SHORT TERM FRANCE WIND PRODUCTION FORECASTING BASED ON METEOROLOGICAL OPEN-SOURCE DATA AND DEEP LEARNING

Hugo Antoine François HAMBURGER – s212364
Supervised by Ole WINTHER - DTU Compute, Technical University of Denmark

GitHub Repository : https://github.com/Paewynn/02456-DEEP-LEARNING-s212634
Google Drive Repository : Drive

ABSTRACT

The main contribution of this project is to partially explore a Dataset that has never been used for machine learning and that consist of a higher number of lower quality data compared to what is usually used. Several conventional model structures for short term wind load factor forecasting (4 hours ahead) were tested and a structure composed of Convolutional, LSTM, Dense and Regularization layers is the one that perform the best.

1. INTRODUCTION

The idea of the project is to use of a huge quantity of non-professional French weather data which allows a strong granularity (as there is a lot of dispersed stations) instead of being restricted to a few weather stations from the national weather agency (~50) to forecast national Wind Energy production on a short term (H+1, H+2, H+3, H+4).

The data consist of temperature, sky visibility, humidity, wind chill, maximum wind speed, pressure and location with an hourly time step (provided by ~800 stations dispersed unequally over the French territory). Each of these stations can have missing data and didn't measure necessarily all the mentioned parameters.

2. DATA PROCESSING

A first step was to limit the number of data used to keep a reasonable computing burden. Here I chose to only consider wind speed to limit the computing burden and as most of the wind production forecasting projects use only wind speed data. If an expansion is possible, the other interesting data would probably be maximum wind speed, pressure, and temperature as they all impact the wind production. This is also the case in reference [3] at the exception of the maximum wind speed that was not investigated. Another restriction was to use approximately 8 months of data to cover the seasonality in the training Data. Using a "short" temporal depth for the data also allows to avoid the handling of the technological improvement of the wind turbines and some climatic cycles that impact the wind.

A second step was to deal with the missing values. Here I was inspired by the reference [4] and choose to not consider the stations with to much missing data and to estimate the remaining missing data by using sklearn IterativeImputer function. It results in the restriction to 178 stations.

Finally, the model is fed with data corresponding of 5086 samples/hourly time steps and each input contain the wind speed measurement between t and t-23h resulting in 24*178 = 4272 features per sample. In a second time, I also included the 24 most recent wind load factor value and the date in the form of the projection of the polar coordinate on a circle. Therefore, it results in 4298 features per input sample.

The output of the model corresponds to the wind power production of the next four hours (H+1, H+2, H+3, H+4) resulting in 5086 * 4 points.

