Environmental Data Science (2025), 4: e19, 1–21
doi:10.1017/eds.2025.10

CAMBRIDGE
UNIVERSITY PRESS

APPLICATION PAPER

WindDragon: automated deep learning for regional wind power forecasting

Julie Keisler[1,2] and Etienne Le Naour[1,3]

[1]EDF R&D, Palaiseau, France
[2]University Lille, INRIA, CNRS, Centrale Lille, UMR 9189 CRIStAL, Lille, France
[3]Sorbonne Université, CNRS, ISIR, Paris, France
Corresponding authors: Julie Keisler and Etienne Le Naour; Emails: julie.keisler@inria.fr; etienne.le-naour@edf.fr

Received: 28 August 2024; **Revised:** 07 February 2025; **Accepted:** 20 February 2025

Keywords: automated machine learning; deep learning; renewable energies; wind power forecasting

Abstract

Achieving net-zero carbon emissions by 2050 necessitates the integration of substantial wind power capacity into national power grids. However, the inherent variability and uncertainty of wind energy present significant challenges for grid operators, particularly in maintaining system stability and balance. Accurate short-term forecasting of wind power is therefore essential. This article introduces an innovative framework for regional wind power forecasting over short-term horizons (1–6 h), employing a novel Automated Deep Learning regression framework called *WindDragon*. Specifically designed to process wind speed maps, *WindDragon* automatically creates Deep Learning models leveraging Numerical Weather Prediction (NWP) data to deliver state-of-the-art wind power forecasts. We conduct extensive evaluations on data from France for the year 2020, benchmarking *WindDragon* against a diverse set of baselines, including both deep learning and traditional methods. The results demonstrate that *WindDragon* achieves substantial improvements in forecast accuracy over the considered baselines, highlighting its potential for enhancing grid reliability in the face of increased wind power integration.

Impact Statement

This article presents an optimization tool to automatically find efficient deep neural networks to forecast aggregated wind power generation at the level of a region or a country. These models are based on wind speed maps from numerical weather prediction (NWP) forecasts and take advantage of their spatio-temporal aspect. These methods could play a crucial role in the smooth operation of power grids in the context of massive renewable energy integration.

1. Introduction

1.1. Global context

To meet the 2050 net zero scenario envisaged by the Paris Agreement [United Nations Convention on Climate Change, 2015], wind power stands out as a critical energy source for the future. Remarkable progress has been made since 2010, when global electricity generation from wind power was 342 TWh, rising to 2100 TWh in 2022 (International Energy Agency, IEA, 2023). The IEA targets approximately 7400 TWh of wind-generated electricity by 2030 to meet the zero-emissions scenario. However, to realize the full potential of this intermittent energy source, accurate forecasts of wind power generation are needed to efficiently integrate it into the power grid.

1.2. Regional wind power forecasting

Most of the work in the literature on wind power forecasting is done at a local scale, that is, an individual wind farm or turbine. In this article, we focus on a more global scale, the aggregated production of a country or a large region. Regional wind power generation forecast is critical in the context of the European electricity market for several reasons. (i) First, a short-term forecast of up to 48 h is useful for the spot (day-ahead) market, which sets the “final” price of electricity hour by hour according to supply and demand. (ii) Second, Short-term forecasts are useful for the TSO (Transmission System Operator), which has to ensure the balance between supply and demand on the transmission network within its perimeter. (iii) Finally, in the longer term, up to a few days, regional wind power forecasts can be used to anticipate downturns. They correspond to a situation in which a large amount of renewable energy is fed into the grid at the same time. Renewable energies indeed have market priority over, for example, nuclear or coal, which are more expensive to produce.

Wind power generation forecast at a global scale can be done in two ways, either by forecasting each farm in the region (or even each wind turbine) and then adding these forecasts together, or by directly forecasting the aggregated signal. The first method is impractical for the majority of operators, as it requires production data for each park, which is confidential. Moreover, even in cases where the data is available, Wang et al. (2017) pointed out that having a forecast system for each wind farm in the region considered can be too costly for some forecast service providers. In this article, we focus on wind power generation forecast at a global scale.

1.3. Contributions

In this study, we propose to leverage the spatial information in NWP wind speed maps for national wind power forecasting by exploiting the capabilities of Deep Learning (DL) models. The overall methodology is illustrated in Figure 1. To fully exploit the potential of the DL mechanisms, we introduce WindDragon, an automated deep-learning framework that uses the tools developed in the DRAGON[1]

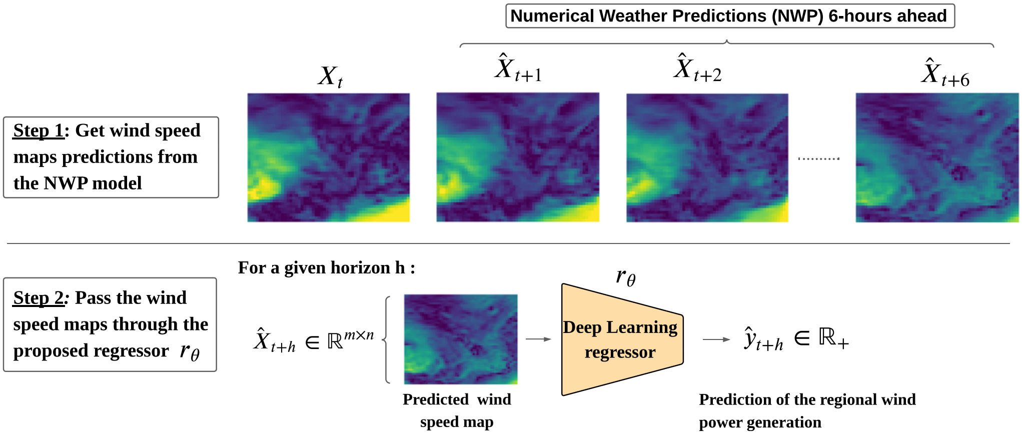

Numerical Weather Predictions (NWP) 6-hours ahead
X_t
$\hat{X}_{t+1}$
$\hat{X}_{t+2}$
$\hat{X}_{t+6}$
Step 1: Get wind speed maps predictions from the NWP model
Step 2: Pass the wind speed maps through the proposed regressor r_θ
For a given horizon h :
$\hat{X}_{t+h} \in \mathbb{R}^{m \times n}$
r_θ
Deep Learning regressor
$\hat{y}_{t+h} \in \mathbb{R}_+$
Predicted wind speed map
Prediction of the regional wind power generation

Figure 1. Global scheme for wind power forecasting. Every 6 h, the NWP model produces hourly forecasts. Each map is processed independently by the regressor which maps the grid to the wind power corresponding to the same timestamp.

[1] https://dragon-tutorial.readthedocs.io/en/latest/index.html.

https://doi.org/10.1017/eds.2025.10 Published online by Cambridge University Press

package (Keisler et al., 2024b). WindDragon attempts to automatically design well-performing neural networks for short-term wind power forecasting using NWP wind speed maps. WindDragon's performance will be benchmarked against conventional computer vision models such as Convolutional Neural Networks (CNNs) as well as standard baselines in wind power forecasting. The contributions of this study can be summarized as follows:

- We develop a novel automated deep learning framework specifically tailored to forecast aggregated wind power generation from wind speed maps.
- The proposed framework, named WindDragon, is designed to fully leverage the spatial information embedded in wind speed maps and can accommodate increases in installed capacity, making it adaptable and reusable.
- We conduct extensive experiments that demonstrate that WindDragon, when combined with Numerical Weather Prediction (NWP) wind speed maps, significantly outperforms both traditional and state-of-the-art deep learning models in wind power forecasting.

2. State-of-the-art

Wind power forecasting at the level of a single wind farm is a mature discipline (Jonkers et al., 2024) on forecast horizons ranging from the next minutes to the next days (see Kariniotakis 2017 for a book on the subject). However, regional forecasting remains largely unexplored in the literature (Higashiyama et al., 2018).

2.1. Regional wind power forecasting

2.1.1. Transfer strategy

Some studies have attempted to take advantage of the wealth of research at the turbine or wind farm scale to forecast regional wind energy. The general idea is to apply a forecasting model to wind turbines or farms whose data are available within the region and use a transfer function to move from local to regional data. For instance, Pinson et al. (2003) mentioned a model based on online persistence scaled with a ratio of the total installed capacity in the region and the capacity of wind farms for which online measures are available. Camal et al. (2024) forecasted the production of any wind farm in the control area of a TSO, taking into account the information collected from other wind farms. The method combines feature selection, regularization, and local-learning via conditioning on recent production levels or expected weather conditions.

2.1.2. Input dimension reduction

Approaches that have attempted to forecast regional wind production directly from meteorological data such as NWP maps, or by incorporating operational variables from the (potentially numerous) wind farms in the region, have quickly run into the problem of the large size of the input data. Camal et al. (2024) noticed that at the scale of a region or of a country, the number of explanatory variables grows linearly with the number of explanatory sites or the number of variables considered per site. Both statistical and Machine Learning models face in this case the curse of dimensionality. Therefore, regularization or feature selection was investigated to mitigate the high dimensionality of the input features. Siebert (2008) used a clustering algorithm based on k-means and a mutual information-based feature selection algorithm to determine the best set of features for the forecast model. Lobo and Sanchez (2012) searched for samples with similar weather conditions. Davò et al. (2016) leveraged the principal component analysis (PCA) method to reduce the dimension of the data sets when forecasting regional wind power and solar irradiance. Wang et al. (2017) reduced the dimension of the NWP grid with the selection of minimum redundancy characteristics (mRMR) and PCA. They then applied a weighted average learning strategy to forecast the production of a Chinese region. In the study from Wang et al. (2018), the spatio-temporal weather data is represented using a distance-weighted kernel density estimation model (DWKDE) which

https://doi.org/10.1017/eds.2025.10 Published online by Cambridge University Press

is the basis for a feature selection method based on mRMR. Finally, Wang et al. (2019) performed a probabilistic forecasts with regular vine copulad to reduce the weather dataset.

