for online applications. Lu et al. [22] used deep RL to develop an energy-trading scheme according to the predicted future renewable energy generation, estimated future power demand, and battery level. This work also depended on forecasted renewable energy power generation. In Bui et al. [25], a double deep neural network (DDQN) was proposed for function approximation of Q-values. The authors claim that this method trains the model faster as compared with a single deep (having one network) RL algorithm. This work also depends on the estimation of future load demand and PV generation. Zhou et al. [24] suggested an algorithm to train the agent in real time using real data profiles instead of forecasted datasets. A fuzzy Q-learning approach is adopted for a system consisting of household users and a local energy pool in which customers are free to trade with the local energy pool and enjoy low-cost renewable energy while avoiding the installation of new energy generation equipment. Another online approach was proposed in Kim et al. [26] in which real-time pricing is used to reduce the system cost. Both Zhou et al. [24] and Kim et al. [26] do not provide information regarding the efficiency of their algorithms with respect to other offline Q-learning techniques. Another study by Kim and Lim [44] applied Q-learning directly in real time.

Optimal cost was achieved for the whole year rather than for a single day. In contrast to other offline approaches, this direct online approach trains the agent in real time using real data by moving from one day to another. In the beginning, the agent experience is low; however, as days progress, the agent starts exploiting more actions, and an optimal policy is developed.

The literature reviewed here suggests that offline policies (including RL) require predicted data in order to produce optimal results if the real data is the same as the predicted data, i.e., zero prediction error. The RL online approach, on the other hand, does not rely on predictions and uses real streaming data. However, it is not clear how effective the online RL algorithm is as compared to the offline approaches when the prediction error increases. Motivated by this shortcoming in the existing literature, this paper provides a comprehensive comparison between the two approaches. Using one year of real PV generation and load data obtained from [45], different profiles for predicted data were created by generating random noise with different standard deviations. The noise was added to the real data to create synthetic predicted data. Then, the 24 h of predicted data were used to train the RL agent, and then, the optimised battery command achieved in this process was applied to the real data (offline RL approach). The online RL, on the other hand, interacted with real data in real time. Then, the energy costs of the two approaches were compared to help users make decisions on the most appropriate approach given the accuracy of the available forecasted data. Finally, the case with zero prediction error was considered in the comparison of MINLP versus RL to establish a benchmark between conventional offline approaches versus the offline RL.

2. Energy Management System

The aim of the EMS is to reduce the cost of the power imported from the grid as shown in Figure 1. Thus, the grid supplies energy only if the RES and BESS do not fulfil the demand, and the time of transfer is chosen in order to minimise the cost. Export of energy to the main grid might also be possible if a feed-in tariff is available. Renewable energy, PV in this study, has a priority to fulfil the load requirement first. If it is not sufficient, then the battery, main grid, or combination of both are used to fulfil the demand. The BESS may charge from the PV directly or charge from or discharge into the main grid if needed. The EMS can make use of different tariff rates within the day by charging the BESS during low-tariff periods and discharging it during high-tariff periods. In this work, a fixed feed-in tariff is assumed, and there are three different tariffs (peak, mid, low) which are assumed to import the energy from the main grid depending on time of the day.

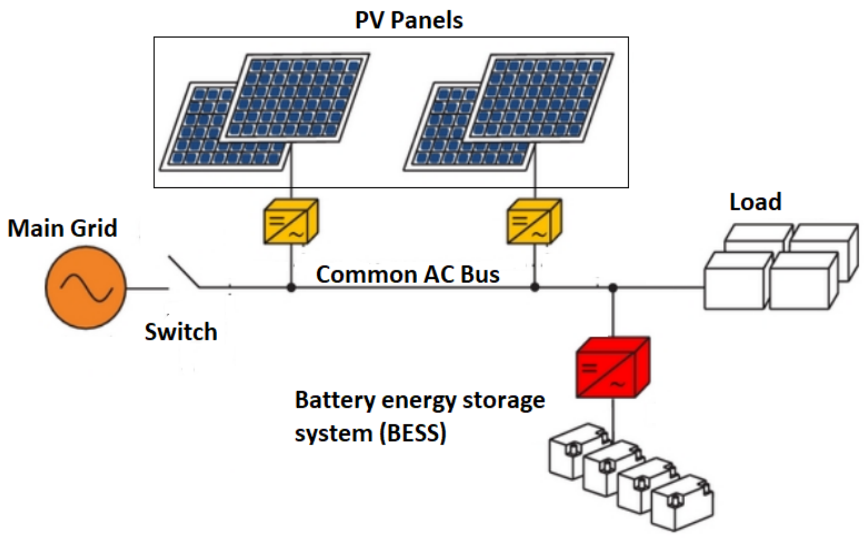

Figure 1. Illustration of microgrid model with BESS.

In this paper, an RL algorithm is used to interact with the microgrid and take automated decisions to control the BESS, taking into account a dynamically changing environment characterised by the available PV output, load demand, and the level of battery charge (SOC). The decisions are in the form of actions for the battery to charge, discharge, or remain idle. The recommended actions for the BESS are developed through Q-learning.

2.1. System States

The state space (S) is discretised at Δt = 30 min, which suggests that the learning agent captures the information related to the dynamics of the microgrid after the time interval of 30 min. In Equation (1), t represents the time period, which has 48 states in 24 h of a day due to its discretisation every 30 min.

$$s_t = [SOC,\, e_t^{Net}, t] \in S, \tag{1}$$

where SOC, e_t^{Net} are the battery state of charge and net power demand, respectively. The e_t^{Net} is the difference between the load demand and the energy generated by PV such that:

$$e_t^{Net} = e_t^{demand} - e_t^{PV}. \tag{2}$$

The SOC should be bounded by maximum and minimum limits such that:

$$SOC_{\text{min}} \leq SOC(t) \leq SOC_{\text{max}}. \tag{3}$$

We discretise the state space as shown in Equation (4) below in which the i, j, k indices represent the SOC, e_t^{Net} and t, respectively as:

$$S_{discrete} = \left\{ S_{i,j,k} \right\}, \tag{4}$$

where each index in the state space has the following levels: i = 3 levels, j = 2 levels, i.e., positive ($e_t^{Net} \geq e_t^{PV}$) or negative ($e_t^{Net} < e_t^{PV}$), and k = 48 levels. Thus, the total number of states is $48 \times 3 \times 2 = 288$.

2.2. Action Space

The action space consists of the charge, discharge, and idle command of the battery such as:

$$A = \{ a | (\text{Discharge, Idle, Charge}) \}. \tag{5}$$

At each time step t, one action is selected from the action space A. If the action "Discharge" is chosen, the battery discharges into the main grid, supplies the load, or both. In case of the action "Idle", the load demand is fulfilled by the PV source, main grid, or both. If the "Charge" action is selected, the battery is charged from the PV, the grid, or both.

2.3. Backup Controller

In this work, we used a backup controller, which acts as a filter for every control action resulting from the policy π to take care of the practical constraint, such as the inability of the battery to charge or discharge beyond its maximum and minimum SOC level, respectively. In addition, there is a certain limit of battery charging or discharging at time t. For example, if the "Charge" action is selected by the RL agent at time t and one of the discrete states is ($\mathrm{e_t^{Net}} > \mathrm{e_t^{PV}}$), then the battery should charge from the main grid up to a certain limit (Δe) defined in Table 1 even if the capacity of the battery is more than Δe. Moreover, if at time t, one of the discrete states is ($\mathrm{e_t^{Net}} < \mathrm{e_t^{PV}}$) and the RL agent selects the action "Charge", the battery will charge from the extra PV available (after fulfilling the load demand) by respecting the charging rate parameter of the battery. If for example the current PV power is more or less than the charging rate of the battery, the battery is charged up to the maximum charging rate (Δe) or current PV power, respectively.

