2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE) | 978-1-7281-9111-9/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICCCEEE49695.2021.9429584

Enhancing Energy Trading Between Different Islanded Microgrids A Reinforcement Learning Algorithm Case Study in Northern Kordofan State

[1st]Moayad ELamin
School of Electrical and Electronic Engineering
University of Khartoum
Khartoum, Sudan
Email: mo2yd99@gamil.com

[1st]Fay Elhassan
School of Electrical and Electronic Engineering
University of Khartoum
Khartoum, Sudan
Email: faymagid4@gamil.com

[2nd]Mahmoud A. Manzoul
Department of Electrical & Computer Engineering
Jackson State University
Jackson, MS 39217
Email: mahmoud.a.manzoul@jmsu.edu

***Abstract*—This paper tackles the problem of rural electrification and the lack of grid connection to large areas of Sudan. It introduces microgrids as an alternative to conventional centralized generation as they provide stability in electricity supply in addition to the environmental benefits accompanied with using renewable energy sources. A new method is introduced to facilitate the fluctuation in energy production when using renewable sources by creating a Reinforcement Learning algorithm to conduct the process of energy trading between different islanded microgrids. The goal of the trading process is to achieve stability and generation-load balance in the microgrids. The paper also presents a case study of three villages in North Kordufan State; Hamza Elsheikh, Tannah and Um-Bader. The study uses real solar irradiance and wind speed data to create a MATLAB simulation for a fully functional microgrid. An RL environment of the grids is created which can be used for future research and modelling in the field of smart grids. The paper explores Vanilla Policy Gradients VPG as a solution algorithm for the problem. The algorithm achieved generation-load stability when applied to data extracted from the MATLAB simulation; satisfying the loads while also achieving profit from the trading process; reducing the return of investment period for the microgrid.**

I. Introduction

Sudan's electricity situation is a challenge that needs to be tackled with excellent efficiency and using innovative solutions. According to governmental sources, the national electricity grid covers less than 40% of Sudan; with more than 60% of it being residential demand. 75% of power generated goes to the capital Khartoum state, which has a 66% connection percentage, while South Darfur and North Kordofan have 2% and 5% connection percentage.

Sudan's centralized system is becoming unusable due to the high maintenance cost and reliability problems and the limitations to connecting new demand areas. Sudan has an astronomical loss rate of 25%. Sudan also suffers when balancing the addition of extra generation to meet the growing demand and spending on adding new areas to the network. Distributed systems are flexible, reliable, economically efficient, and environmentally responsible. They end the need for the high cost of new additions to the network as the distance between generation and consumption areas is small enough to reduce losses.

Microgrids are suitable for Sudan with its high number of villages and lack of major cities. They are a variation of smart grids with small scale grids to work on villages, islands, and small residential areas. They can work as standalone islanded grids or be grid-connected. They use distributed-renewable energy sources to generate electricity locally and then fulfil the local demand and use storage units for night demand and fault cases. Grid Management System is a necessity for microgrids. Reinforcement learning is the technique of choice to control the trading process; it works with sequential decision making and is good with environment interacting systems.

Section 2 will talk about the theoretical background for Microgrids and Reinforcement learning as well as looking at the previous work applied to relevant fields. **Section 3** will see a discussion about the methodology used for the solution, exploring the data used, the design procedure for the Reinforcement Learning environment and the algorithm used. **Section 4** will list and discuss the results achieved from the paper and explains their importance when compared to a system without the proposed solution. **Section 5** will conclude the paper and discuss the future work to be done building on the work proposed.

II. Theoretical Background and Related Work

A. Microgirds

Microgrids are defined as "A group of interconnected loads and distributed energy resources $DERs$ with set electrical boundaries that are used as a single controllable entity concerning the grid that can connect and disconnect itself from the grid based on the mode required."

Fig. 1. A Simple microgrid

The microgrid has a small-scale power supply network for a small community. One of its major advantages is its ability to work alone during utility grid disturbance or outage; meaning that microgrids can operate in two modes:

- ON-grid
- OFF-grid

A microgrid in on-grid mode is connected to the primary utility grid and synchronizes with it. This mode enables bidirectional power flow. If any disturbance happens to the primary grid, the microgrid will switch to the off-grid mode or what is known as a standalone grid (islanded). In the off-grid mode, the microgrid acts as the primary provider to the specified geographical area, working autonomously with high-quality service by acting as local voltage and frequency regulator [1].

