Energy and AI 13 (2023) 100241

Contents lists available at ScienceDirect

Energy and AI

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

Maintaining flexibility in smart grid consumption through deep learning and deep reinforcement learning☆

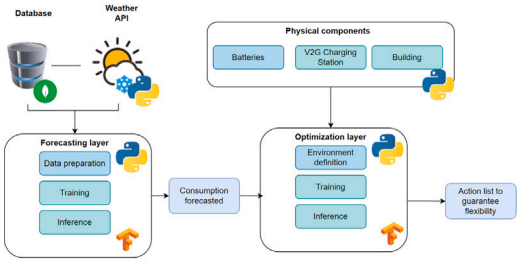
Fernando Gallego *, Cristian Martín, Manuel Díaz, Daniel Garrido

ITIS Software Institute, University of Málaga, Málaga, Spain

HIGHLIGHTS

- Deep reinforcement learning and deep learning for guaranteeing energy flexibility.
- Best DQN model achieves a complete action listing for the next hour 90% of the time.
- Optimization of smart power grids with novel technologies.
- New solution that incentivizes the use of distributed energy sources.

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Multi-agent based system
Smart grid
Distributed energy resources

ABSTRACT

The smart grid concept is key to the energy revolution that has been taking place in recent years. Smart Grids have been present in energy research since their emergence. However, the scarcity of data from different energy sources, hardware power, or co-simulation environments has hindered their development. With advances in multi-agent-based systems, the possibility of simulating the behavior of different energy sources, combining real building consumption, and simulated data, storage batteries and vehicle charging points, has opened up. This development has resulted in much research published using both simulated and physical data. All these investigations show that the main problem is that the machine learning algorithms do not fully match the real behavior, it is complex to use them to replicate the different actions to be performed. This paper aims to combine the approach of behavior prediction with state-of-the-art techniques, such as deep learning and deep reinforcement learning, to simulate unknown or critical system scenarios. A very important element in smart grids is the possibility of maintaining consumption within specific ranges (flexibility). For this purpose, we have made use of Tensorflow libraries that predict energy consumption and deep reinforcement learning to select the optimal actions to be performed in our system. The developed platform is flexible enough to include new technologies such as smart batteries, electric vehicles, etc., and it is oriented to real-time operation, being applied in an on-going real project such as the European ebalance-plus project.[1]

☆ Acknowledgments: This work is funded by the H2020 ebalanceplus project (grant agreement 864283), and the Spanish project PY20_00788 (“IntegraDos: Providing Real-Time Services for the Internet of Things through Cloud Sensor Integration”). This project has also received funding from the European Union’s Horizon Europe research and innovation programme under the Marie Skodowska-Curie grant agreement No 101086218.

* Corresponding author.
E-mail addresses: fgdc2f3@uma.es (F. Gallego), cristian@uma.es (C. Martín), mdiaz@uma.es (M. Díaz), dgm@uma.es (D. Garrido).

[1] Ebalance-plus: https://www.ebalanceplus.eu.

https://doi.org/10.1016/j.egyai.2023.100241
Received 18 November 2022; Received in revised form 25 January 2023; Accepted 14 February 2023
Available online 2 March 2023

1. Introduction

With the emergence of distributed energy sources (DER) [1] and the improvement of energy storage systems (ESS) [2,3], the relevance of the smart grid field has increased, rising the necessity to search for new solutions that optimize the efficiency of the electrical grid [4], thus increasing energy savings while satisfying the demand of the system consumers and new devices capable of both consuming and producing energy, known as prosumers. These solutions must address both the challenges of component communication(and system security), and the evaluation of scenarios with a wide variety of approaches [5]. The inclusion of renewable energy sources in today's systems offers new possibilities to meet this customer demand; a demand that is growing and where the diversity of components is increasing dramatically, leading to solutions that work with microgrids [6,7]. These grids rely on small energy sources such as solar panels, batteries, and other generators to support small electrical load groups. Numerous researchers have studied the improvement of these grids in places where distribution is more limited. These solutions not only present benefits on an economic level, reducing the number of interruptions in supply to customers the cost of energy (with the inclusion of other sources), but also increases energy efficiency and reduces blackouts. In addition, it is a proposal that is in line with Green Technologies as it reduces the human footprint, greenhouse gases, and damage to the environment. It also allows for greater scalability, offering a possible fractal view at various levels.

On the other hand, not only have DERs and ESSs improved, but changes have emerged in sectors where components were previously static, known, but where new features are appearing, such as charging stations for electric vehicles. Those technologies admit cars with batteries capable of both consuming and supplying energy [8], thus becoming another component to be managed within the grid. The variability of the devices in these grids is constantly growing, requiring that all systems that are developed be endowed with the flexibility to address their inclusion.

The availability of new grid components increases the complexity of power supply management. One of the most interesting approaches to this problem is active network management (ANM) [9]. ANM offers the possibility of controlling the power flow in such a way as to provide the amount of power needed to meet the demand requirements of the consumers participating in the system in real time [10]. This technique aims to increase the use of systems that include renewable energy sources, such as wind turbines or photovoltaic panels, alongside traditional methods of energy supply, ANM is becoming increasingly common in this type of system.

This approach solves the problem of monitoring complex systems with energy from different sources. However, this approach does not allow anticipating decisions on the actions to be taken, making it so difficult to provide the system with flexibility, time slots in which consumption can be maintained between a series of peaks (consumption pattern).

In order to find an intelligent solution that actively manages the energy of a grid, a platform based on deep learning for prediction and deep reinforcement learning for optimization of grid actions is proposed in this work. To the best of our knowledge, this is the first time both techniques are used in a smart grids context, combining the advantages of each approach, predicting the future consumption of the system on a regression problem where these algorithms stand out for their performance [11,12], and optimizing the possible actions to be performed in the next time frame, thus always meeting the expected consumption target, reducing or increasing it if necessary, and anticipating the decision making. Concretely, we will solve the problem of managing a network composed of batteries, electric vehicle charging stations, and buildings, predicting its consumption with neural networks and the actions to be performed with the multi-agent system [13,14].

The rest of the article is organized as follows: We will start by addressing the necessary background concepts in Section 2, continue with the structure, operation, and details of the platform in Section 3 and evaluate them with a use case using a real scenario in Section 4. Finally, we will end with an evaluation and discussion of the results obtained in Section 5 and Section 6.

1.1. Related works

The active management of the grid is an efficient and high-performance way in Smart Grid systems, particularly those that incorporate various energy sources such as renewables. In [15], the authors define a framework, based on reinforcement learning, that allows the management of electrical grids without the need for extensive knowledge in this type of system, they can deploy a series of environments based on hyperparameters. Our platform allows a close simulation, but it is also based on the consumption forecast of the following quarter-time slots, anticipating the decision-making process and guaranteeing the possibility of maintaining flexibility. In [16], a distributed algorithm capable of solving the decision-making problem through user–supplier communication and real-time electricity pricing is proposed. Its main objective is to maintain a balance between energy supply and demand. In this case, the authors present a series of strategies to satisfy the users' demands at all times, while our framework seeks to maintain as long as possible, making use of the feasible actions of the system, the energetic flexibility according to the users' demand.

On the other hand, the approach presented in [17] covers the prediction of the behavior of an electrical network from the data obtained from the system, making use of deep learning. The main difference with this paper is one of use dynamic systems for long-term prediction, while our prediction is focused on the short term to be very accurate and to be able to get actions as close to reality as possible. The researchers [18] define a prediction model of the energy generated by one of the Smart Grid components, a photovoltaic panel, by means of mathematical probabilities and meteorological data. In our case, at least at this time, since we do not have sources that generate energy other than batteries, we do not predict the state of our system, but rather, given the system, we seek to obtain the consumption that the network will have to maintain flexibility, although, like them, we will use deep learning. In [19], the authors present the improvement the accuracy in predicting future loads, in this case of electric vehicle stations, through the combination of grey models and recurrent neural networks, Long Short-Term Memory (LSTM). As in the previous paper, we do not predict the consumption of these vehicles specifically but of the total smart grid, and, from this, we estimate the optimal actions, among which the parked electric car's batteries charging or discharging.

2. Background

2.1. Reinforcement learning

Since the beginnings of reinforcement learning, it has been a field that has stood out within machine learning for its great potential [20]. Its main advantage lies in the possibility of reducing or detailing the problem as much as possible at each moment, being able to work with a group of agents where each can be made up in turn of another agent's set. This characteristic has allowed it to appear in numerous fields of research. The first of these, and the one that managed to highlight its great performance, is games [21], specifically chess. However, it has not been the only field [22], and so, it can be found in robotics [23] as well as in autonomous driving [24]. All these fields share the scarcity of simulators capable of modeling existing problems, and working with real physical models involves a higher cost, even more so with the technological advances in computing.