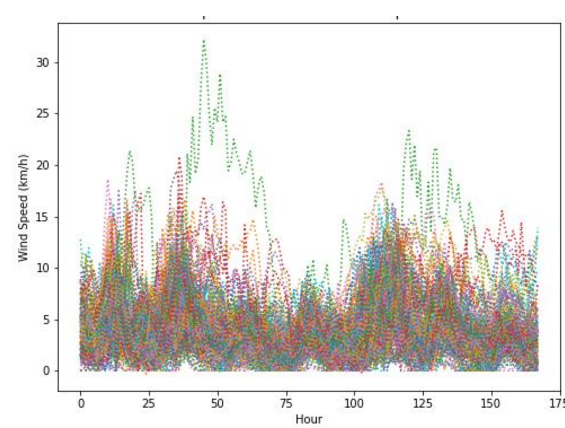

Figure 1: Input Data – One Week Wind Speed

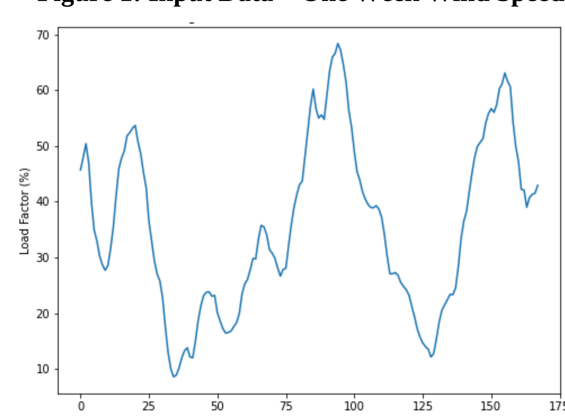

Figure 2: Target Data – One Week Wind Load Factor

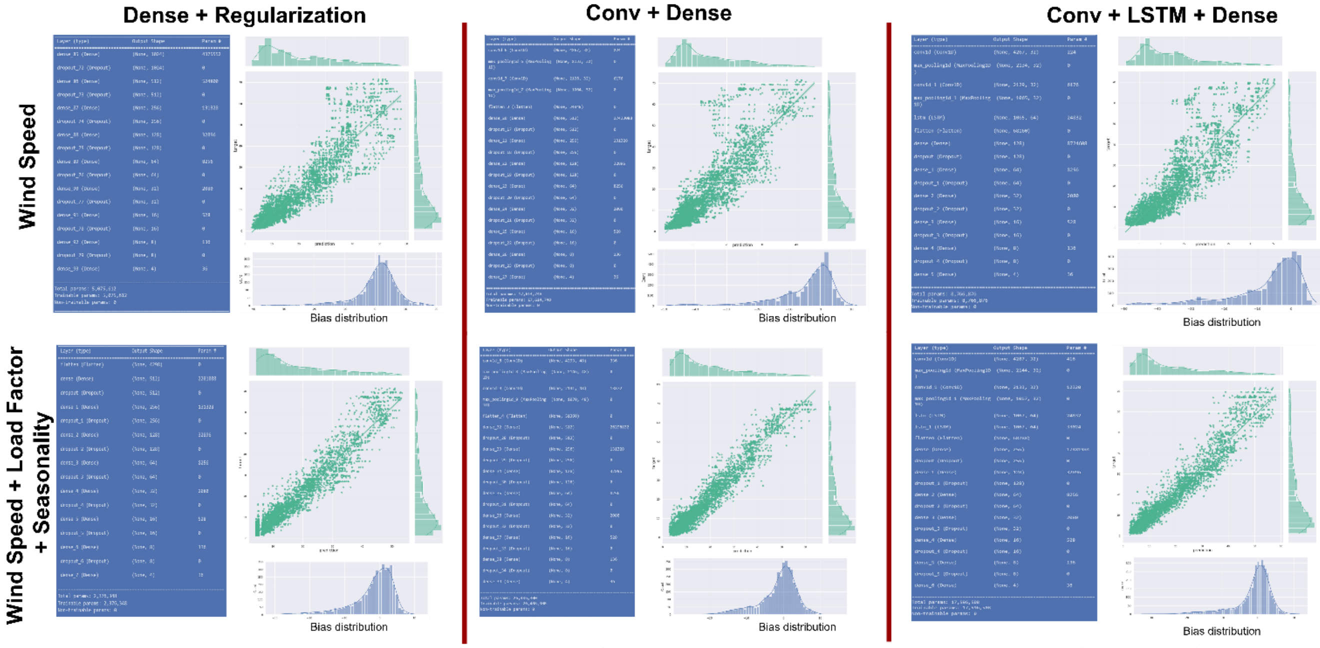

Figure 3: Best model structure of each type of model and their associated prediction in function of their target values and error distribution

3. SRUCTURE OF THE EVALUATED MODELS

For this project, I essentially considered one structure of model being a mix of convolutional layers, LSTM layers, dense layers, and regularization layers. This structure was mainly inspired by the work presented in the references [1] and [2].

This project differs from the one presented in the references; it uses a lot of data scattered throughout the country to forecast a national wind load factor opposed to on site measurement used for the prediction of a single windfarm. Therefore, to have a better understanding of the contribution of each type of layer, I splitted and tested the model by adding the different blocks incrementally. It results in three neural network structures:

1. A combination of **Dense Layers** and **Regularization Layers** (Dropout)
2. A combination of a first stage of **Convolutional Layers** followed by **Dense Layers** and **Regularization Layers** (Dropout and Max Pooling)
3. A combination of a first stage of **Convolutional Layers** followed by **LSTM Layers** then by **Dense Layers** and **Regularization Layers** (Dropout and Max Pooling)

The assumption behind the use of this types of layers to handle weather time series with a lot of data to forecast wind production are the following one:

RNNs are a type of neural network that are particularly well-suited for processing sequential data, such as time series. This is because RNNs have a "memory" that allows them to retain information about past inputs in the sequence. This allows RNNs to make use of context and dependencies between the data points in the time series, which is verry interesting with the data used in this project.

