Enhancing hybrid renewable energy performance through deep Q-learning networks improved by fuzzy reward control

Chahinaze Ameur[1], Sanaa Faquir[2], Ali Yahyaouy[1], Sabri Abdelouahed[1]
[1]LISAC Laboratory, Computer Science Department, Faculty of Sciences DharMehraz, Sidi Mohamed Ben Abdallah University, Fes, Morocco
[2]Faculty of Engineering Sciences, Private University of Fes, Fes, Morocco

Article Info

Article history:

Received Jul 26, 2021
Revised Dec 24, 2021
Accepted Jan 4, 2022

Keywords:

Fuzzy logic controller
Photovoltaic panels
Reinforcement Q-learning
Renewable energy
Wind turbine

ABSTRACT

In a stand-alone system, the use of renewable energies, load changes, and interruptions to transmission lines can cause voltage drops, impacting its reliability. A way to offset a change in the nature of hybrid renewable energy immediately is to utilize energy storage without needing to turn on other plants. Photovoltaic panels, a wind turbine, and a wallbox unit (responsible for providing the vehicle's electrical need) are the components of the proposed system; in addition to being a power source, batteries also serve as a storage unit. Taking advantage of deep learning, particularly convolutional neural networks, and this new system will take advantage of recent advances in machine learning. By employing algorithms for deep Q-learning, the agent learns from the data of the various elements of the system to create the optimal policy for enhancing performance. To increase the learning efficiency, the reward function is implemented using a fuzzy Mamdani system. Our proposed experimental results shows that the new system with fuzzy reward using deep Q-learning networks (DQN) keeps the battery and the wallbox unit optimally charged and less discharged. Moreover confirms the economic advantages of the proposed approach performs better approximate to +25% Moreover, it has dynamic response capabilities and is more efficient over the existing optimization approach using deep learning without fuzzy logic.

Corresponding Author:

Chahinaze Ameur
LISAC Laboratory, Computer Science Department, College of Science DharMehraz, Sidi Mohamed Ben Abdallah University
Fes, Morocco
Email: chahinaze.am@gmail.com

1. INTRODUCTION

In recent years, governments and industries around the globe have recognized the danger of global warming and are actively seeking alternatives to minimize fossil fuel greenhouse gas emissions. In reality, renewable energy represents a promising solution to reduce greenhouse gas emissions, and its use can be quite useful for a range of applications, including off-grid stand-alone systems [1]. In view of the fluctuating nature of renewable energy production, remote sites may benefit from storage devices.

Among the renewable energy resources, both solar and wind power are derived indirectly or not from the weather [2]. Power is generated using hybrid renewable energy systems (HRES) by combining different types of renewable energy. HRES typically uses solar and wind energies alongside storage units such as batteries to store energy excess from input sources [3], [4]. The present article discusses an unbundled system consisting of a power production system, a power consumption system, and a power storage system.

The present article discusses an unbundled system consisting of a power production system, a power consumption system, and a power storage system: photovoltaic (PV) panels and wind turbines belong to the power production group; in the power consumption group are variable electric loads and hybrid electric vehicles; storage of power is limited to the battery bank. An agent-based collaborative energy management system for stand-alone systems is presented. Ultimately, our objective is to balance power between the wallbox units and cover the demand simultaneously in order to increase system reliability. Reinforcement learning enables the agents to learn the optimal policy. A continuous state-action space is dealt with by each agent by implementing deep Q-learning networks methods.

A fuzzy logic approach to energy management was used before [5], [6] a network of neural connections is described in [7] and genetic algorithm in [8]. Fuzzy models demonstrated good estimations of the PV and wind turbine (WT) power output after applying them and as shown in [8], the proposed method was applied successfully to the analysis of a hybrid system supplying power for a telecommunications relay station. An agent-based system was suggested by the author [9] with a central coordinator to respond optimally to emergency power needs. Several studies have identified the benefits of autonomous multiagent systems for managing buying and selling power [10]. In study [11], an agent-based system is presented for generating, storing, and scheduling energy.

Currently, many researchers are working on designing and building fuzzy logic controllers. Based on supervised learning, using training data is the most common method of designing fuzzy logic controllers (FLC). Real-world applications, however, sometimes make it impossible to obtain a piece of training data or extract an expert's knowledge, especially when the cost of doing so is very high. Designing the inference rules is an important part of the FLC process. Experimental data is needed for this part. Developing fuzzy rules' consequent parts is more challenging than developing their antecedent parts. Globally, not all the time it is easy to acquire specialized information as a priori expert knowledge is required to derive these fuzzy rules. Additionally, control based on fuzzy rules strategy that works under some conditions might not work under others.

This is why reinforcement learning is utilized to increase the quality and performance of the system. Algorithm class is known as reinforcement learning, which allows computer systems to learn from experience. Currently, deep Q-learning networks (DQN) and Q-learning are among the most popular methods for reinforcement learning [12]. No matter what the environment looks like, these algorithms useful for learn from experiences by trial and error. In order to implement reinforcement learning over a continuous input/output domain, it is necessary to integrate fuzzy rules into deep Q-learning algorithms.

Multiple engineering applications used the Q-learning algorithm. It is used in [13] to learn how to control a hybrid electric vehicle (HEV) online. Simulated fuel economy results of the combined control strategy are good. In study [14], a game-based fuzzy Q-learning technique was employed to detect and protect wireless sensor networks from intrusions. In contrast to the Markovian game theoretic. According to [15], there is a two step ahead Q-learning algorithm for scheduling the batteries in a wind turbine. On the other hand, [16] suggests a three step ahead Q-learning algorithm for scheduling batteries in a solar energy system. To reduce the power consumption of solar panels from the grid, [17] suggested a multi-agent system utilizing Q-learning, as far as successful defenses are concerned, the proposed model works better.

Scaling is the problem with Q-learning. Whenever we are talking about complex environments, like designing a video game, a great deal of states and actions may exist. This problem becomes complicated when dealing with table state and action. It is here that artificial neural networks come in handy.

Recently, deep learning becomes apparent as a powerful tool in solving complex problems and is rapidly becoming the state of the art in many fields, including speech recognition, natural language processing, and robotics. The proposed model was used to route traffic [18], showing that the results are generally fast paths that avoid frequent traffic stops at red lights. Qu *et al.* [19] proposes a deep reinforcement learning approach that incorporates double Q-learning, and demonstrate that to reduce overestimation observed over several games, as hypothesized, as well as to achieve much better performance overall. A radar signal based on deep Q-learning networks was implemented in [20] and showed that the resultant algorithm not only reduced overestimations as hypothesized, however, it improved performance on multiple games as well. Deep learning benefited from technological advances in processing central unit (PCU) and graphics processing unit (GPU) for fast calculations that are needed during training.