Moreover, in some cases, the scenarios to be evaluated are completely inaccessible for the device, either because of a risk to its

integrity or because the environment is not capable of reaching those conditions. A digital model capable of simulating its behavior in this type of situation allows both to explore them and to obtain knowledge about the actions to be performed and their possible consequences, being able to anticipate options in decision making, a complex problem addressed by reinforcement learning.

In smart grids, it becomes even more complex to define efficient and, above all, reliable algorithms in terms of behavior to solve the decision-making problem. This is because electricity grids are in a moment of transition with the inclusion of new, more sustainable components, such as renewable energy sources or energy storage systems.

Initially, many systems were based on physical models, evaluating real scenarios faced on a day-to-day basis. The next step was the development of systems composed of physical and mathematical models, the latter defined through a series of equations, usually provided by manufacturers or obtained through the results of physical devices. The need for accurate models that predict future behavior led to a breakthrough in the simulation of Smart Grid environments, focusing two parallel ways. The first one sought to predict their behavior with automatic learning models: Autoregressive, Autoregressive Integrated Moving Average (ARIMA) [25,26], Seasonal Autoregressive Integrated Moving Average (SARIMA), Prophet or (LSTM) [27,28], based on deep learning. All of them are highly efficient algorithms, especially when a large number of data are available; however, in situations where data are scarce, their simulation is far from the real behavior. In this article, we will focus mainly on the latter, LSTM, which will be detailed later.

Reinforcement learning has been used in this article to obtain the optimal actions to guarantee flexibility. This line has a great performance, especially when the operation of the components is known, whether or not sufficient data are available [29]. If each agent is known in detail, the environment can be designed, so the problem will pass to the solving method. These methods have grown and more and more options are becoming available for solving multi-agent problems [30]. All of them seek to obtain the maximum reward from an initial to a final state through a series of actions. These methods are mainly classified into two groups: value iteration V(s) and policy iteration $\pi(s)$, and these in turn on-policy and off-policy. Both algorithms offer great results, although the most used is the policy iteration since it tends to converge in fewer iterations. The former seeks the reward with a greedy attitude, understanding that the optimal policy is other than the one it has, while the latter updates the current one as it thinks it is the optimal one.

Lately, the Q-learning algorithm [31], a development based on policy iteration and off-policy type, has gained some relevance thanks to new papers in which a new method, deep reinforcement learning, was used [32,33]. Such a development has been used as an optimization method for the platform presented in this paper (Fig. 1).

Specifically, we have focused our solution on a concept known as deep reinforcement learning using deep Q-learning, which is widely used lately and which we will explain in the next section. This method has been used mainly in the security sections of Smart Grids [34,35]. It has also been used in the evaluation of the optimal load required for electric vehicles, taking into account the traffic conditions at that moment [36].

2.1.1. Deep reinforcement learning

Deep reinforcement learning consists in the application of techniques based on neural networks to problems modeled with reinforcement learning (Fig. 2). With this procedure, the actions that the agent can perform are decomposed in the neurons of the network, giving weight to each of the equations seen in the policy iteration: V(s, θ) and π(s, θ).

For these weights to approach the optimal ones, training is needed. In each step, the action to be performed is evaluated with the current policy of the Deep Q-Network (DQN) agent by weighting the neurons with their experiences according to the rewards. Once the indicated number of steps has been performed, we evaluate and obtain the current reward.

Their strengths lie in the high computational capacity currently available, the ease of access to it, and the great performance that neural networks have in them, being able to explore numerous scenarios earlier than with other types of algorithms, thus achieving faster convergence.

Based on the large results shown by Deep Mind [37], a new line of development with great potential appeared. This technique consisted of learning from the repetition of experiences, training the algorithm at all times which actions would have the greatest rewards. At first, it seemed to be limited to problems with few actions; however, it has adapted to solve some of them with high complexity, such as chess. Nevertheless, it is also necessary to mention that it is a solution restricted to a certain type of problem and not applicable to all, especially in those cases where a solution that satisfies the requirements is not sufficient, and only the optimal one is worthwhile.

Moreover, to apply reinforcement learning, it is necessary to define expressly and in detail all the components involved in the system, the actions that they can perform, knowing that the one that is not detailed will not be considered at any time, and the reward that will be obtained for each of them.

In the case of Smart Grids, it requires sufficient knowledge of the devices involved in the electrical network to define the different interventions that they can perform, limiting the problem to the context in which it is evaluated.

In our case, we have evaluated the environment consisting of a storage battery with a capacity of 120 kW and three options of different charges and discharges, a V2G-type electric vehicle charging station that allows a charge and discharge up to 15 kWh, thus being able to use the battery of the vehicles as an injection to the grid, and the air conditioning system of a building as we will detail in Section 4 (Fig. 3).

2.2. Deep learning

Since its appearance with the simple perceptron [38], it has been possible to observe its great potential. However, it has not been until the 21st century that, with the advance in technology and, thus, in computational capacity, it has been used massively. This field has been very important for the development of other less closely related fields, such as medicine [39], civil infrastructure [40], security [41], or robotics [42].

The use of these algorithms is due to the high performance they offer [43] provided that a sufficient amount of data is available. It is mainly used for training. The usefulness of this type of model in the Smart Grid field covers a wide range of opportunities and challenges that lack a solution or whose solution does not have the desired result be it because its development is costly, its behavior is far from the real one, or its execution time is not suitable. One of these challenges is, mainly, the application of neural networks for cybersecurity or demand prediction, a field closely related to deep learning, especially in recent years.

This connection is due to the marked boom both fields are experiencing in recent years, and the improvement in deep learning usually leads to the study of its impact on the Smart Grid. With the emergence of the first fully connected models [44], great approximations of the behaviors of power grid components were achieved. After the development of convolutional layers, these approximations were even greater [45]. Currently, recurrent neural networks, i.e. networks whose output from a higher layer can feed a neuron that is in a previous layer, obtaining great results [46] when calculating behaviors of objects whose tendency is to repeat over the seasons in such a manner that only the intensity in each of them varies [47].

In this study, we will work both with fully connected networks and Long Short-Term Memory to predict the demand of an environment consisting of a building, an external battery, and an electric vehicle

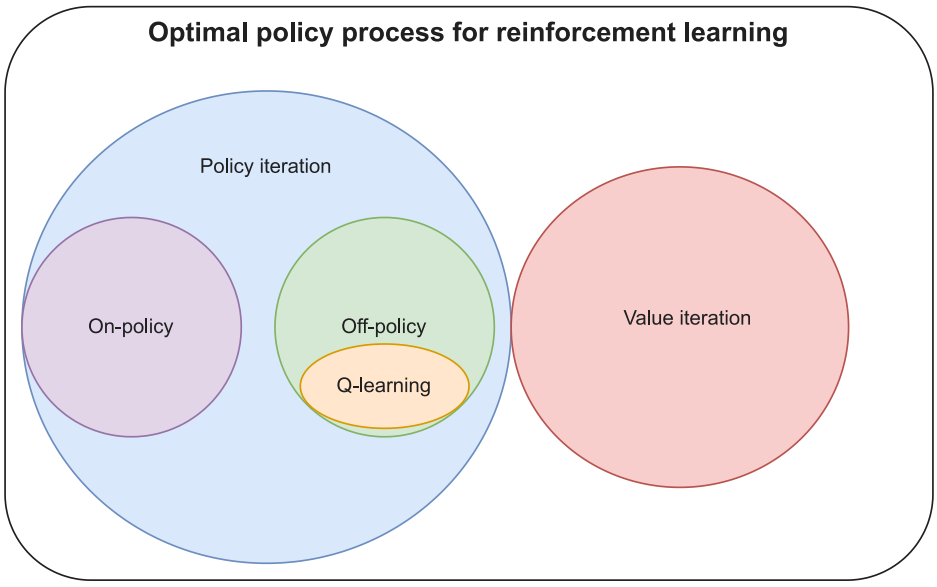


Fig. 1. Optimal policy acquisition.