Microgrids have a wide range of benefits and noticeably more flexible than a backup generator. The Microgrid's main components include Loads, DERs, master controller, smart switches, protective devices, communication, control, and automation. The microgrid load is of two categories; critical and non-critical (fixed and flexible). Critical load (Fixed) must be satisfied at all conditions and is not altered. In contrast, the non-critical load (flexible) can differ and be adjusted based on the economic incentives or the grid (islanded requirements).

$DERs$ consist of distributed generation units DG, and Energy Storage System ESS installed on the utility or consumer premises. The distributed generation units are either dispatchable or non-dispatchable. Dispatchable units can be controlled by the central controller and are subjected to technical constraints depending on the unit type. Non-dispatchable cannot be controlled by the microgrid controller as its input is changeable, and unrestrained. These types of units include solar and wind, mainly renewable sources. [2]

Electricity demand varies based on the time of day and time of year. While in the traditional power system, electricity can not be stored, which leads to a gap between supply and demand. Microgrid have a mixed power generation which will fill in the mismatch as some generations have significant response times, and others have little flexibility. Provided late reasons, the energy storage system is quite beneficial in managing such a system. ESS synchronizes with DGs as an assurance to microgrid generation capability. Its inclusion within the microgrid system allows the excess energy generated to be stored or, in the typical scenario, put into the utility grid.

The master controller in the microgrid performs the microgrid's dual-mode scheduling based on economic and security considerations. Usually, the master controller is responsible for interaction with the utility grid, switching between two modes.

With that been said, microgrids benefits include improving reliability by introducing self-healing at the local distribution network and managing local loads due to higher power quality. They also provide benefit with carbon emission reduction due to diversification usage in renewable energy sources, economically reducing the Transmission and Distribution (T&D) costs. [2].

B. Reinforcement Learning

Reinforcement Learning RL is the third type of machine learning next to Supervised and Unsupervised Learning. It is learning what to do to maximize a digital reward signal [see 3]. When an agent (i.e., a player in a game) is traversing an environment, he takes actions and collects rewards as he goes.

The whole problem can be described as an environment env that can be described on a state-space S consisting of states s that describe fully the world that can affect or be affected by the agent's decisions. The agent can take action a from an action space A where $a \in A$ that will change its state and receive a reward r where $r \in R$. Any RL problem can be described as a Markov Decision Process MDP [4]. In an MDP, there is the concepts of a reward function $R(s)$, which can map states to rewards achieved when reaching that state.

In an episodic process (one with an actual starting state and a final state), the total reward is the accumulation of each reward received through the journey traversing the environment until reaching the final state. This total reward is the value to be maximized in the RL problem.

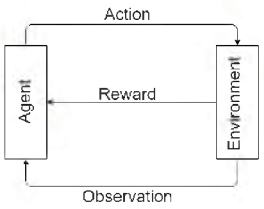

Fig. 2. Reinforcement Learning Parts

Another concept apparent in MDP is a policy π, which defines the path (also known as trajectory τ) that the agent will take during the episode. The policy $\pi(s)$ maps state-action pairs, i.e., what action to take when an agent is in state s. There are two types of policies, either a deterministic policy, where the agent is told precisely what action to make when arriving at state s. The other type of policy is a stochastic policy; here, the agent is told the probability of taking each available action a when in-state s.

This paper will focus on Policy gradients, which are a select type of solution methods that can learn a parametrized policy π_θ that can select an action without returning to a value function[introduced 5]. It can be used to learn policy parameters but not to select exact actions. Updating the policy parameters

can be achieved either using gradient-based or gradient-free methods to maximize expected return. Policy gradients update the policy parameter on each step in the direction of an estimate of the gradient of performance compared to the policy parameter.

C. Related Work

The idea of microgrids replacing conventional power grids in rural areas has been the subject of research. B. M. Sivapriya et al. [6] worked with the problem of microgrid design using the centre of moment approach to the placement of PV panels network, providing case studies for their designs on villages in India. Murenzi et al. [7] worked in Africa, introducing Microgrids as a viable method to electrify sub-Saharan Africa. They showed that in a typical Rwandan village, the installation of a microgrid with PV, batteries, and a micro-hydro is a better financial alternative than extending the national power grid transmission to reach the village.