Table 1. Chosen parameters of the microgrid used in this paper.

Name	Values
Total capacity of the battery	12,000 KWh
Max/Min charging rate of battery (Δe)	(2300/2) KW
SOCmax	100%
SOCmin	70%
Initial SOC of the battery (SOC(0))	[min, max]
Time step lenght (Δt)	30 min
α, γ, ε	0.5, 0.5, 0.6
Total iterations (Offline)	10,000–15,000

2.4. Reward

The reward function $r(s_t,a_t)$ is the immediate incentive gained by taking a specific action a at time t in state$s$. The reward function is chosen to minimise the running cost of importing power from the grid and maximise the revenue of selling power to the grid. The cost is calculated every 30 min (as Δt = 30) by multiplying the respective tariff rates, as mentioned in Section 2.5. The reward function is the negative of the cost of imported energy or the cost of exported energy. Hence, the reward function can be formulated as follows:

$$r(s_t,a_t) = \left\{ \begin{array}{ll} -P_t^{grid} \times \Delta t \times Tariff_{imp}, & P_{grid} \geq 0 \\ P_t^{grid} \times \Delta t \times Tariff_{exp}, & P_{grid} < 0 \end{array} \right\}, \tag{6}$$

where $Tariff_{imp}$ and $Tarriff_{exp}$ are the import and export tariffs, respectively. $P_t^{grid}$is the grid power and is given by:

$$P_t{}^{grid} = e_t{}^{Net} + P_t{}^{batt}, \tag{7}$$

where P_t^{batt} is the power used to charge the battery.

2.5. Tariff

The import tariff has three different values depending on the time of use:

$$Tariff_{imp} = \left\{ \begin{array}{ll} 0.05£/\text{kWh} & \text{low peak, } 22:00 \text{ to } 8:00 \\ 0.08£/\text{kWh} & \text{medium peak, } 9:00 \text{ to } 12:00 \\ 0.171£/\text{kWh} & \text{high peak, } 19:00 \text{ to } 21:00 \end{array} \right\}. \tag{8}$$

The export tariff does not vary and it is $Tariff_{exp} = 0.033£/\text{kWh}$.

3. Q-Learning Algorithm

The backbone of the Q-learning algorithm is based on the two components described in Equation (9) as:

$$R_t{}^{\pi} = r(s_t, a_t) + \sum_{i=1}^{\infty} \gamma^i . r(s_{t+i} . a_{t+i}). \tag{9}$$

The first component shows the impact of the current action on future rewards, and the second component is the total discounted rewards at time step t under a given policy π. Therefore, R_t^{π} is defined as the sum of the instant reward at time step t plus the future discounted rewards. The parameter γ is the discount factor used to determine the importance of future rewards from the next time step (t + 1) up to infinity. If γ = 0, the algorithm considers the current reward only, while if γ = 1, both current and future rewards have equal weight. In *Q*-learning, the policy is learned implicitly without any prior knowledge. This is done by approximating the action-value function by repeatedly updating the $Q(s_t, a_t)$ through experience such as:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r(s_t, a_t) + \gamma \max Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]. \tag{10}$$

To approximate the Q-table, it is important to estimate all state–action pairs of the table. Parameter α is the learning rate that determines how much the newly obtained value overrides the old value of $Q(s_t, a_t)$. If $\alpha = 0$, the newly obtained information is ignored during training, whereas if $\alpha = 1$, only the latest information is used. Therefore, in Q-learning, the selection of α (ranging between 0 and 1) is very important to keep a balance between the old and new information. The RL agent takes random actions in the beginning if it follows an ε-greedy policy, which is adopted in this work. The idea behind the purely ε-greedy approach is to try every decision once and then keep picking the one that results in the highest reward as learning progresses. After certain iterations and by performing different actions in each state of the Q-table, the agent learns to maximise the value (state–action) of the Q-table by taking greedy actions. Random and greedy actions correspond to exploration (ε) versus exploitation, respectively. In this work, taking a greedy action, the decision is based on:

$$\varepsilon \leftarrow \varepsilon / \sqrt{M(s) - M_{\max}} \tag{11}$$

where $M(s)$ is the number of times a certain action is taken in a specific state. $M_{\max}$ is the maximum constant value selected after which greedy actions are selected by the Q-learning algorithm.

3.1. Offline RL Implementation

In this section, the implementation of offline RL to control BESS in microgrids will be discussed. At the beginning of each day, the forecasted PV and load data are gathered as inputs to RL. Then, Q-learning is run using the same input data until convergence is achieved. The policy developed at the end of this phase is used to generate the charging, discharging and idle commands for the next 24 h. This strategy is repeated for each day. The backup controller monitors the control parameters of the battery. After selecting the control actions of the battery from Q-learning, the backup controller ensures that all physical constraints and limitations are met before actually applying the battery actions on the physical system. Figure 2 illustrates the offline implementation of RL-based EMS.

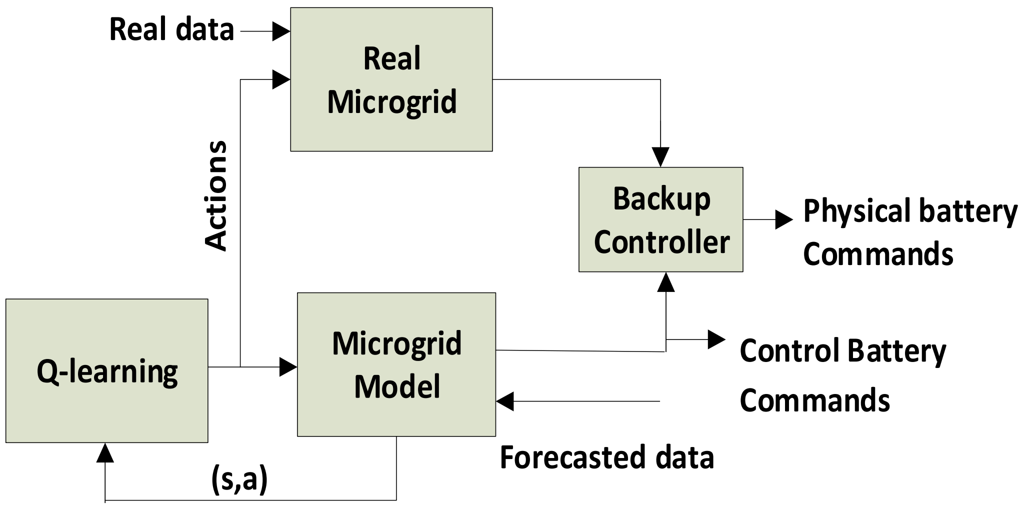

Figure 2. Offline implementation of the RL algorithm.

This offline RL implementation is similar to the traditional EMS approaches such as MILP, where estimated data are used by the optimiser to produce decision variables, such as charging/discharging/idle commands of the battery. The estimated synthetic data for PV generation and load consumption for the next 24 h are used by RL to schedule the battery command. Each episode of one day consists of 48 steps (30 min time interval). The RL keeps on using the same data until convergence is achieved. A total of 10,000–15,000 iterations were employed for better convergence. The optimised battery commands are dispatched at the start of the following day for the battery to operate in real time. The same process is repeated for every day of the year. As we are interested to find the total cost of a complete year (365 days), for the initialization of offline Q-learning, the Q-table simply initialises the action-value function at time step 0 with the value of 0 or ∞.