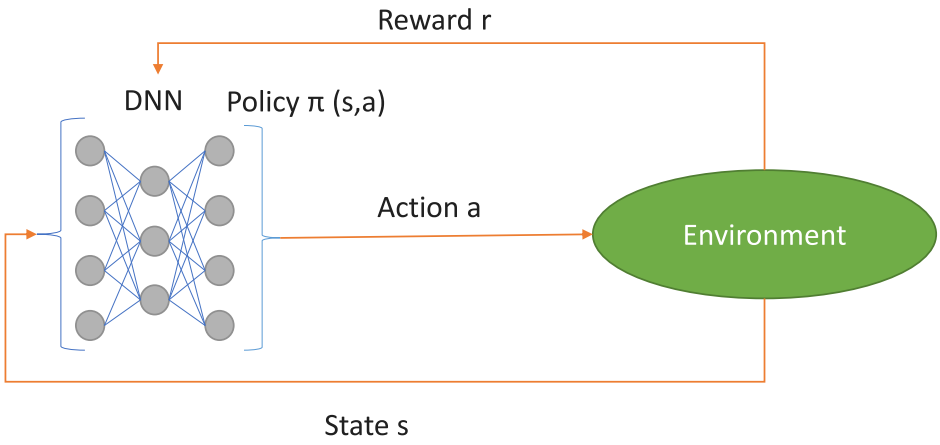


Fig. 2. Deep reinforcement learning.

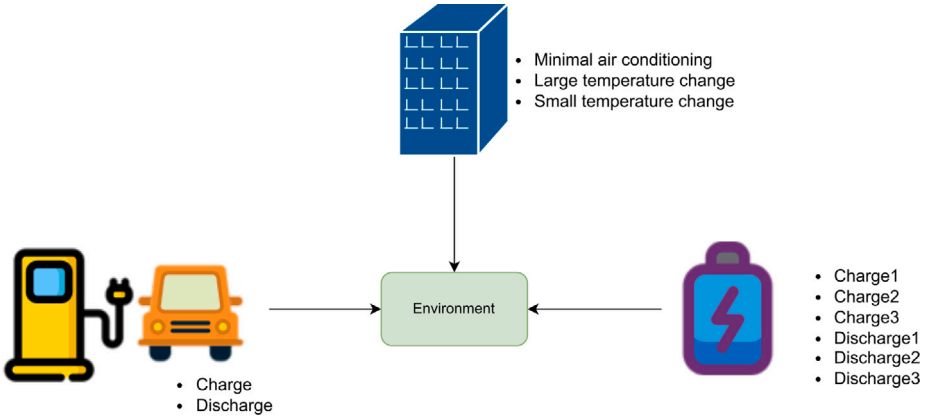


Fig. 3. Multiagent-based system environment.

charging station equipped with the ability to both charge and discharge the building battery.

The selection of these models is mainly due to the fact that LSTM is an algorithm that uses as input the output of a higher layer or of the same layer, managing to generate certain memory. This network stands out for its excellent performance in numerous works when it is sought to optimize the prediction and where there is a relationship between the current value and one of the most recent values that have passed through the network [48,49]. On the other hand, we will also work with fully connected models since, in spite of not having memory,

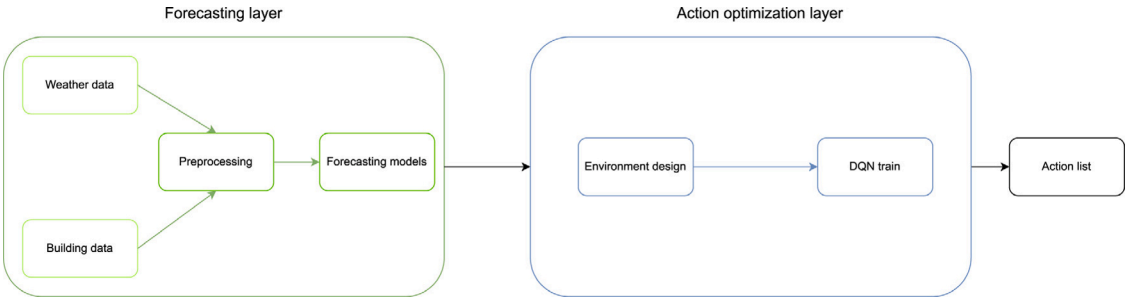


Fig. 4. Platform architecture.

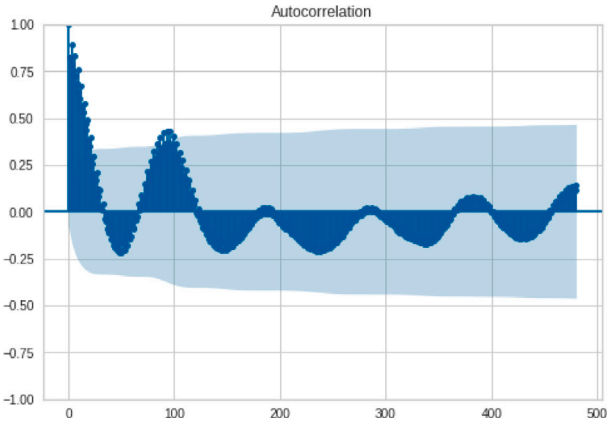


Fig. 5. Autocorrelation of consumption obtained in quarter-hour bands regarding 500 lags.

they obtain great results. All of this will be detailed and exemplified in Section 4, where the platform will be evaluated with a real use case.

3. Platform deployment

The main objective of the system is to obtain a series of actions to be performed in the following quarter-hour slots, thus offering the possibility of maintaining energy flexibility and optimal planning of the system components. Flexibility implies guaranteeing consumption within established thresholds at any given moment, called consumption pattern, for as long as possible. To achieve this, the platform is made up of two main components (Fig. 4).

The first of these is responsible for the prediction of consumption in the following time slots. To this end, it is necessary to monitor the consumption of the variables that are most closely linked to this. After a first evaluation, in our case, it has been determined that the weather, the season, and the current date are the most relevant features, having a higher value in the correlation with the consumption variable. The generated data have been preprocessed by decomposing the input into the current date into year, month, day, hour, minute, second, and day of the week. Additionally, the relevance of previous consumption values in the current data was explored using autocorrelation. The results obtained show that all 384 records (4 days * 24 h * 4 records per hour) are related to this value, especially the previous four days, since they are in quarter-hour bands (Fig. 5).

From this point on, we have studied the deep learning models that perform better in predicting the following values, modifying both the number of layers and the number of neurons per layer or the type of network, recurrent or fully connected. The deployment of the models was performed using the Tensorflow framework. The selected models have been those that stand out within deep learning in the resolution of regression problems, both recurrent and convolutional networks [50].

The results of the different models selected were monitored at all times using the Grafana platform, allowing us to evaluate them in real-time. In this platform, only the value of the next hour was checked since it is likely to have the greatest dispersion within the prediction, although all the predicted values were stored for the measurement of the errors.

For the evaluation of consumption prediction models, both the mean squared error (MSE) and the difference in the absolute value of the predicted and actual data (MAE) have been selected. Both metrics stand out for being very efficient for the optimization of logistic regression and time series models.

The first one shows the variation between the predictions of our model concerning the behavior of actual consumption, while the second is a smoother metric, in the sense that it does not penalize this difference as much as the MSE does. Additionally, both normalized MSE (NMSE) and normalized MAE (NMAE) have been added to size the existing difference in the problem.

$$NMSE = \frac{\frac{1}{n}\sum_i^n (y_i - \hat{y}_i)^2}{y_{max} - y_{min}}, NMAE = \frac{\frac{1}{n}\sum_i^n |y_i - \hat{y}_i|}{y_{max} - y_{min}} \quad (1)$$

Thus obtained will be discussed in the following section.

Once the consumption forecast for the following time slots has been obtained, and a consumption pattern (a percentage that represents the tolerance of the system) has been established; the second component has been defined. This consists of the design of a system based on multi-agents that allows solving the decision-making problem at the time of knowing the actions that will maintain flexibility in that time interval. For this purpose, an environment composed of the actions of each component involved in the system has been developed:

$$A = a_1, a_2, a_3 ... a_n \quad (2)$$

The total of the actions is bounded by the sum of the combinations of “n” over “i”, being equal to this when all the actions can coexist with the others simultaneously, as can be seen below:

$$\sum_i^n C(n,i) \rightarrow C(n,i) = \frac{n!}{i!(n-i)!} \quad (3)$$

The environment definition requires as much knowledge of the system as possible since the more detailed the information is provided, in the form of actions and rewards, the more precise its behavior will be, which can be a limitation in terms of scalability. The next thing to define has been the observations of the environment. Observations are understood as those variables known to the system and influence the decision of the next action, being in our case the consumption of the next time slot. Finally, the different ways of defining the reward have been explored, starting from the difference for the expected consumption, always negative, and obtaining that the problem converges more quickly in the following way:

$$R = \sum_{t=0}^{N-1} r_{t+1} \qquad \begin{cases} c_{real} = c_{exp} & r_{t+1} = 1 \\ c_{real} \neq c_{exp} & r_{t+1} = -abs(c_{real} - c_{exp}) \end{cases} \quad (4)$$