Although this input reduction is necessary for most Machine Learning models, deep learning models have demonstrated high capacities for extracting complex features from high-dimensional data.

2.2. Deep learning for wind power forecasting

Deep learning models have been highly investigated for wind power forecasting both at the turbine level and at the regional aggregation level. A large variety of architectures have been used, depending on the input data available and the features that are sought to be extracted.

Yu et al. (2021) recognized the abilities of the deep learning model for non-linear mapping and massive data handling and used a feedforward neural network based on historical wind power and NWP information for regional wind power forecasting. To model the time dependencies of the wind power time series, many works leveraged recurrent neural networks and their variants (long short-term memory or gated recurrent unit) such as Liu et al. (2021) or Alkabbani et al. (2023). The interactions between several wind farms have been investigated using the Transformer model by Lima et al. (2022) and using graph neural networks by Qiu et al. (2024). The direct use of DNN directly on wind speed maps has been tackled using convolutional neural networks (CNNs) which have shown strong capabilities for extracting relevant features from image data. Higashiyama et al. (2018) used 3-dimensional CNNs to forecast the production of a single wind farm based on NWP grids. Bosma and Nazari (2022) and Jonkers et al. (2024) proposed day-ahead regional wind power forecasting CNNs whose architecture was inspired by Computer Vision models such as ResNet (see He et al. 2016).

The challenge of wind power forecasting is that it combines dependencies to weather variables but remains a time series. Therefore, architectures mixing various types of layers have been investigated to capture various dependencies. Miele et al. (2023) compared the performance of CNN-LSTM with a multimodal neural network with two branches: one for the NWP grid and one for past data, for a single wind farm. Zhou and Lu (2023) combined convolution, LSTM, and attention layers to forecast the production of a wind farm. Given this large variety of possible architectures, one might want to use automated tools to find the best one for the dataset at hand.

2.3. Automated deep learning

2.3.1. Main concepts

The research field related to the automation of deep neural network design is called Automated Deep Learning (AutoDL). It belongs to a more global research area called Automated Machine Learning (AutoML) which studies the automatic design of high-performance Machine Learning models. As with any AutoML approach, AutoDL systems consist of three main components: the *search space*, the *search strategy*, and the *performance evaluation*. The *search space* should contain all the considered neural network architectures and hyperparameters which is the set of all available design choices, like the number and type of layers in the neural network, the connection between the layers, or the training parameters, like the learning rate. The *search strategy* will determine how to navigate within the search space to select promising configurations. The bigger the search space, the more sophisticated the search strategy should be for effective exploration. The *performance evaluation* will assess the performance of the candidate configurations until the search strategy finds a suitable neural network (usually the best configuration found after a given number of evaluations).

2.3.2. AutoDL for wind power forecasting

A few works have applied AutoDL to wind power forecasting, such as Tu et al. (2022) or Jalali et al. (2022). However, these approaches are limited to optimizing the hyperparameters of one type of architecture, possibly integrating a few architectural hyperparameters such as the number of layers. The AutoDL community has developed a large number of tools to optimize neural network architectures

https://doi.org/10.1017/eds.2025.10 Published online by Cambridge University Press

more broadly, but as Tu et al. (2022) points out, the search spaces used by these approaches are tailored to Computer Vision and Natural Language Processing tasks. For example, Hutter et al. (2019) reviewed many approaches based on (hierarchical) cell-based search spaces, where the neural networks are represented as a sequence of small iterated Directed Acyclic Graphs (DAGs) called cells. The architecture of the cell is optimized and then the pattern is repeated throughout the network. Such an approach is efficient for Computer Vision tasks, where models that repeat sequences of convolutional pooling layers and skip connections are very powerful. Another popular approach is DARTS, proposed by Liu et al. (2018), which uses a meta-architecture that is designed to include all possible architectures. The general structure of the network is fixed, and for each layer several candidate operations are possible. Each is associated with a probability of being chosen, which is optimized by gradient descent. This approach, which is effective for generating architectures based on 3 × 3 or 5 × 5 convolutions, has a very limited search space and assumes that the subgraph obtained by keeping only the operation with the highest probability for each layer is the optimal graph. More diverse tasks have been tackled by the AutoDL framework AutoPytorch, which offers a version for tabular data, described in Zimmer et al. (2020), and for time series forecasting, see Deng et al. (2022), providing search spaces of MLPs and residual connections for the tabular version, and various encoder/decoder blocks for the time series version to cover several state-of-the-art architectures in time series (e.g., TFT from Lim et al., 2021, NBEATS from Oreshkin et al., 2019, or DeepAR from Salinas et al., 2020). All search spaces for the above AutoDL approaches have been restricted to allow effective searching. This observation is shared more generally by recent reviews such as White et al. (2023) on AutoDL and Baratchi et al. (2024) on AutoML. In the case of wind production forecasting, as indicated by Tu et al. (2022), we would like to have a search space for designing architectures that combine different types of layers such as MLPs, CNNs, or attention, that also have computational graphs that are more complex than a linearly sequential architecture, and whose hyperparameters can be optimized, as they are crucial in this type of task. The AutoDL package DRAGON, recently introduced in Keisler et al. (2024b), provides tools for designing such search spaces. The package has already been used to create EnergyDragon (see Keisler et al., 2024a), an AutoDL framework for forecasting load consumption.

2.4. DRAGON package

DRAGON, or DiRected Acyclic Graphs optimizatioN, is an open-source Python package[2] offering tools to conceive Automated Deep Learning frameworks for diverse tasks. The package is based on three main elements: building bricks for search space design, search operators for those bricks, and search algorithms.

2.4.1. Search space

DRAGON offers several building bricks to encode deep neural network architectures and hyperparameters. The network structures are represented as Directed Acyclic Graphs, where the nodes represent the layers and the edges the connection between them. The layers are encoded by a succession of three elements: a combiner, an operation, and an activation function. As no constraint is made on the graph structure, each node may receive an arbitrary number of incoming inputs of various sizes. They are gathered into a single input through the combiner. The operation can be any *PyTorch* building block parametrized by a set of hyperparameters. The DRAGON user has to specify which kind of building blocks the search space should contain, and for each, the associated hyperparameters. Besides the DAGs, the user can choose to optimize other hyperparameters such as the learning rate, the output shape of the last layer, etc. The hyperparameters may be numerical or categorical. The graph encoding can be used to represent the entire structure, but it is also possible to design more specific search spaces for certain applications. For example, it is possible to combine different graphs for a Transformer-type structure (see

[2] https://dragon-tutorial.readthedocs.io/en/latest/.

https://doi.org/10.1017/eds.2025.10 Published online by Cambridge University Press

Vaswani et al., 2017 for an introduction to the Transformer model), with one graph for the encoder part and another graph for the decoder part, in order to impose a two-part structure. In the process of creating an AutoDL framework based on DRAGON, the selection of appropriate building blocks from the package is essential for generating a suitable search space.

2.4.2. Performance evaluation

The search space has been designed for a specific performance evaluation strategy, which will assess the score of a given configuration from the search space. DRAGON does not provide any default performance evaluation, which depends on the task at hand. Therefore, it should be implemented within the created AutoDL framework. Given an element from the search space, the performance evaluation should at least build a model and perform any type of training/validation process on the data.

2.4.3. Search Operators

Each building block from DRAGON comes with a *neighbor* attribute that defines how to create a neighboring value from a representation. Those operators can be seen as mutations in the case of an evolutionary algorithm or neighborhood operators for a simulated annealing or a local search. In the case of an integer, for example, the *neighbor* attribute will pick the new value in a range surrounding the actual one. For the DAGs, it is possible to add or delete nodes, or to modify the edges and the node's contents.

2.4.4. Search Algorithms

The package implements several search strategies which may use the search operators and can be distributed in a high-performance computing (HPC) environment. Besides the random search, Hyperband (see Li et al. (2018)), an evolutionary algorithm and Mutant-UCB presented in Brégère and Keisler (2024) are available. They take as input the search space and the performance evaluation designed by the user and return the best configuration.

For more information on the DRAGON package see the original article Keisler et al. (2024b) or the documentation online[3].

3. WindDragon

We used the tools provided by DRAGON to create WindDragon, an AutoDL framework for regression on wind speed maps toward regional wind power forecasting. The framework takes as input two datasets $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{valid}}$. Each dataset $\mathcal{D}$ is made up of pairs (X_t, Y_t) for several time steps t, where $X_t \in \mathbb{R}^2$ is a wind speed map and $Y_t \in \mathbb{R}^R$ are the associated wind production values, one for each of the R regions. First, the framework creates wind speed maps by region r: X_t^r. Two datasets $\mathcal{D}^r_{\text{train}} = (X^r, Y^r)$ and $\mathcal{D}^r_{\text{valid}} = (X^r, Y^r)$ are put together for each region r with these regional wind speed maps and the associated regional production. WindDragon aims at finding, for each region r, the optimal model $\hat{f}^r$ from a search space Ω with respect to a loss function ℓ such that:

$$\hat{f}^r \operatorname*{arg\,min}_{f \in \Omega} \ell\left(f_{\hat{\delta}}, \mathcal{D}^r_{\text{valid}}\right), \tag{1}$$

where the model $f_{\hat{\delta}}$ corresponds to the model $f \in \Omega$ trained on $\mathcal{D}^r_{\text{train}}$.