CNNs are also often used for time series data, although they are primarily designed for image processing. CNNs can be effective for time series data because they are able to automatically learn features from the data and can handle large amounts of input data efficiently. They are also well-suited for learning local patterns in the data, which can be useful for identifying trends or patterns which could also be interesting for weather-based data.

Dense layers can handle a large amount of data effectively and learn a potentially complex, non-linear relationship between the input and output data. Which is clearly the case for the power curve of a wind turbine as presented in **Figure 4**.

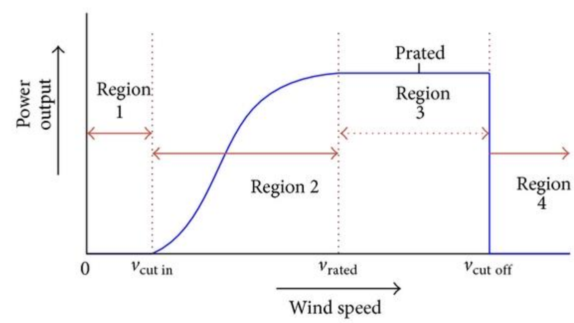

Figure 4: Generic Power Curve of a wind turbine

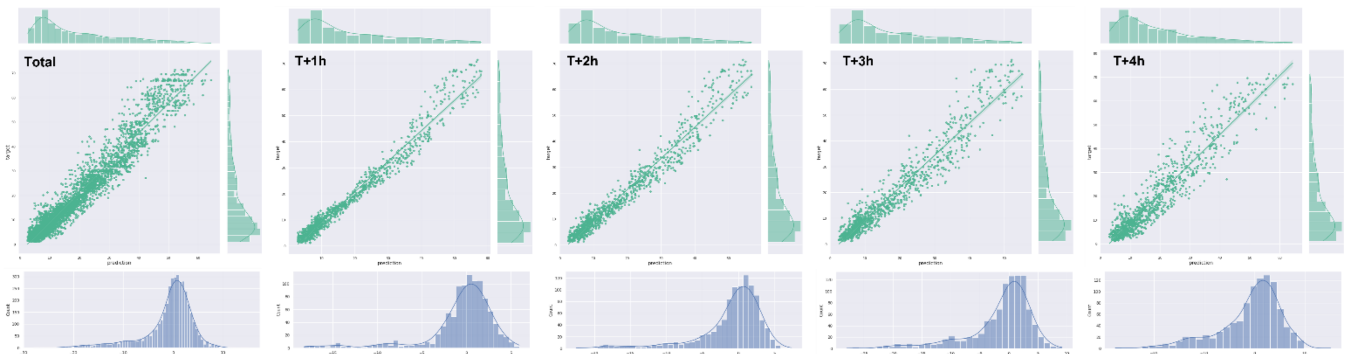

Figure 5: Best model and its associated predictions in function of their target values and error distribution

4. RESULTS AND ANALYSIS

As explained in the **part 2**, In a first part of the project, I used only wind data as input to have a better understanding of the impact of each layer and each type of data. In a second part of the project, I also included the 24 last values of the wind load factor and the date in the form of the 2D projection of the polar coordinate of a circle.

The main metrics used in this project are the R^2, RMSE, and MAPE as it allows me to compare my results to what I found in my references

The models that obtain the best results for each structure types are presented in **Figure 3**. Each model type structure was tested for different layers organization and hyperparameters. It is important to notice that the model using CNN and RNN layers include a lot more of parameters and were trained under more constraining parameters (batch size, kernel size and epoch number) to keep a reasonable computational time and memory usage, especially in with wind data only as I kept the training time below 1 hour. This may be the main reason behind the results obtained and presented in **Table 1** as the constraints may have been to restrictive and the models could have been trained a bit more.

With these constraints, the model structure using only dense and regularization layers was clearly the better which was a bit surprising compared to the scientific literature. We can also suppose that dense layers are well suited to handle time series with a lot of features per step.

Model	R2	RMSE	MAPE
Dense	0.799	56.684	44.082
Conv + Dense	0.551	126.593	38.217
Conv + LSTM + Dense	0.364	179.452	40.84

Table 1: Results with wind data only

In a second part of the project, I included extra data and I released the constraints on the hyperparameters to push the training of the model until they reach stable loss. Nevertheless, I was still limited by the memory usage for the model using LSTM and tried to limit its training below 6 hours.