As can be seen, the difference between expected consumption and real consumption got by current action in absolute value has been

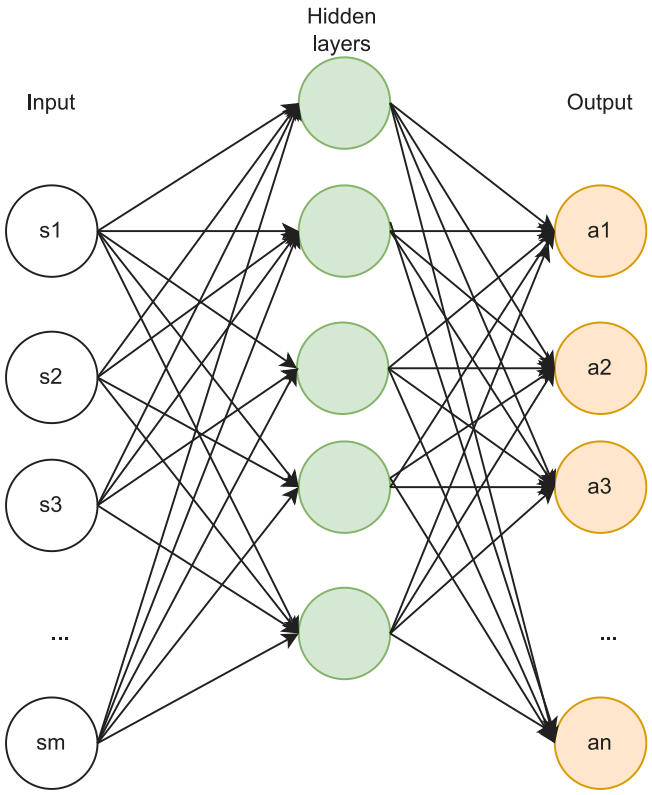


Fig. 6. Deep Q-Network architecture.

taken, unless it coincides, in which case it will be 1. This process will be repeated until the last step of the prediction (N).

Once the complete multi-agent system was defined, different algorithms that could solve this problem were explored. At first, the possibility of using Q-learning, a reinforcement learning algorithm with high performance, was studied. However, due to the results that deep reinforcement learning has been showing recently [51], the decision was made to use DQN . This algorithm was born as an evolution of Q-learning since it solves this problem by combining it with neural networks.

This method of solving multi-agent systems gives weight, indicated by the current state of the neural network training, to each iteration of the system, modifying the initial Bellman equation [52] $Q(s,a) = r + \gamma max_{a'} Q(s',a')$ to the following $Q(s,a;\theta) = r + \gamma max_{a'} Q(s',a';\theta)$. Given that we were using Tensorflow for prediction and that this library has support for multi-agent systems, it has also been used for this second component (TF-agents). The structure of the DQN can be seen in Fig. 6.

This network takes as input the different states through which the system can pass and, through the measuring of weights of the different neurons in the hidden layers of the network, obtains as output one of the "n" actions that the system can perform, mapping on the network the different Action-Q-value combinations, which have the rows of a Q-table. In this way, it will give more or less relevance, using the weights of the network, to the actions to be performed by the system for each of the states it goes through.

Also, it is necessary to mention a key component of the implementation of the DQN model, which is the experience repetition buffer. This component allows storing the most recent experiences and sampling them for use during training.

Finally, by unifying these two approaches, we conclude that, from the data of the previous four days, we can predict the actions to be performed in the next hour, optimally with the following workflow, seen in Fig. 7.

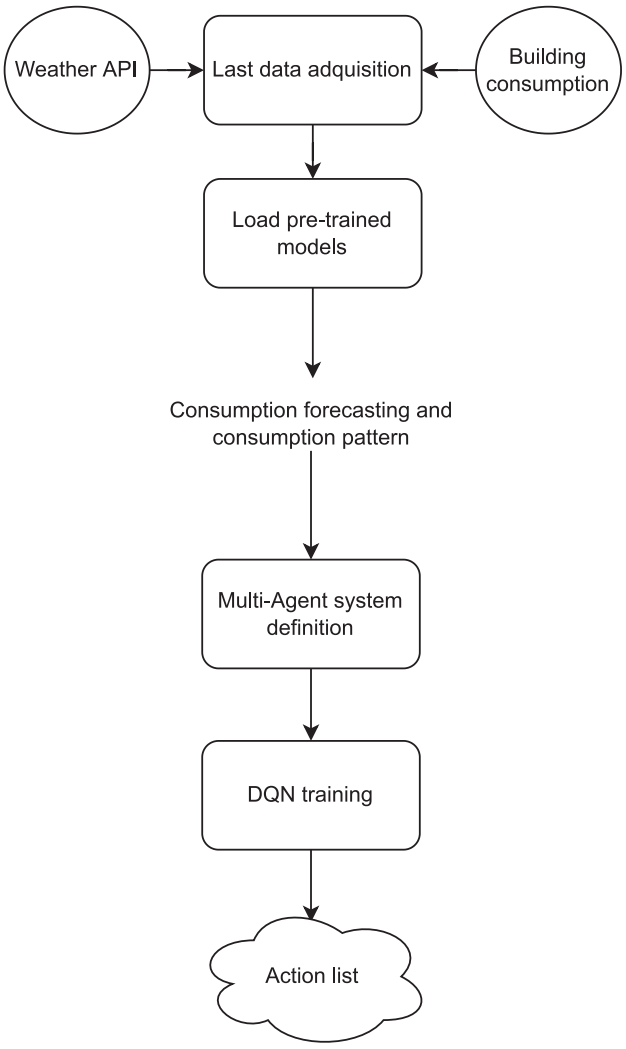
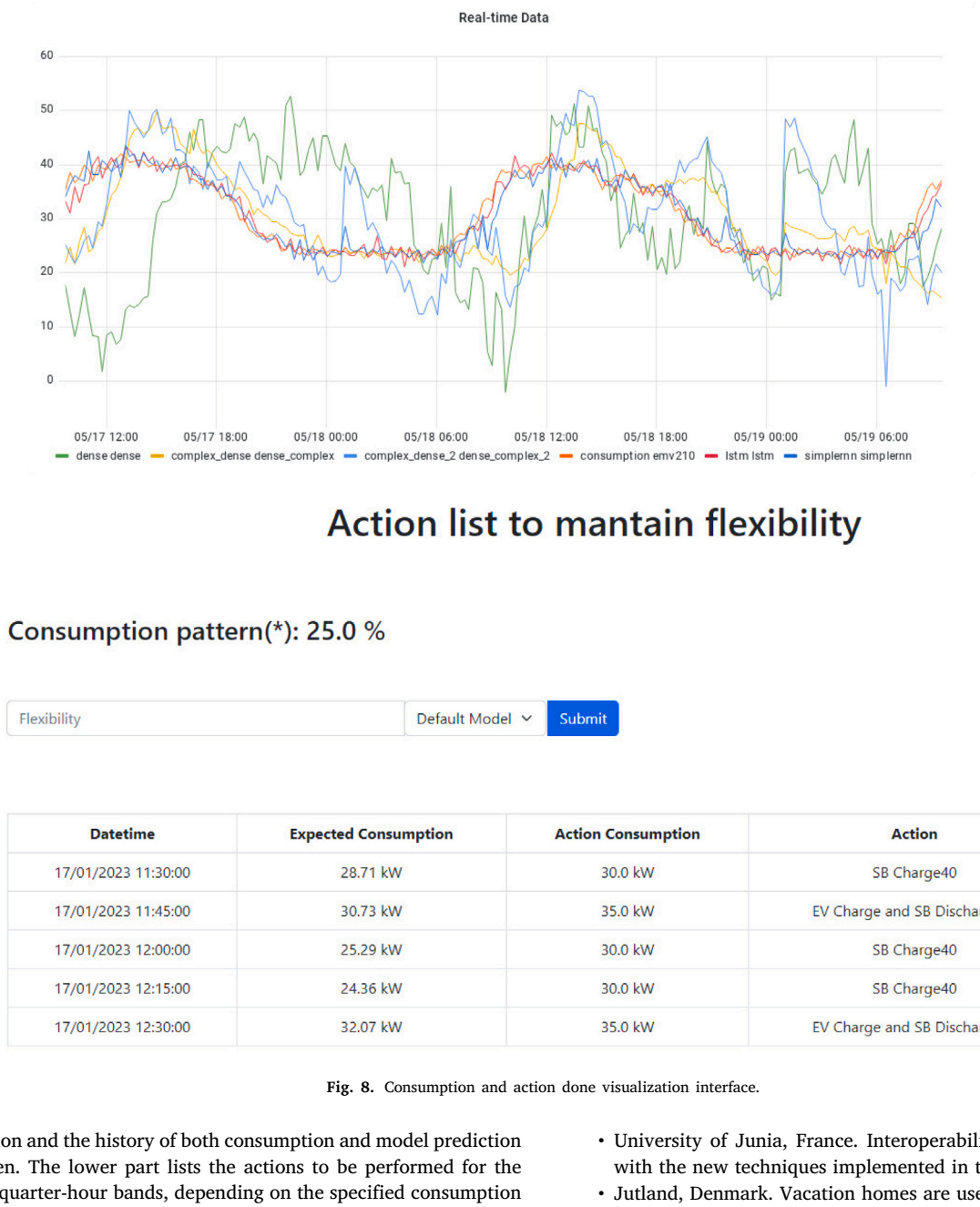


Fig. 7. System workflow.