3.1. Search space and performance evaluation

3.1.1. Data processing

The input data X_t contains the wind speed map corresponding to the whole country and has to be divided into regional data. As shown Figure 2 for a specific region (here Auvergne-Rhône-Alpes), wind turbines

[3] https://dragon-tutorial.readthedocs.io/en/latest/index.html.

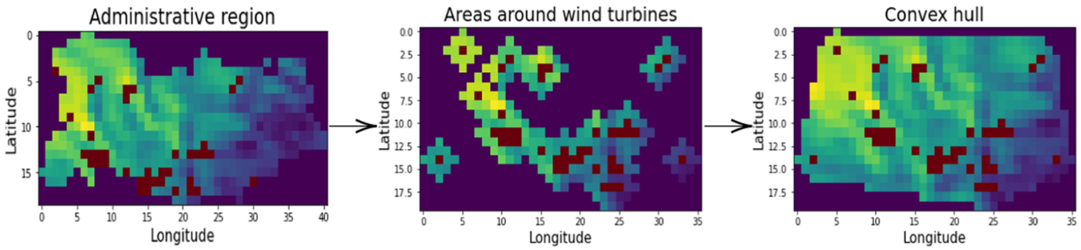


Figure 2. *Data preparation for the region Auvergne-Rhône-Alpes. The wind farms are represented in red. The first image shows the distribution of wind farms across the administrative region.*

are not evenly distributed across the administrative regions. Therefore, instead of using them, we draw areas around each wind farm in the region and took the convex hull of all the considered points. The result is a seamless map $X_t^r \subset X_t \in \mathbb{R}^2$ that includes local wind turbines with no gaps to disrupt the models. The areas surrounding the wind farms are drawn according to a distance parametrized by a parameter called $g \in \mathbb{N}^\star$. When g gets higher, the convex hull becomes larger. Installed capacity data—corresponding to the maximum wind power a region can produce—for each region and each time step t is available and updated every 3 months. It was collected and used to scale the wind power target to train the models. Training the model f on the region r with respect to the training loss ℓ_{train}, means finding the model optimal weights $\hat{\delta} \in \Delta$ such that:

$$\hat{\delta} \in \underset{\delta \in \Delta}{\operatorname{argmin}} \, \ell_{\text{train}} \left(f_\delta(X^r), \frac{Y^r}{c^r} \right), \tag{2}$$

where $c^r \in \mathbb{R}$ is the installed capacities for the region r and $\mathcal{D}^r_{\text{train}} = (X^r, Y^r)$. The evaluation of the model f on $\mathcal{D}^r_{\text{valid}}$ is made on the denormalized value Y^r.

3.1.2. Search space

Each model $f \in \Omega$ has to forecast a one-dimensional output $Y_t^r \in \mathbb{R}$ from a two-dimensional input: the wind speed map $X_t^r \in \mathbb{R}^2$. Therefore, each neural network from Ω is made of two Directed Acyclic Graphs as represented in Figure 3. A first graph Γ_1 processes 2D data and can be composed of convolutions, pooling, normalization, dropout, and attention layers. Then, a flattened layer and a second graph Γ_2 follow. This one is composed of MLPs, self-attention, convolutions, and pooling layers. A final

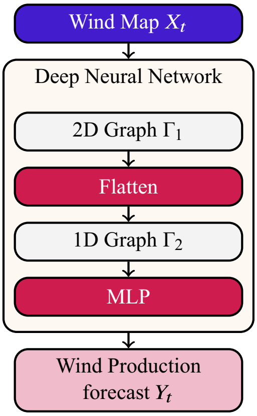


Figure 3. *WindDragon's meta-model for wind power forecasting.*

https://doi.org/10.1017/eds.2025.10 Published online by Cambridge University Press

Table 1. *Layers available and their associated hyperparameters in the WindDragon search space (for the first and the second graph)*

Layer type	Graph concerned	Optimized hyperparameters	
Identity	Both	-	
Fully connected (MLP)	Both	Output shape	Integer
Self-Attention	Both	Initialization type	[convolution, random]
		Heads number	Integer
		Output dimension	Integer
1D Convolution	1D Graph Γ_2	Kernel size	Integer
		Output dimension	Integer
2D Convolution	2D Graph Γ_1	Kernel size	Integer
		Output dimension	Integer
1D Pooling	1D Graph Γ_2	Pooling size	Integer
		Pooling type	[Max, Average]
2D Pooling	2D Graph Γ_1	Pooling size	Integer
		Pooling type	[Max, Average]
1D Normalization	1D Graph Γ_2	Normalization type	[Batch, Layer]
2D Normalization	2D Graph Γ_1	Normalization type	[Batch, Layer]
Dropout	Both	Dropout rate	Float

MLP layer is added at the end of the model to convert the latent vector to the desired output format. The detailed operations and hyperparameters available within WindDragon are detailed in Table 1. Regarding the parameters that are external to the architecture, the weather map size parameter g is also optimized. The search space is then: $[\Gamma_1, \Gamma_2, o, g]$ where o represents the final MLP layer, which is a constant.

3.1.3. Performance Evaluation

The performance evaluation takes as input a region r and a configuration from the search space and will:

- Construct the datasets $\mathcal{D}^r_{\text{train}}$ and $\mathcal{D}^r_{\text{valid}}$ from $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{valid}}$ according to the parameter g parameterizing the grid size, from the configuration.
- Build the model f^r with the elements from the configuration and train the model on $\mathcal{D}^r_{\text{train}}$ according to Equation (2).
- Evaluate the performance of $f^r_{\hat{\delta}}$ on $\mathcal{D}^r_{\text{valid}}$ according to Equation (1).

3.2. Search algorithm

Regarding the search algorithm, four are available within DRAGON: the Random Search, HyperBand (Li et al., 2018), an Evolutionary Algorithm, and Mutant-UCB. In Brégère and Keisler (2024) introducing this last algorithm, the four are compared and Mutant-UCB appears as the most efficient one.

3.2.1. Mutant-UCB

This algorithm combines a multi-armed bandits approach with evolutionary operators. Each model $f \in \Omega$ corresponds to an arm, an choosing arm corresponds to a partial training of the model. Indeed, training a neural network takes a lot of time, and a lot of algorithms such as the Random Search or the Evolutionary Algorithms give the same amount of resources for all the evaluated configurations. It means such algorithms are losing a lot of time and computational resources on bad configurations. Resource allocation strategies used for example by HyperBand, allows to gradually attribute resources to the most promising solutions. A partial training can then be, for example, a training on a small set of data or with a small number of epochs. In short, Mutant-UCB generates a population of $K \in \mathbb{N}^\star$ of random configurations. For each arm k from this population, a partial training is made to get a first loss ℓ_k. Then, at each iteration i, an arm I_i from the population is drawn following an Upper-Confidence-Bound strategy:

$$I_i \in \underset{k \in \{1,\dots K\}}{\operatorname{argmin}} \left\{ \widehat{\ell}_k - \sqrt{\frac{E}{N_k}} \right\},$$

where $\hat{\ell}_k$ is the average loss for all the previous partial training of the model associated to the arm k, E is the exploration parameters and N_k the number of times the arm k has been picked. Once the arm I_i is chosen, with a probability $1 - \overline{N}_{I_i}/N$, the model is mutated. Otherwise, a new partial training is done. The value N corresponds to the maximum number of partial training a model can have (to prevent overfitting) and $\overline{N}_{I_i}$ corresponds to the number of times the model associated to I_i has been trained. In the case of a mutant creation, the number of arms K increases, and the new model is partially trained for the first time. For more information on Mutant-UCB please refer to Brégère and Keisler (2024).

3.2.2. Partial training

In the original article, the partial training were done on a small number of epochs. For WindDragon, we changed it to be a small number of epochs on a given region. Instead of running one version of Mutant-UCB, we performed one optimization for all regions. We indeed make the assumption that a similar architecture will fit for all the regions, even if some layers or hyperparameters might change from one region to another. The input X^r might be of different shapes for different regions. This shape change is handled by DRAGON when building the neural network f. The layers and DAGs from the package may be adapted by weight cropping or padding to any new shape during the network initialization. Splitting the training between different regions follows the spirit of Mutant-UCB, where the loss minimized to pick the future arm relies on the empirical mean of the various partial trainings of a model f. The performance across the regions might be different, and converging towards a model generally good over all regions can be done by taking this empirical mean. To reduce the variance between the performance of the region, the loss ℓ considered to evaluate a model f on a given region would be an error function (such as the mean squared error, the mean absolute error or a variant) of f, divided by this same error function but of a reference model. See Section 4 for more information.

4. Experiments

4.1. Datasets

The wind speed maps used are 100-m high forecasts at a 9 km resolution provided by the HRES[4] model from the European Centre for Medium-Range Weather Forecasts (ECMWF). The maps are provided at an hourly time step and there are four forecast runs per day (every 6 h). Only the six more recent forecasts are used here as the forecasting horizon of interest is 6 h. The hourly French regional and national wind power generation data as well as the French TSO hourly forecasts and the installed capacities values come from the ENTSOE-E Transparency Platform[5].

4.2. Baselines

We use the following baselines to compare hourly forecasts for a horizon h $(h \in \{1,\dots,6\})$:

- **Persistence**: Given access to forecasts every 6 h derived from the ground truth situation, the wind power value is also available at the same intervl. Persistence involves replicating this value for the

[4] https://www.ecmwf.int/en/forecasts/datasets/set-i.