3.2. Online RL Implementation

Online RL is applied directly to real data in real time. Therefore, the agent learns the optimal policy by interacting with the real system. There is no pre-training in this online approach, unlike offline techniques. Figure 3 shows the online RL for EMS. The online RL algorithm updates the actions of the battery and dispatches them every 30 min in real time regardless of the status of convergence. Learning can be very slow, especially in first few days. Before convergence, the performance would be suboptimal. With time, the agent develops an optimal policy. The function of the backup controller in the online RL implementation is the same as for offline RL, as described in Section 2.3.

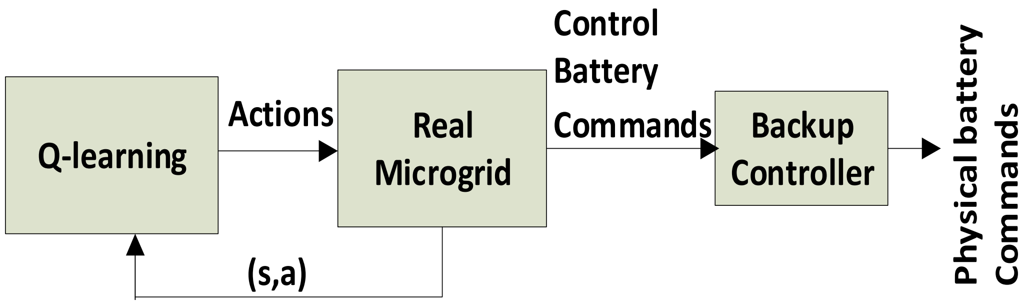

Figure 3. Online implementation of the RL algorithm.

There is no separate training stage in online RL; rather, the agent gathers experiences during real-time interaction with the environment. Therefore, in the beginning, the Q-table is initialised with a shortsighted future reward with the algorithm hyper-parameter γ set to 0. Then, the table will be updated in real time by interacting with the real system. This simple initialisation step reduces the convergence time substantially as per [44].

3.3. Prediction Error Generation

To compare the performance of both offline and online Q-learning, there is a need to create a difference between forecasted data (PV and load) and real data to represent the prediction error. We use an algorithm to add random noise to the real net power demand given by $e_t^{Net} = e_t^{demand} - e_t^{PV}$. The real PV and load profiles for a complete year are obtained from [45]. The noise is generated using normally distributed white Gaussian noise having different standard deviation (σ) values. For each σ, five noise profiles are generated, averaged, and then added to the real net power demand data to produce the forecasted net power demand, as shown in Figure 4. The increase in the σ value will increase the standard deviation error. Thus, the forecasted net demand with higher sigma values represents an increasing trend of deviation with respect to the real power demand and vice versa.

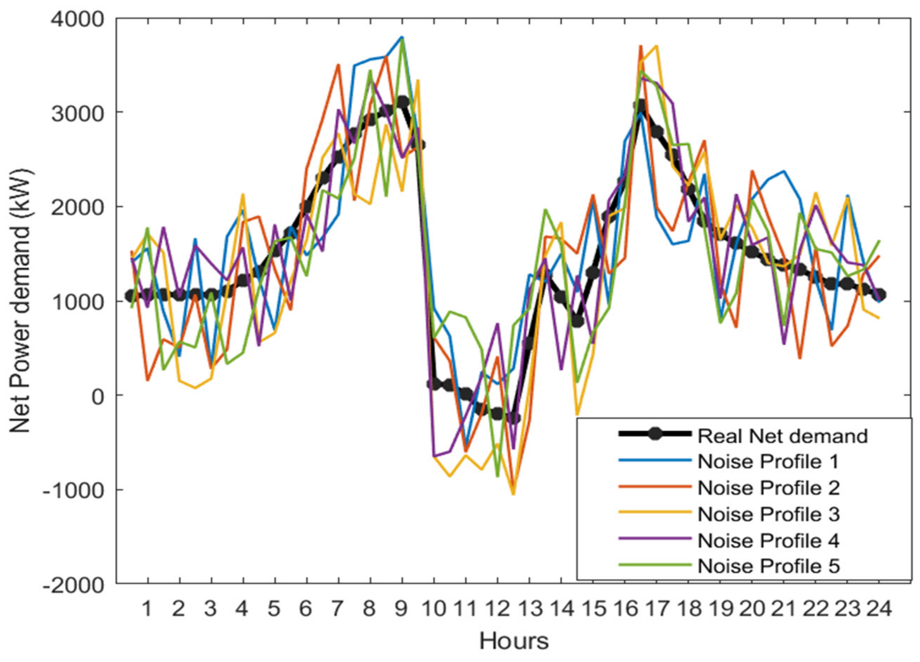

Figure 4. Real and forecasted net demand per hour for one day after noise addition (at σ =100).

4. Simulation Results

This section evaluates the performance of the offline and online optimisation techniques. The offline and online RL were both applied on a daily basis with an interval of 30 min. Firstly, we compared the results of offline RL with MILP to analyse the efficiency of both algorithms in terms of cost saving in a grid-tied microgrid system. Then, we compared offline RL with online RL after establishing a benchmark between the offline RL and MILP techniques. In this regard, both the offline and online RL optimisation techniques need to be investigated in terms of cost saving per year. This work compares the behaviour of both the approaches when there is a different percentage of errors present between forecasted and real data profiles (PV and load). This information can be used to decide between offline and online RL when the real data (PV and Load) profiles deviate from the forecasted data. The real data assumed in this work are gathered from the online open-source data platform in reference [45]. The chosen parameters used to simulate the behaviour of the microgrid are provided in Table 1.

Below, Figure 5 shows the average net forecasted $e_{forecast}^{sum/year}/5$ demand per year using all five samples at each σ. Then, Equation (12) is used to find the percentage error between the real and forecasted power demand per year:

$$Net^{error/year} = \frac{1}{n}\sum_{1}^{n}\left(\frac{e_{Real}^{sum/year} - e_{forecast}^{sum/year}}{e_{Real}^{sum/year}}\right). \tag{12}$$

The error bars in Figure 5 above show the $Net^{error/year}$ for each sigma using the highest and lowest sample of $e_{forecast}^{sum/year}$. The arrows indicating 1 and 2 in Figure 5 indicate the constant $e_{real}^{sum/year}$ and varying $Net^{error/year}$ in Equation (12), respectively.

Figure 6 depicts the difference between the generated $e_{forecast}^{sum/year}/5$ and the real power demand. The difference of the total real power demand per year with the highest and lowest sum of the generated forecasted power demand per year out of five samples at each sigma is described using the error bars on the secondary axis of Figure 6.