The first step in the workflow is the collection of the most recent meteorological and consumption data. After that, the pre-trained models are loaded, and a prediction is made, which passes, along with a consumption pattern entered by the user, to the multi-agent system environment. When defining the environment of the multi-agent system, it is necessary to know the actions performed by each component and to detail carefully the result of each one of them. In addition, the reward for each one and the output conditions must be specified. It is necessary to clarify that for each use case, an environment development must be made with the corresponding implementation of the agents involved. From this point on, the environment is defined with these values, and the DQN model is trained. At the end, the set of actions to be performed in the following time slots is obtained. If the model finds a solution, each of the actions found to be optimal for quarter-hour slot will be listed.

If no solution is found, algorithm will indicate all the actions until reaching the time slot where it is not possible, it is understood, then, that a new execution is needed with a more flexible consumption pattern or with more recent data in case the new prediction is easier to satisfy. Failure to achieve an optimal solution is recorded with a large negative reward and measured by the number of steps taken. In this way, we could detect situations in which the model has not found a solution for further evaluation and analysis.

To facilitate data visualization, an interface has been developed (Fig. 8). In the upper part, the most recent records of the building's



Datetime	Expected Consumption	Action Consumption	Action
17/01/2023 11:30:00	28.71 kW	30.0 kW	SB Charge40
17/01/2023 11:45:00	30.73 kW	35.0 kW	EV Charge and SB Discharge60
17/01/2023 12:00:00	25.29 kW	30.0 kW	SB Charge40
17/01/2023 12:15:00	24.36 kW	30.0 kW	SB Charge40
17/01/2023 12:30:00	32.07 kW	35.0 kW	EV Charge and SB Discharge60

Fig. 8. Consumption and action done visualization interface.

consumption and the history of both consumption and model prediction can be seen. The lower part lists the actions to be performed for the following quarter-hour bands, depending on the specified consumption pattern. Also, the interface has two more views, one for the consumption prediction of the next 24 h and another with the history of the model errors in the consumption prediction. Through this interface, users can easily visualize the current prediction and actions to be carried out in the system.

4. Study case

The use case under which the platform developed in this article is studied is the European ebalance-plus project. This project was born as a solution concerning the flexibility within the distributed grid options of the retail market. It aims to offer solutions linked to smart grid technologies to promote a new market. Ebalance-plus has four demosite:

- University of Calabria, Italy. Techniques studied to increase energy flexibility, network automation processes, and failure control recovery technologies.
- University of Junia, France. Interoperability of existing systems with the new techniques implemented in the project.
- Jutland, Denmark. Vacation homes are used for the optimization of network loads and improvement in local distribution.
- University of Malaga, Spain. The university complex is formed by four buildings, including two energy solutions, one at the building level and one at the district level.

In our case, we will concentrate on the latter, which focuses on creating a DC Micro Grid, although the idea is to extend the developments to the rest of the demonstrators. The components of this system are a storage battery, an electric vehicle charging station, and the consumption of the Ada Byron Research building. This building belongs to the University of Malaga and has both private companies and public research groups inside. It has five floors, a usable area of 6492 m^2, an average occupancy of 42 people per day, and an annual consumption of approximately 1,154,264 kWh. Its main energy expenditure consists of the operation of the air conditioning, an agent that has been simulated to evaluate the actions to be taken. Since only the actual consumption of the building and specifications of the operation of the other components are available, models have been

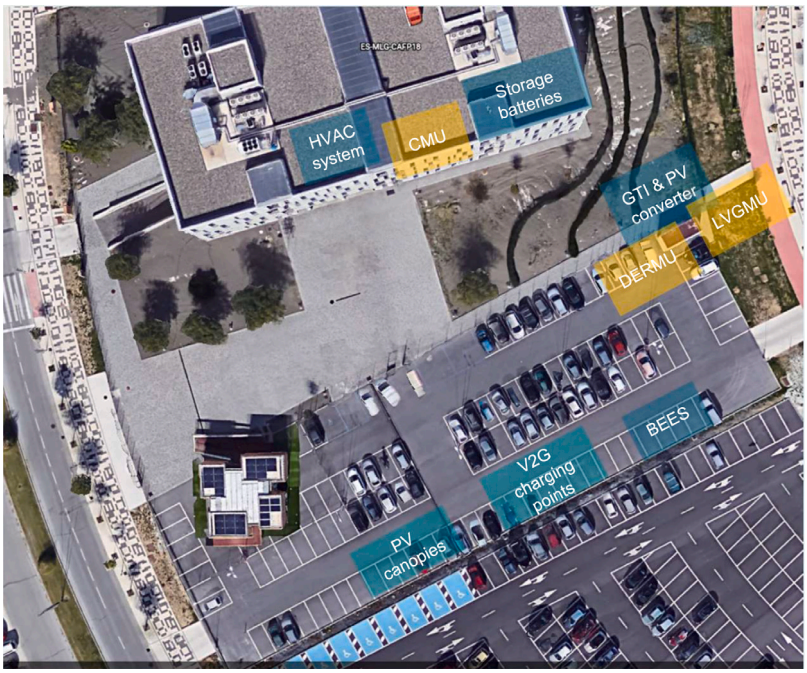


Fig. 9. Ebalance-plus components study case.

developed based on these requirements to simulate their behavior and to explore future scenarios prior to installation (see Fig. 9).

A variety of models have been explored looking for those that could offer better performance, increasing the number of layers, the number of neurons, or the type of network. The models selected for this test were a simple dense model with one layer of 64 neurons, two complex dense models (one complex single-layer model with 1024 neurons and another two-layer model with 512 neurons), a SimpleRNN, and an LSTM. The training performed had varying the numbers of epochs, detecting that overtraining occurred in the dense models from 8000 epochs onwards and in the recurrent models from 3000 epochs onwards. Once the models have been trained with the aforementioned preprocessing, we have the consumption prediction for the following four quarter-hour bands. For the training, we have used equipment provided with two GPUs, one of them with Tesla V100 and the other with RTX 3090.

The next step has been the development of the multi-agent system. The components have been the three mentioned above, leaving the possible actions to be performed as follows:

- Building
  - Perform a small temperature change
  - Perform a large temperature change
  - Turn off the air conditioning (default)
- Battery
  - Charge at 36 kWh
  - Charge at 24 kWh
  - Charge at 12 kWh
  - Discharge at 36 kWh
  - Discharge at 24 kWh
  - Discharge at 12 kWh
  - None (default)
- V2G electric vehicle charging station
  - Charge.
  - Discharge
  - None (default)

The set of actions that can be performed by the system in each iteration totals 51 actions, since some of them cannot be performed simultaneously. The experience repetition buffer has been initialized with a size of 100,000, and the number of iterations and evaluations during training has been modified.

This approach has made it possible to obtain a list of actions of the components involved in the system that maintains the flexibility of an electrical network, supplying and consuming electricity from various sources such as electric vehicle charging stations. It also opens the possibility of including, in the future, additional renewable sources as power generators without the need for a great effort and solving one of the most common problems, namely, the active management of a network while maintaining flexibility during consumption. Our platform allows developers of predictive consumption pattern algorithms to find out how to guarantee flexibility, having only to enter these data and automatically get the set of actions that maximize the quarter-time slots. Moreover, it serves as an API Rest to provide predictions or actions in case you want to develop future implementations based on these data or decisions (Fig. 10). Code is available in our repository with the instructions for its correct execution.[2] Additionally, it has been deployed in Docker containers to scale the application and provide consistency to the system. However, since it deals with private consumption data, it requires credentials to exploit its functionality.[3,4]

In Section 5, we will study and analyze the results obtained for each of the platform components.

5. Results and discussion

Once the selected metrics has been defined to evaluate the performance of the different models, whose structure can be seen in Table 1, we will evaluate the performance of each one of them. The epochs used for the models have varied as recurrent networks reached overfitting earlier than fully connected ones.

