23rd International Conference on Renewable Energies and Power Quality (ICREPQ'25)
Tenerife (Spain), 25th to 27th June 20245
Renewable Energy and Power Quality Journal (RE&PQJ)
ISSN 2172-038 X, Volume No.23, September 2025

First Approximation of Application of Federated Learning to Wind Turbines

A. Gil Macia[1], J. Enrique Sierra-García[2] and Matilde Santos[3]

[1] Department of Computer Science and Automatic Control, UNED, 28040-Madrid (Spain), agil945@alumno.uned.es

[2] Department of Digitalization, University of Burgos, 09006–Burgos (Spain), jesierra@ubu.es

[3] Institute of Knowledge Technology, Complutense University of Madrid, 28040–Madrid (Spain), msantos@ucm.es

Abstract. This work investigates the application of Federated Learning techniques to reduce the training time of wind turbine power stabilization controllers on a wind farm. A Reinforcement Learning controller based on Q-learning is implemented and the results of the individual controller are compared with a system of 4 wind turbines using Federated Learning. The simulation results show how this technique significantly improves the convergence time of the controller when compared to control strategies without federated learning. The preliminary results demonstrate how Federated Learning has great potential for improving the effectiveness of wind turbine controllers while maintaining the privacy and security of their operational data.

Key words. Q-Learning, Federated Learning, Intelligent Control, Reinforcement Learning, Wind Turbine.

1. Introduction

Renewable energy is the bet of most countries to reduce their carbon footprint and decrease their impact on climate change. Among clean energy sources, wind energy stands out due to its great capacity to produce energy and the high development of its technology [1].

Nevertheless, the nonlinear dynamics and modeling uncertainties of wind turbines pose significant challenges when developing effective and robust control strategies, which are, on the other hand, essential for them to be efficient and profitable [2].

To improve the results obtained with classical control techniques, data-driven approaches have been proposed in recent years that allow the controller to learn optimal control laws without the need for an accurate representation of the plant [3]. Among these techniques, Reinforcement Learning (RL) has shown good control characteristics regarding reduction of structural loads and stabilization of the power output [4]. These results are mainly attributed to the ability of reinforcement learning controllers to optimize system performance by dynamically adapting to changing operational conditions [5].

Despite the promising results obtained when applying Reinforcement Learning control techniques to wind turbines, their adoption in the industry has been relatively limited due to the large amount of data required by these algorithms to learn the optimal control strategy. Although a possible solution to this problem would be to combine the data from multiple wind turbines to train a global model, decreasing in this way the training times, the sensitivity of operational data in wind turbines and the cybersecurity risks associated with transiting the data over a network make this solution impractical.

In this context, Federated Learning (FL) techniques emerge as a possible solution to this problem. Federated learning is a decentralized machine learning approach where multiple devices or nodes collaboratively train a shared model without sharing raw data. Instead, each device trains the model locally using its own data, and only model updates (e.g., gradients) are sent to a central server for aggregation, preserving data privacy and reducing communication overhead [6].

Inspired by the work presented in [7], we propose to apply federated learning to train a global model using the individual data from each turbine without the need to share it outside its local system. This method preserves data privacy while also leveraging information from multiple wind turbines to increase controller performance.

To do so, this paper explores the integration of Federated Learning (FL) into a system of wind turbines with individual Reinforcement Learning (RL) controllers, with the objective of overcoming some of the current limitations of data-driven controllers while preserving the privacy of the data involved in the process.

Simulation results prove that the use of reinforcement learning enables a reduction in the training time of the Q-learning algorithm applied to power control in wind turbines.

The structure of the paper is as follows. Section 2 summarizes some contributions on related topics. In section 3, the Q-learning algorithm is presented, and the control architecture of a Federate Learning scheme is presented. Section 4 discusses the results obtained with federated learning-based model control and compares it with non-federated control. The paper ends with the conclusions and future work.

2. Related Work

Since its original introduction in 2016, Federated Learning (FL) has gained increasing popularity in multiple industries due to its ability to train machine learning models on a distributed system without the need to share sensitive data in the process, thereby preserving the privacy and security of the data involved [8]. However, despite its growing adoption in multiple fields [9], even in renewable energy [10], its application in the field of control is still relatively limited.