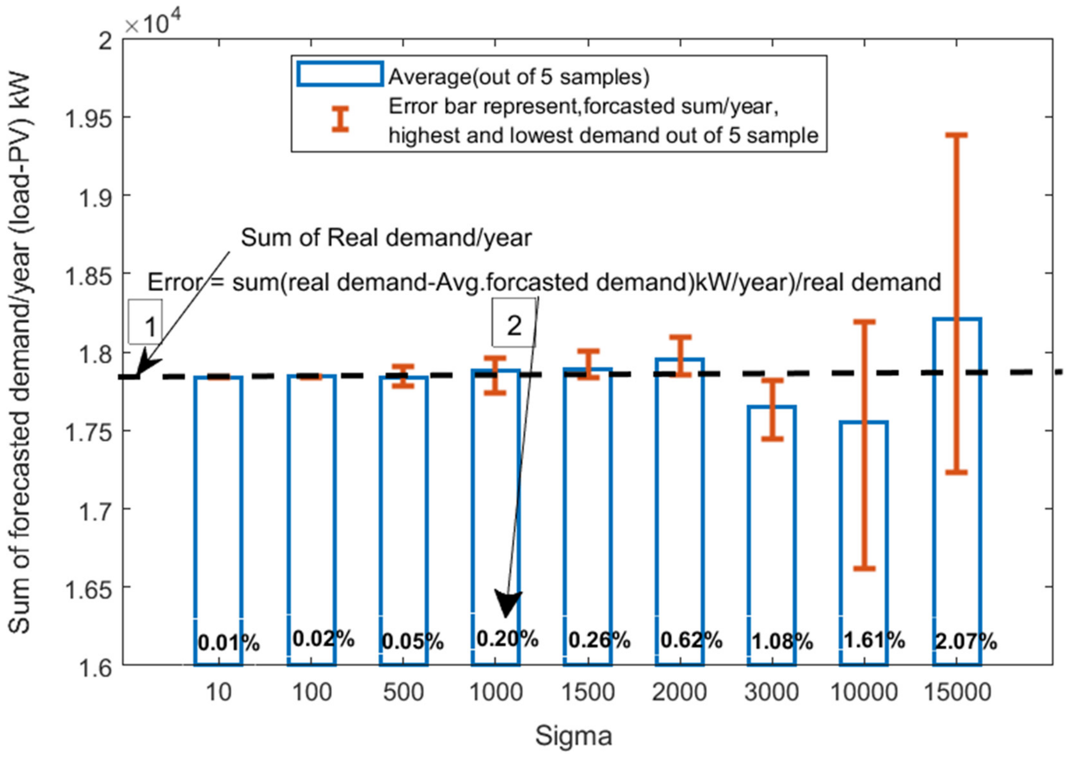

Figure 5. Average sum of the net forecasted demand per year out of five random samples.

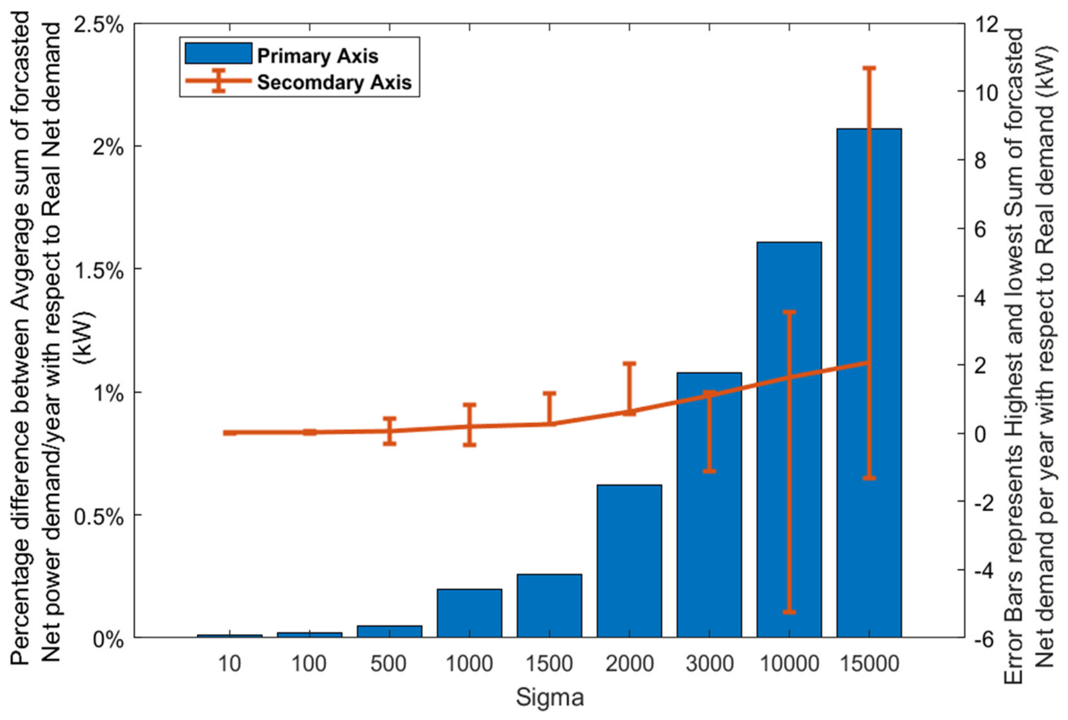

Figure 6. Average sum of forecasted demand with respect to real demand per year.

The two datasets, real (original) and forecasted (synthetic), are related to the net power demand per year and used to compare the performance of offline and online RL. Firstly, we compare offline MILP with offline RL by considering both the forecasted and real data profiles (PV, load demand) to be the same for both approaches. The results (Figure 7) show that both approaches have almost identical performance in terms of cost optimisation of the microgrid, with negligible difference. Over that small difference, MILP behaves slightly better due to the convergence requirements of RL. Therefore, the offline approaches, such as MILP and RL, which use the same forecasted and real data (having 0% error) for training, are equally good in real time.

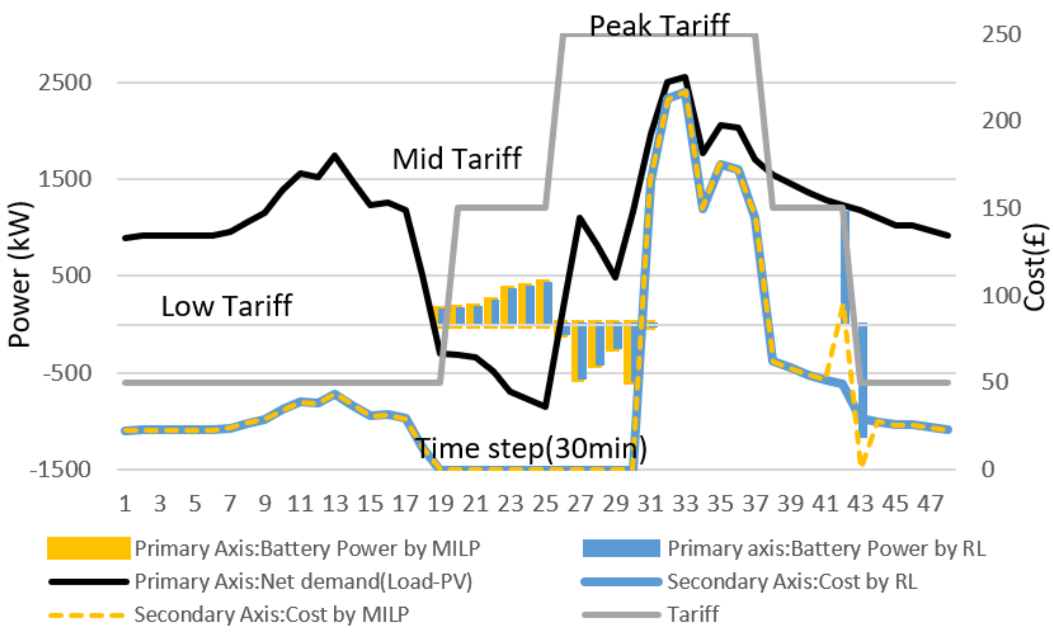

Figure 7. Comparison between MILP and RL in terms of cost per hour.

Figure 8 shows the convergence in terms of cost using offline RL. In this figure, only real data (PV and load) profiles were employed to analyse the convergence pattern of the offline RL as an ideal case. In the beginning, the cost is high and the curve shows random behaviour. As the number of episodes increases, the learning ability of the agent improves until the Q-table converges to show the optimal cost.