[2] Github: https://github.com/ertis-research/MultiAgentSystem.
[3] https://hub.docker.com/repository/docker/fernandogallego/mas-platform
[4] https://hub.docker.com/repository/docker/fernandogallego/mas-training

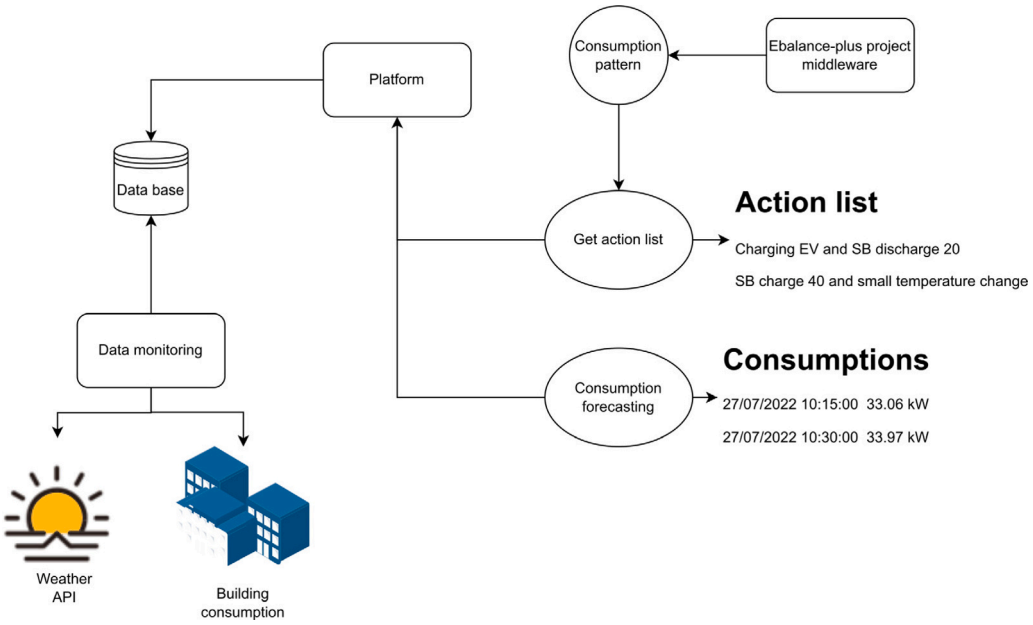


Fig. 10. Platform integration with ebalance-plus middleware.

Table 1
Neural networks architecture used.

Neural network	Hidden layers	Epochs
Basic Dense	64	8000
Complex Dense	512 × 512	8000
Complex Dense 2	1024	8000
SimpleRNN	64 × 64 × 32	3000
LSTM	64 × 64 × 32	3000

Table 2
Values obtained from the metrics for the last 500 entries.

Last registers	MSE	MAE	NMSE	NMAE
Basic Dense	242.79	8.40	108.65	5.10
Complex Dense	57.23	3.70	22.43	2.02
Complex Dense 2	121.63	5.09	50.22	3.05
SimpleRNN	6.44	1.45	1.77	1.36
LSTM	5.90	1.40	1.64	1.13

Table 3
Values obtained from the metrics for the last 150 entries.

Last registers	MSE	MAE	NMSE	NMAE
Basic Dense	102.88	8.01	60.03	5.01
Complex Dense	38.74	4.47	17.78	2.37
Complex Dense 2	76.28	6.96	42.32	4.21
SimpleRNN	3.12	1.25	1.27	0.66
LSTM	2.33	1.18	1.28	0.72

The results obtained for the last 500 records are shown in Table 2. The basic dense network is far from optimal in predicting consumption for the following four quarter-hour bands. On the other hand, the fully connected complex networks are more similar to a real value, although the recurrent models are the closest. SimpleRNN and LSTM achieve high performance, allowing their combination to be used to predict consumption values in the next hour that are virtually identical to what they will be.

By implementing online learning, the performance of the models is getting better and better, as can be seen in Figs. 11 and 12, which represent the error obtained for the last two days of the NMAE and NMSE (see Table 3).

Table 4
Accuracy of multiple DQN training by modifying total steps.

Name	Total steps	Accuracy
Model 1	50	25%
Model 2	100	25%
Model 3	500	100%
Model 4	1000	100%

Table 5
Error of multiple iterations for different consumption patterns by modifying total steps for model 4.

Pattern Cons.	Total steps	Accuracy	Time(s)
7.5%	100	90%	2.62
7.5%	1000	100%	11.50
7.5%	10000	100%	230.15
10%	100	85%	2.57
10%	1000	100%	11.51
10%	10000	100%	228.58
20%	100	95%	2.63
20%	1000	100%	11.53
20%	10000	100%	230.15
30%	100	100%	2.65
30%	1000	90%	11.50
30%	10000	100%	232.71

To quantify the improvement that has occurred with the online training, the performance of the models has been obtained but only for the last 150 records.

In these 350 measurements of the difference, the models managed to reduce the error presented by both dense and recurrent networks by almost half.

These great results allow us to obtain a reliable prediction, for which we have used the average between SimpleRNN and LSTM, the two most accurate models, to anticipate decision-making with the multi-agent system.

On the other hand, once we have a prediction, with a low error as we have seen, we have evaluated the behavior of the multi-agent system when obtaining the actions to be performed by DQN, modifying the steps to be taken and the consumption pattern (Tables 4 and 5). It is worth mentioning that the "accuracy" field represents the percentage of times the model finds a solution that satisfies the requirements.

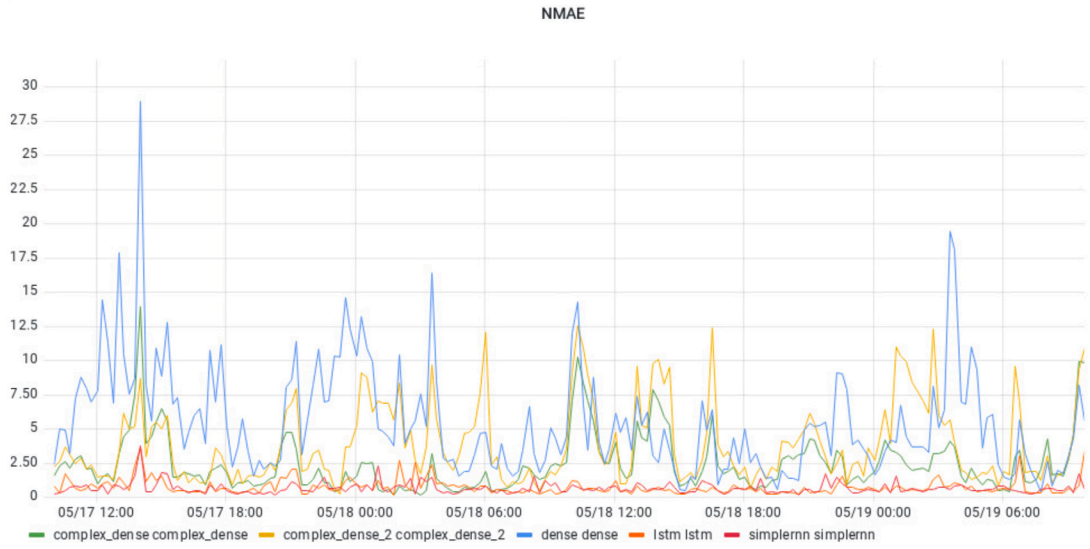


Fig. 11. NMAE obtained for the last 192 records.

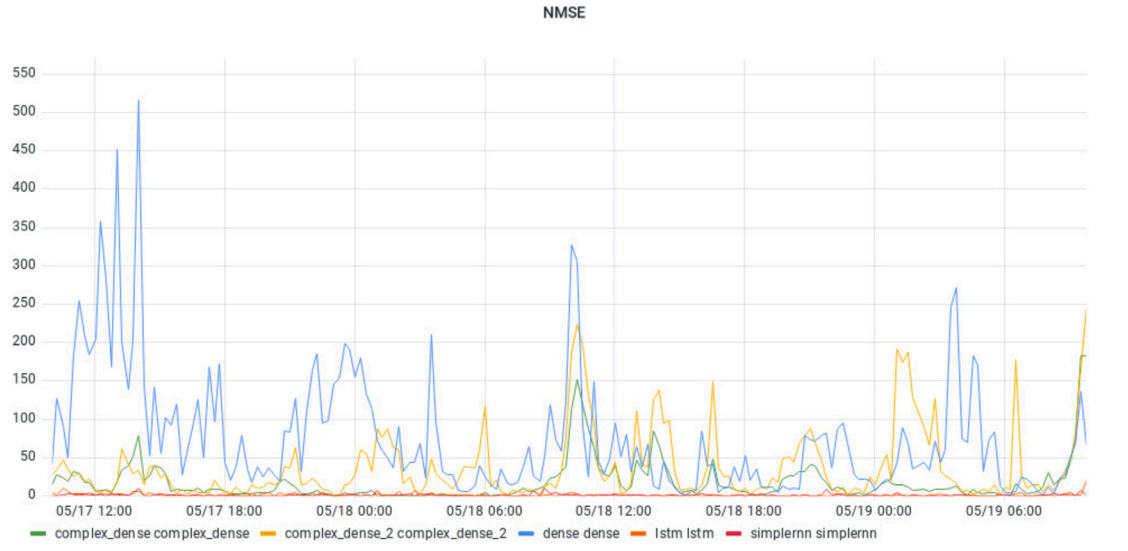


Fig. 12. NMSE obtained for the last 192 records.