In our previous work [21], using multi-agent technology, an intelligent system for the optimization and management of renewable energy systems has been developed. In addition to providing quality services for energy consumers, this work aims to satisfy the energy demand from solar and wind turbines, optimize battery usage, and enhance the battery-related performance. In some scenarios, simulation results revealed that the charger failed to meet the power demand, and that deep discharges were often observed even at a low charger state. Furthermore, a comparison between a multi agent system and fuzzy logic controller is made in [22], and it was found that a multi agent system has a more efficient implementation and a quicker response time than a FLC. An improved exploration/exploitation strategy is presented for the agent in this study, depending on how often he is in a certain state.

The paper has the following structure: the section 2 explains how reinforcement learning was used in this study. In the section 3 of this study examines a hybrid system of renewable energy. In section 4, we discuss the systems that were studied. In section 5, we discuss the results, and we close by reviewing the findings and providing some perspectives.

2. REINFORCEMENT LEARNING

Figure 1 illustrates how an agent interacts with its environment using perception and action according to the standard reinforcement learning model. The agent receives information about the current environment for each interaction step; the agent determines how to produce an output based on the information received. An agent is informed of the state change by a scalar reinforcement signal. A good agent's behavior should seek to increase the value of reinforcement signal over time. By using algorithmic trial and error, this can be accomplished [23].

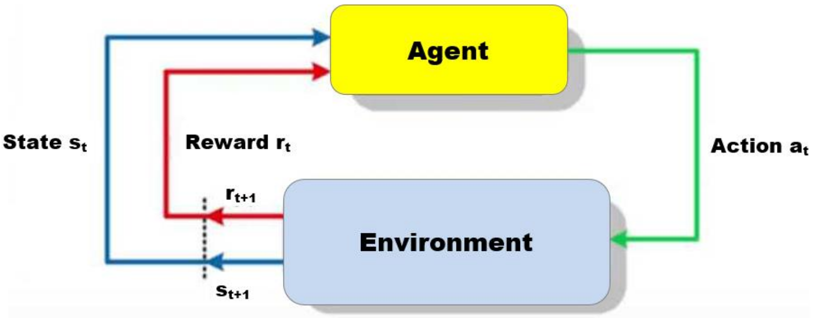

Figure 1. Standard model of reinforcement learning

2.1. Q-learning algorithm

A recent form of reinforcement learning algorithm, Q-learning (Watkins in 1989), Models of its environment are not necessary and can be implemented online. This makes it ideal for repeated games with unknown opponents. Trial and error is the basis of Q-learning. A controller (or an agent) selects an appropriate action (or output) according to its Q-value at each stage. Q-learning applied to identify the optimal policy using delayed rewards, we attempt to determine the optimal value of the action: $Q*(s,a) = max\pi\ Q\pi(s,a)$, i.e. the best value achievable by any policy. This algorithm produces the amount of the Q table in which is stored the quantity of the dual action-state. In the standard online Q-learning process, the Bellman equation describes a one-step value updating, which is just the depth-1 expansion of definition of $Q: Q(s,a) = r + \gamma\ maxa'\ Q(s',a')$. The following algorithm includes the Q-learning process:

```
For each state-action pair (s, a), Initiate the table entry Q(st,at) to zero and observe the current state s.
Do forever:
---Pick an action a and execute it
---Immediately receive a reward r
---Observe the new state s'
---Update the table entry for Q (st, at) as (1):
        $Q(st,at) = Q(st,at) + \alpha.(R\ (st,at) + \gamma\ maxa'Q(st',at') - Q(st,at))$        (1)
---s=s'
```

In this case, $Q(st,at)$ represents dual action-state in the Q table at time step t, α represents learning rate, which affects speed of convergence to final $Q\ (Q\ *)$, a function reward $r\ (st,at)$ for state s and action a at time step t, where γ is the discount factor, the value included in the interval [0,1] that determines the value of a future reward function; and a t' is the action at time step t' [24]. The algorithm involves a number of stages, from determining an initial state to performing, it consists of a series of actions to rewarding when the goal state is reached. A stochastic approximation will lead to the true Q-learning function (i.e., to optimal Q and π).

2.2. Deep Q-learning networks algorithm

With deep Q-learning, deep learning is harnessed through so-called deep Q-networks. In order to calculate the Q value of a network, the neural network is a standard feed-forward model is used. A new Q-value can be generated by using the maximum output of the neural network and experiences from the past stored in a local memory. The illustration compares deep Q-learning and Q-learning in Figure 2.

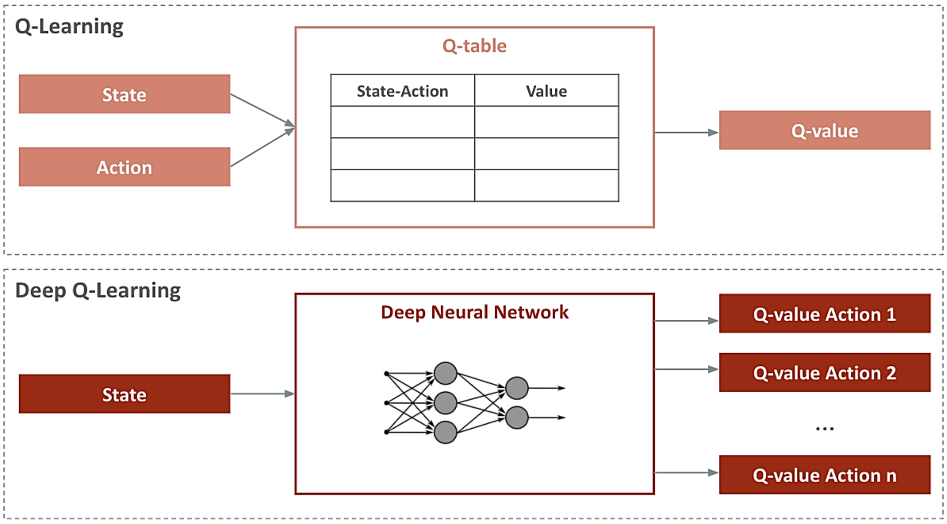

Figure 2. An analysis of the differences between Q-learning and deep Q-learning

One important thing to notice is that deep Q-networks do not rely on traditional supervised learning methods, due to the lack of labeled expected output. Reinforcement learning depends on policy or value functions, which means the target continuously changes with each round of implementation. The agent utilizes two neural networks, for this reason, rather than just one. One network, called Q-network, calculates Q-values in state St, while another network, called target network, calculates them in state $St + 1$. The Q-network retrieves, considering the current state of St, and an action values $Q\ (St, a)$. In order to calculate $Q(St + 1, a)$ for the temporal difference target, next state St+1 is used by the target-network. The N-th iteration of two networks were trained during this program produces measures of the Q-network, the target network's data is copied. Figure 3 illustrates all the steps involved in the process.