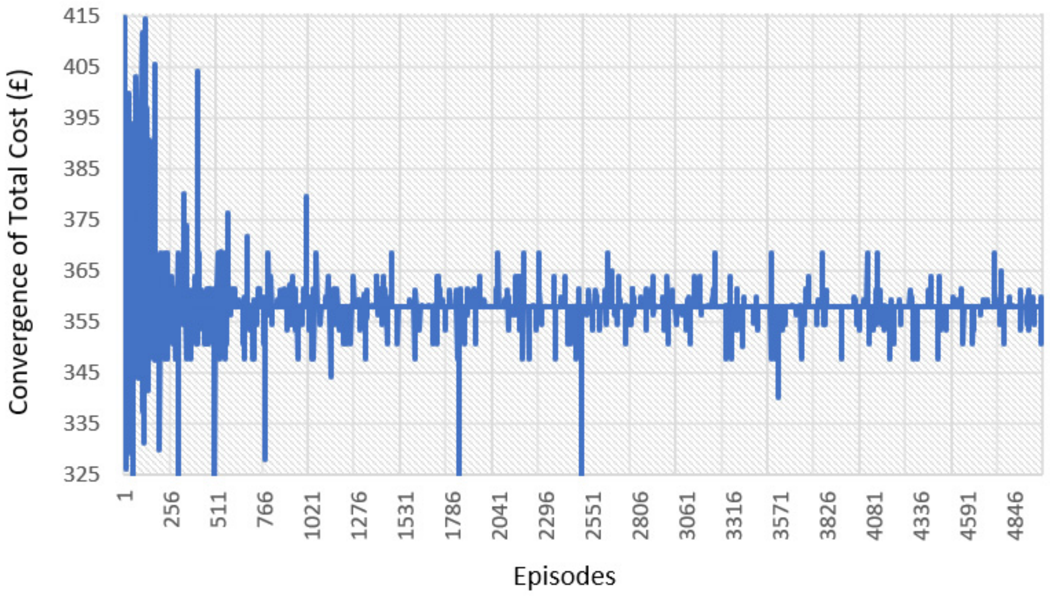

Figure 8. Convergence of the cost during the offline RL algorithm training.

4.1. Offline vs. Online RL with No Forecast Error

The performance of the two RL approaches is first compared when there is no forecast error; i.e., the forecast data are the same as real data. Of course, this ideal situation does not exist in practice, but it provides an initial benchmark for the results. Both RL approaches are implemented as explained earlier in Section 3.1 and Section 3.2. In addition, MILP was also used to optimise the battery operation. Figures 9 and 10 show the energy imported from the grid with respective cost on a daily basis, respectively. It can be seen that when the forecast error is zero, offline RL produces superior results. In Figures 9 and 10, it can be seen that before the convergence of the online RL algorithm, a higher amount of grid energy is imported; therefore, the cost is also higher with respect to offline RL in the initial days. However, as the days progress and online RL learning converges to the optimal policy, the controller follows the same pattern as offline RL in terms of cost saving and reducing the imported energy. In this work, convergence was achieved in between 75 and 90 days in the case of online RL.

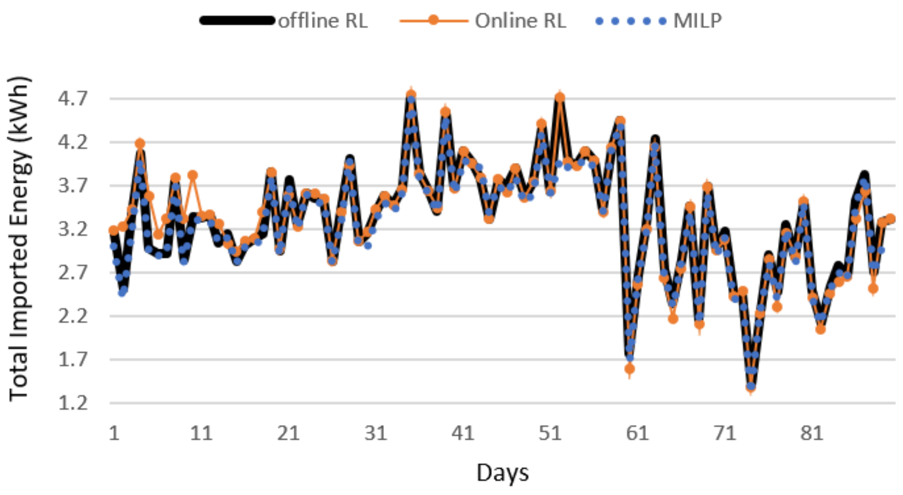

Figure 9. Performance of offline (ideal scenario) vs. online RL in terms of imported energy per day.

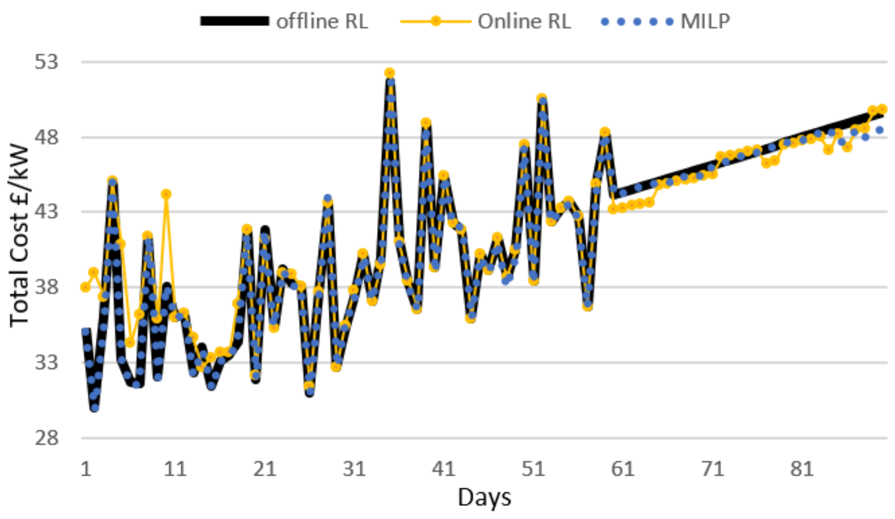

Figure 10. Performance of offline (ideal scenario) vs. online RL in terms of cost per day.

Figure 11 shows the overview of the battery actions generated in an average day after applying the Q-learning algorithm. We used the offline RL for a single day in Figure 11 to show the different states of the battery after convergence during different time intervals of the day with respect to load demand and PV availability. The difference between the load and PV (e_t^{Net}) is shown in the graph below at each time step. The e_t^{Net} can be negative if the PV power generated is greater than the load demand at time t. The suggested battery actions by the RL agent pass through the backup controller to accommodate all physical constraints, as described in Section 2.3. As shown in Figure 11, when $e_t^{PV} > e_t^{Net}$ at time t, the battery charges from the current PV power up to the maximum level of Δe after fulfilling the load demand. The outstanding PV power is sold to the main grid. During discharging, the battery discharges up to maximum level of Δe either to fulfil the load demand or sell power to the utility grid.

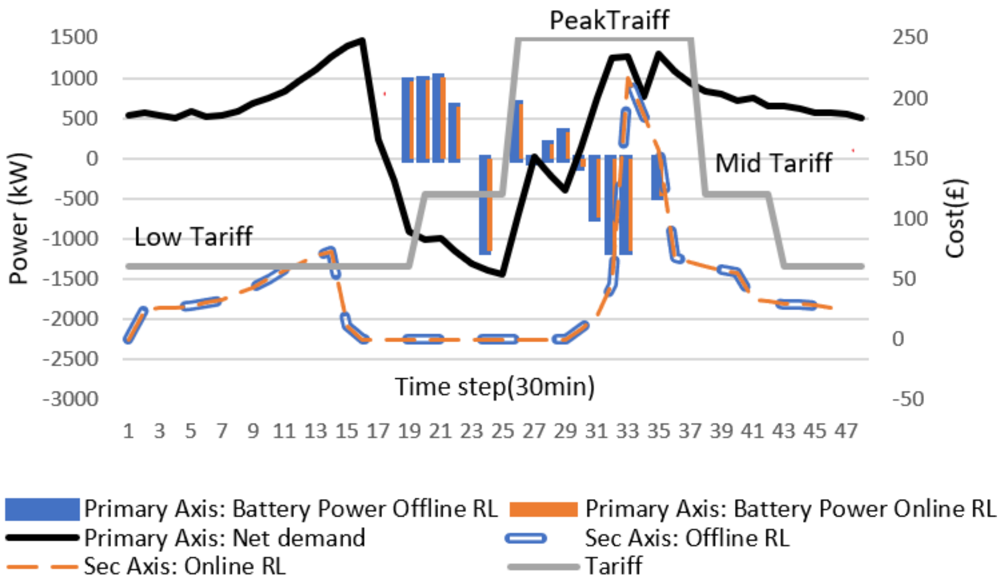

Figure 11. Comparison of offline vs. online RL after convergence.