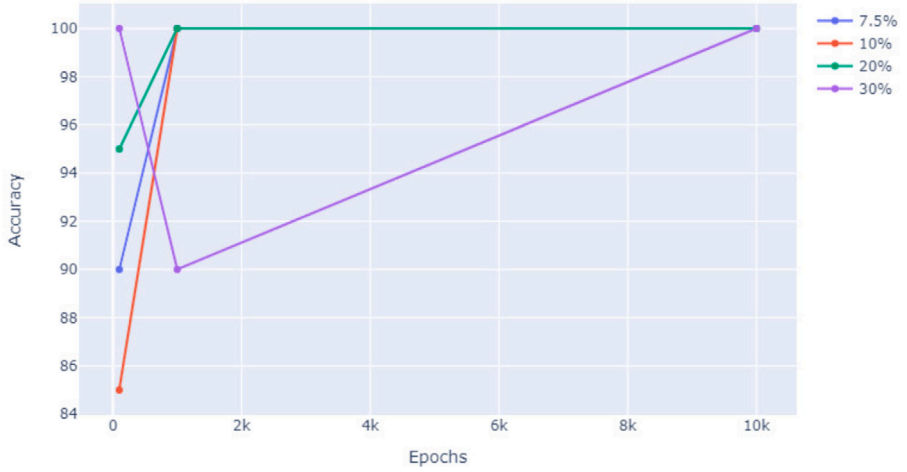


Fig. 13. Accuracy per total steps depending on pattern consumption.

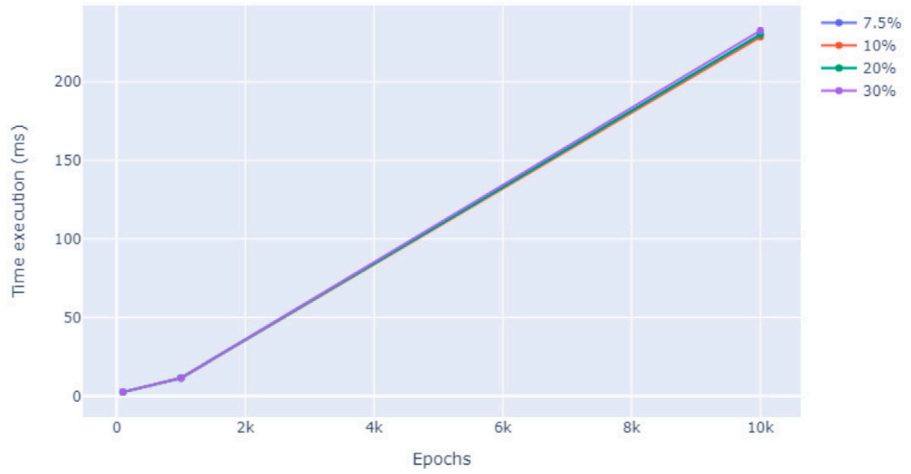


Fig. 14. Training execution time per total steps depending on pattern consumption.

Table 6
Overall performance of model used in platform.

Neural Network	NMAE	DQN Model	Accuracy mean
LSTM	0.72	Model 4	96,66%

The performance of the latter model has been very good, reaching 90% accuracy for a consumption pattern and with only 100 steps, thus reducing its execution time to 2.62 s and having been previously trained only with 1000 steps which is a low value. This accuracy can be increased to 100% but, in turn, will increase its execution time, and knowing that this prediction is designed for the next immediate hour, it will be of no interest if it takes too long (Fig. 13). The peak in the performance of the 20.0% consumption pattern is due to the fact that, in that iteration of the training, it did not obtain the optimal value, but its result is close to the others.

Additionally, it has been observed that by multiplying the total number of steps by ten, the execution time rises to eleven seconds, and if it is multiplied by one hundred, the estimation of the actions takes about four minutes. Also, the consumption pattern does not usually affect the execution time but it does reduce the possibility of the algorithm finding a solution as it is a more stringent requirement (Fig. 14).

The overall performance of the model, including both the best-performing model for prediction (LSTM) and optimization (Model 4), can be seen in Table 6.

6. Conclusion

This paper has focused on combining deep learning with reinforcement learning using multi-agent-based system for the active management of a smart grid. This kind of systems has to deal with multiple distributed data sources where traditional machine learning techniques do not work as expected. A key element in the smart grid is the flexibility that can be provided by users maintaining consumption in specific ranges during a time period. Our system makes predictions of energy consumption and try to select the optimal actions to be performed in next intervals. Using this information, our solution can indicate how much flexibility can be provided to other elements of the system such as energy algorithms. The provided results show the feasibility of this solution, which is being used in a real-time system in the European project ebalance-plus. At this point, our main objective is to continue with improvements on the platform to increase its scalability and flexibility, such as including methods of resolution by reinforcement learning other than the DQN algorithm.

The main limitation of this work is related to the system's non-scalability due to the definition of the environment. Since a precise environment specification is required, the platform will not guarantee good results in systems whose composition is not the components used. To solve this limitation, we would like to develop modifications to the platform that would allow a more dynamic definition of the environment and provide the platform with scalability.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Fernando Gallego Donoso reports financial support was provided by Horizon Europe. Fernando Gallego Donoso reports financial support was provided by Government of Andalusia.

Data availability

Data will be made available on request.

References

[1] Akorede MF, Hizam H, Pouresmaeil E. Distributed energy resources and benefits to the environment. Renew Sustain Energy Rev 2010;14(2):724–34.
[2] Xu G, Yu W, Griffith D, Golmie N, Moulema P. Toward integrating distributed energy resources and storage devices in smart grid. IEEE Internet Things J 2017;4(1):192–204. http://dx.doi.org/10.1109/JIOT.2016.2640563.
[3] Tushar W, Chai B, Yuen C, Smith DB, Wood KL, Yang Z, et al. Three-party energy management with distributed energy resources in smart grid. IEEE Trans Ind Electron 2015;62(4):2487–98. http://dx.doi.org/10.1109/TIE.2014.2341556.
[4] Li W, Luo M, Zhu L, Monti A, Ponci F. A co-simulation method as an enabler for joint analysis and design of MAS-based electrical power protection and communication. Simulation 2013;89(7):790–809.
[5] Siano P. Demand response and smart grids - A survey. Renew Sustain Energy Rev 2014;30:461–78. http://dx.doi.org/10.1016/j.rser.2013.10.022.
[6] Ghorashi Khalil Abadi SA, Habibi SI, Khalili T, Bidram A. A model predictive control strategy for performance improvement of hybrid energy storage systems in DC microgrids. IEEE Access 2022;10:25400–21. http://dx.doi.org/10.1109/ACCESS.2022.3155668.
[7] Mutarraf MU, Guan Y, Terriche Y, Su C-L, Nasir M, Vasquez JC, et al. Adaptive power management of hierarchical controlled hybrid shipboard microgrids. IEEE Access 2022;10:21397–411. http://dx.doi.org/10.1109/ACCESS.2022.3153109.
[8] Sovacool BK, Hirsh RF. Beyond batteries: An examination of the benefits and barriers to plug-in hybrid electric vehicles (PHEVs) and a vehicle-to-grid (V2G) transition. Energy Policy 2009;37(3):1095–103.
[9] Pérez-Olvera J, Green TC, Junyent-Ferré A. Self-learning control for active network management. In: 2021 IEEE madrid PowerTech. 2021, p. 1–6. http://dx.doi.org/10.1109/PowerTech46648.2021.9494928.
[10] Gill S, Kockar I, Ault GW. Dynamic optimal power flow for active distribution networks. IEEE Trans Power Syst 2014;29(1):121–31. http://dx.doi.org/10.1109/TPWRS.2013.2279263.