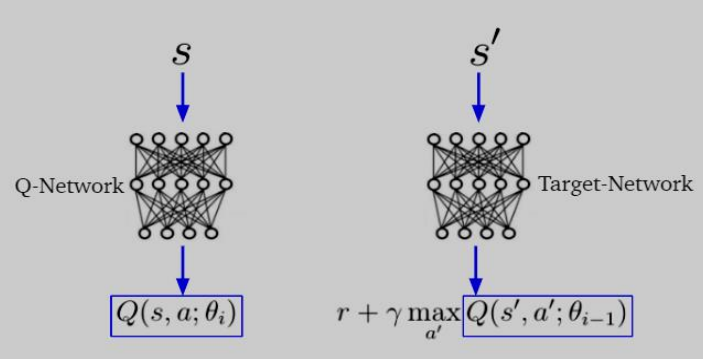

Figure 3. Target and Q-network

The measures θ (i-1) (weights, biases) of the target-network according to the parameter θ (i) of the Q-network at a previous point in time. So, the measures of the target-network are frozen in time. Each iteration updates them with the Q-network parameters. Essentially, an experience replay is just a memory in the form of a series of tuples «a, r s, s’» as shown in Figure 4.

In Figure 4, agents are trained through interaction with their environment and receiving data that is used when learning the Q-network. In order to build a complete picture of transitions, the agent discovers the environment. Initial decisions are made randomly by the agent, and this becomes insufficient over time. The agent examines the environment and the Q-network to decide what to do. Our approach (combination of random behavior and Q-network behavior) is called the epsilon greedy method for the reason that we switch between random and Q policy using the probability hyper parameter epsilon.

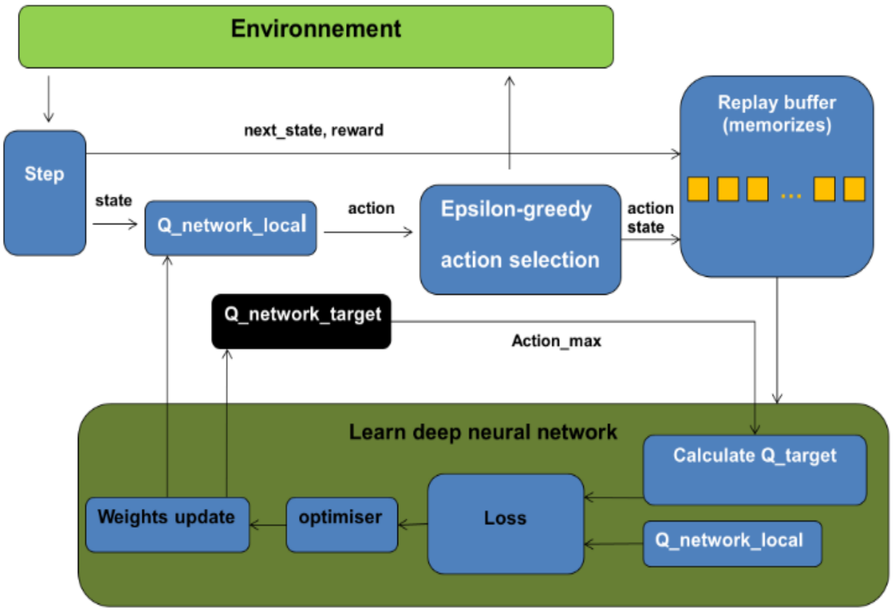

Figure 4. DQN algorithm concept

The predicted value and target value are calculated using two distinct Q-networks (Q_network_local and Q_network_target) during learning. By copying the weights from the actual Q-network, the target network weights are updated after being frozen for several time steps. Stabilizing the training process by freezing the target Q-network for a while and updating its weights with the actual Q-network weights. We find that our training process is more stable when we apply a replay buffer that stores agent experience. After that, random samples are used for training.

For both networks of «a, r s, s'» random batches are used to calculate Q-values from the experience replay, and then backpropagation is performed. Calculating loss as the square of the difference between Q-value target and Q-value predicted. In other words, we want to reduce the distance between the predicted and target Q-value. This distance is expressed by the squared error loss function. By applying gradient descent algorithms, one can minimize this loss function.

$$(\theta_i) = \mathbb{E}_{a\sim\mu}[(y_i - Q(s, \mathrm{a}\,; \theta_i))^2] \tag{2}$$

Where

$$y_i := \mathbb{E}_{a'\sim\mu}[r + \gamma max_{a'} Q(s', a'; \theta_{i-1}) | S_t = s, A_t = a] \tag{3}$$

When Q-network is being trained, this occurs while parameters are transferred to target network later. Q-learning is a multi-stage process that involves several steps:

```
Initiate replay memory D to its maximum capacity
Assign random weights to the action-value function Q
For episode=1; M do
Initial sequence $S_1=\{x_1\}$ and preprocessed sequenced $\varphi_1=\varphi_{(s1)}$
for t=1; T do
Pick a random action $a_t$ using probability ε
```

```
Otherwise select $a_t = max_a Q^*(s_t, a; \theta)$
Execute action $a_t$in emulator and observe reward $r_t$ and image $x_{t+1}$
Set $s_{t+1} = s_t, a_t, \varphi_{t+1}$and preprocess $\varphi_{t+1} = \varphi_{(s_{t+1})}$
Store transition ($\varphi_t$;$a_t$;$r_t$;$\varphi_{t+1}$) in D
Sample random mini batch of transitions ($\varphi_j$;$a_j$;$r_j$;$\varphi_{j+1}$) from D
Set
    $y_j = \{ \begin{matrix} r_j & for\ terminal\ S_{t+1} \\ r_j + \gamma\ max_{a'} Q(S_{t+1}, a'; \theta) & for\ non-terminal\ S_{t+1} \end{matrix}$
Perform a gradient descent step on $(y_j - Q(\varphi_j, a_j; \theta))^2$
End for
End for
```