In the very recent paper in [11], the authors detail the latest advances regarding the application of Federated Learning techniques in the field of control. This study highlights the potential benefits of this technique in terms of adaptability, scalability, generalization, and privacy preservation. While also making a comprehensive analysis of its current challenges such as communication overhead, non-IID (non-independent and identically distributed) data, and the limitations of current aggregation techniques with regard to robustness.

In the paper in [7], Federated Learning is used in combination with Reinforcement Learning to enhance performance in Autonomous Guided Vehicles (AGVs). The article demonstrates how Federated Learning techniques can be used to increase performance in Reinforcement Learning controllers without the need to share sensitive data in the process. Finally, the paper suggests possible applications of FL in other control domains, such as wind turbine control systems.

Nevertheless, FL has been recently applied in the wind energy field for forecasting [12]. As an example, the paper by [13] shows a Federated Learning-based model for wind power prediction of different locations in Pakistan using wind speed and wind direction. It uses different machine learning techniques, such as Linear Regression (LR), Support Vector Regression (SVR), Random Forest Regression (RFR), Extreme Gradient Boosting Regression (XGBR), and Multilayer Perceptron Regression (MLPR) models.

Finally, some investigation has been conducted into the use of Reinforcement Learning (RL) techniques for the power stabilization of wind turbines. To name a few, the work presented in [14] explores the use of reinforcement learning (RL) for optimizing wind turbine control to maximize energy capture and minimize structural loads. Specifically, the authors apply a double deep Q-learning as an agent to control wind turbine production using the rotor velocity, blade angle, and the yaw of the nacelle orientation as control variables. The paper by Xie et al. [4] addresses the torque and pitch control problems of wind energy converters. Their design applies in real-time a reinforcement learning-based control that combines deep neural networks (DNNs) and model predictive control (MPC).

While there are recent studies exploring the use of Federated Learning in control systems. As well as the use of Reinforcement Learning controllers for power stabilization and load reduction of wind turbines. The combination of these techniques in the domain of wind turbines remains largely unexplored. To the best of our knowledge, no prior work has applied Federated Learning techniques to Reinforcement Learning-based control of wind turbines.

3. Federated Learning Control Proposal

Wind turbines operate in different regions depending on wind speed, and each region requires specific control strategies to optimize performance and ensure safety. In this paper, we focus on the full load region (between rated and cut-out wind speed), where wind speeds are between the rated speed and the cut-out speed. In this mode, the turbine operates at its rated power output, that is, at its maximum capacity. Control in this region aims to maintain constant power by adjusting the angle of the blades, and the pitch angle, to shed excess aerodynamic load.

Therefore, the control variable is the blade pitch angle of the wind turbine, and the control objective is to maintain the output power of the turbine as close as possible to the optimal value for which the wind turbine has been rated. This way energy efficiency is maximized while the control limits the power output to protect the wind turbine.

To achieve this objective, an individual control system based on reinforcement learning is implemented [15].

Fig.1. shows the general architecture of a reinforcement learning controller. In this approach, the optimal control strategy is learned from the interaction between the controller and his environment. The controller, represented by the agent, modifies the environment by performing an action $\boldsymbol{a_t}$. The execution of this action will give as a result a change in the state of the environment $\boldsymbol{s_{t+1}}$ and a reward value $\boldsymbol{r_t}$, representing how beneficial the action was with regards to the control objectives. Finally, the reinforcement learning algorithm uses this information to adapt its policy in order to promote future actions that maximize the reward over time.

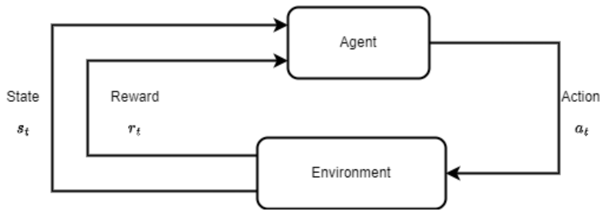

Fig.1. Architecture of Reinforcement Learning methods. Adapted from [16].