Figure 11 is the randomly selected day of the year when both online and offline RL converge. The plot shows the same battery actions for both offline and online RL. Therefore, the cost achieved in both approaches are also same.

4.2. Offline vs. Online with Varying Forecast Error

The forecasted profiles generated in Section 3.3 are used by the offline RL to create the battery charge/discharge/idle commands, which are then applied to the real data as was explained in Section 3.1 and Figure 2. For online RL, real data are used to generate the battery commands that are directly applied to the physical microgrid system. Figure 12 shows the overview of the battery actions (kW) generated in an average day after introducing 1.6% forecasted error with respect to the real net demand. Figure 12 also showed the daily cost of offline and online RL. The offline RL cost is higher than that of online RL at the time steps 31 to 35. Therefore, the overall average cost of offline RL in a day is higher than that of online RL (after convergence).

Figure 13 shows the optimal average cost achieved per year for both offline and online RL.

The results show that the average cost and imported energy of the offline Q-learning increase as the relative error between the forecasted and real power demand grows or vice versa. The error between real and synthetic predicted profiles are calculated using Equation (12). A rise in the value of the standard deviation reflects the increase in the error (in percent), as shown in Figure 5. Therefore, the noise level (σ) and the relative error are proportional to each other. At the start, when the error of the forecasted demand with respect to the real demand is low, the offline Q-learning performs better in comparison with the online RL in terms of cost optimisation per year. However, as the error increases, for example at 1.61% (between forecasted and real net demand), the online Q-learning begins to perform better and results in a lower cost than the offline Q-leaning.

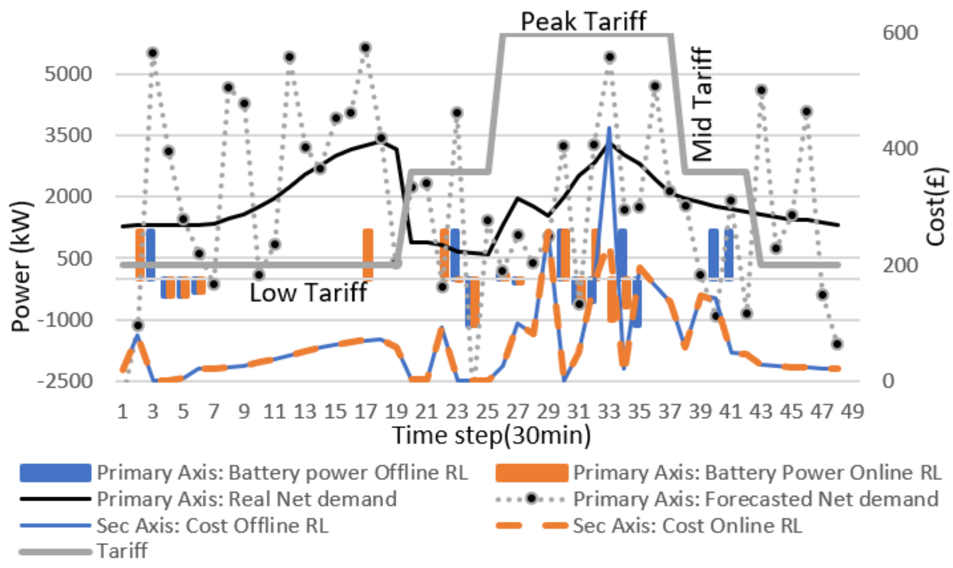

Figure 12. Comparison of offline vs. online RL (with forecast error) after convergence.

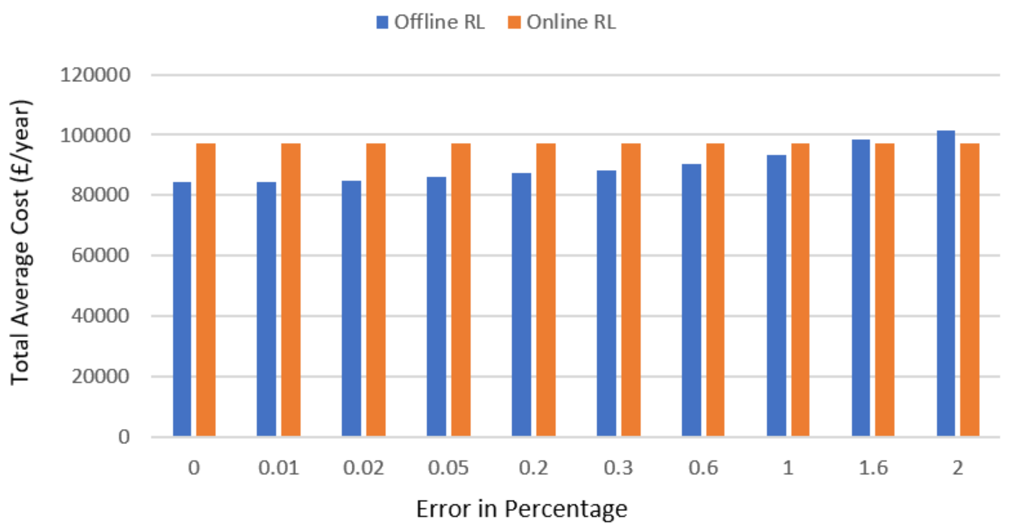

Figure 13. Performance of online vs. offline RL in terms of average total cost per year.

5. Discussion

In this study, we report a comparison between the offline and online optimisation approaches to reduce the operational cost of a grid-connected microgrid by optimally managing the BESS in the presence of forecasting error. A variation between the forecasted and real-time demand may occur due to a change in the weather, ultimately affecting the suggested real-time optimal battery actions obtained through offline training. Hence, the results may not be optimal in terms of reducing cost. To avoid the complication of using forecasted data profiles (PV, load) offline, the literature suggested the online Q-learning [44]. However, online Q-learning needs some days to converge, as its training happens online after exploring real-time data. This knowledge gap in the existing research indicated a need for thorough analysis and comparison of both offline and online approaches. Therefore, we proposed the comparison between offline versus online algorithms on an annual basis. Conventional MILP and offline RL show approximately similar behaviour in terms of saving cost in microgrid operation. Then, the average annual costs of offline and online RL approaches were analysed using different forecasted data profiles. The synthetic forecasted data with respect to real data were produced by adding random white Gaussian noise with specified standard deviation.

Table 2 suggested the possible implementation of offline or online RL on modern research to optimise the economy of power systems. For example, different type of optimisation problems were solved by using different methods in the past, as mentioned in Section 1. These solutions faced a range of challenges, including convergence, inefficient optimisation (cost, Co2 Emission), and computational time under different or varied conditions. The scope of this article suggests that reinforcement learning (offline/online) has full potential to deal with the different types of optimisation problems and challenges.

Table 2. Proposed offline and online RL on the current application scenario.

