Learning to Run a Power Network
Energies of the future and carbon neutrality

TER Report
Gaëtan Serré
gaetan.serre@universite-paris-saclay.fr

Abstract

As part of the first year of the Artificial Intelligence master of Paris-Saclay, I had to do a TER (Travail d'Étude et de Recherche). It consists to work one day a week during two months on a research topic in a laboratory. My TER was about the creation of the new edition of the L2RPN (*Learn to Run a Power Network*) [6] series of challenges in partnership with INRIA and RTE (Réseau de Transport de l'Électricité). It took place from March to April. I was officially supervised by Isabelle Guyon and Adrien Pavao, but I worked a lot with Benjamin Donnot, Antoine Marot and Eva Boguslawski who all work at RTE.

RTE is a company that manages the French power network. One of their goals is to prevent a line from being overloaded (we will call it an *overflow*) because it can cause a huge power outage (we will call it a *blackout*). L2RPN is thus a series of challenges which consists in creating the best algorithm (that we will call *agent*) able to manage a simulation of an power network while avoiding overflows and blackouts. The long-term goal is to be able to use such an agent as a decision support tool.

My role in this TER was to generate the data for this year's edition and to design a simple agent in order to prepare myself to create a more complex one that will serve as a baseline for the competition during my internship on the same topic, with the same team.

Contents

1 Background

1.1 Context

Power Systems are huge structures that transport energy from where it is produced (nuclear, fossil, wind, solar, hydro-electric power plants, etc...) to where it is consumed (houses, factories, etc...). These structures are extremely complex, they often rely on thousands of kilometers of power lines and are vital to our society. We need people and tools to maintain and control them. Recently, Deep Learning techniques and more particularly Deep Reinforcement Learning have attracted the attention of experts thanks to their ability to learn power network representations but also thanks to their their parallelizable architecture. It is in this context that the L2RPN series of challenges was created, so that participants can create reinforcement learning agents capable of managing a power network simulation, while avoiding blackouts. This year's edition, and consequently my TER, is part of a energy transition and ecological context because the power network will be inspired by optimistic predictions for 2050, i.e, with as few CO_2 emitting power plants (which we will call *thermal*) possible and as much as renewable power plants as possible.

1.2 2022 challenge stakes

This challenge is the fourth edition of L2RPN. The great novelty of this year is that the power network will be made up of many renewable power plants and as few thermal power plants as possible in order to project ourselves towards an ideal scenario of 2050. In previous years, the energy mix was more actual, with almost one fifth of the energy produced by thermal power plants. The agent's goal remains the same, to avoid blackouts. The stakes for this year is to see if it is possible to control a such network and to highlight future issues.
The massive use of renewable energies raises a new problem. Indeed, for the network to operate, electricity production and consumptions must be equal at all times. However, renewable energies are subject to weather changes which makes it impossible to have a stable energy production. Thus, to overcome this problem, we have added batteries in the network in order to be able to store electricity when, for example, the renewable power plants produce too much energy compared to consumptions and to be able to deliver the previously stored electricity when the weather does not allow the renewable power plants to produce sufficiently.
To create and participate in a L2RPN challenge, RTE has created a set of tools that constitutes a pipeline. An environment (Grid2Op [2]) will use data representing the power network as well as the production and consumptions injections (previously generated by another RTE tool) in order to create the electricity flows on each line of the network at each time step. An agent will then interact with the environment to decide on the action to be taken at each time step. Finally, a tool is used to visualize the agent's performance. This whole process is illustrated by the Figure 1.