Depending on the method used to update the policy, it is possible to distinguish between different types of reinforcement learning techniques. In this case, Q-learning is used. In this approach, an expected reward (Q-Value) is assigned to each possible combination of action and state, resulting in a table of Q-Values referred to as Q-Table. Which is updated based on the rewards obtained from the environment. Finally, the controller policy is obtained from the Q-Table by choosing the action with the maximum expected reward for the current state.

Due to the discrete nature of this approach, the state and action spaces need to be discretized into a finite number of states.

In the developed controller, the state of the system consists of the difference between the target power and the power output of the wind turbine $\boldsymbol{P_{error}}$. And the current wind speed $\boldsymbol{v_{wind}}$.

In order to decrease the number of states while allowing finer control when the output power is close to the reference power, the value of $\boldsymbol{P_{error}}$ is discretized using the following geometric exponential function:

$$\boldsymbol{S = sign(x)e^{\left|\frac{x}{1.5}\right|}} \qquad (1)$$

Where $\mathbf{P_{error}} \in$ [-1097, 1097].

On the other hand, the value of $\boldsymbol{v_{wind}}$ is discretized in increments of 0.1 m/s for an expected wind interval between 6 m/s and 15 m/s.

$$\{\boldsymbol{x} \in \boldsymbol{R} \mid \boldsymbol{6 \le x \le 15},\ \boldsymbol{x = 6 + 0.1k, k \in Z}\} \quad (2)$$

The reward function is used to incentivize the agent to learn optimal control strategies by encouraging behaviors that minimize the error between the power output $\boldsymbol{P_{out}}$ and the desired power $\boldsymbol{P_{ref}}$. In this case, the reward function is defined as the absolute value of the error, that is:

$$\boldsymbol{r = -|P_{error}|} \qquad (3)$$

Additionally, the action of the controller $\boldsymbol{a_t}$, representing the pitch angle that will be applied to the wind turbine, is restricted to integer values within an interval of [0,20] degrees. This discretization of the action space is done to reduce the problem's complexity and decrease the time required to learn the optimal policy.

Finally, the Q-Table is updated based on the reward signal received, as well as the previous and current action and state of the system. The update rule in Q-Learning is given by the following formula [17]:

$$\boldsymbol{Q_t} \leftarrow \boldsymbol{Q_t} + \alpha[\boldsymbol{r_t} + \boldsymbol{\gamma Q}_{\max} - \boldsymbol{Q_t}] \qquad (4)$$

Where $\boldsymbol{Q_t = Q(s_t, a_t)}$, $\boldsymbol{Q_{max} = max_a Q(s_{t+1}, a)}$, $\boldsymbol{\alpha}$ is the learning rate and $\boldsymbol{\gamma}$ is a discount factor used to balance the importance of future rewards with respect to immediate rewards. In this case, a learning rate of $\boldsymbol{\alpha = 0.01}$ and a discount factor of $\boldsymbol{\gamma = 0.95}$ are used.

Finally, the next action is selected from the Q-Table based on the current state of the system. To explore the search space, an epsilon factor of 0.01 is used, giving a 1% chance to select a random actuation value from the search space during the training process.

A. Federated learning

With the objective of reducing training time and improving controller performance, a federated learning scheme is implemented. In this approach, each wind turbine represents an independent agent controlled by a Reinforcement Learning controller. When a wind turbine receives a new reward signal, it will update its local Q-Table $\boldsymbol{Q(s,a)}$, while also storing the changes on a temporary table $\boldsymbol{\Delta Q(s,a)}$. After a given time, all the individual systems will send their temporary table $\boldsymbol{\Delta Q(s,a)}$ to the federated server, which will aggregate the information and update the global model $\boldsymbol{Q_{global}(s,a)}$. Finally, the global model is distributed to all the individual wind turbines in order to update their local Q-Tables.