Applications of RL in smart grids and microgrids vary, a smart building energy management algorithm [8] that uses a Markov decision process to model the smart building. The algorithm controlled included interactions with the utility grid and internal RES. The algorithm used Q-Learning to make decisions on energy dispatch actions achieved better energy costs in the building against multiple pricing policies. Mocanu et al. [9] created a deep belief network that improved the performance of standard reinforcement learning algorithms. They namely worked on SARSA and Q-learning; in the context of predicting energy in a smart building, the algorithm can generalize a learned behaviour model into any other building without any specific history of that building. Leo Raju et al. [10] proposed a model-free reinforcement learning algorithm (Q-learning) to solve the optimal dispatch problem, which concerns finding the best combination of available power resources to provide the required load with minimal cost. Their algorithm converged to the optimal solution and provided adaptability in dynamic situations and unforeseen load management.

Fabrice et al. [11] proposed an algorithm to control power flow between a multi-storage Microgrid mapping fully it as a Multi-Agent System (MAS) and using Multi-Agent Reinforcement Learning to solve the problem. They produced results showing that a centralized control; unit for the microgrid is not needed. The algorithm can achieve the minimal cost of drawing power from the primary grid and achieve most grid independence. Finally, Xiao et al. [12] proposed an energy trading game between different microgrids intending to achieve the Nash Equilibrium without knowing the generation and load demand of the other microgrids using a DQN-based energy trading strategy, achieving an improvement of 22.3% in the utility of the microgrid.

III. Methodology

A. Microgrid

In this paper two control stages are applied; the first stage applies the classical microgrid control strategy on a local level within the microgrid itself (at the primary control), the second stage happens at a more global level where n microgrids will be communicate and the reinforcement learning algorithm will be used at the secondary and tertiary controls. See figure below

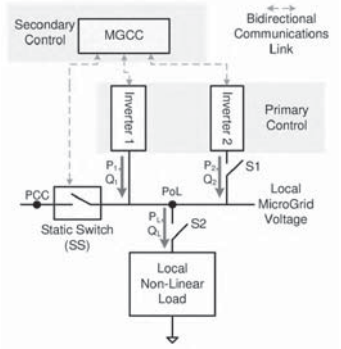

Fig. 3. Control Methodology

The first stage is what is known as the primary control of the system. It works at a local level of the microgrid to regulate the current and voltage and preliminary power-sharing. In contrast, the second stage of this paper's methodology is known in classical microgrids control methods as the secondary and tertiary control levels with the difference that the traditional control theories are not applied. A reinforcement learning algorithm is used to supervise the n microgrids performance from different dimensions such as power quality enhancement, power management,EMS, and economic dispatch.

The Primary controller of the simulation is shown below :

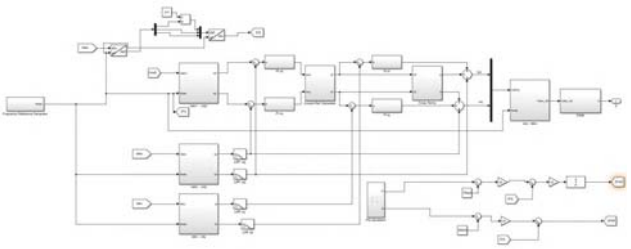

Fig. 4. Inverter Controller

Figure 4 shows the block diagram stages of the primary controller that are Inverter Output Control and Power Sharing Control. The Power Sharing Control, in general, has various theories to be applied; this paper uses the droop control theory for its eligibility with the case study to achieve the balance in the microgrid. The power-sharing part of the controller consists of the PQ calculation, and the droop control block diagram, the inverter output control consists of the outer and inner loop for the current and voltage regulation.

Illustration to the controller methodology:

1) Using Three-phase V-I measurement, take voltage and current readings as abc dimensions.
2) using abc to dq transformation and as part of Phase-Locked Loop PLL

3) The transformation output is then inputted into to Power Sharing Control loop and into the Inverter Control Loop.
4) In Power Sharing Control Loop the readings are inputted for PQ calculation using

$$P = V_d * I_d + V_q * I_q \tag{1}$$

$$Q = V_d * I_q - V_q * I_d \tag{2}$$

5) In Inverter Output Control Loop, use step 2 and step 4 after processing it through a voltage reference generator.
6) After passing the controller's inner loop, a signal is sent to the VSI to stimulate the flow between the microgrid supply and demand.

B. The Data

PVGIS[13] was used to get full time-series of hourly values of both solar radiation and PV performance, which was used to extract PV solar irradiance data.The PVGIS data is the core of the solar part of the generation. No real life data from solar farms or systems was found in rural Sudan areas so an alternative was used. A similar problem was apparent in the wind generation part of the generation.The Wind speeds data was obtained using the windprospecting tool[14]. It provided wind speed readings at different altitudes, for an extended period of time. These parts are installed together in the simulation to extract data for the RL environment.For the load profile, the NERL project[15] provided the Rural African Load Profile Tool which was used to get load profiles for the regions and the loads. No real time or recorded load profile data was found for Sudanese rural grids. The NERL project gives an alternative set of values for rural loads which can be scaled to different villages and rural areas.