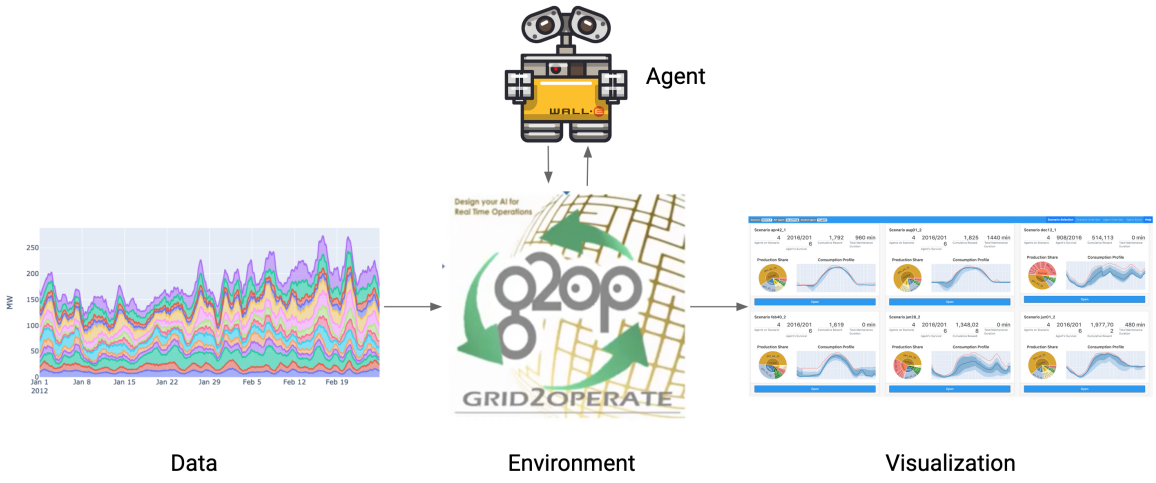


Figure 1: L2RPN pipeline

The goal of my TER was to simulate what the injections could be in 2050, using an energy mix approaching carbon neutrality. In addition, I made preliminary explorations to implement agents solving the problem.

2 Data generation

As stated in the introduction, for this year's edition, we wanted data that have an energy mix similar to the Figure 2 that is, which that do not include thermal power plants.

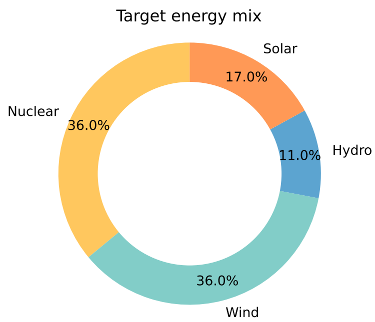


Figure 2: Target energy mix. Borrowed from [4]

As illustrated in Figure 3, the data themselves are electricity injections represented by time series that describe for each power plant how much electricity it produces at each time step.

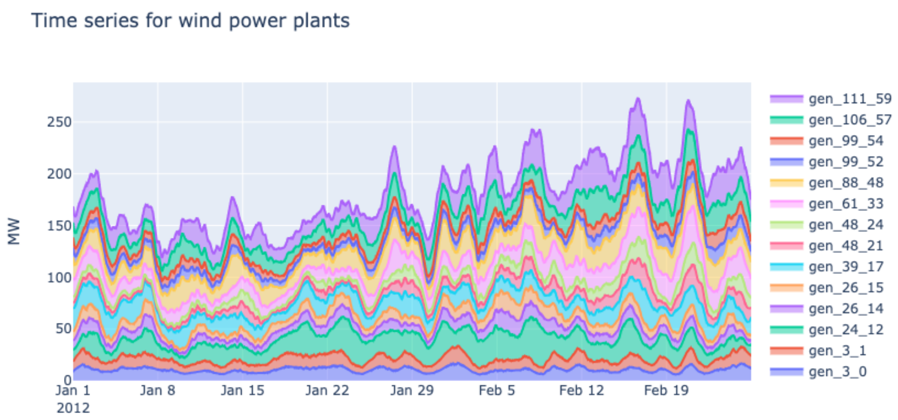


Figure 3: Time series representing the energy produced by each wind power plant at each time step

To generate these time series, we need data concerning the architecture of the power network, the weather, the consumptions and the power plants (their types and maximum production). These data, specially about the consumptions and the weather come from RTE studies. Then, a library created by RTE named *Chronix2grid* [1], will use these data to generate the time series and from these, we deduce the energy mix (simply by summing the electricity produced by each type of power plant). This whole process is detailed in Figure 4. Chronix2grid tries to find the perfect amount of electricity to inject in the grid for each power plant to match the consumptions at each time step. Chronix2grid can be seen as a solver of an optimization problem with many constraints where the objective is to produce the same amount of electricity as consumed at each time step. Some configurations of our initial data can therefore lead Chronix2grid to not find a solution.