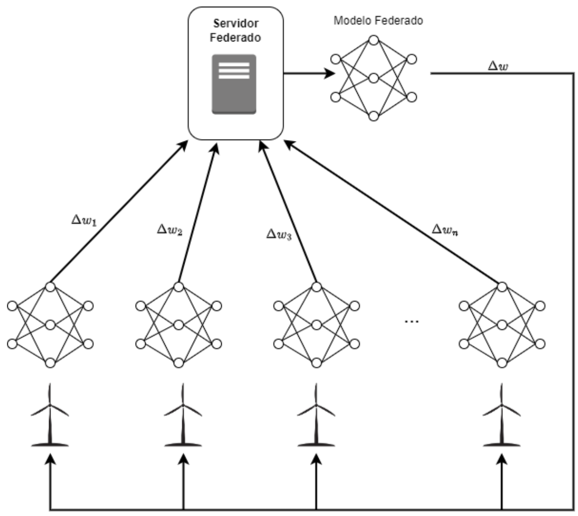

Fig.2. Federated learning scheme applied to wind turbines.

Each wind turbine sends its Q-Table updates to the Federated Server with a frequency of $\boldsymbol{f_{sync} = \frac{1}{t_s} Hz}$. On the other hand, the global model is distributed to the wind turbines with a frequency of $\boldsymbol{f_{update} = \frac{1}{t_u} Hz}$. For our implementations, values of $\boldsymbol{t_s = 10s}$ and $\boldsymbol{t_u = 50s}$ are used.

Due to the probabilistic information represented in the Q-Table, the information from the wind turbines cannot be aggregated by averaging the values of each wind turbine. This is because, on a given iteration, the combination of a state and action may have been explored by a single wind turbine. Because of this, averaging the results of all the agents in the system would lead to incorrect probability values. As a result, the use of traditional reinforcement learning aggregation techniques like Federate Averaging [18] does not yield satisfactory results. Instead, the federated server aggregates the updates from all the turbines using the following aggregation function:

$$Q_{global\_new}(s,a) = Q_{global}(s,a) + \sum_{i=0}^{n} \Delta Q_i(s,a) \quad (5)$$

where n is the number of wind turbines in the system and $\Delta Q_i(s,a)$ represents the local updates of the i-th turbine. This aggregation method adds the individual knowledge gained by each of the wind turbines into a global model $Q_{global}(s,a)$ without the need to share raw operational data of the system.

During the update process, the only data shared with the federated learning are the updates in the Q-table. As such, the privacy of sensitive data is preserved while leveraging the collective knowledge of all the systems.

4. Results

To test the hypothesis, the learning times of an individual wind turbine were compared with the results of a system of 4 wind turbines using the Federated Learning architecture described earlier.

The wind turbine dynamics were simulated using the 1.5MW model introduced in [19]. In this model, the Tip Speed Ratio (TSR) is related to the rotor angular speed (w) and the wind speed (V) by the following expression:

$$TSR = \frac{Rw}{V} \quad (6)$$

, where R is the rotor radius expressed in meters and w is the rotor angular speed in rad/s.

Based on the Tip Speed Ratio (TSR) and the blade angle θ_{pitch} values, the power coefficient C_p is calculated from a lookup table that defines the efficiency of the wind turbine for each operational point.

Once the power coefficient C_p of the wind turbine has been determined, the aerodynamic power is calculated using the following expression:

$$P_{aero} = 0.5\rho AV^3 C_p(TSR, \theta_{pitch}) \quad (7)$$

, where ρ is the air density in Kg/m^3 and $A = \pi R^2$ is the rotor sweep area in m^2.

The aerodynamic power P_{aero}, together with the rotor speed w can be used to calculate the aerodynamic torque T_a. To do that, the following equation is used:

$$T_a = \frac{P_{aero}}{w} \quad (8)$$

The generator reaction torque is related to the aerodynamic torque and the rotor speed by the following expression:

$$J\dot{w} = T_a - T_g \quad (9)$$

Where J is the rotor inertia in $kg \cdot m^2$ and T_g is the reaction torque of the generator in $N \cdot m$.