3. HYBRID SYSTEM ADOPTED FOR RENEWABLE ENERGY

This study examines the use of a stand-alone hybrid PV/Wind/Battery hybrid renewable energy system (HRES) to supply electrical needs of a residential home or apartment on an isolated local. Our system combines the advantages of wind and solar energies to maximize efficiency. During the planning process for the installation of an HRES at an isolated site, it was determined that as well as weather changes, output and input sources required to satisfy the load demand were analyzed: a maximum of 1 kWc of energy per day, 16 PV panels made up of 36 cells need to be combined in series with a generic wind turbine having a 1 kW peak rated power under standard conditions (25 °C and 300 W/m^2 of lighting). The system delivers a maximum power of 2 kW. Additionally, There are several components to the system like a number of batteries that can be used as either sources or consumers, and other consumers include the wallbox unit and the dynamic load [5]. The following Figure 5 shows how the HRES was used in this study:

Figure 5. Energy system hybrid

4. ENVIRONMENT OF DEEP Q-LEARNING AGENT

4.1. State space and action

The agent works in a standalone environment, as described previously. As a way of determining the system's current state, data is collected from the system by the agent: It gathers data about the power demand of the consumers, the power powered by the photovoltaic panels, and the power created by wind turbines, wallbox unit (battery of vehicle) and the secours of the battery. For the agent's states to be defined, the new power (Pnew) between the generated power (Psys) and the power demand (Pload), SOC, and the amount of energy entering the wallbox must be considered.

Actions can be taken on both the battery and the wallbox by the agent. As shown in Figure 6, these actions are charging/discharging the battery and enabling/disabling the wallbox. This enables the agent to decide whether to recharge or discharge the battery as well as whether the wallbox will work. In our study, Q-learning networks are used to find an optimal relationship between system states and actions, as well as exploring the use of actions in different states through an exploration/exploitation strategy: Agents are responsible for managing the stand-alone system's energy so as to fully recharge the batteries and to ensure that the wall box has enough electricity to meet the needs of the unit. Figure 7 depicts a general overview architecture of our DQN approach, which comprises of an input layer that represents state space input, three fully connected inner product layers (hidden layers), and an output layer that represents loss. Based on each possible action, the output is a Q-value.

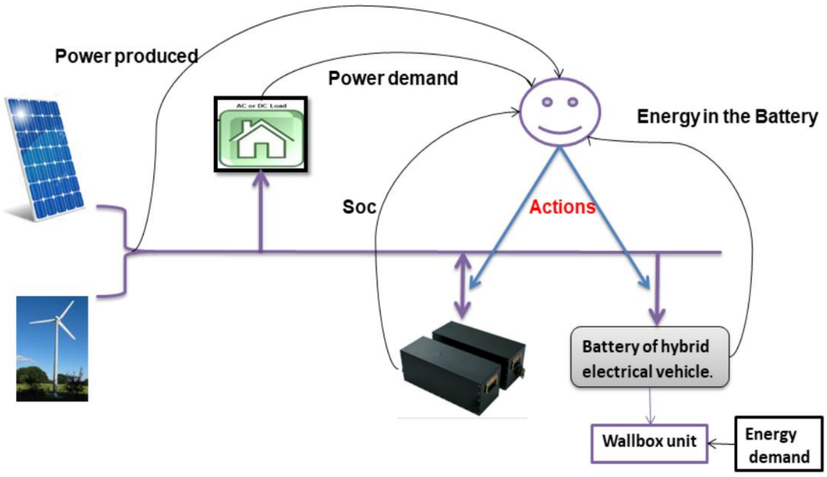

Figure 6. Agent interaction with the environment

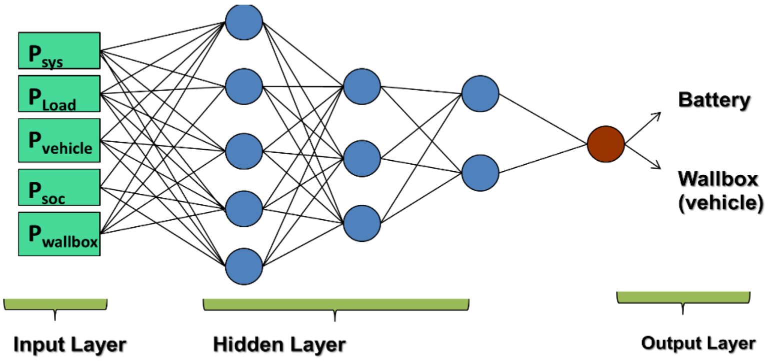

Figure 7. An overview of the neural network used in deep Q-learning

4.2. Fuzzy reward

In reinforcement learning, the reward signal comes from the environment, indicating whether an agent is performing a ‘good’ action. As part of the reward function, all the constraints that should be respected need to be specified, for improved learning efficiency, consider factors affecting decisions as well as interrelationships. An accurate reward function also impacts the computations required, as it results in a more complex function. Furthermore, flexible requirements for optimality are typically included in multi-criteria decision making. It is proposed that a fuzzy reward based on fuzzy system applied to strike the right balance between flexible requirements and implementation complexity [25].

By using fuzzy logic models, complex processes can be modeled in general terms, without using complicated models. According to classical set theory, there are crisp sets and fuzzy sets, while an element may belong to one or both. As defined by a number ranging from [0, 1] according to fuzzy set theory, a fuzzy set element's degree of membership is a function of its participation in the fuzzy set. By using fuzzy if/then rules, logic floue can express relationships between fuzzy variables in linguistic terms. Generally, such rules follow a generic format:

R: if (k1 is Im) and/or (k2 is Im)....and/or (km is Im) then (y is O), where Im There are a number of fuzzy inputs, k=… (k1, k2,…km) is the crisp input vector, y is the output variable and O an expert’s fuzzy set. There are a variety of fuzzy implications implemented by the Mamdani method [26] due to computational efficiency, this method has been used in this article to find the value of total reward. As illustrated in Figure 8, by using four blocks, it is possible to specify the preceding procedure: The input vectors are first transformed into fuzzy values by the fuzzifier. Knowledge base and data base is the second block, where membership functions are defined and fuzzy rules are stored. Finally, a fuzzy inference engine is used to make approximate decisions based on fuzzy rules, followed by a defuzzifier to determine a crisp result. The reward is based on three factors: battery status (SOC), the amount of power flowing through the wallbox unit and the load (power demand).