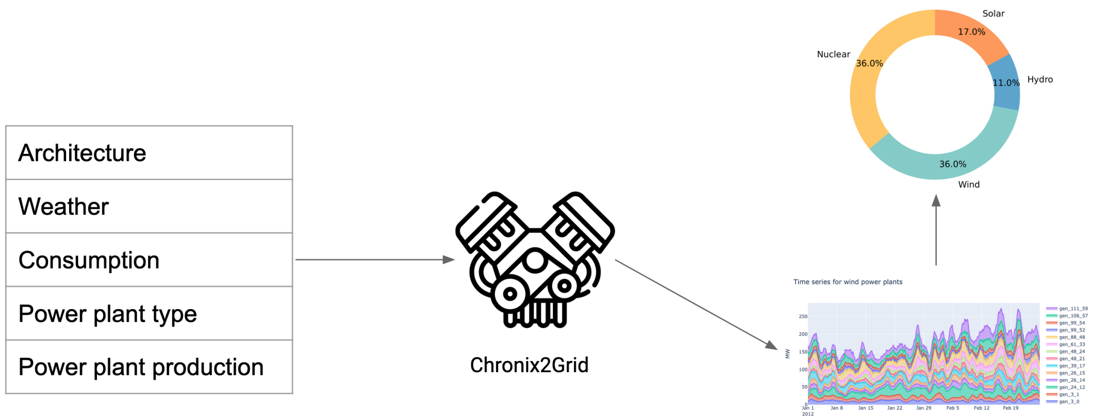


Figure 4: Chronix2grid pipeline

2.1 Problem state

To generate the data for this year's edition, I started with the initial data of the 2020 edition. We assumed that the weather and consumptions will not change between now and 2050. Also, we will use almost the same power network as the 2020 and 2021 editions. As shown in Figure 5, it consists of 62 power plants and 91 consumptions. The only difference with the previous editions is that we have added 7 batteries.

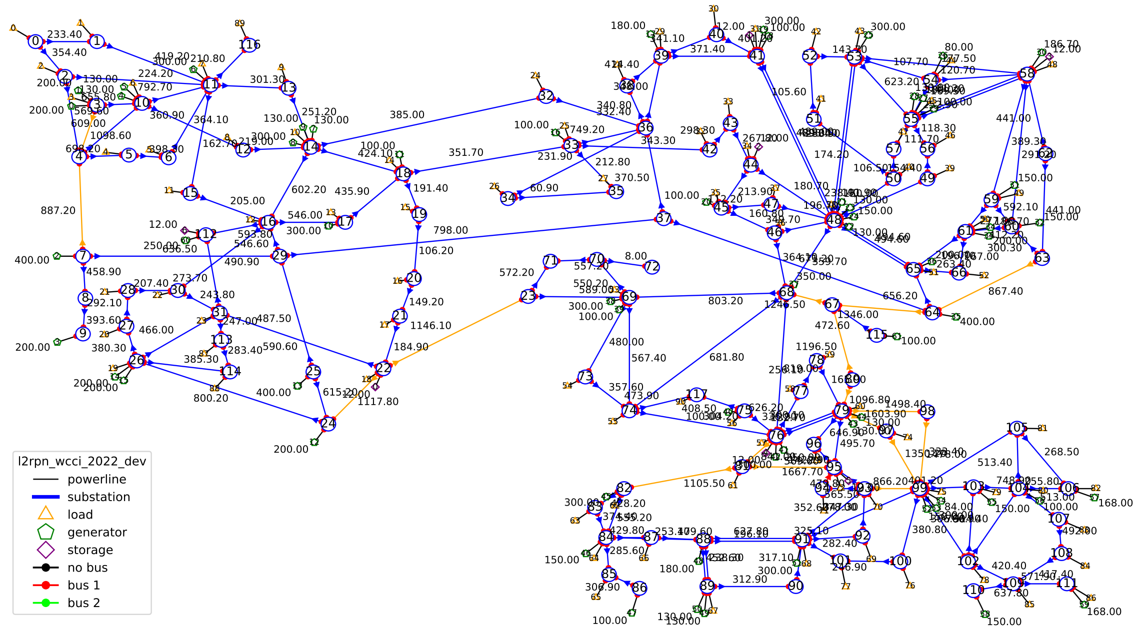


Figure 5: 2022 power network

The goal is therefore to modify the 2020 data that initially produce the energy mix in Figure 6 to have an energy mix similar to the one in Figure 2, that is, with as little carbon emissions as possible and as much renewable energy as possible.