On the other hand, the generator torque T_{gen} is defined as the reaction torque scaled by the gear ratio N:

$$T_g = \frac{T_{gen}}{N} \quad (10)$$

Finally, the output power generated by the turbine is calculated using the following expression:

$$P_{gen} = \eta_g \cdot P_{mech} \quad (11)$$

Where η_g is the generator efficiency and P_{mech} is the input mechanical power to the generator defined as:

$$P_{mech} = T_{gen} w \quad (12)$$

Additionally, a dynamic rate limiter of $0.17\, rad/s$ is added on the plant input to simulate some actuator delay in θ_{pitch}.

The same model parameters and C_p values presented in the base model are used [19]. The most relevant of these parameters are summarized in the following table:

Table I. - Wind turbine model parameters.

PARAMETER	VALUE	UNITS
R	35	m
J	3410432	$kg \cdot m^2$
N	88	-
ρ	1.225	kg/m^3
η_g	0.95	-

Each wind turbine was simulated with a random wind model generated using a Weibull probability distribution with a scale parameter λ=11 and a shape parameter *k*=10. Based on this distribution, the wind model was created by taking a weighted average between the current wind value and the new value obtained from the distribution.

$$v_{wind} = \alpha v_{wind} + \beta f(x; \lambda, k) \quad (14)$$

Where $f(x; \lambda, k)$ is the Weibull distribution and α and β are weighting factors.

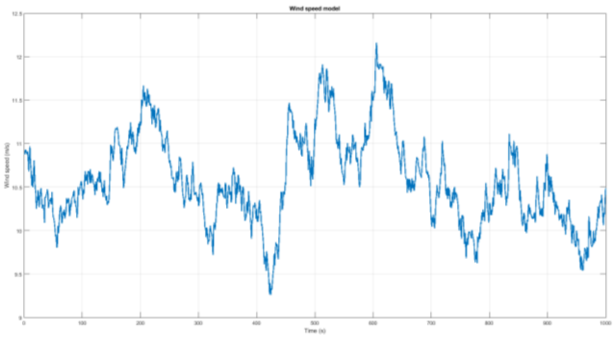

Fig.3. Wind model simulation using a Weibull probability distribution with λ=12 and k=2.

Additionally, the reference power output P_{ref} was set to 800 GW/hour, being the maximum rated output power of the simulated wind turbine at 1.5 MW/hour.

Both federated and non-federated systems were trained for 130 iterations, each iteration of 800 seconds, resulting in the following Mean Squared Error (MSE) per iteration:

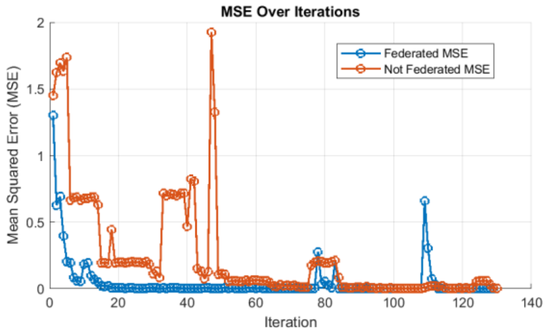

Fig.4. Mean squared error comparison between federated (blue line) and non-federated (red line) algorithms over 130 iterations

Fig.4. shows how the Federated Learning-based control was able to learn a better control strategy in significantly fewer iterations when compared to the non-federated approach. The fluctuations in the MSE values between iterations after the algorithm has converged can be explained due to the epsilon value used by the controller to search the action space and changes in the transient response of the controller.

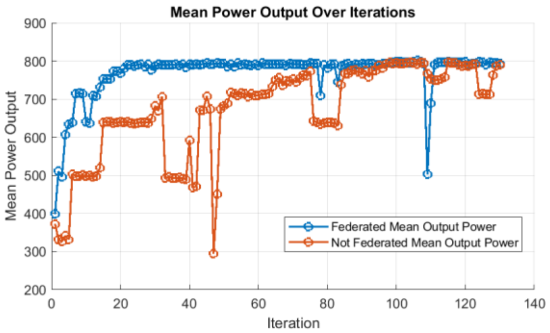

Fig.5. Mean squared error comparison between federated (blue line) and non-federated (red line) algorithms over 130 iterations

Fig.5. shows how the Federated Learning-based controller achieves the reference mean power output in significantly fewer iterations than the non-federated approach.