C. The Enviroment

The environment consists of three microgrids, the main grid controlled by the agent and two other microgrids for trading, each consists of loads, battery, and generation. The battery is controlled using the supply and charge methods to either supply or be charged from the microgrid itself using loads and generation or do both with another microgrid's battery depending on the trading action. The microgrid generation is the summation of the Wind and solar generations.

A single microgrid loads are the village's houses, schools, mosques, health centres, and water pumps at the simulated village. Each of the grids has a different configuration of these types of loads and are given at hourly intervals. A microgrid's state is its total load, total generation, and remaining capacity in the battery at any given time.

An optimal and worst limit for a microgrid trade are set. For a buyer, the maximum price is the national grid price (network price) while the optimal price is any price less than the microgrid's generation cost. This cost is the kWh price that will give a return of investment at the time that was set. Meanwhile, for a seller microgrid, the minimum price is its generation cost while the optimal price is higher than the network price.

The environment observation is the total load, generation, and current battery capacity of the microgrid as well as the last transaction's price. The action that the agent will take consists of the type of the action (buy, sell, hold), the target microgrid, the amount to trade, and the price. Given the type of action, the action is not discrete as it can take any value in the range specified. Therefore Vanilla Policy Gradients will be used as a continuous action method.

D. Vanilla Policy Gradient

The idea behind policy gradients is to increase the probability of the actions that lead to high reward while decreasing the probabilities of actions with lower reward until an optimal policy is reached. Policy gradients is an on-policy algorithm applicable to both discrete and continuous action spaces. Let π_θ be a policy parameterzed by θ, and $J(\pi_\theta)$ the expected finite-horizon undiscounted return of the policy. The gradient of $J(\pi_\theta)$ is:

$$\nabla_\theta J(\pi_\theta) = E_{\tau\sim\pi_\theta}[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) A^{\pi_\theta}(s_t, a_t)] \tag{3}$$

Where τ is a trajectory and A^{π_θ} is the advantage function for the current policy.

The policy gradient algorithm updates policy parameters using stochastic gradient ascent on policy performance:

$$\theta_{k+1} = \theta_k + \alpha \nabla_\theta J(\pi_{\theta_k}) \tag{4}$$

Policy gradient implementations compute advantage function estimates based on the infinite-horizon discounted return, despite otherwise using the finite-horizon undiscounted policy gradient formula.

VPG uses a stochastic policy on an on-policy method, meaning that it explores using action sampling according to the latest stochastic policy. The amount of randomness for selecting the action is a function of the initial conditions and the training process. As training continues, the policy becomes more deterministic, as it is pushed to exploit the rewards already received, which can cause the policy to be trapped in local optima.

The algorithm for VPG is as follows:

1. Initialize policy parameters θ_0 and value function parameters ϕ_0.
2. for $k = 0, 1, 2...$ do
3. Collect trajectories $\mathcal{D}_k = \tau_i$ by running $\pi_k = \pi(\theta_k)$.
4. Compute reward-to-go $\hat{R}_t$
5. Compute Advantage, $\hat{A}_t$ based on current value function V_{ϕ_k}
6. Estimate the policy gradient as

$$\hat{g}_k = \frac{1}{|\mathcal{D}_k|} \sum_{\tau\in\mathcal{D}_k} \sum_{t=0}^{T} \nabla_\theta \log \phi_\theta(a_t|s_t)|_{\theta_k} \hat{A}_t$$

7. Compute policy update, either using standard gradient policy ascent,

$$\theta_{k+1} = \theta_k + \alpha_k \hat{g}_k$$

or via any other gradient ascent algorithm

8. Fit value function by regression on mean squared error,

$$\phi_{k+1} = argmin_{\phi} \frac{1}{|\mathcal{D}_K T|} \sum_{\tau \in \mathcal{D}_K} \sum_{t=0}^{T} (V_{\phi}(s_t) - \hat{R}_t)^2,$$

Typically via some gradient descent algorithm.