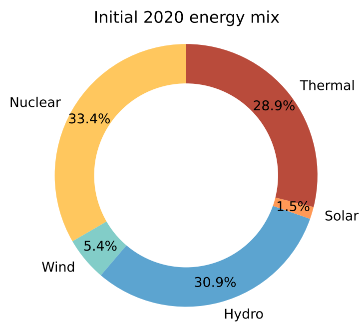


Figure 6: 2020 energy mix

Ideally, I would like to replace all thermal power plants with renewable ones. However, this would prevent Chronix2grid from converging because, at certain time steps, all the power plants would produce too much or not enough electricity. So I created two more complex methods that will each change one of the following parameters: the type of power plants or their maximum electricity production.

2.2 Method 1: optimization

My first intuition was to change the type of power plants in order to obtain an energy mix as close as possible to the one of the Figure 2. So I created an optimization problem which consists in finding the configuration where the difference between its energy mix and the target energy mix is minimal. This problem can be roughly written as follows (detailed version 2):

$$min_{config} \sum_{t \in \{\text{nuclear, wind, thermal, solar, hydro}\}} [em_t(TARGET) - em_t(config)]^2 \tag{1}$$

Where $em_t(config)$ is the percentage of electricity produced by type t in configuration $config$ and $em_t(TARGET)$ is the percentage of electricity produced by type t in the target energy mix (Figure 2).

I then implemented the simulated annealing algorithm [5] to solve this problem and after several million iterations, I obtained the results shown in Figure 7. However, to compute $em_t(config)$, I used many approximations because it would have been too long to use Chronix2grid. Therefore, the optimal configuration returned by my solver does not allow Chronix2grid to generate time series. For example, the weather data not being taken into account in the solver, 48 of the 62 power plants of the optimal configuration were wind power plants. Therefore, the remaining 14 power plants are not able to generate enough electricity when there is no wind.
After discussion with RTE experts, we concluded that this method was too brutal and that we needed to find a gentler way of reducing the share of thermal power plants.

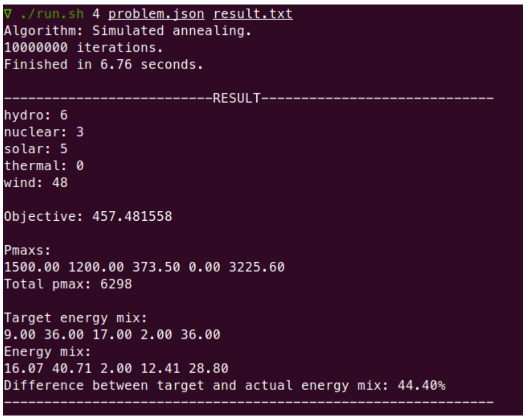


Figure 7: Result of the solver

2.3 Method 2: Incremental increase of renewable production

The second method was to increase the maximum production of renewable power plants in order to reduce the share of electricity produced by thermal power plants. Indeed, in Chronix2grid, the use of the latter is penalized. Thus, Chronix2grid uses these power plants only when there is no alternative and therefore increasing the production of renewable power plants indirectly reduces the share of thermal power plants in the energy mix. Moreover, it is a soft method which leaves Chronix2grid a lot of freedom and does not change the number of power plants of each type. Also, we can justify this choice by assuming that in 2050, the renewable power plants will have a better efficiency than today. I have then multiplied the maximum production of all the wind and solar power plants by 1.5 as shown in Figure 8.