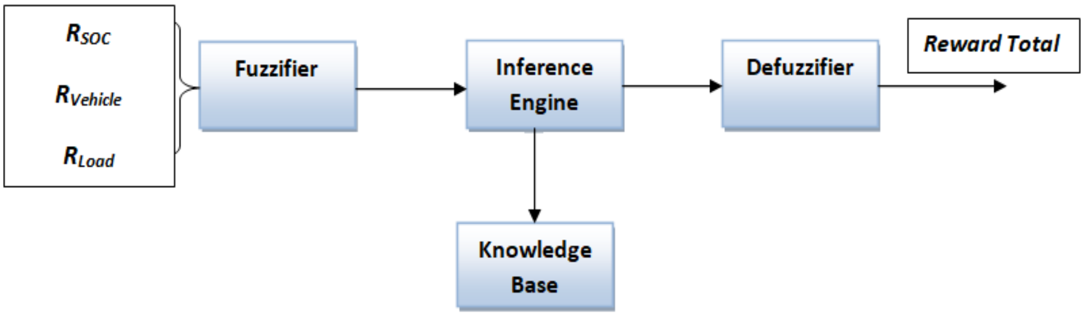

Figure 8. Block diagram of an FLS type Mamdani

In the resulting state, charge state (R_{soc}) of the battery, the vehicle ($R_{Vehicle}$), and the load (R_{load}) must be taken into consideration, the factor of reward was calculated:

$$R_{SOC} = \frac{soc - soc_{min}}{soc_{max}} \tag{4}$$

$$R_{vehicle} = \frac{Vehicle - vehicle_{min}}{Vehicle_{max}} \tag{5}$$

0

$$R_{load} = \frac{Load - Load_{min}}{Load_{max}} \tag{6}$$

0

Scaling is done by normalizing the value $R_{soc,}R_{Vehicle,}R_{load,}$ within the range [-1,1]. As determined by the values of the soc, vehicle, and load at maximum and minimum levels.

A reward is calculated by using the fuzzy system input vector, which is composed of quantities obtained from (4), (5) and (6). Figure 9 uses three membership functions for each input. To cover the range of each input, quantifying the inputs in three areas provides sufficient detail. The P, A, G denote for Poor, average and good. Figure 10 illustrates rules and their results. The negative very big, negative big, negative small, zero, positive small, positive big, and positive very big denote by NVB, NB, NS, Z, PS, PB, and PVB respectively.

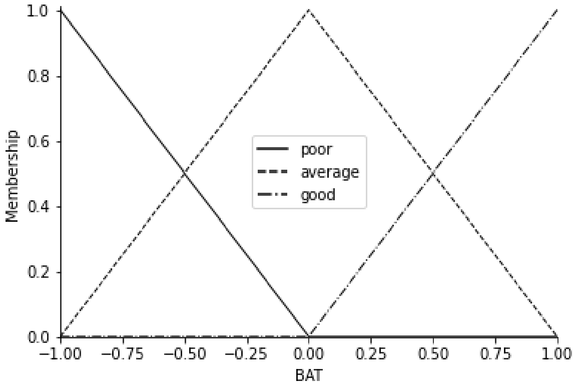

Figure 9. Membership functions in input

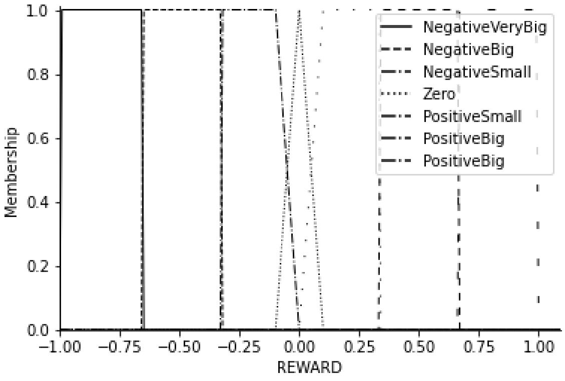

Figure 10. Membership functions in output

This is done to assure that the charge state (SOC) and percentage of energy in the battery of hybrid vehicles are maintained at their maximum levels, while simultaneous covering the hybrid's power consumption of the vehicle. When the battery and percentage of energy archived in the battery of a vehicle are not at their maximum values, it is necessary to serve the power demand by increasing the SOC and percentage of energy located in the battery at the same time. In case this is not possible, covering the energy demand should be the goal.

5. RESULTS AND DISCUSSION

5.1. Workings of the system

As can be seen in Figure 11, when photovoltaic panels (Ppv) and wind turbines (Pw) are added together, they create system power (Psys). Throughout every period of time (t). The formula for the system P_{sys} is:

$$P_{sys}(t)=P_{pv}(t)+P_{w}(t) \tag{7}$$

Batteries should be charged between 25% and 85% to ensure longer battery life. The second parameter is the SOC which is calculated by the formula (8) at each interval of time t (1 hour).

$$SOC=P_{bat}/BC \tag{8}$$

Where $B_{capacity}$ represents the capacity of the battery and P_{bat} represents its power. Maintain the battery charge state (SOC) between $SOC_{minimum}$ and $SOC_{maximum}$. When: SOC_{min}=25% (at its minimum level, the battery cannot be discharged) SOC_{max}=85% (when a battery approaches its maximum capacity, it cannot be charged). Following is the formula for determining the maximum and minimum levels of the batteries:

$$Bat_{minimum}=SOC_{minimum}*B_{capacity} \tag{9}$$

$$Bat_{maximum}=SOC_{maximum}*B_{capacity} \tag{10}$$

Bat_{new}, which indicates the battery level, is the third parameter considered, each time period (t), Bat_{new} is calculated according to the following formula:

$$Bat_{new}=Battery+P_{photovoltaic}+P_{wind}-P_{need} \tag{11}$$

The load needs (P_{need}) and the consumption of hybrid vehicle demand (P_{veh}) for 3 days is represented by Figure 12.