	Scope	R2	RMSE	MAPE
Dense	**Total**	**0.855**	**41.036**	**32.863**
	t+1h	0.887	31.816	29.733
	t+2h	0.876	35.189	29.721
	t+3h	0.847	43.378	34.167
	t+4h	0.81	53.763	37.832
Conv + Dense	**Total**	**0.898**	**28.9**	**31.068**
	t+1h	0.92	22.516	24.192
	t+2h	0.914	24.248	27.172
	t+3h	0.893	30.181	33.156
	t+4h	0.863	38.656	39.75
Conv + LSTM + Dense	**Total**	**0.895**	**29.705**	**26.232**
	t+1h	0.943	16.062	21.583
	t+2h	0.922	21.927	23.183
	t+3h	0.882	33.534	27.69
	t+4h	0.833	47.3	32.47
Combination of the best models for each timestep	**Total**	**0.903**	**27.545**	**28.052**
	t+1h	0.943	16.062	21.583
	t+2h	0.922	21.927	23.183
	t+3h	0.882	33.534	27.69
	t+4h	0.863	38.656	39.75

Table 2: Results of each model with expended data

The model structure and the results are presented in **Figure 3** and **Table 2**. Here, I also distinguished the metrics for each time step. It is important to notice that in this part, the training of the models with CNN and RNN layers was significantly longer than the one with Dense layers only (4 to 10 time longer). With the extra data and a longer training time, I significantly increased the performance of the CNN and the RNN model and reach reasonable results that are more coherent with the scientific literature. The results obtain using the best model for each timestep are presented in **Figure 5**. It corresponds to the Conv + LSTM + Dense model to t+1h, t+2h, t+3h and to the Conv + Dense model for t+4h. This model provides good results, especially for t+1h and t+2h predictions, however the prediction metrics for t+4h are probably a bit high for a real usage. We can also observe that all the models have some difficulties to forecast high load factor (>40%), this is probably linked with the low probability of having these events. To fix this, a potential improvement could be to feed the model with more data to have more situations with high load factor.

5. DISCUSSION

Overall, I obtained good results in the end of the projects with a model that combine Convolutional, LSTM, Dense and Regularization Layers. The model still has some difficulties to forecast high load factor and t+4h but it could probably have some usage in the current state. Furthermore, it provides really good results to forecast at t+1h and t+2h. Nevertheless, I still think that there is room for some improvements.

The potential extensions of this work would be to include other types of weather data types related to the wind production. To keep a reasonable amount of data, it could be interesting to consider only the stations that contribute the most to the model.

It could also be interesting to explore more complex model structure, for example by connecting the dense layer to both the LSTM output and to the initial input. It would also be interesting to create wider structure including LSTM on a more powerful hardware to do not be limited by the RAM usage as I still expect this structure potential to not being fully exploited.

It is also possible to try to forecast the electricity demand or the PV production of the country in the same way by using the original dataset (electricity demand in France is highly thermosensitive).

6. REFERENCES

[1] Hong, Y.-Y., & Rioflorido, C. L. P. P. (2019). A hybrid deep learning-based neural network for 24-h ahead wind power forecasting. In Applied Energy (Vol. 250, pp. 530–539). Elsevier BV.

[2] Wu, Z., Luo, G., Yang, Z., Guo, Y., Li, K., & Xue, Y. (2022). A comprehensive review on deep learning approaches in wind forecasting applications. In CAAI Transactions on Intelligence Technology (Vol. 7, Issue 2, pp. 129–143). Institution of Engineering and Technology (IET).

[3] Li, P., Wang, X., & Yang, J. (2020). Short-term wind power forecasting based on two-stage attention mechanism. In IET Renewable Power Generation (Vol. 14, Issue 2, pp. 297–304). Institution of Engineering and Technology (IET).

[4] Deng, X., Shao, H., Hu, C., Jiang, D., & Jiang, Y. (2020). Wind Power Forecasting Methods Based on Deep Learning: A Survey. In Computer Modeling in Engineering & Sciences (Vol. 122, Issue 1, pp. 273–301). Computers, Materials and Continua (Tech Science Press).