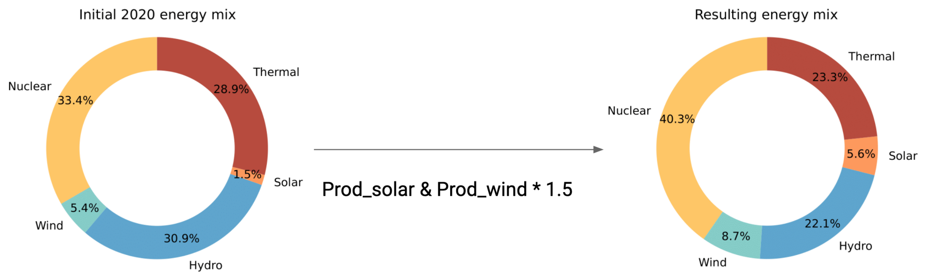
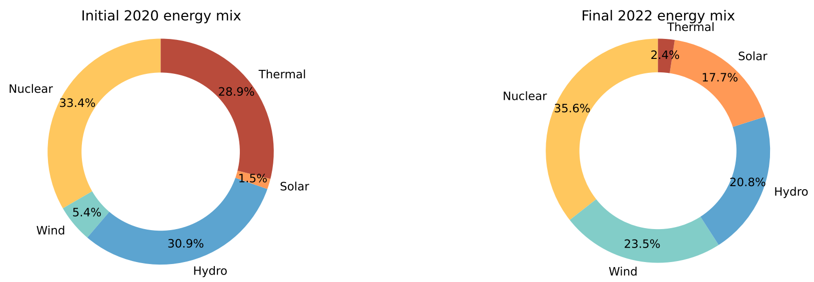


Figure 8: Increase renewable production

We can see that the resulting energy mix includes less thermal energy and more renewable energy. It is therefore closer to what we want to obtain. However, almost a quarter of the electricity is still produced by thermal power plants. Therefore, we would like to multiply the maximum production of renewable power plants by a number greater than 1.5. Unfortunately this is not possible because otherwise, at certain time steps, the renewable power plants will produce too much electricity compared to the consumptions and therefore Chronix2grid will not able to generate the time series.

2.4 Final data

After discussion with RTE experts, it turned out that the second method was the one to use. Indeed, in reality, solar and wind power plants can be ordered to produce less electricity. This is called curtailment. This would allow to increase even more the maximum production of the renewable power plants because Chronix2grid could "throw away" the excess of electricity when the consumptions are lower than the production. However, curtailment was not implemented in Chronix2grid. So Benjamin Donnot, a RTE expert, added it and it allowed him to generate the data that will be used in this year's edition and whose energy mix is described by the Figure 9.



(a) 2020 Energy mix (b) 2022 Energy mix

Figure 9: Comparison energy mix 2020-2022

We can see that 2.4% of the electricity is still produced by thermal power plants but this is a very small part of the energy mix. Moreover, according to the RTE experts, we cannot do better without deeply changing Chronix2grid. These data are therefore more than satisfactory for the 2022 edition of L2RPN.

Now that we have our data, I can focus on creating a simple agent.

3 Expert Agent

As said in the introduction, the second part of my TER consisted in the creation of a simple agent in order to prepare myself to create a more complex one which will serve as the baseline for this year's edition. To create these two agents, I will work with Eva Boguslawski who is a PhD student at RTE. The reinforcement learning approach is privileged in the L2RPN challenges, however, not being experts, Eva and I decided to create a first agent that uses only expert rules to build our intuition about the actions and their consequences on the network. To do this, we started with the power network shown in Figure 10, which is much smaller than the one that will be used for this year's edition in order to facilitate the research of problems and their solutions.