8. end for

E. The reward

Reinforcement learning works on the notion of a reward for its actions; it is the value the policy tries to maximize, and it is the metric used in this paper. Reward design is a significant part ofRL; designing a good reward that pushes the agent in the desired action direction is a design and configuration problem. The reward was optimized to achieve grid stability and to profit from the trading interaction at each point. The reward is a function of total generation G, amount of energy stored at battery B, total load L, network price NP, Microgrid unit generation price UP, trading price$P$, trading amount A, remaining amount after making the action R and the type of action:

When action is to buy

$$reward = \frac{NP - P}{UP} - \frac{R}{A} \tag{5}$$

$$reward = reward - 1, \ \ if \ P < UP \tag{6}$$

When action is to Sell

$$reward = \frac{P - UP}{UP} - \frac{R}{A} \tag{7}$$

$$reward = reward - 1, \ \ if \ P < UP \tag{8}$$

When action is to Hold

Reward changes if load is not satisfied

$$reward = -10, \ \ if A < 0 \tag{9}$$

This reward pushes the agent to stabilize the load and generation, causing no power outages in the network; the reward is set to -10 if, after any action, the load is unstable, that is, the load is more significant than generation and battery.

IV. Results

The system was implemented for the three locations and generated a full year's worth of data for each location. First, figure (5) shows a generation network of Solar battery and load at hamza Elsheikh, which shows the same change in daily generation and provides viability to trading. The y-axis as the rate of energy production within each microgrid for 75 days represented by the x-axis.

A simulation for Wind and battery generation connected with the three locations' load was performed. This shows a 1.5MW generation network with varying generation along a single day "the figure shows a 75 day period" which shows that for each location, there is enough variation in the generation for trading to be viable. Figure (6) shows the y-axis as the rate of energy production within each microgrid for 75 days represented by the x-axis.

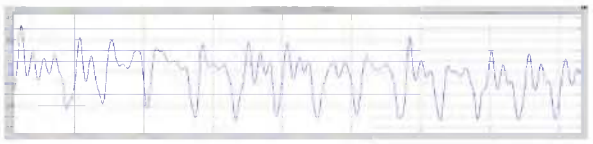

Fig. 5. Solar Generation at Hamza Elshiekh

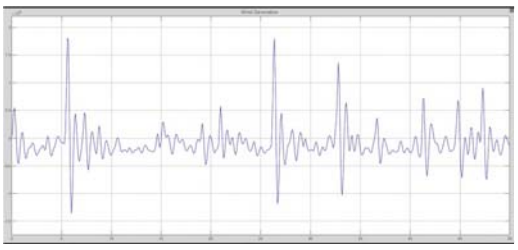

Fig. 6. Wind Generation at Hamza Elshiekh

A fully working microgrid was simulated with all distributed generation connected, i.e., Solar, Wind, and the batteries at Hamza Elsheikh, which was able to sustain the load for the same 75 day period. Throughout the 75 days the microgrid generate energy beyond its need by 100 per cent, which will help to achieve the energy trading goal between the microgrids see fig 7 below:

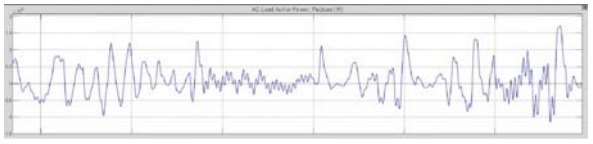

Fig. 7. Fully Functional Microgrid at Hamza Elsheikh

The algorithm was run for 50 epochs, each epochs being 4000 timesteps, which is the equivalent of a year in the data obtained from the MatLab simulation producing 200_000 time steps per experiment. When running VPG for the first run, with matching configuration to the MatLab simulation at Hamza Elsheikh, the following results were achieved:

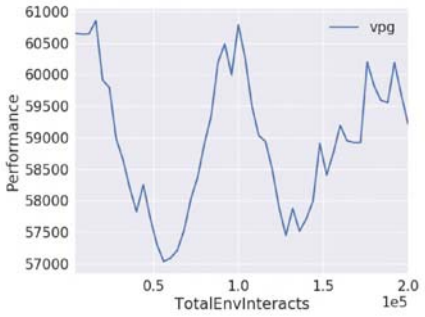

Fig. 8. VPG performance on original design

These simulation results are a clear indicator that at each time step, the grid is achieving profit from trading and

never reaching zero rewards meaning that the grid is always stable. These promising results introduced an opportunity to reduce some of the generations to provide a more challenging environment to the algorithm, so it was decided to remove the Microgrid wind generation and stabilize a solar-based Microgrid. The solar only generation microgrid was a more challenging configuration for the microgrid, but the algorithms could achieve the next results.

