WIND POWER FORECASTING USING CNN AND LSTM MODELS

Alvaro Carrera Cardeli (s172254), Marcos Ivorra Peleguer (s202947), Santiago Ribelles Garcia (s202948)

Danmarks Tekniske Universitet (DTU)
DTU Compute
Richard Petersens Plads, Build. 324, DK-2800 Kgs. Lyngby

ABSTRACT

Wind energy has experienced a huge development over the last decade due to its importance in decarbonising the world's economy and making a green energy transition. The main goal of this paper is to implement and test deep learning algorithms that could perform a proper forecast of the power generated by one park, *Park Power*, with 48-hours in advance. The data is provided by a Chinese company that recorded information every 15 minutes during almost 4 years. Based on the nature of the data used and the time-dependant patterns that govern the power curves, it was considered Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) as the deep learning algorithms used for forecasting. Thus, searching for a generalized model for each of the algorithms and comparing the performance of both algorithms will be the aim of this report, concluding that the CNN outperforms LSTM forecasting reaching to 93.857% of accuracy rate. Also, an individualized temporal analysis will be done for each of the algorithms, this concludes that the bigger the time window for LSTM predication, the better performs.

***Index Terms*—** CNN, LSTM, wind energy, Park Power, deep learning, power forecasting

Github Repository:
https://github.com/Marcoscos/windpowerforecastDTU

1. INTRODUCTION

Wind energy has experienced a huge development over the last decade due to its importance in decarbonising the world economy and making a green energy transition. The unpredictable nature of wind energy due to the high variability of wind can compromise the efficiency and reliability of this technology. Hence, the motivation of this paper is to help utility companies to forecast wind energy production 48-hours ahead, which will allow to making wind energy a more reliable energy source as well as for the optimal price selection to selling electricity produced from it.

This paper is focused on deep learning algorithms for the wind power output predictions due to the importance and high development of neural networks in this field in the last years, as recent works have proven to obtain accurate predictions. The study centres on the development of a Convolutional Neural Network (CNN) and a Long Short-Term Memory (LSTM) model, that will be used to defining a benchmark of wind power forecasting. Hence, both models are compared in order to identify what is the best approach for conducting the power predictions. In order to comparing both neural networks, the forecasts were conducted following a 5-fold cross-validation, looking for the generalization error (i.e. the most general model), and a temporal validation, as a time-series procedure.

In this paper, a state of art related to the topic will be presented in section 2. Then, in Section 3 it will described the data set and the algorithms implemented, CNN and LSTM. The final sections will focus in presenting the results, discussing about them and providing some conclusions, section 4, 5, and 6, respectively. Also, in section 6 it will be purposed some future lines.

2. STATE OF ART

Deep learning for wind power prediction has experienced a wide development in the last years, with new methods and scopes that have been able to improve the neural networks' predictions as well as their performance.

A study from the Central South University, in China, Hui Liu, Xi Wei Mi and Yan-fei Li (in November 2017) developed a hybrid deep-learning wind speed prediction model, that merges the empirical wavelet transformation, which decomposes the raw wind speed data into different sub-layers; and two different kinds of recurrent neural networks, which are an LSTM neural network, that predicts the low-frequency wind speed sublayers, and the Elman Neural Network, which predicts the high-frequency sublayers. This paper compares the proposed neural network to eleven main-stream forecasting models and concludes that the EWT-LSTM-Elman model that they proposed has satisfactory multi-step forecasting results in high-precision wind prediction [1].

Wen-Hui Lin et.al., from Khun Shan University in Taiwan (October 2021), proposes in a study a Deep Learning Network to obtain the correlations between meteorological features and wind power generation using multilayer neural convolutional architecture with gradient descent algorithms that minimize estimation errors compared to statistic-based prediction models. The study focuses on 24-72h ahead wind power prediction with a Mean Absolute Percentage Error (MAPE) less than 10% with the use of a Temporal Convolutional Network (TCN). In the case study performed for the mentioned report, the experimental results obtained a MAPE of 5.13% for 72h wind power prediction. In addition, the proposed model was compared LSTM, RNN and GRU neural networks, where the TCN outperforms these three models in terms of data input volume, stability of error reduction and forecast accuracy [2].

In October 2019, Sana Mujeeb et al. developed a model with two stages. The first stage is based on the Wavelet Packet Transform (WPT), which is used to decompose past wind power signals, that together with Numerical Weather Predictions are used as input to the wind power forecast model. In the second stage, the Efficient Deep Convolutional Neural Network (EDCNN) is employed to forecast the wind power 24h in advance for the Main wind farm ISO NE. The EDCNN model obtains an average of 2.6% of MAPE for the 4 seasons of the year. Hence, the model showed accurate predictions for the day-ahead hourly wind power and outperformed the CNN and SELU CNN models [3].