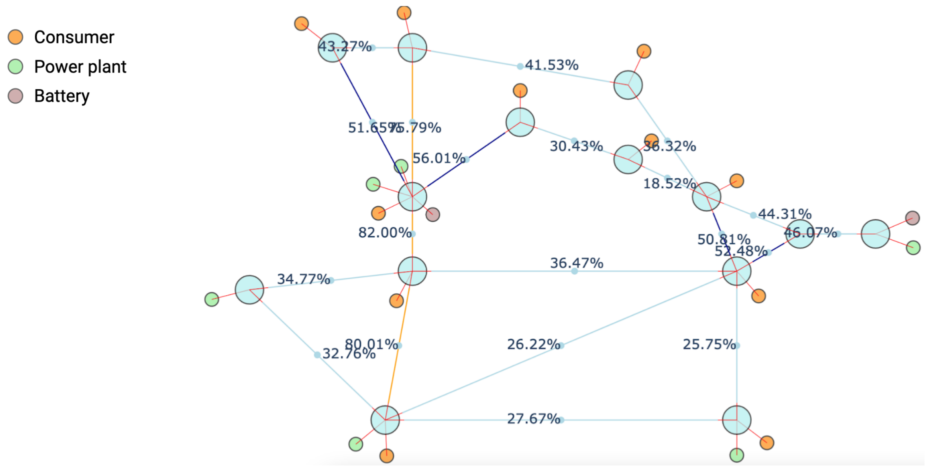


Figure 10: Used network for the expert agent

3.1 Environment details

Before implementing an agent, we must define the information which the agent has at its disposal at each time step (we call it an *observation*) as well as the possible actions.
An observation is here a vector of 425 dimensions containing information about the state of the power network, such as the date, the load of the lines, the state of the power plants, the state of the batteries, etc... The agent can use all this information to choose which action to do. At each time step, the possible actions are curtail renewable power plants (reduce the amount of electricity produced by these power plants) or charge/discharge batteries.
Moreover, knowing that the L2RPN challenges are focused on reinforcement learning, we need to define a reward i.e. a number given by the environment to the agent at each time step allowing the agent to know if the action it has just done seems good. In the old competitions, the reward took into account many parameters like the distance to the original network, the maximum network load,

etc... For this edition, the reward is still to be determined because, contrary to the other editions, we use batteries.

3.2 First intuition

The first thing we noticed with Eva is that we can split the network horizontally into two parts. The lower part contains the majority of the electricity production and the upper part the majority of the consumptions. So the electricity will flow from the bottom to the top. Thus, our first intuition is that the line highlighted in Figure 11 will cause overflows and therefore blackouts because its capacity is way lower than the amount of electricity that will want to go through.

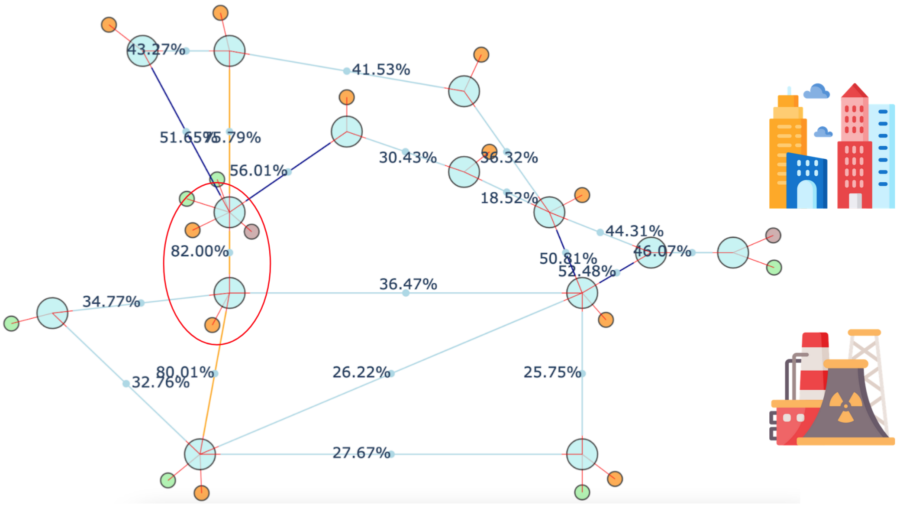


Figure 11: Problematic line

To confirm our intuition, we simulated a scenario of the network with a Do Nothing agent i.e., an agent that does not perform any action.
A scenario is composed of two time series: one for the production and one for the consumptions which represents a standard duration (e.g. one day, one week or one month). Here, a scenario represents one day.
Not surprisingly, this line causes an overflow and then a blackout during the peak of consumptions at 7pm. Therefore, we had to define rules to solve this problem.

3.3 Expert rules

Our idea to make sure that the problematic line does not cause any more overflows was to use the batteries that are located on the upper part of the of the network. By charging them when consumptions are low and using them when they are high, it would limit the risk of overflow.
Thus, we have defined two states of the network: the state **DANGER** and the state **NO DANGER**. To define in which state the network is at each time step, we look for the most loaded line and if its load is above a certain threshold, we consider that the network is in the state **DANGER**. According to the state, our agent will make several actions:

- If the network is in the state **NO DANGER**, charge the batteries
- If the network is in the state **DANGER**, use the batteries and reduce the energy produced by the power plants located in the lower part of the network.