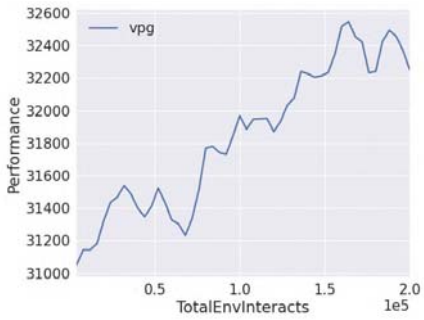

Fig. 9. VPG Performance without wind generation

V. Conclusion

This paper presented a solution to a growing problem in the Sudanese energy situation: the low inclusion of the national grid to the number of potential users and the high cost of adding new consumers to the network in the context of growing demand. It presented advancement to conventional microgrid design and implementation. The paper also described the design process of a microgrid emphasizing on the control techniques and their various levels. The environment designed to give its user the ability to change its load, generation, and the ability to install extra microgrids to the system and easily change the main grid location within the system.

The paper discusses a new methodology for microgrid control by applying a state of the art technique of sequential control to the presented problem. Deep reinforcement learning was introduced and one of the most popular algorithms in the fields of sequential machine learning was implemented in the context of smart grids and energy control. The algorithm showed great potential as it achieved the set threshold of grid stability and trading profit.

VPG's success at the task at hand is an excellent indication of the scalability, and lack of need of context-based optimization for the problem as on-policy algorithms generally tend to be less complicated and less computationally and storage capacity demanding.

VI. Future Work

A simulation that runs both the microgrid and the RL algorithm in real time needs to be created to further check the performance of the model. As shown in the results, the model can achieve great results in a short amount of time. DRL algorithms give the ability to save a model with its parameters and policies to be used with minimal processing; this can be used to transfer the model into low computation real-life usage.

Other algorithms are to be explored to check for better results and lower complexity and computational needs as some of the DRL field is continuously introducing new algorithms. These algorithms include PPO, $DDPG$, $TRPO$ and SAC.

A soft implementation of a small scale solar system can further test the performance of the propossed system and will act as a low cost replacement to implementing the system in a full scale microgrid.

References

[1] Mushtaq N Ahmed et al. "An overview on microgrid control strategies". In: *International Journal of Engineering and Advanced Technology (IJEAT)* 4.5 (2015), pp. 93–98.

[2] Sina Parhizi et al. "State of the art in research on microgrids: A review". In: *Ieee Access* 3 (2015), pp. 890–925.

[3] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[4] Oguzhan Alagoz et al. "Markov decision processes: a tool for sequential decision making under uncertainty". In: *Medical Decision Making* 30.4 (2010), pp. 474–483.

[5] Richard S Sutton et al. "Policy gradient methods for reinforcement learning with function approximation". In: *Advances in neural information processing systems*. 2000, pp. 1057–1063.

[6] Sivapriya Mothilal Bhagavathy and Gobind Pillai. "PV Microgrid Design for Rural Electrification". In: *Designs* 2.3 (2018), p. 33.

[7] Jean Pierre Murenzi and Taha Selim Ustun. "The case for microgrids in electrifying Sub-Saharan Africa". In: *IREC2015 The Sixth International Renewable Energy Congress*. IEEE. 2015, pp. 1–6.

[8] Sunyong Kim and Hyuk Lim. "Reinforcement learning based energy management algorithm for smart energy buildings". In: *Energies* 11.8 (2018), p. 2010.

[9] Elena Mocanu et al. "Unsupervised energy prediction in a Smart Grid context using reinforcement cross-building transfer learning". In: *Energy and Buildings* 116 (2016), pp. 646–655.

[10] Leo Raju et al. "Reinforcement learning in adaptive control of power system generation". In: *Procedia Computer Science* 46 (2015), pp. 202–209.

[11] Fabrice Lauri et al. "Managing power flows in microgrids using multi-agent reinforcement learning". In: *Agent Technologies in Energy Systems (ATES)* (2013).

[12] Liang Xiao et al. "Reinforcement learning-based energy trading for microgrids". In: *arXiv preprint arXiv:1801.06285* (2018).

[13] *Photovoltaic Geographical Information System (PVGIS)*. 2020. URL: https://ec.europa.eu/jrc/en/pvgis (visited on 08/28/2020).

[14] *Sudan Wind Prospecting*. 2020. URL: http://sudan.windprospecting.com (visited on 08/28/2020).

[15] *National Renewable Energy Laboratory*. 2020. URL: https://data.nrel.gov (visited on 08/28/2020).