The results show that the federated learning output was significantly better in all the iterations. Particularly, wind turbines using Federated Learning stabilized the power output around the reference power after 28 iterations. Meanwhile, the non-federated system required 95 iterations to achieve similar results consistently.

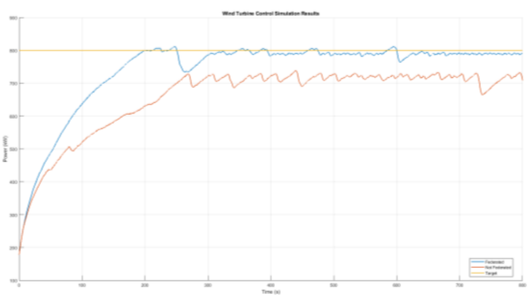

Fig.6. Power output of federated (blue line) and non-federated (red line) wind turbines at iteration 81.

Fig.6. compares the results obtained with and without Federated Learning after 81 iterations of training. As can be observed, the Federated Learning approach achieves significantly better performance, with output power values closer to the reference power than the non-federated controller.

The previous results compared the average power output of all the wind turbines using Federated Learning with a wind turbine implemented without Federated Learning. However, to have a better understanding of how the different agents on the Federated Learning system evolve, the results of each wind turbine used in the Federated Learning are shown.

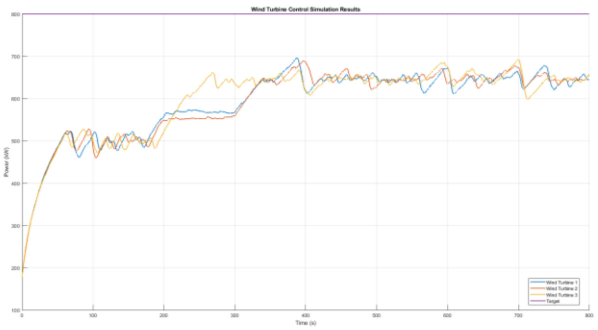

Fig.7. Power output of each wind turbine after 6 iterations of training.

Fig.7. shows how the power output of all the wind turbines in the Federated Learning model evolves in a similar way. When one of the wind turbines finds a better control strategy, this knowledge is shared with the federated server and, after the global model is distributed, the rest of the agents incorporate these improvements in their control strategy.

These results demonstrate the effectiveness of Federated Learning in improving performance and reducing training times in Reinforcement Learning controllers. Showing his potential use in optimizing power output in multi-turbine systems.

5. Conclusion and further works

This paper demonstrates how Federated Learning techniques can be used to reduce training times of Reinforcement Learning controllers based on Q-learning when applied to the power output stabilization of wind turbines. A system of four wind turbines using federated learning was compared with a non-federated learning system, obtaining a significant reduction in the convergence time. The simulation showed how the federated learning approach learned an optimal control strategy in fewer iterations than the non-federated approach. In particular, the mean output power of the federated learning approach converged to the reference power output after 28 iterations, compared to the 95 iterations required by the non-federated approach.

This approach to reinforcement learning has the potential to reduce the training times of Q-Learning controllers used for the power stabilization of wind turbines while preserving the security and data privacy of the data involved during the training.

However, while the proposed controller has been tested with a simulated model, its generalization to real systems remains unknown. Additionally, the application of federated learning to other reinforcement learning controllers and reward strategies remains largely unexplored, as well as a comprehensive analysis of the robustness of the proposed solution.

Finally, future work that remains open is the exploration of different reward functions to reduce rotor vibrations while maximizing power output, a robustness analysis of the obtained controllers, and the use of more advanced reinforcement learning techniques such as Deep Deterministic Policy Gradient (DDPG). Additionally, the use of other federated learning techniques such as Clustered Federated Learning (CFL) remains an open area of exploration.