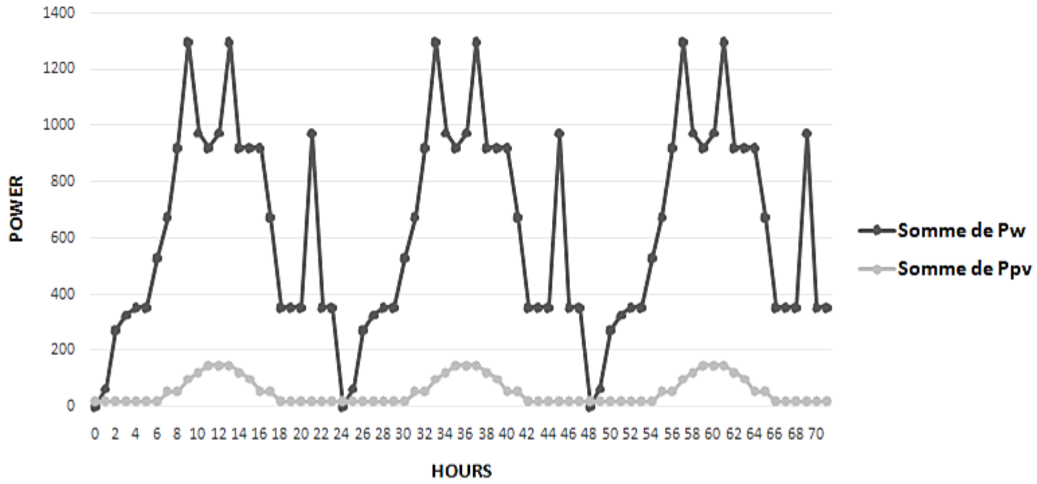

Figure 11. Standard model of Power produced by the PW/PPV source for 3 days

5.2. Results and discussion

Several batteries were tested with their minimum charged levels and some were charged to their maximum levels to determine the effectiveness of the deep Q-learning network: Empty batteries: As a starting point, the battery is at its lowest value (PBattery=$Bat_{minimum}$=1000 W). The chart shows the battery level variation over three days. The Figure 13 present's efficiency indicators regarding our approach using fuzzy reward deep Q-learning. Another approach does not use fuzzy reward signals, so the immediate reward is calculated according to the formula (12).

$$R_{total} = \frac{R_{soc} + R_{vehicle} + R_{load}}{3} \quad (12)$$

Full batteries: When the battery is charged to its maximum, it starts with its maximum value (PBattery=Bat$_{maximum}$=2800 W). The Figure 14 shows the battery level variation over three days:

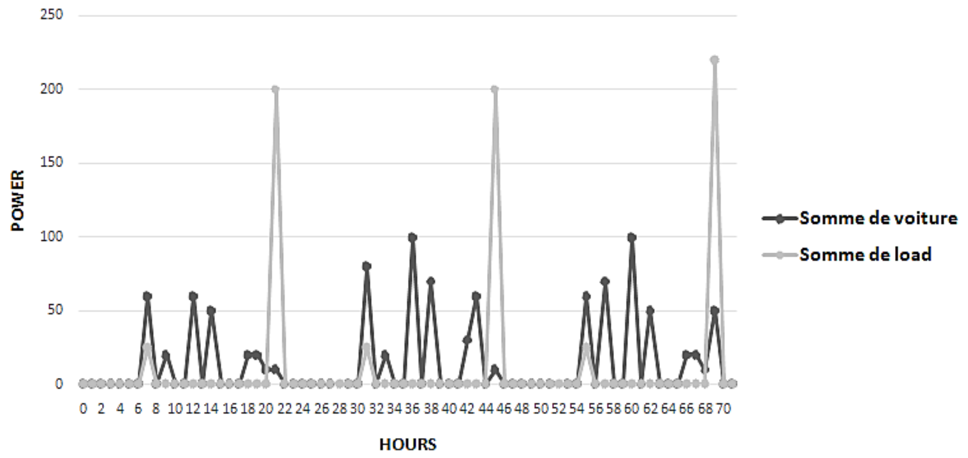

Figure 12. Standard daily load and the consumption of hybrid vehicle demand variation for 3 days

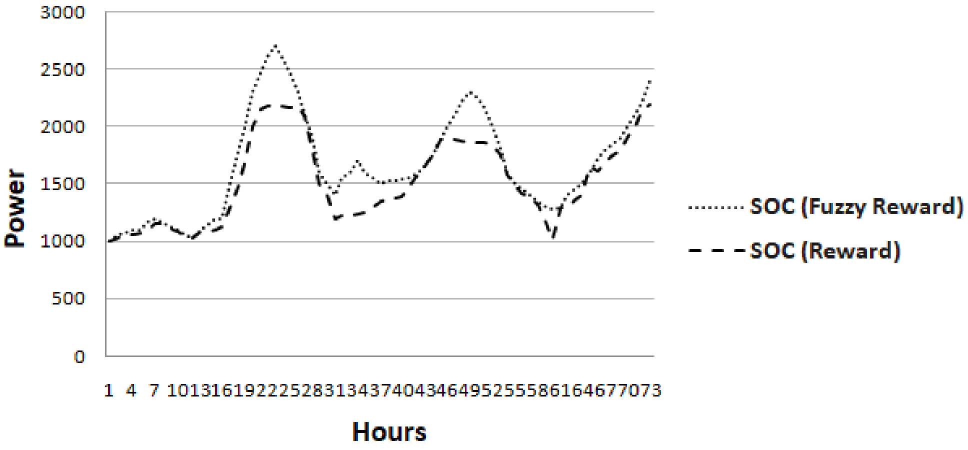

Figure 13. Battery power curve obtained when an empty battery is used to begins a system

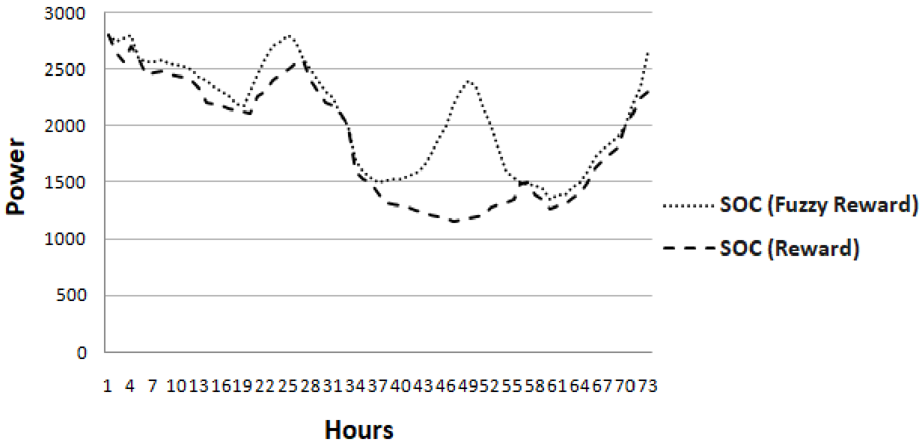

Figure 14. Battery power curve obtained when a full battery is used to begins a system

An agent can perform one action for the battery of the system at any time, while another action can be performed for the electrical vehicle (depending on the state in the wallbox unit). The battery's action can be either charging ('C_B') in case a surplus of energy provided by renewable energy sources and a charge of load is satisfied, or discharging ('D_B) when the energy provided is not enough to satisfy the loads shown in Figure 15. For the action of the electrical vehicle, it will be only charging on demand if there is an excess of energy provided by renewable energy sources, alternatively from the system's battery after satisfying the needs of load. In other words, the agent is able to take any combination of two actions at any time.