[5] https://transparency.entsoe.eu/.

subsequent 6 h. Therefore, the model predicts wind power generation at future times $t+h$ as equal to the observed generation at the current time *t*.

- **XGB on Wind Speed Mean**: Forecasts wind power at $t+h$ using a two-step approach as depicted Figure 4: (i) Compute the mean wind speed for the considered region at $t+h$ using NWP forecasts. (ii) Apply an XGBoost regressor (Chen and Guestrin, 2016) to predict power generation based on the computed mean wind speed.
- **Convolutional Neural Networks (CNNs)**. Use the same training setup as WindDragon: forecasts wind power at $t+h$ using the NWP forecasted wind speed map. CNNs can efficiently regress a structured map on a numerical value by learning local and spatial patterns (LeCun et al., 1995). In addition, the weight sharing induced by the convolutional mechanism reduces the number of learned weights compared to alternative deep learning mechanisms like dense (Haykin, 1994) or self-attention layers (Vaswani et al., 2017). This feature makes CNNs particularly effective when dealing with relatively small amounts of data. Figure 5 shows the architecture of the CNN baseline we implemented. We used a simple grid search to optimize the hyperparameters (e.g., the number of layers, the kernel sizes, the activation functions).
- **French TSO (RTE).** The European TSOs have to provide *Current*, *IntraDay*, and *Day-Ahead* wind and solar forecasts. We have used the *Current* forecast within our baseline to put the results into perspective with operational values. The forecasting methods and horizons are not detailed. The regulatory article[6] only states that the published "Current" forecast is the latest update of the forecast. The information is regularly updated and published during intra-day trading. It is the closest setup from our experiments.

4.2. *Experimental setup*

We used the years from 2018 to 2019 to train the models, and the data from 2020 is used to evaluate how the models perform. All the neural networks were trained using the Adam optimizer. The CNN was

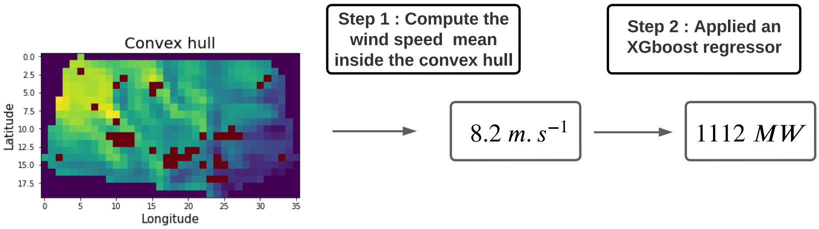

Convex hull
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Latitude
0
5
10
15
20
25
30
35
Longitude
Step 1 : Compute the wind speed mean inside the convex hull
Step 2 : Applied an XGboost regressor
$8.2\ m.s^{-1}$
$1112\ MW$

Figure 4. *Visual illustration of the XGB two-step approach on the Auvergne-Rhône-Alpes region.*

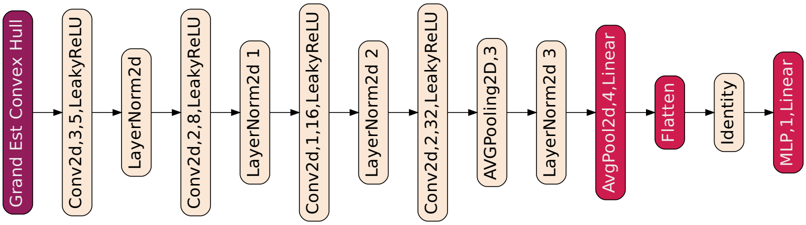


Figure 5. *CNN architecture applied to the Grand Est region.*

[6] https://transparencyplatform.zendesk.com/hc/en-us/articles/16648445340180-Generation-Forecasts-for-Wind-and-Solar-14-1-D.

https://doi.org/10.1017/eds.2025.10 Published online by Cambridge University Press

trained for 200 epochs. Mutant-UCB was parametrized with $N=10$, $K=600$, $E=0.01$ and 20 epochs by partial training. The CNN model was given as input to the search algorithm. Among the first K models initialized, 10 had the CNN architecture, with values of g ranging from 1 to 10. The CNN losses were used to scale the regional errors for WindDragon. Mutant-UCB was distributed over 20 V100 GPUs and ran for 72 h.

4.3. Results

We computed two scores: **Mean Absolute Error (MAE)** in Megawatts (MW), showing the absolute difference between ground truth and forecast, and **Normalized Mean Absolute Error (NMAE)**, a percentage obtained by dividing the MAE by the average wind power generation for the test year. The MAE gives an idea of the amount of energy contained in the errors, while the NMAE enables performance to be compared between regions. We run experiments for each of the 12 French metropolitan regions and then aggregate the forecasts to derive national results. Let us have $\hat{y}^r_{t,m}$ the forecast of the baseline m on the region r at time t. We get the national forecast $\hat{Y}_m = \left\{\hat{y}_{t,m}\right\}_{t=1}^{N}$ by aggregating the forecasts of the 12 French metropolitan regions:

$$\hat{y}_{t,m} = \sum_{r=1}^{12} \hat{y}^r_{t,m}.$$

Then, the national metrics for each baseline m are retrieved between the national value Y and the national forecast of this baseline: $\hat{Y}_m$. The national results are presented in Table 2, while detailed regional results can be found in Table 3. It is interesting to note that the sum of the regional errors is greater than the national error for each model. This is due to the fact that the regional errors offset each other when the signals are aggregated.

The results in Table 2 highlight three key findings:

i. **Improved performance with aggregated NWP statistics.** Using the average of NWP-predicted wind speed maps coupled with an XGB regressor significantly outperforms the naive persistence baseline. It shows that the signal is closer to a regression problem than to a time series forecasting one. It is also interesting to note that this simple model is already better than the signal produced by the French TSO.

ii. **Gains from full NWP map utilization**. More complex patterns can be captured by using the full predicted wind speed map, as opposed to just the average, thereby improving forecast accuracy. In this context, the CNN regressor applied to full maps yielded gains of 47 MW (11.5%) over the mean-based XGB.

iii. **WindDragon's superior performances**. WindDragon outperforms all baselines, showing an improvement of 69 MW (19%) over the CNN. On an annual basis, this corresponds to approximately 600 GWh. The average French citizen consumes between 2500 and 3000 kWh[7] of

Table 2. *National results: metrics computed on the aggregation of the regional forecasts for each model. The best results are highlighted in bold and the best second results are underlined*

	WindDragon		RTE		CNN		XGB on mean		Persistence	
	MAE (MW)	NMAE	MAE (MW)	NMAE	MAE (MW)	NMAE	MAE (MW)	NMAE	MAE (MW)	NMAE
France	**300.0**	**6.6%**	482.1	10.6%	369.0	8.1%	416.7	9.2%	779.7	17.3%

[7] Based on the average European per capita consumption [Statista Research Department, 2022].

Table 3. *Regional results. The best results are highlighted in bold and the best second results are underlined*

	WindDragon		CNN		XGB on mean		Persistence	
Region	MAE (MW)	NMAE	MAE (MW)	NMAE	MAE (MW)	NMAE	MAE (MW)	NMAE
Auvergne-Rhône-Alpes	**19.3**	**14.8%**	19.6	15.0%	29.2	22.4%	28.7	22.0%
Bourgogne-Franche-Comté	**30.0**	**13.6%**	34.1	15.4%	42.3	19.1%	58.7	26.6%
Bretagne	**33.7**	**13.2%**	38.0	14.9%	47.1	18.4%	67.2	26.3%
Centre-Val de Loire	**50.5**	**14.2%**	57.3	16.1%	61.9	17.5%	96.7	27.3%
Grand Est	**108.2**	**10.8%**	130.5	13.1%	148.8	14.9%	251.2	25.1%
Hauts-de-France	**140.7**	**10.6%**	167.6	12.7%	178.8	13.5%	320.1	24.2%
Île-de-France	**6.2**	**20.5%**	7.2	23.7%	7.5	24.9%	9.5	31.5%
Normandie	**27.4**	**11.8%**	30.8	13.2%	36.8	15.8%	55.9	24.0%
Nouvelle-Aquitaine	**37.8**	**13.8%**	44.0	16.4%	53.7	19.6%	77.9	28.4%
Occitanie	**51.1**	**12.3%**	55.8	13.5%	91.6	22.1%	96.3	23.2%
PACA	**3.2**	**29.7%**	3.5	32.4%	4.5	41.4%	4.3	39.5%
Pays de la Loire	**34.1**	**12.5%**	39.0	14.3%	41.9	15.4%	74.9	27.5%

electricity per year. Therefore, 600 GWh per year is equivalent to the consumption of around 200,000 French inhabitants. The results underscore WindDragon's effectiveness in autonomously discovering the optimal deep-learning configurations for wind power regression. Moreover, Table 3 indicates that the improvement is effective in all regions. During optimization, WindDragon managed to find, for each region, a model that outperformed each other from the baseline. The architectures found vary a bit from one region to another. Examples of the models produced by WindDragon for various regions can be found Figures A1–A5 The architectures mix various layers such as convolutions, pooling, and normalization layers. The structures are, for the majority, composed of a large two-dimensional graph, efficiently extracting spatial information from the input wind speed map and a small one-dimensional graph. The hyperparameters are however unique for each model.