Acknowledgments

This work has been partially funded by the Spanish Ministry of Science and Innovation, under the MCI/AEI/FEDER project number PID2021-123543OBC21.

References

[1] Hassan, Q., Viktor, P., Al-Musawi, T. J., Ali, B. M., Algburi, S., Alzoubi, H. M., ... & Jaszczur, M. (2024). The renewable energy role in the global energy Transformations. Renewable Energy Focus, 48, 100545.

[2] Zhou, Bowen & Zhang, Zhibo & Li, Guangdi & Yang, Dongsheng & Santos, Matilde. (2023). Review of Key Technologies for Offshore Floating Wind Power Generation. Energies. 16. 710. 10.3390/en16020710.

[3] Pandit, R., Infield, D., & Santos, M. (2022). Accounting for environmental conditions in data-driven wind turbine power models. IEEE Transactions on Sustainable Energy, 14(1), 168-177.

[4] Xie, J., Dong, H., & Zhao, X. (2023). Power regulation and load mitigation of floating wind turbines via reinforcement learning. IEEE Transactions on Automation Science and Engineering.

[5] Dong, H., Xie, J., & Zhao, X. (2022). Wind farm control technologies: from classical control to reinforcement learning. Progress in Energy, 4(3), 032006.

[6] Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., & Gao, Y. (2021). A survey on federated learning. Knowledge-Based Systems, 216, 106775.

[7] Sierra-García, J. E., & Santos, M. (2024). Federated discrete reinforcement learning for automatic guided vehicle control. Future Generation Computer Systems, 150, 78-89. https://doi.org/10.1016/j.future.2023.08.021.

[8] McMahan, H.B., Moore, E., Ramage, D., Hampson, S., & Arcas, B.A. (2016). Communication-Efficient Learning of Deep Networks from Decentralized Data. International Conference on Artificial Intelligence and Statistics.

[9] Wen, J., Zhang, Z., Lan, Y., Cui, Z., Cai, J., & Zhang, W. (2023). A survey on federated learning: challenges and applications. International Journal of Machine Learning and Cybernetics, 14(2), 513-535.

[10] Grataloup, A., Jonas, S., & Meyer, A. (2024). A review of federated learning in renewable energy applications: Potential, challenges, and future directions. Energy and AI, 100375.

[11] Weber, J., Gurtner, M., Lobe, A., Trachte, A., & Kugi, A. (2024). Combining federated learning and control: A survey. IET Control Theory & Applications, 18(18), 2503-2523.

[12] Arooj, Q. (2024). FedWindT: Federated learning assisted transformer architecture for collaborative and secure wind power forecasting in diverse conditions. Energy, 309, 133072.

[13] Alshardan, A., Tariq, S., Bashir, R. N., Saidani, O., & Jahangir, R. (2024). Federated learning (FL) model of wind power prediction. IEEE Access.

[14] Soler, D., Mariño, O., Huergo, D., de Frutos, M., & Ferrer, E. (2024). Reinforcement learning to maximize wind turbine energy generation. Expert Systems with Applications, 249, 123502.

[15] Sierra-García, J. E., & Santos, M. (2020). Exploring reward strategies for wind turbine pitch control by reinforcement learning. Applied Sciences, 10(21), 7462.

[16] Amiri, Roohollah & Mehrpouyan, Hani & Fridman, Lex & Mallik, Ranjan & Nallanathan, Arumugam & Matolak, David. (2018). A Machine Learning Approach for Power Allocation in HetNets Considering QoS. 10.48550/arXiv.1803.06760.

[17] Watkins, Christopher & Dayan, Peter. (1992). Technical Note: Q-Learning. Machine Learning. 8. 279-292. 10.1007/BF00992698.

[18] T. Sun, D. Li and B. Wang, "Decentralized Federated Averaging," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 4, pp. 4289-4301, 1 April 2023, doi: 10.1109/TPAMI.2022.3196503.

[19] The MathWorks, Inc. “Control design for wind turbine.” mathworks.com. Accessed: February 12, 2025. [Online]. Available: https://nl.mathworks.com/help/control/ug/wind-turbine-control-design.html