Figures 13, 14 and 16 illustrate the way efficiency indicators change over time for each case. Three days are required to run the simulation. At the start of the simulation, the indicators were rising rapidly as more exploration was performed. In using fuzzy rewards, it is apparent that the proposed approach provides a better performance around 25% and provides a more dynamic response and is more efficient than the alternative method, which stabilizes at lower levels and less quickly. This comparison verifies that the system with fuzzy reward using deep Q-learning keeps the battery and the wallbox unit optimally charged and less discharged. The performance of the algorithm has been approved in simulations, a summary of the results confirm the economic advantages of the suggested approach over the existing optimization approach using deep learning without fuzzy logic.

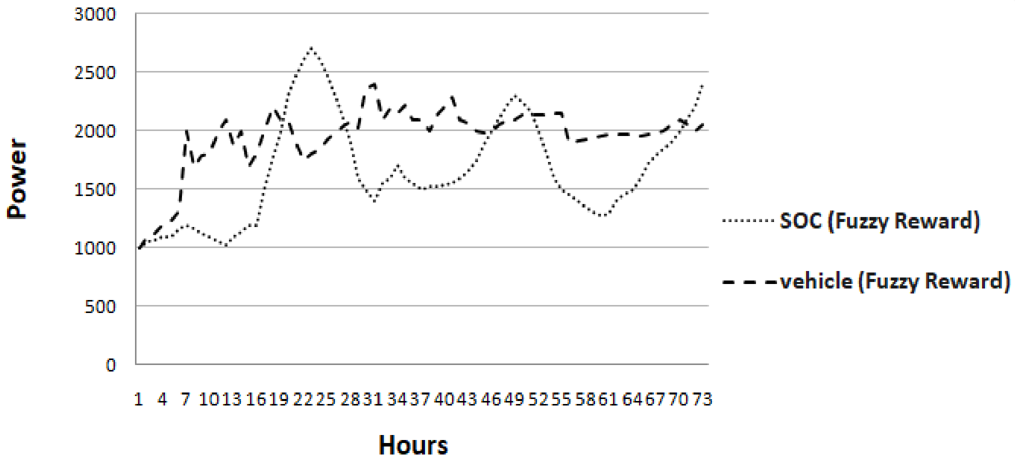

Figure 15. Soc and wallbox unit (vehicle)

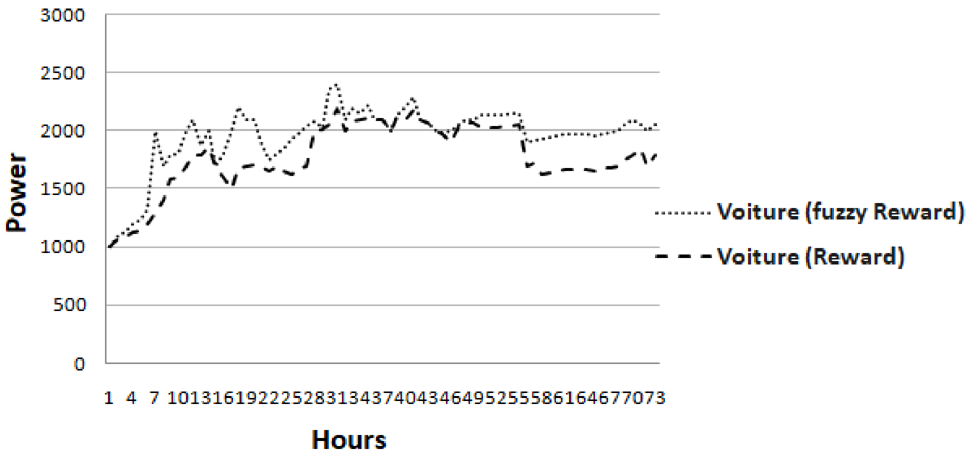

Figure 16. Power consumption of wallbox unit (vehicle)

6. CONCLUSION

Through controlling the energy flow in a standalone system, this paper demonstrates how deep Q-learning networks can provide insights into complex energy management problems. The independent learner's approach is used to minimize the state space and for improving the learning mechanism. To take advantage of state variables, the modified version of this strategy uses state-specific rewards and information that is local to each agent. A combination of exploration and exploitation algorithm enables each agent to learn quickly, converge to a policy very quickly, and demonstrate excellent performance. Future experiments will compare different MAS approaches. In addition, these techniques will be applied in real world settings where a standalone system contains multiple units. Our university has been actively support research on energy management problem solved by machine learning.

REFERENCES

[1] B. K. Koua, P. M. E. Koffi, P. Gbaha, and S. Touré, "Present status and overview of potential of renewable energy in Cote d'Ivoire," *Renewable and Sustainable Energy Reviews*, vol. 41, pp. 907–914, Jan. 2015, doi: 10.1016/j.rser.2014.09.010.

[2] R. E. H. Sims, "Renewable energy: a response to climate change," *Solar Energy*, vol. 76, no. 1–3, pp. 9–17, Jan. 2004, doi: 10.1016/S0038-092X(03)00101-4.

[3] J. Godson, M. Karthick, T. Muthukrishnan, and M. S. Sivagamasundari, "Solar PV-WIND hybird power generation system," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 2, no. 11, pp. 5350–5354, 2013.

[4] M. Asaduz-Zaman, M. H. Rahaman, M. S. Reza, and M. M. Islam, "Coordinated control of interconnected microgrid and energy storage system," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 6, pp. 4781–4789, Dec. 2018, doi: 10.11591/ijece.v8i6.pp4781-4789.

[5] J. Sabor, H. Tairi, A. Yahyaouy, and S. Faquir, "Energy management in a hybrid PV/wind/battery system using a type-1 fuzzy logic computer algorithm," *International Journal of Intelligent Engineering Informatics*, vol. 4, no. 3/4, 2016, doi: 10.1504/IJIEI.2016.10001293.

[6] W.-M. Lin, C.-M. Hong, and C.-H. Chen, "Neural-network-based MPPT control of a stand-alone hybrid power generation system," *IEEE Transactions on Power Electronics*, vol. 26, no. 12, pp. 3571–3581, Dec. 2011, doi: 10.1109/TPEL.2011.2161775.

[7] H. Yang, W. Zhou, L. Lu, and Z. Fang, "Optimal sizing method for stand-alone hybrid solar-wind system with LPSP technology by using genetic algorithm," *Solar Energy*, vol. 82, no. 4, pp. 354–367, Apr. 2008, doi: 10.1016/j.solener.2007.08.005.

[8] I.-Y. Chung, "Distributed intelligent microgrid control using multi-agent systems," *Engineering*, vol. 05, no. 01, pp. 1–6, 2013, doi: 10.4236/eng.2013.51B001.