After implementing an agent using these rules, we simulated the previous scenario of the network with our agent and this time, no blackouts or overflow happens. So we compared our agent with the Do Nothing agent on all the scenarios available (seven days in January) and we can see on the Figure 12 that our agent is either better or similar to the Do Nothing agent.

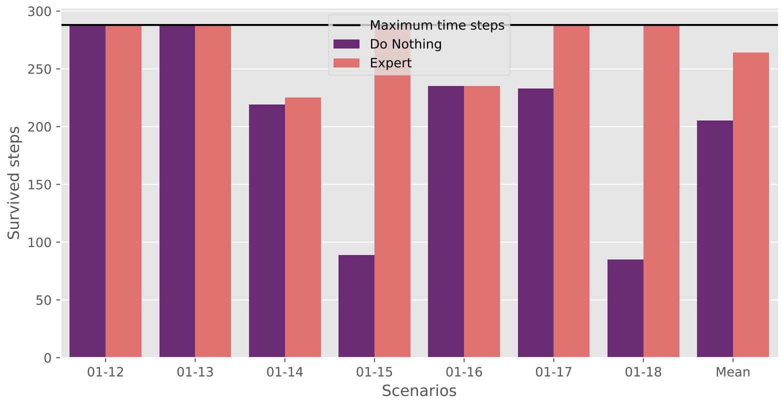


Figure 12: Comparison between our Expert agent and the Do Nothing agent

However, we can see that, for some scenarios, our agent did not succeed to avoid a blackout. This is due to the fact that, sometimes, it is not the problematic line that is overloaded but the one just above. Since our solution does not foresee this case, the agent is unable to avoid this problem. This is one of the many limitations of our approach.

3.4 Limitations

Thanks to our expert rules, we were able to solve the main problem of our seven scenarios. However, this problem is dependent on many factors (the grid architecture, the weather, the type of power plants, ...). Consequently, so are our rules because if we change one of these factors, the problem potentially disappears, another one appears and our solution becomes useless. We would have to enumerate all the potential problems that could appear and create a solution for each one of them. It is therefore very complicated to create an expert agent efficient on many scenarios. Moreover, we have many parameters, such as the threshold determining the state of the network, that are very hard to optimize. This is why the reinforcement learning approach is privileged by RTE as opposed to the expert rules approach which requires a lot of technical knowledge and time.

4 Further work

As said in the introduction, I am doing my internship on this subject and more particularly on the design of a more complex agent that will use reinforcement learning and expert rules. We are currently working on it with Eva. We were able to define the actions available for our agent which turn out to be on continuous spaces. This limits our choice of reinforcement learning algorithm. On the advice of our supervisors, we chose to use the PPO (Proximal Policy Optimization) algorithm [9]. The PPO is an algorithm created by OpenAI that is part of the family of Policy Gradients algorithms. These algorithms consist of use a neural network to predict the probability of the agent to do each of the possible actions. To train the network, the algorithm will wait for the end of an episode (in our case a scenario). Depending on whether the agent has won or lost, it will modify the weights in order to increase/decrease the probability of doing the actions it has done for each of the observations of this episode. PPO is a variant that accepts continuous action spaces. More specifically, we use the PPO implementation of the Stable Baselines 3 library [8].

The only expert rule we are currently using is to do nothing if the network is in the state **NO DANGER**. Otherwise, we use the reinforcement learning part of the agent. This rule comes from the solutions of the winners of the previous editions of L2RPN. However, we think that it could be ill adapted to our problem since we might want to charge the batteries when the network is in the state **NO DANGER**, as we did for our expert agent.

First, we wanted to verify that our agent was learning something by limiting the number of scenarios available to make it overfit. It seems to work, as illustrated in Figure 13, our agent is much better than the Do Nothing agent on these scenarios. At the moment, we are looking for the best hyper-parameters of our agent namely the threshold of the state **DANGER**, the learning rate and the network architecture of the PPO. After that we want to see if our agent will be better if we remove the expert rule from previous editions.