4.4. Forecasts comparison

In Figure 6, we present the aggregated national wind power forecasts using both WindDragon and the CNN baseline during a given week. While both models deliver highly accurate forecasts, it is important to highlight that DRAGON demonstrates superior accuracy, particularly during the high production level at the end of the signal. Figure A6 shows visual comparisons of all baseline performances on this same week. It appears that the models perform well at different times. For example, the RTE forecast is best for the small production spike in the middle of the day on 11 January, but worst for the production dip on the night of 10 January. These differences in performance open the way to mixtures of models to further improve forecasts.

4.5. Performance analysis

We compared the performance of the two best baselines, CNN and WindDragon, in more detail. Figure 7 shows the absolute errors and the normalized absolute errors by hour of the day and by month. In general, WindDragon is significantly better than CNN at all times of the day and for all months. In Figure 7a,b, the

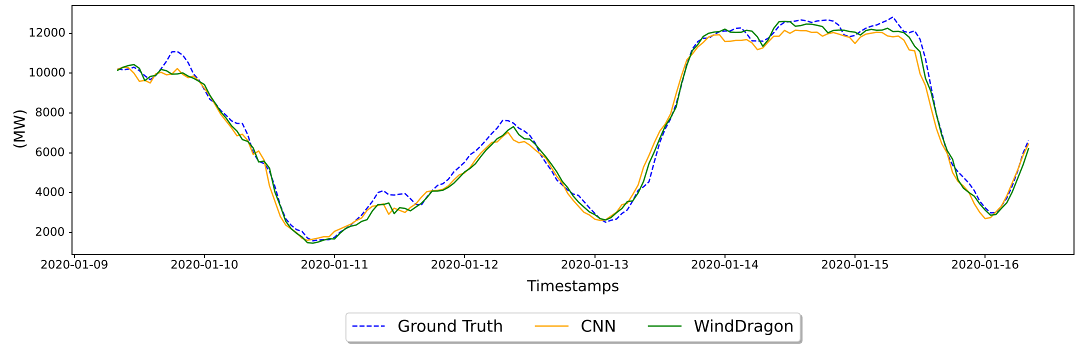
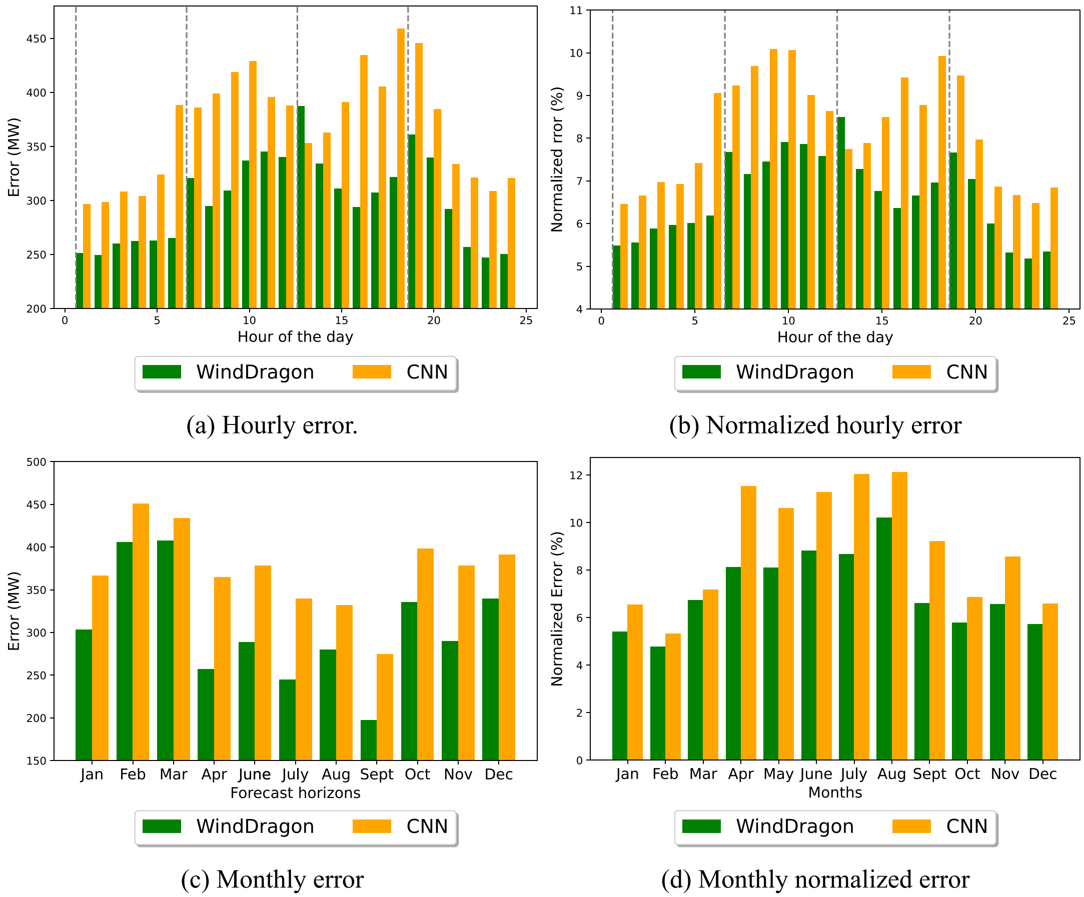


Figure 6. *Wind power forecasts for a week in January 2020. The figure displays the ground truth as dotted lines, and the forecasts from the two top-performing models, WindDragon and CNN.*



(a) Hourly error.

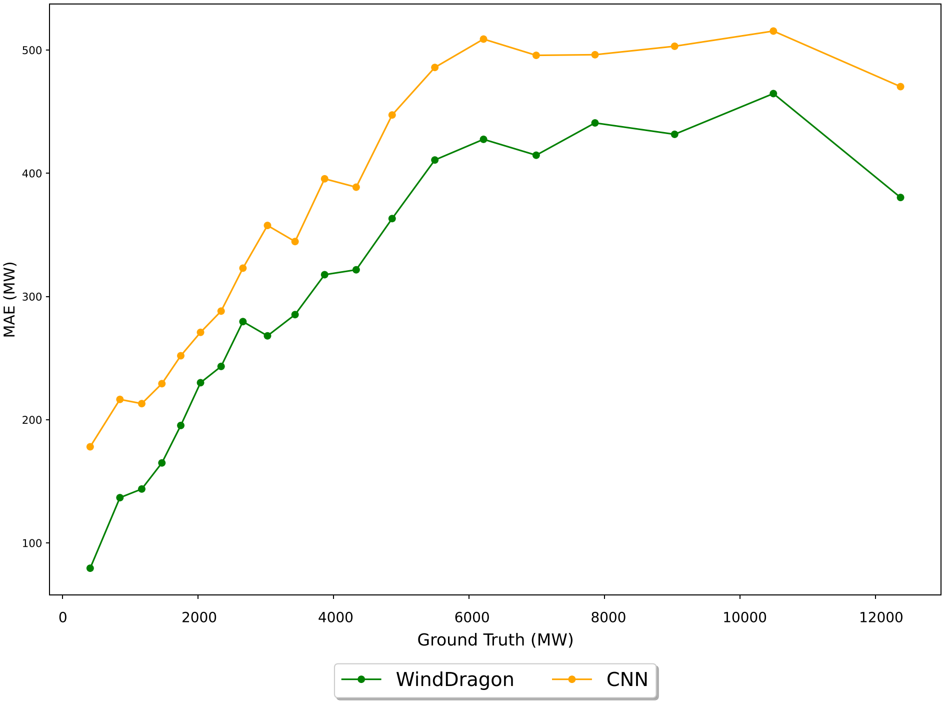
(b) Normalized hourly error

(c) Monthly error

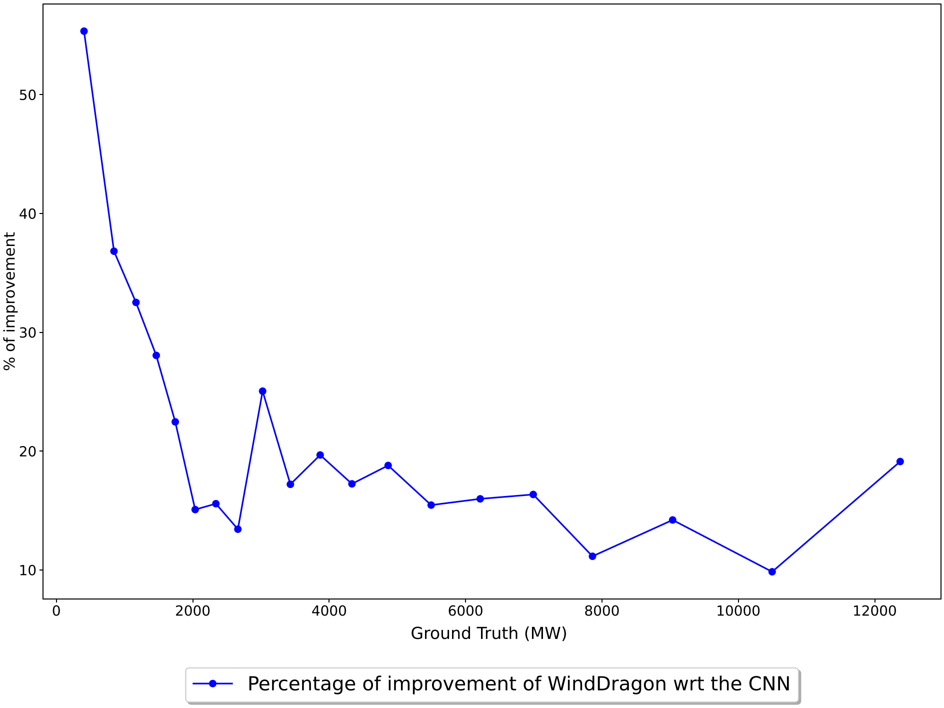
(d) Monthly normalized error

Figure 7. *Errors comparison between WindDragon and the CNN. The dotted vertical lines in Figure 7a,b represent the beginning of the new NWP forecast.*

dotted line represents the hour when a new NWP forecast arrives (every 6 h). For the first two forecasts of the day (at midnight and 6 a.m.), the performance of both models decreases as the forecast horizon increases. This is much more marked in the case of CNN, whose performance deteriorates dramatically,



(a) MAE repartition of the CNN and WindDragon over 20 quantiles.