[9] G. R. Prudhvi Kumar, D. Sattianadan, and K. Vijayakumar, "A survey on power management strategies of hybrid energy systems in microgrid," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 2, pp. 1667–1673, Apr. 2020, doi: 10.11591/ijece.v10i2.pp1667-1673.

[10] Y. S. Foo. Eddy, H. B. Gooi, and S. X. Chen, "Multi-agent system for distributed management of microgrids," *IEEE Transactions on Power Systems*, vol. 30, no. 1, pp. 24–34, Jan. 2015, doi: 10.1109/TPWRS.2014.2322622.

[11] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, Sep. 2015.

[12] Y. Hu, W. Li, H. Xu, and G. Xu, "An online learning control strategy for hybrid electric vehicle based on fuzzy Q-learning," *Energies*, vol. 8, no. 10, pp. 11167–11186, Oct. 2015, doi: 10.3390/en81011167.

[13] S. Shamshirband, A. Patel, N. B. Anuar, M. L. M. Kiah, and A. Abraham, "Cooperative game theoretic approach using fuzzy Q-learning for detecting and preventing intrusions in wireless sensor networks," *Engineering Applications of Artificial Intelligence*, vol. 32, pp. 228–241, Jun. 2014, doi: 10.1016/j.engappai.2014.02.001.

[14] E. Kuznetsova, Y.-F. Li, C. Ruiz, E. Zio, G. Ault, and K. Bell, "Reinforcement learning for microgrid energy management," *Energy*, vol. 59, pp. 133–146, Sep. 2013, doi: 10.1016/j.energy.2013.05.060.

[15] R. Leo, R. S. Milton, and S. Sibi, "Reinforcement learning for optimal energy management of a solar microgrid," in *2014 IEEE Global Humanitarian Technology Conference-South Asia Satellite (GHTC-SAS)*, Sep. 2014, pp. 183–188, doi: 10.1109/GHTC-SAS.2014.6967580.

[16] L. Raju, S. Sankar, and R. S. Milton, "Distributed optimization of solar micro-grid using multi agent reinforcement learning," *Procedia Computer Science*, vol. 46, pp. 231–239, 2015, doi: 10.1016/j.procs.2015.02.016.

[17] Y. Zhou, H. Nejati, T.-T. Do, N.-M. Cheung, and L. Cheah, "Image-based vehicle analysis using deep neural network: A systematic study," in *2016 IEEE International Conference on Digital Signal Processing (DSP)*, Oct. 2016, pp. 276–280, doi: 10.1109/ICDSP.2016.7868561.

[18] D. Silver *et al.*, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, Oct. 2017, doi: 10.1038/nature24270.

[19] Z. Qu, C. Hou, C. Hou, and W. Wang, "Radar signal intra-pulse modulation recognition based on convolutional neural network and deep Q-learning network," *IEEE Access*, vol. 8, pp. 49125–49136, 2020, doi: 10.1109/ACCESS.2020.2980363.

[20] C. Ameur, S. Faquir, and A. Yahyaouy, "Intelligent optimization and management system for renewable energy systems using multi-agent," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 8, no. 4, pp. 352–359, Dec. 2019, doi: 10.11591/ijai.v8.i4.pp352-359.

[21] C. Ameur, S. Faquir, and A. Yahyaouy, "A study of energy reduction strategies in renewable hybrid grid," in *Artificial Intelligence and Industrial Applications*, Springer International Publishing, 2021, pp. 14–25.

[22] C. Ameur, S. Faquir, A. Yahyaouy, and M. A. Sabri, "An approach for revising a fuzzy logic controller using Q-learning algorithm," in *2017 Intelligent Systems and Computer Vision (ISCV)*, Apr. 2017, pp. 1–6, doi: 10.1109/ISACV.2017.8054949.

[23] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An introduction to deep reinforcement learning," *Foundations and Trends® in Machine Learning*, vol. 11, no. 3–4, pp. 219–354, 2018, doi: 10.1561/2200000071.

[24] F. S. Melo, "Convergence of Q-learning: a simple proof," *Institute Of Systems and Robotics, Tech. Rep*, pp. 1–4, 2001.

[25] D. Srinivasan and M. A. Lee, "Survey of hybrid fuzzy neural approaches to electric load forecasting," in *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, vol. 5, pp. 4004–4008, doi: 10.1109/ICSMC.1995.538416.

[26] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1–13, Jan. 1975, doi: 10.1016/S0020-7373(75)80002-2.

BIOGRAPHIES OF AUTHORS

Chahinaze Ameur PhD student received master degree in computer science: business intelligence from Faculty of Sciences DharMehraz, Sidi Mohamed Ben Abdallah University, Fes, Morocco. Her main research interests include electric energy management systems, energy storage, fuzzy control, learning systems, power transmission lines, renewable energy sources, artificial intelligence and deep learning. She can be contacted at email: chahinaze.am@gmail.com.

Sanaa Faquir Professor in Faculty of Engineering Sciences Fes, Morocco, Her main research interests include fuzzy control, secondary cells, air pollution control, data mining, demand side management, electric potential, energy management systems, energy storage, global warming, graph theory, hybrid power systems, learning systems, load flow control, power consumption, power generation control, power generation reliability, power supply quality, power transmission lines, renewable energy sources, road accidents, road safety, road traffic, solar cell arrays, traffic engineering computing, wind turbines, artificial intelligence and deep learning. She can be contacted at email: sanaa.faquir@usmba.ac.ma.

Ali Yahyaouy Professor in Faculty of Sciences DharMehraz, Sidi Mohamed Ben Abdallah University, Fes, Morocco. His main research interests include learning (artificial intelligence), image classification, image segmentation, neural nets, convolutional neural nets, data mining, feature extraction, medical image processing, pattern clustering, traffic engineering computing, Big Data, cancer, fuzzy set theory, history, object detection, object recognition, ontologies (artificial intelligence), recommender systems, road accidents, road safety, road traffic, semantic Web, skin, video signal processing, artificial intelligence and deep learning. He can be contacted at email: ayahyaouy@yahoo.fr

Sabri Abdelouahed Professor in Faculty of Sciences DharMehraz, Sidi Mohamed Ben Abdallah University, Fes, Morocco. His main research interests include image classification, learning (artificial intelligence), image segmentation, traffic engineering computing, cancer, feature extraction, medical image processing, skin, object detection, convolutional neural nets, image color analysis, image motion analysis, image texture, neural nets, object recognition, road traffic, video signal processing, accident prevention, computer vision, convolution, driver information systems, face recognition, fatigue, feature selection, feedforward neural nets, artificial intelligence and deep learning. He can be contacted at email: abdelouahed.sabri@gmail.com.