Reference	Application	Used Method	Future Approach and Strategy
[3]	Sizing large-scale thermal energy storage (TES)	MILP	Apply offline day beforeApply online in real time
[4]	Minimise the use of fossil fuels	LP and MILP	Online RL
[7]	Reduce the cost	GAMSCPLEX	Online RL
[27]	Multi-objective Optimisation	MINLP guided by Q-learning	Multi-objective RL (online/offline)
[46]	Control load shedding	IOT, mathematical modelling	Online RL
[47]	Energy trading and security	Blockchain based	Compare block-chain based mechanism with RL (online/offline)
[48]	Power management	Fuzzy logic controller	Offline RL
This work	BESS management in MG	Offline and online RL	Apply offline RL for training, online RL at real time

6. Conclusions

The following are the key findings of this paper:

- When the error is in between 0 and 1.5%, the offline RL algorithm performs better in terms of cost with respect to the online RL.
- In the first few days, RL performs better as it converges from day 1, while online RL shows better results in terms of cost after a few days. The number of days may vary depending upon the difference between the real and forecasted demand.
- The operating cost of the microgrid is proportional to the imported energy from the main grid by considering PV and battery operating cost equal to zero.
- The computational cost and time of offline RL is higher than that of online RL.
- In the literature [10–18], it was evident that the forecasting of PV has less accuracy than load forecasting.

Therefore, a higher difference between forecasted and real PV and load profiles suggests adopting an online Q-learning approach. For example, there are certain countries and areas where the forecasting of PV and load demand are not certain due to abrupt changes in the weather condition or the user behaviour. The actual energy demand at run times may change a lot in contrast to the predictions. The online Q-learning for cost optimisation provides a better solution in these regions. While operating in favourable prediction conditions, the offline Q-learning performs better.

In the future, there may be other quantification methods such as root mean square error (RMSE) that can be used to introduce errors in forecasted data with respect to real data to compare these two approaches for other type of noise distributions. In the future, both offline and online Q-learning approaches may be employed as a two-layer structure. This can provide a better and more efficient solution for the cost optimisation in a real microgrid.

Author Contributions: Conceptualisation, M.A. and K.H.A.; methodology, K.H.A.; validation, M.A. and S.D.; formal analysis, E.A.; investigation, K.H.A. and M.S.; resources, A.A.T.; writing—

original draft preparation, K.H.A.; writing—review and editing, M.A., E.A., S.D., A.A.T. and K.H.A.; supervision, M.A. and A.A.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Engineering and Physical Sciences Research Council (EPSRC) grant number EP/T025875/1 and The APC was partially funded by the editor allowance.

Data Availability Statement: The data is available from the lead or the corresponding author upon reasonable requests.

Acknowledgments: The financial support was provided by the Engineering and Physical Sciences Research Council (EPSRC), U.K. under research Grant No. EP/T025875/1. K.H.A. would also like to thank Sukkur Institute of Business Administration University, Pakistan for fully funded Ph D studentship.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

ANN	Artificial Neural Networks
BESS	Battery Energy Storage System
CHP	Combined Heat and Power
DDQN	Double Deep Neural Network
DG	Diesel Generator
DP	Dynamic Programming
DSP	Discretised Step Transformation
EMS	Energy Management System
FC	Fuel Cell
GAMS	General Algebraic Modelling System
LP	Linear Programming
MC	Micro Turbine
MILNP	Mixed-Integer Nonlinear Programming
MILP	Mixed-Integer Linear Programming
MIP	Mixed Integer Programming
PV	Photovoltaic
RES	Renewable Energy Sources
RL	Reinforcement Learning
SOC	State of Charge

References

1. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; MIT Press: Cambridge, UK, 2018.
2. Chen, Y.-H.; Lu, S.-Y.; Chang, Y.-R.; Lee, T.-T.; Hu, M.-C. Economic analysis and optimal energy management models for microgrid systems: A case study in Taiwan. *Appl. Energy* **2013**, *103*, 145–154. [CrossRef]
3. Benalcazar, P. Optimal sizing of thermal energy storage systems for CHP plants considering specific investment costs: A case study. *Energy* **2021**, *234*, 121323. [CrossRef]
4. Dolara, A.; Grimaccia, F.; Magistrati, G.; Marchegiani, G. Optimization Models for Islanded Micro-Grids: A Comparative Analysis between Linear Programming and Mixed Integer Programming. *Energies* **2017**, *10*, 241. [CrossRef]
5. Luna, A.C.; Diaz, N.L.; Graells, M.; Vasquez, J.C.; Guerrero, J. Mixed-Integer-Linear-Programming-Based Energy Management System for Hybrid PV-Wind-Battery Microgrids: Modeling, Design, and Experimental Verification. *IEEE Trans. Power Electron.* **2016**, *32*, 2769–2783. [CrossRef]
6. Cosic, A.; Stadler, M.; Mansoor, M.; Zellinger, M. Mixed-integer linear programming based optimization strategies for renewable energy communities. *Energy* **2021**, *237*, 121559. [CrossRef]
7. Li, Y.; Yang, Z.; Li, G.; Zhao, D.; Tian, W. Optimal Scheduling of an Isolated Microgrid With Battery Storage Considering Load and Renewable Generation Uncertainties. *IEEE Trans. Ind. Electron.* **2018**, *66*, 1565–1575. [CrossRef]
8. Stluka, P.; Godbole, D.; Samad, T. Energy management for buildings and microgrids. In Proceedings of the 2011 50th IEEE Conference on Decision and Control and European Control Conference, Orlando, FL, USA, 12–15 December 2011; pp. 5150–5157. [CrossRef]
9. Mosa, M.A.; Ali, A. Energy management system of low voltage dc microgrid using mixed-integer nonlinear programing and a global optimization technique. *Electr. Power Syst. Res.* **2020**, *192*, 106971. [CrossRef]