(b) Percentage improvement of WindDragon compared to CNN over 20 quantiles.

Figure 8. *Comparison of the CNN and WindDragon performance over 20 quantiles. The two figures show WindDragon's superiority over CNN over the entire distribution, but particularly over the distribution tails.*

https://doi.org/10.1017/eds.2025.10 Published online by Cambridge University Press

particularly at 6 a.m. (when the forecast horizon is therefore 6 h). This observation is less true for the later hours of the day. As for the months, the differences are more pronounced in summer, when wind power production is lower. Finally, we have plotted Figure 8a the mean absolute errors of CNN and WindDragon per quantile of the wind power distribution. We can see from this distribution that the two curves diverge particularly at the first quantile, where the production values are extremely low, and at the last quantile, where they are extremely high. The two curves never cross, demonstrating the homogeneous superiority of WindDragon over CNN. Figure 8b shows the skill score between the MAE of WindDragon and the MAE of the reference model, the CNN, which confirms the impression given by Figure 8a.

4.6. WindDragon search algorithm (Mutant-UCB) time convergence

Mutant-UCB ran for 72 h on 20 GPUs. However, we saved the losses of the models found by the algorithm as it ran so that we could analyze its convergence time. Figure 9a shows the best NMAE found per time step for each region. We can see that the performance converges very quickly during the first 2 h of the algorithm before stabilizing. Only a few regions such as Ile-de-France, Auvergne-Rhône-Alpes, and Centre-Val de Loire show improvements in the last hours. Figure 9b zooms in on the first 3 h of the algorithm. Except for PACA and Ile-de-France, most regions fall below 15% of NMAE in about an hour. Thus, although Mutant-UCB has run for a long time to achieve very good performance, it was possible to obtain correct models in just 1 h.

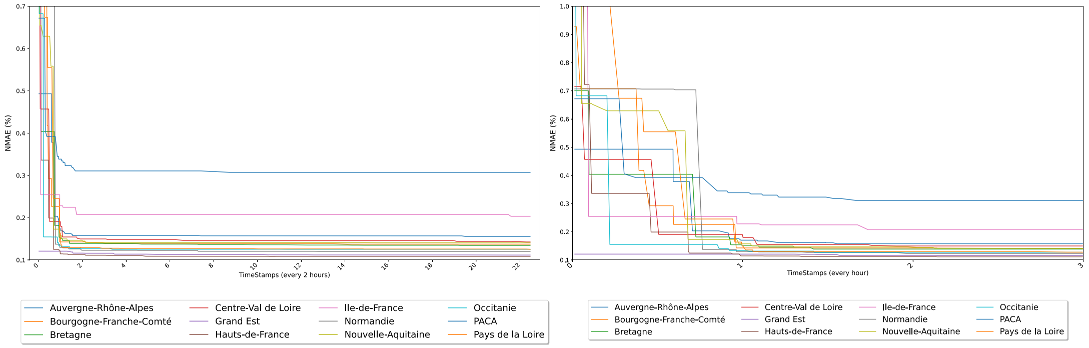
5. Conclusion and impact statement

5.1. Summary

This article presents WindDragon, an Automated Deep Learning framework for forecasting regional wind power. WindDragon automated the creation of performing Deep Neural Networks leveraging Numerical Weather Prediction wind speed maps to deliver wind production forecasts. We demonstrate on the French national and regional wind production data that WindDragon can find deep neural networks outperforming traditional and state-of-the-art deep learning models in regional wind power forecasting. Compared to the handcrafted deep learning model inspired by the state of the art in computer vision, WindDragon allows us to find models that perform particularly well in winter and at high wind values, which is all the more interesting in the context of wind power forecasting.

5.2. Limitations

WindDragon, like many AutoML systems, is limited by its high running time compared to handcrafted baselines. However, this duration should be compared to the time spent creating powerful models by



(a) NMAE through time for each region.

(b) NMAE through time: zoom on the 3 first hours

Figure 9. *WindDragon search algorithm (Mutant-UCB) convergence: NMAE through time for each region.*

https://doi.org/10.1017/eds.2025.10 Published online by Cambridge University Press

hand, which is often hard to measure. Besides, once the model has been found, the inference speed remains competitive with other deep learning models. However, future study could focus on reducing this running training time through even more efficient search algorithms or reducing the search space. This gained efficiency could also be achieved by reducing the input weather map dimension, for example, using unsupervised representation techniques. The high number of model training and evaluations could be leveraged by creating a mix of models instead of just identifying the best one by region. Section 4 highlighted that the baseline models produced quite different forecasts. These differences, if complementary, could enable a mix of models to achieve better performance.

5.3. Future study

Finally, with the rise of data-driven weather forecasting tools, the accuracy of weather forecasting has increased at various forecast horizons (Ben Bouallègue et al., 2024) and for multiple weather variables. With its non-dependency on past data, our methodology could easily be applied to longer forecast horizons (to be used for other industrial use cases) but also for photovoltaic (PV) regional forecasting, by applying it to solar radiation maps generated by NWP models.

Acknowledgments. We are grateful for the technical advice and careful proofreading of Ghislain Agua and Yannig Goude.

Author contribution. Conceptualization: J. K.; E. L. N., Methodology: J. K.; E. L. N., Data curation: J. K.; E. L. N., Data visualization: J. K.; E. L. N., Writing original draft: J. K.; E. L. N., All authors approved the final submitted draft.

Competing interest. The authors declare none.

Data availability statement. We use open-source data for wind power generation given by the French TSO: https://www.rte-france.com/eco2mix. However, NWP maps are not open source.

Ethics statement. The research meets all ethical guidelines, including adherence to the legal requirements of the study country.

Funding statement. This research was supported by grants from EDF (Electricité de France).

References

Alkabbani H, **Hourfar F**, **Ahmadian A**, **Zhu Q**, **Almansoori A and Elkamel A** (2023) Machine learning-based time series modelling for large-scale regional wind power forecasting: A case study in Ontario, Canada. *Cleaner Energy Systems 5*, 100068.

Baratchi M, **Wang C**, **Limmer S**, **van Rijn JN**, **Hoos H**, **Bäck T and Olhofer M** (2024) Automated machine learning: Past, present and future. *Artificial Intelligence Review 57*(5), 1–88.

Bouallègue ZB, **Clare MC**, **Magnusson L**, **Gascon E**, **Maier-Gerber M**, **Janoušek M**, **Rodwell M**, **Pinault F**, **Dramsch JS**, **Lang ST**, et al. (2024) The rise of data-driven weather forecasting: A first statistical assessment of machine learning–based weather forecasts in an operational-like context. *Bulletin of the American Meteorological Society 105*(6), E864–E883.

Bosma SB and Nazari N (2022) Estimating solar and wind power production using computer vision deep learning techniques on weather maps. *Energy Technology 10*(8), 2200289.

Brégère M and Keisler J (2024) A bandit approach with evolutionary operators for model selectionarXiv preprint arXiv: 2402.05144.

Camal S, **Girard R**, **Fortin M**, **Touron A and Dubus L** (2024) A conditional and regularized approach for large-scale spatiotemporal wind power forecasting. *Sustainable Energy Technologies and Assessments 65*, 103743.

Chen T and Guestrin C (2016) Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, pp. 785–794.

Davò F, **Alessandrini S**, **Sperati S**, **Monache LD**, **Airoldi D and Vespucci MT** (2016) Post-processing techniques and principal component analysis for regional wind power and solar irradiance forecasting. *Solar Energy 134*, 327–338.

Deng D, **Karl F**, **Hutter F**, **Bischl B and Lindauer M** (2022) Efficient automated deep learning for time series forecasting. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 664–680.

Haykin S (1994) *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR.

He K, **Zhang X**, **Ren S and Sun J** (2016) Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.

Higashiyama K, **Fujimoto Y and Hayashi Y** (2018) Feature extraction of nwp data for wind power forecasting using 3d-convolutional neural networks. *Energy Procedia 155*, 350–358.

Hutter F, **Kotthoff L and Vanschoren J** (2019) *Automated Machine Learning: Methods, Systems, Challenges*. Springer Nature.

International Energy Agency (IEA). *Wind Power Generation*, 2023. https://www.iea.org/energy-system/renewables/wind. IEA, Paris.

https://doi.org/10.1017/eds.2025.10 Published online by Cambridge University Press

Jalali SMJ, **Ahmadian S**, **Khodayar M**, **Khosravi A**, **Shafie-khah M**, **Nahavandi S and Catalao JP** (2022) An advanced short-term wind power forecasting framework based on the optimized deep neural network models. *International Journal of Electrical Power & Energy Systems 141*, 108143.

Jonkers J, **Avendano DN**, **Van Wallendael G and Van Hoecke S** (2024) A novel day-ahead regional and probabilistic wind power forecasting framework using deep cnns and conformalized regression forests. *Applied Energy 361*, 122900.

Kariniotakis G (2017) *Renewable Energy Forecasting: From Models to Applications*. Woodhead Publishing.