[11] Gao L, Liu T, Cao T, Hwang Y, Radermacher R. Comparing deep learning models for multi energy vectors prediction on multiple types of building. Appl Energy 2021;301:117486. http://dx.doi.org/10.1016/j.apenergy.2021.117486, URL https://www.sciencedirect.com/science/article/pii/S0306261921008734.

[12] Lin L, Gao L, Kedzierski MA, Hwang Y. A general model for flow boiling heat transfer in microfin tubes based on a new neural network architecture. Energy AI 2022;8:100151. http://dx.doi.org/10.1016/j.egyai.2022.100151, URL https://www.sciencedirect.com/science/article/pii/S266654682200012X.

[13] Zhang L, Gao Y, Zhu H, Tao L. A distributed real-time pricing strategy based on reinforcement learning approach for smart grid. Expert Syst Appl 2022;191:116285.

[14] Zhang Y, Yang Q, An D, Li D, Wu Z. Multistep multiagent reinforcement learning for optimal energy schedule strategy of charging stations in smart grid. IEEE Trans Cybern 2022.

[15] Henry R, Ernst D. Gym-ANM: Open-source software to leverage reinforcement learning for power system management in research and education. Softw Impacts 2021;9:100092.

[16] Zhang L, Gao Y, Zhu H, Tao L. A distributed real-time pricing strategy based on reinforcement learning approach for smart grid. Expert Syst Appl 2022;191:116285. http://dx.doi.org/10.1016/j.eswa.2021.116285.

[17] Li J, Luo Y, Wei S. Long-term electricity consumption forecasting method based on system dynamics under the carbon-neutral target. Energy 2022;244:122572.

[18] Rodríguez F, Galarza A, Vasquez JC, Guerrero JM. Using deep learning and meteorological parameters to forecast the photovoltaic generators intra-hour output power interval for smart grid control. Energy 2022;239:122116. http://dx.doi.org/10.1016/j.energy.2021.122116.

[19] Feng J, Yang J, Li Y, Wang H, Ji H, Yang W, et al. Load forecasting of electric vehicle charging station based on grey theory and neural network. Energy Rep 2021;7:487–92. http://dx.doi.org/10.1016/j.egyr.2021.08.015, 2021 The 4th International Conference on Electrical Engineering and Green Energy.

[20] Wang J, Xu W, Gu Y, Song W, Green TC. Multi-agent reinforcement learning for active voltage control on power distribution networks. 2021, http://dx.doi.org/10.48550/ARXIV.2110.14300, URL https://arxiv.org/abs/2110.14300.

[21] Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, et al. Playing atari with deep reinforcement learning. 2013, arXiv preprint arXiv:1312.5602.

[22] Sineglazov V, Dolgorukov S. Reinforced learning for navigation equipment test table CAD systems integration. In: 2021 IEEE 6th international conference on actual problems of unmanned aerial vehicles development. IEEE; 2021, p. 90–4.

[23] Modares H, Ranatunga I, Lewis FL, Popa DO. Optimized assistive human–robot interaction using reinforcement learning. IEEE Trans Cybern 2015;46(3):655–67.

[24] Jakubowski A, Ziecina P, Miłos P, Galias C, Homoceanu S, Michalewski H. Simulation-based reinforcement learning for real-world autonomous driving. In: 2020 IEEE international conference on robotics and automation. IEEE; 2020, p. 6411–8.

[25] Rizkya I, Syahputri K, Sari R, Siregar I, Utaminingrum J. Autoregressive integrated moving average (ARIMA) model of forecast demand in distribution centre. IOP Conf Ser: Mat Sci Eng 2019;598:012071. http://dx.doi.org/10.1088/1757-899X/598/1/012071.

[26] Singh RK, Rani M, Bhagavathula AS, Sah R, Rodriguez-Morales AJ, Kalita H, et al. Prediction of the COVID-19 pandemic for the top 15 affected countries: advanced autoregressive integrated moving average (ARIMA) model. JMIR Publ Health Surveill 2020;6(2):e19115.

[27] Chen K, Zhou Y, Dai F. A LSTM-based method for stock returns prediction: A case study of China stock market. In: 2015 IEEE international conference on big data (big data). IEEE; 2015, p. 2823–4.

[28] Kong W, Dong ZY, Jia Y, Hill DJ, Xu Y, Zhang Y. Short-term residential load forecasting based on LSTM recurrent neural network. IEEE Trans Smart Grid 2017;10(1):841–51.

[29] Lu R, Hong SH, Zhang X. A dynamic pricing demand response algorithm for smart grid: reinforcement learning approach. Appl Energy 2018;220:220–30.

[30] Henry R, Ernst D. Gym-ANM: Open-source software to leverage reinforcement learning for power system management in research and education. Softw Impacts 2021;9:100092. http://dx.doi.org/10.1016/j.simpa.2021.100092.

[31] Wei Q, Liu D, Shi G. A novel dual iterative Q-learning method for optimal battery management in smart residential environments. IEEE Trans Ind Electron 2014;62(4):2509–18.

[32] Mocanu E, Mocanu DC, Nguyen PH, Liotta A, Webber ME, Gibescu M, et al. On-line building energy optimization using deep reinforcement learning. IEEE Trans Smart Grid 2018;10(4):3698–708.

[33] Yang Q, Wang G, Sadeghi A, Giannakis GB, Sun J. Two-timescale voltage control in distribution grids using deep reinforcement learning. IEEE Trans Smart Grid 2019;11(3):2313–23.

[34] Wei F, Wan Z, He H. Cyber-attack recovery strategy for smart grid based on deep reinforcement learning. IEEE Trans Smart Grid 2019;11(3):2476–86.

[35] An D, Yang Q, Liu W, Zhang Y. Defending against data integrity attacks in smart grid: A deep reinforcement learning-based approach. IEEE Access 2019;7:110835–45.

[36] Qian T, Shao C, Wang X, Shahidehpour M. Deep reinforcement learning for EV charging navigation by coordinating smart grid and intelligent transportation system. IEEE Trans Smart Grid 2020;11(2):1714–23. http://dx.doi.org/10.1109/TSG.2019.2942593.

[37] Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. Science 2018;362(6419):1140–4.

[38] Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev 1958;65(6):386.

[39] Bejnordi BE, Veta M, Van Diest PJ, Van Ginneken B, Karssemeijer N, Litjens G, et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. JAMA 2017;318(22):2199–210.

[40] Bao Y, Tang Z, Li H, Zhang Y. Computer vision and deep learning–based data anomaly detection method for structural health monitoring. Struct Health Monit 2019;18(2):401–21.

[41] Diro AA, Chilamkurti N. Distributed attack detection scheme using deep learning approach for Internet of Things. Future Gener Comput Syst 2018;82:761–8.

[42] Levine S, Pastor P, Krizhevsky A, Ibarz J, Quillen D. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. Int J Robot Res 2018;37(4–5):421–36.

[43] Hafeez G, Alimgeer KS, Khan I. Electric load forecasting based on deep learning and optimized by heuristic algorithm in smart grid. Appl Energy 2020;269:114915.

[44] Iliadis M, Spinoulas L, Katsaggelos AK. Deep fully-connected networks for video compressive sensing. Digit Signal Process 2018;72:9–18.

[45] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention. Springer; 2015, p. 234–41.

[46] Mikolov T, Karafiát M, Burget L, Cernockỳ J, Khudanpur S. Recurrent neural network based language model. In: Interspeech, vol. 2, no. 3. Makuhari; 2010, p. 1045–8.

[47] Connor JT, Martin RD, Atlas LE. Recurrent neural networks and robust time series prediction. IEEE Trans Neural Netw 1994;5(2):240–54.

[48] Li Y, Zhu Z, Kong D, Han H, Zhao Y. EA-LSTM: Evolutionary attention-based LSTM for time series prediction. Knowl-Based Syst 2019;181:104785.

[49] Alazab M, Khan S, Krishnan SSR, Pham Q-V, Reddy MPK, Gadekallu TR. A multidirectional LSTM model for predicting the stability of a smart grid. IEEE Access 2020;8:85454–63. http://dx.doi.org/10.1109/ACCESS.2020.2991067.

[50] Li P, Abdel-Aty M, Yuan J. Real-time crash risk prediction on arterials based on LSTM-CNN. Accid Anal Prev 2020;135:105371. http://dx.doi.org/10.1016/j.aap.2019.105371, URL https://www.sciencedirect.com/science/article/pii/S0001457519311108.

[51] Yang Y, Juntao L, Lingling P. Multi-robot path planning based on a deep reinforcement learning DQN algorithm. CAAI Trans Intell Technol 2020;5(3):177–83.

[52] Bellman R. Dynamic programming. Science 1966;153(3731):34–7.