10. Dong, J.; Olama, M.M.; Kuruganti, T.; Melin, A.M.; Djouadi, S.M.; Zhang, Y.; Xue, Y. Novel stochastic methods to predict short-term solar radiation and photovoltaic power. *Renew. Energy* **2019**, *145*, 333–346. [CrossRef]
11. Soubdhan, T.; Ndong, J.; Ould-Baba, H.; Do, M.-T. A robust forecasting framework based on the Kalman filtering approach with a twofold parameter tuning procedure: Application to solar and photovoltaic prediction. *Sol. Energy* **2016**, *131*, 246–259. [CrossRef]
12. Pedro, H.; Coimbra, C.F. Assessment of forecasting techniques for solar power production with no exogenous inputs. *Sol. Energy* **2012**, *86*, 2017–2028. [CrossRef]
13. Kushwaha, V.; Pindoriya, N.M. A SARIMA-RVFL hybrid model assisted by wavelet decomposition for very short-term solar PV power generation forecast. *Renew. Energy* **2019**, *140*, 124–139. [CrossRef]
14. Lv, Z.; Wang, L.; Guan, Z.; Wu, J.; Du, X.; Zhao, H.; Guizani, M. An Optimizing and Differentially Private Clustering Algorithm for Mixed Data in SDN-Based Smart Grid. *IEEE Access* **2019**, *7*, 45773–45782. [CrossRef]
15. Voyant, C.; Notton, G.; Kalogirou, S.; Nivet, M.-L.; Paoli, C.; Motte, F.; Fouilloy, A. Machine learning methods for solar radiation forecasting: A review. *Renew. Energy* **2017**, *105*, 569–582. [CrossRef]
16. Nti, I.K.; Teimeh, M.; Nyarko-Boateng, O.; Adekoya, A.F. Electricity load forecasting: A systematic review. *J. Electr. Syst. Inf. Technol.* **2020**, *7*, 1–19. [CrossRef]
17. Kuster, C.; Rezgui, Y.; Mourshed, M. Electrical load forecasting models: A critical systematic review. *Sustain. Cities Soc.* **2017**, *35*, 257–270. [CrossRef]
18. Jahan, I.S.; Snasel, V.; Misak, S. Intelligent Systems for Power Load Forecasting: A Study Review. *Energies* **2020**, *13*, 6105. [CrossRef]
19. Mbuwir, B.V.; Ruelens, F.; Spiessens, F.; Deconinck, G. Battery Energy Management in a Microgrid Using Batch Reinforcement Learning. *Energies* **2017**, *10*, 1846. [CrossRef]
20. Liu, W.; Zhuang, P.; Liang, H.; Peng, J. Distributed Economic Dispatch in Microgrids Based on Cooperative Reinforcement Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 2192–2203. [CrossRef] [PubMed]
21. Jiang, B.; Fei, Y. Smart Home in Smart Microgrid: A Cost-Effective Energy Ecosystem with Intelligent Hierarchical Agents. *IEEE Trans. Smart Grid* **2014**, *6*, 3–13. [CrossRef]
22. Lu, X.; Xiao, X.; Xiao, L.; Dai, C.; Peng, M.; Poor, H.V. Reinforcement Learning-Based Microgrid Energy Trading With a Reduced Power Plant Schedule. *IEEE Internet Things J.* **2019**, *6*, 10728–10737. [CrossRef]
23. Foruzan, E.; Soh, L.-K.; Asgarpoor, S. Reinforcement Learning Approach for Optimal Distributed Energy Management in a Microgrid. *IEEE Trans. Power Syst.* **2018**, *33*, 5749–5758. [CrossRef]
24. Zhou, S.; Hu, Z.; Gu, W.; Jiang, M.; Zhang, X.-P. Artificial intelligence based smart energy community management: A reinforcement learning approach. *CSEE J. Power Energy Syst.* **2019**, *5*, 1–10. [CrossRef]
25. Bui, Y.-H.; Hussain, A.; Kim, H.-M. Double Deep *Q* -Learning-Based Distributed Operation of Battery Energy Storage System Considering Uncertainties. *IEEE Trans. Smart Grid* **2019**, *11*, 457–469. [CrossRef]
26. Kim, B.-G.; Zhang, Y.; van der Schaar, M.; Lee, J.-W. Dynamic Pricing and Energy Consumption Scheduling With Reinforcement Learning. *IEEE Trans. Smart Grid* **2015**, *7*, 2187–2198. [CrossRef]
27. Yoldas, Y.; Goren, S.; Onen, A. Optimal Control of Microgrids with Multi-stage Mixed-integer Nonlinear Programming Guided Q-learning Algorithm. *J. Mod. Power Syst. Clean Energy* **2020**, *8*, 1151–1159. [CrossRef]
28. Watkins, C.J.C.H.; Dayan, P. Q-learning. *Mach. Learn.* **1992**, *8*, 279–292. [CrossRef]
29. Perera, A.; Kamalaruban, P. Applications of reinforcement learning in energy systems. *Renew. Sustain. Energy Rev.* **2020**, *137*, 110618. [CrossRef]
30. Kuznetsova, E.; Li, Y.-F.; Ruiz, C.; Zio, E.; Ault, G.; Bell, K. Reinforcement learning for microgrid energy management. *Energy* **2013**, *59*, 133–146. [CrossRef]
31. Ji, Y.; Wang, J.; Xu, J.; Fang, X.; Zhang, H. Real-Time Energy Management of a Microgrid Using Deep Reinforcement Learning. *Energies* **2019**, *12*, 2291. [CrossRef]
32. Fujimoto, S.; Meger, D.; Precup, D. Off-policy deep reinforcement learning without exploration. *Proc. Mach. Learn. Res.* **2019**, *97*, 2052–2062.
33. Wang, Y.; Jin, H. A Boosting-based Deep Neural Networks Algorithm for Reinforcement Learning. In Proceedings of the 2018 Annual American Control Conference (ACC), Milwaukee, WI, USA, 27–29 June 2018; pp. 1065–1071. [CrossRef]
34. Zhang, K.; Yang, Z.; Başar, T. Multi-agent reinforcement learning: A selective overview of theories and algorithms. In *Handbook of Reinforcement Learning and Control*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 321–384.
35. Staddon, J.E.R. The dynamics of behavior: Review of Sutton and Barto: Reinforcement Learning: An Introduction (2 nd ed.). *J. Exp. Anal. Behav.* **2020**, *113*, 485–491. [CrossRef]
36. Das, A.; Ni, Z. A Computationally Efficient Optimization Approach for Battery Systems in Islanded Microgrid. *IEEE Trans. Smart Grid* **2017**, *9*, 6489–6499. [CrossRef]
37. Vinyals, O.; Babuschkin, I.; Czarnecki, W.M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D.H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* **2019**, *575*, 350–354. [CrossRef]
38. Boait, P.; Advani, V.; Gammon, R. Estimation of demand diversity and daily demand profile for off-grid electrification in developing countries. *Energy Sustain. Dev.* **2015**, *29*, 135–141. [CrossRef]
39. Hernandez-Aramburo, C.; Green, T.; Mugniot, N. Fuel Consumption Minimization of a Microgrid. *IEEE Trans. Ind. Appl.* **2005**, *41*, 673–681. [CrossRef]

40. Bui, V.-H.; Hussain, A.; Kim, H.-M. Q-Learning-Based Operation Strategy for Community Battery Energy Storage System (CBESS) in Microgrid System. *Energies* **2019**, *12*, 1789. [CrossRef]
41. Rancilio, G.; Lucas, A.; Kotsakis, E.; Fulli, G.; Merlo, M.; Delfanti, M.; Masera, M. Modeling a Large-Scale Battery Energy Storage System for Power Grid Application Analysis. *Energies* **2019**, *12*, 3312. [CrossRef]
42. Hernández, L.; Baladrón, C.; Aguiar, J.M.; Carro, B.; Sánchez-Esguevillas, A.; Lloret, J. Artificial neural networks for short-term load forecasting in microgrids environment. *Energy* **2014**, *75*, 252–264. [CrossRef]
43. Castronovo, M.; François-Lavet, V.; Fonteneau, R.; Ernst, D.; Couëtoux, A. Approximate bayes optimal policy search using neural networks. In Proceedings of the ICAART 2017—9th International Conference on Agents and Artificial Intelligence, Porto, Portugal, 24–26 February 2017; Volume 2, pp. 142–153. [CrossRef]
44. Kim, S.; Lim, H. Reinforcement Learning Based Energy Management Algorithm for Smart Energy Buildings. *Energies* **2018**, *11*, 2010. [CrossRef]
45. Data Platform—Open Power System Data. Available online: https://data.open-power-system-data.org/ (accessed on 8 February 2021).
46. Hussain, M.M.; Siddique, M.; Raees, A.; Nouman, M.; Javed, W.; Razaq, A. Power Management through Smart Grids and Advance Metering Infrastructure. In Proceedings of the 2020 6th IEEE International Energy Conference (ENERGYCon), Gammarth, Tunisia, 28 September–1 October 2020; pp. 767–772. [CrossRef]
47. Ahmad, R.F.; Siddique, M.; Riaz, K.; Hussain, M.M.; Bhatti, M. Blockchain based Secure Energy Trading Mechanism for Smart Grid. *Pak. J. Eng. Technol.* **2021**, *4*, 100–107. [CrossRef]
48. Al Badwawi, R.; Issa, W.R.; Mallick, T.K.; Abusara, M. Supervisory Control for Power Management of an Islanded AC Microgrid Using a Frequency Signalling-Based Fuzzy Logic Controller. *IEEE Trans. Sustain. Energy* **2018**, *10*, 94–104. [CrossRef]