Keisler J, **Claudel S**, **Cabriel G and Brégère M** (2024a) Automated deep learning for load forecasting. In *Proceedings of the Third International Conference on Automated Machine Learning, Volume 256 of Proceedings of Machine Learning Research*. PMLR, 16/1–28.

Keisler J, **Talbi E-G**, **Claudel S and Cabriel G** (2024b) An algorithmic framework for the optimization of deep neural networks architectures and hyperparameters. *Journal of Machine Learning Research 25*(201), 1–33.

LeCun Y, **Bengio Y**, et al. (1995) Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks 3361*(10), 1995.

Li L, **Jamieson K**, **DeSalvo G**, **Rostamizadeh A and Talwalkar A** (2018) Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research 18*(185), 1–52.

Lim B, **Arık SÖ**, **Loeff N and Pfister T** (2021) Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting 37*(4), 1748–1764.

Lima F, **Ren TI and Costa A** (2022) Wind power forecast based on transformers and clustering of wind farms with temporal and spatial interdependence. In *International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp. 01–06.

Liu H, **Simonyan K, and Yang Y** (2018) Darts: Differentiable architecture search. In *International Conference on Learning Representations*.

Liu X, **Zhou J and Qian H** (2021) Short-term wind power forecasting by stacked recurrent neural networks with parametric sine activation function. *Electric Power Systems Research 192*, 107011.

Lobo MG and Sanchez I (2012) Regional wind power forecasting based on smoothing techniques, with application to the spanish peninsular system. *IEEE Transactions on Power Systems 27*(4), 1990–1997.

Miele ES, **Ludwig N and Corsini A** (2023) Multi-horizon wind power forecasting using multi-modal spatio-temporal neural networks. *Energies 16*(8), 3522.

Oreshkin BN, **Carpov D**, **Chapados N and Bengio Y** (2019) N-beats: Neural basis expansion analysis for interpretable time series forecastingarXiv preprint arXiv:1905.10437.

Pinson P, **Siebert N and Kariniotakis G** (2003) Forecasting of regional wind generation by a dynamic fuzzy-neural networks based upscaling approach. In *EWEC 2003 (European Wind Energy and Conference)*, 5.

Qiu H, **Shi K**, **Wang R**, **Zhang L**, **Liu X and Cheng X** (2024) A novel temporal–spatial graph neural network for wind power forecasting considering blockage effects. *Renewable Energy 227*, 120499.

Salinas D, **Flunkert V**, **Gasthaus J and Januschowski T** (2020) Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting 36*(3), 1181–1191.

Siebert N (2008) *Development of Methods for Regional Wind Power Forecasting*. Theses, École Nationale Supérieure des Mines de Paris. https://pastel.hal.science/tel-00287551.

Statista Research Department (2022) *Europe: Electricity Demand Per Capita*. https://www.statista.com/statistics/1262471/per-capita-electricity-consumption-europe/.

Tu R, **Roberts N**, **Prasad V**, **Nayak S**, **Jain P**, **Sala F**, **Ramakrishnan G**, **Talwalkar A**, **Neiswanger W and White C** (2022) Automl for climate change: A call to actionarXiv preprint arXiv:2210.03324.

United Nations Convention on Climate Change. Paris Agreement. *Climate Change Conference (COP21)*, 2015. https://unfccc.int/sites/default/files/english_paris_agreement.pdf.

Vaswani A, **Shazeer N**, **Parmar N**, **Uszkoreit J**, **Jones L**, **Gomez AN**, **Kaiser Ł and Polosukhin I** (2017) Attention is all you need. *Advances in Neural Information Processing Systems 30*.

Wang Z, **Wang W and Wang B** (2017) Regional wind power forecasting model with nwp grid data optimized. *Frontiers in Energy 11*, 175–183.

Wang Z, **Wang W**, **Liu C**, **Wang B and Feng S** (2018) Short-term probabilistic forecasting for regional wind power using distance-weighted kernel density estimation. *IET Renewable Power Generation 12*(15), 1725–1732.

Wang Z, **Wang W**, **Liu C and Wang B** (2019) Forecasted scenarios of regional wind farms based on regular vine copulas. *Journal of Modern Power Systems and Clean Energy 8*(1), 77–85.

White C, **Safari M**, **Sukthanker R**, **Ru B**, **Elsken T**, **Zela A**, **Dey D and Hutter F** (2023) Neural architecture search: Insights from 1000 papersarXiv preprint arXiv:2301.08727.

Yu Y, **Yang M**, **Han X**, **Zhang Y and Ye P** (2021) A regional wind power probabilistic forecast method based on deep quantile regression. *IEEE Transactions on Industry Applications 57*(5), 4420–4427.

Zhou L and Lu R (2023) Attention-based convolutional neural network-long short-term memory network wind power forecasting. In *2003 3rd New Energy and Energy Storage System Control Summit Forum (NEESSC)*. IEEE, pp. 294–297.

Zimmer L, **Lindauer M and Hutter F** (2020) Auto-pytorch tabular: Multi-fidelity metalearning for efficient and robust autodl. *CoRR*, abs/2006.13799. URL https://arxiv.org/abs/2006.13799.

https://doi.org/10.1017/eds.2025.10 Published online by Cambridge University Press

A. Appendix

A.1. Models found by WindDragon for various regions

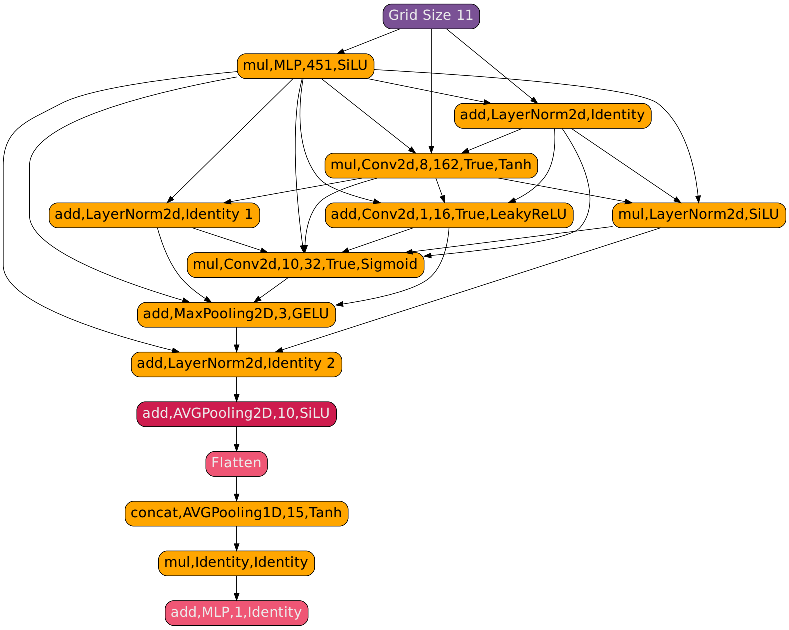


Figure A1. *Architecture found by WindDragon on Grand Est.*

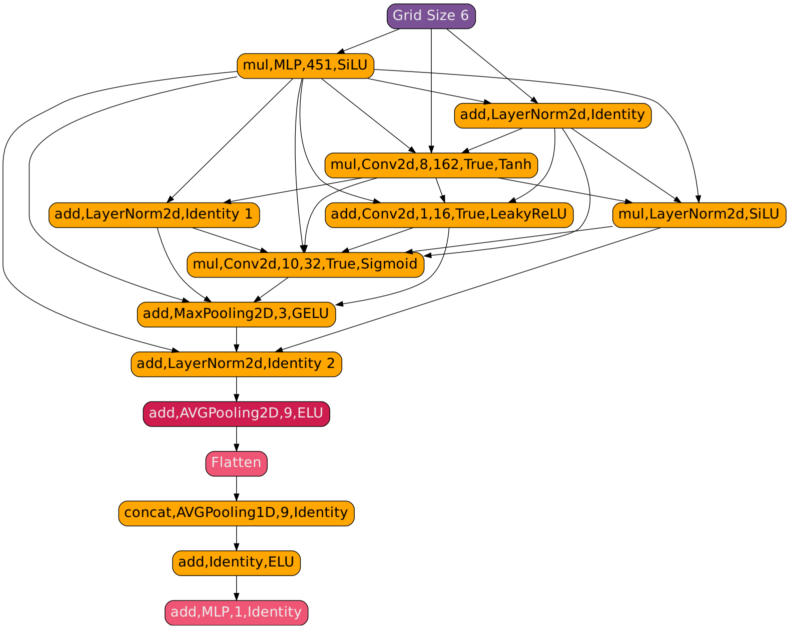

Grid Size 6
mul,MLP,451,SiLU
add,LayerNorm2d,Identity
mul,Conv2d,8,162,True,Tanh
add,LayerNorm2d,Identity 1
add,Conv2d,1,16,True,LeakyReLU
mul,LayerNorm2d,SiLU
mul,Conv2d,10,32,True,Sigmoid
add,MaxPooling2D,3,GELU
add,LayerNorm2d,Identity 2
add,AVGPooling2D,9,ELU
Flatten
concat,AVGPooling1D,9,Identity
add,Identity,ELU
add,MLP,1,Identity

Figure A2. *Architecture found by WindDragon on Auvergne-Rhône-Alpes.*

https://doi.org/10.1017/eds.2025.10 Published online by Cambridge University Press