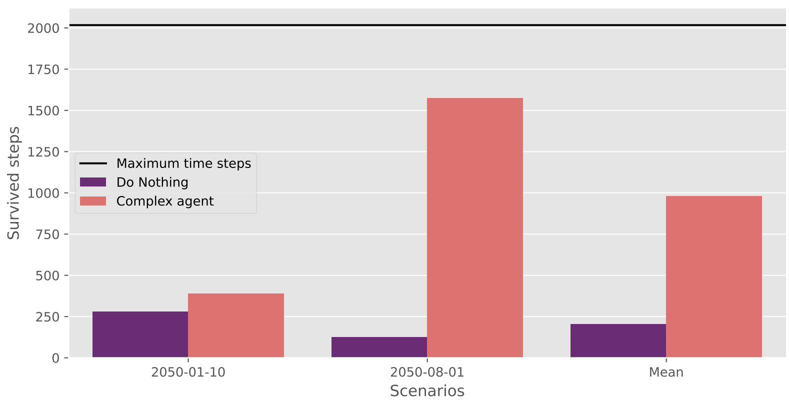


Figure 13: First results of our complex agent

5 Conclusion

To generate the data, I could see that the only method that worked is a soft one, with as few architectural modifications as possible. However, I needed help from RTE experts to modify Chronix2grid so that it is more realistic and can generate the time series for this year. Moreover, this part helped me to learn the libraries that will be used during the challenge.
Creating an expert agent allowed me to build my intuition about the actions and consequences on the network. I was able to learn how to highlight a problem on the power network and to find a solution. On the other hand, I was able to see the limits of this approach: its low adaptability and the and the huge amount of time and knowledge required to list all the potential problems and the search for the best parameters. Creating this agent allowed me to understand why the reinforcement learning approach was privileged and motivated me to work on a more complex agent, mixing reinforcement learning and expert rules.
All of this new knowledge will undoubtedly be useful during my internship, to create the baseline agent but also to participate in the launch of the competition which will take place on June 15, 2022. Moreover, this TER allowed me to apply my skills in mathematics, computer science and artificial intelligence acquired during my academic career in an industrial research problem, which will be very useful in my future professional experiences.

References

[1] *Chronix2grid* developers. Chronix2grid documentation. https://chronix2grid.readthedocs.io/.

[2] *Grid2Op* developers. Grid2op documentation. https://grid2op.readthedocs.io/.

[3] Balthazar Donon. *Deep statistical solvers & power systems applications.* Theses, Université Paris-Saclay, March 2022.

[4] RTE France. Futurs énergétiques 2050 principaux résultats, 2021. https://assets.rte-france.com/prod/public/2021-10/Futurs-Energetiques-2050-principaux-resultats_0.pdf.

[5] S Kirkpatrick, C D Gelatt, Jr, and M P Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983.

[6] Antoine Marot, Benjamin Donnot, Gabriel Dulac-Arnold, Adrian Kelly, Aidan O'Sullivan, Jan Viebahn, Mariette Awad, Isabelle Guyon, Patrick Panciatici, and Camilo Romero. Learning to run a power network challenge: a retrospective analysis, 2020. http://proceedings.mlr.press/v133/marot21a/marot21a.pdf.

[7] Adrien Pavao, Eva Boguslawski, Benjamin Donnot, Isabelle Guyon, Alessandro Leite, Antoine Marot, Gaëtan Serré, Marc Schoenauer, and Sébastien Treguer. Energies of the future and carbon neutrality challenge proposal, 2022.

[8] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. https://jmlr.org/papers/volume22/20-1364/20-1364.pdf.

[9] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. https://arxiv.org/pdf/1707.06347.pdf.

A Appendix

$$min_x \quad \sum_{i}^{d} [Tem_i - em_i(x)]^2 \tag{2}$$

Where:

- $d \in \mathbb{N}$ is the number of types of power plants
- $n \in \mathbb{N}$ is the number of power plants in the grid
- $x \in \mathbb{N}^n$ represents the type of each power plant
- $em_i(x) \in \mathbb{R}$ is the percentage of electricity produced by the type i in the configuration x
- $Tem_i \in \mathbb{R}$ is the target energy mix of the type i