A.2. Forecasts comparison

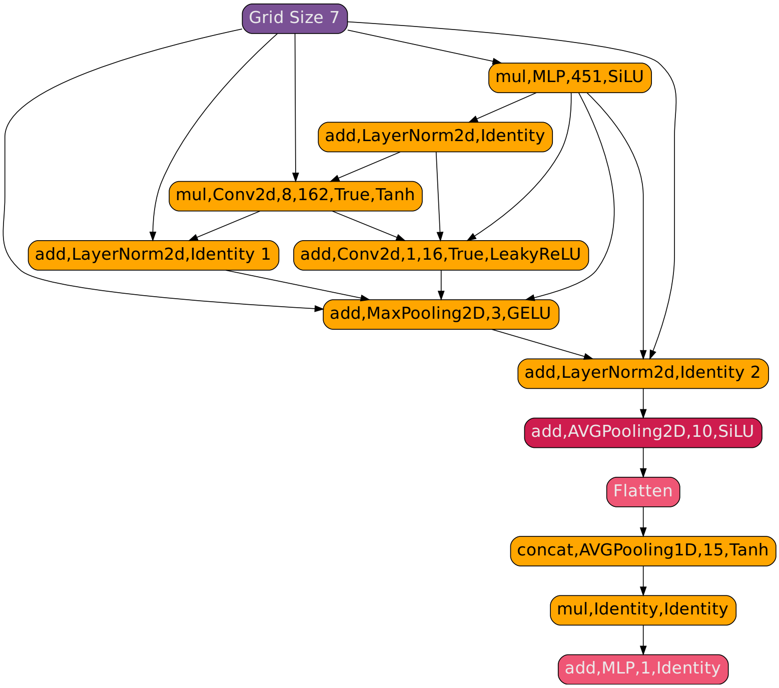


Figure A3. *Architecture found by WindDragon on Hauts-de-France.*

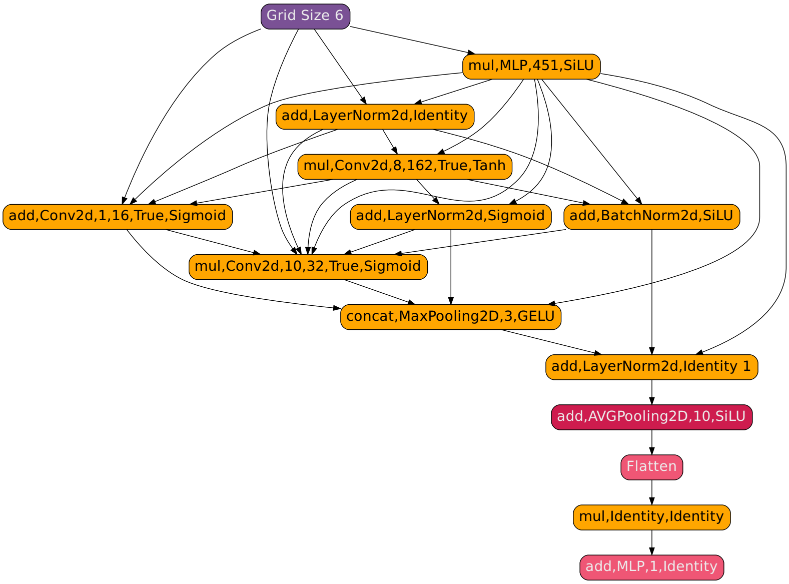


Figure A4. *Architecture found by WindDragon on Île-de-France.*

https://doi.org/10.1017/eds.2025.10 Published online by Cambridge University Press

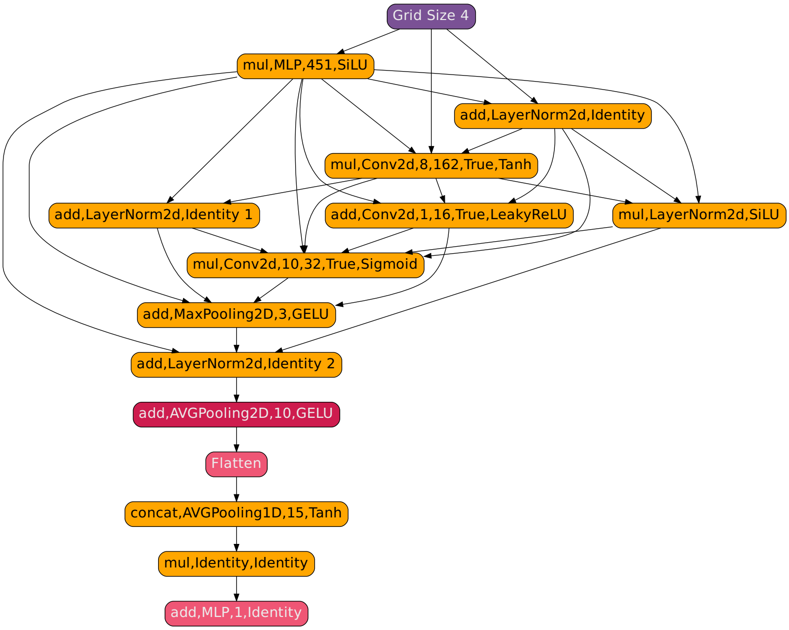
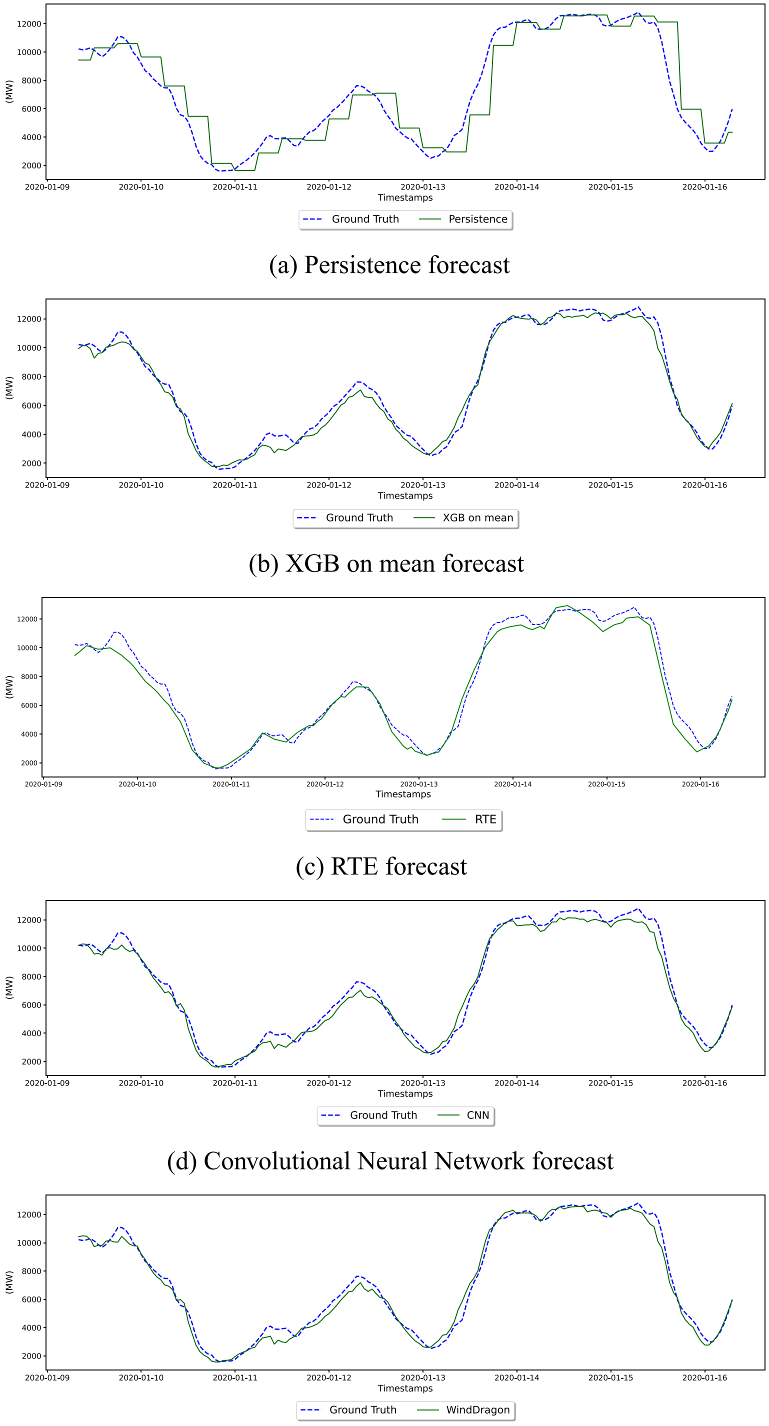

Grid Size 4
mul,MLP,451,SiLU
add,LayerNorm2d,Identity
mul,Conv2d,8,162,True,Tanh
add,LayerNorm2d,Identity 1
add,Conv2d,1,16,True,LeakyReLU
mul,LayerNorm2d,SiLU
mul,Conv2d,10,32,True,Sigmoid
add,MaxPooling2D,3,GELU
add,LayerNorm2d,Identity 2
add,AVGPooling2D,10,GELU
Flatten
concat,AVGPooling1D,15,Tanh
mul,Identity,Identity
add,MLP,1,Identity

Figure A5. *Architecture found by WindDragon on Occitanie.*

https://doi.org/10.1017/eds.2025.10 Published online by Cambridge University Press



(a) Persistence forecast

(b) XGB on mean forecast

(c) RTE forecast

(d) Convolutional Neural Network forecast

(e) WindDragon forecast

Figure A6. Weekly comparative visuals.

Cite this article: Keisler J and Le Naour E (2025). WindDragon: automated deep learning for regional wind power forecasting. *Environmental Data Science*, 4: e19. doi:10.1017/eds.2025.10