In the journal Future Generation Computer Systems, a paper done by Ruiguo Yu et al. in April 2019, presents an improved Long Short-Term Memory-enhanced forget-gate (LSTM-EFG) network model that is used for wind power forecasting. Based on the correlation, the features data of turbine groups that are in a certain distance are filtered to do a further optimization of the forecasting by applying clustering. The model enhances the effect of the forget-gate and changes the activation function to optimize convergence speed. The experimental results obtained show that the Mean Square Error (MSE) for the LSTM-Eodel is lower than existing methods such as LSTM, SVR and KNN, and by using Spectral Clustering to get the temporal correlation characteristics the predictions are the most effective with a MSE of 5.53. This makes wind power prediction for a certain time more accurately [4].

3. METHODS AND MATERIALS

3.1. Data set

The data used for this project was provided by Meteodyn China, and consists of two datasets with information of a wind farm located in China of 49500 KW capacity. The windfarm has a total of 33 wind turbines which have a hub height of 80 m, blade length of 43 and a rated power of 15000KW.

The first dataset contains date/time of recorded numerical weather predictions of the park's location from January 2016 until September 2020 in 15-minute timesteps. The weather data provides measures of wind direction, wind speed, temperature and air density at different heights, being 10m, 50m, 100m and 150m, as well as the air pressure at sea level.
In addition, the second dataset provided contains historical data of the park's measurements ranging from May 2017 to September 2020 also in 15-minute timesteps. The measurements supplied are wind speed and wind direction at 10m, 30m, 50m and 80m, as well as the park's power output, in KW. The data is visualized with the help of seaborn, matplotlib and pandas-profiling python libraries.

3.1.1. Data set preparation

In order to have the data ready to be inputted to the LSTM and CNN neural networks, both datasets are combined. The dataset made will be a combination of the parameters present in the weather forecast data and the power output of the park which is present in the historical measurements' dataset.
As the historical measurements dataset considers data starting from 1st May 2017, all data points present in the weather forecast dataset prior to that date are omitted. Furthermore, there was a 1.05% of missing values for the power output data of the park during the studied time period, therefore the timesteps containing these missing datapoints are also removed from the dataset. Hence, by merging the timesteps with power output data and the historical weather forecast data, the resulting data frame contains a total of 116270 entries.

Since there is a total of 18 columns with the wind speed, wind direction, temperature and air density at 4 different heights, as well as the pressure at sea level and park power output; a reduction of parameters is needed in order to have an optimal performance of the neural networks. Due to the collinearity of the parameters present in the dataset (see fig. 1), a high number of them can be omitted and not being inputted in the neural network.
The objective is to choose the parameters which are more correlated with Park Power [KW], which is the target variable, but also taking into account the high collinearity of the data. Hence, the Speed at 100 m is chosen as it is the variable with

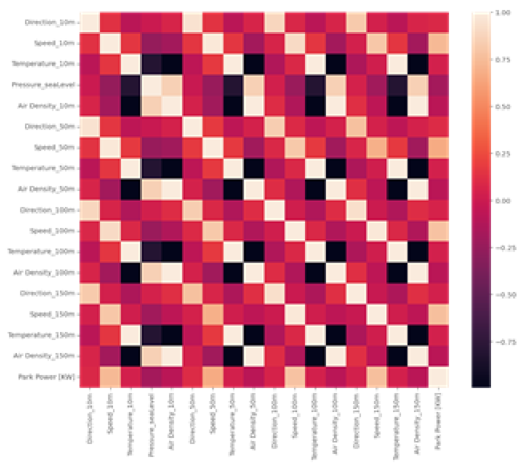


Fig. 1. Correlation Matrix

highest correlation to the power output. Also, the Speed at 10 m is selected to be part of the dataset as it is also high degree of correlation with Park Power [KW] and it is less correlated to the Speed at 100 m than the ones at 50 m and 150 m.

In addition, due to the high collinearity of the rest of the features, which have low correlation towards the target variable, only two of them will be selected for the final dataset. The parameters selected are: Direction at 50 m and Temperature at 100 m.

Once having selected the final features to be inputted in the neural networks, the removal of the outliers present in the data is done. As seen in Fig. 2, there are certain outliers that do not match the speed/power curve. Hence, the data points which are above percentile 99.7% are removed in order to assure that the network is fed with good quality data.

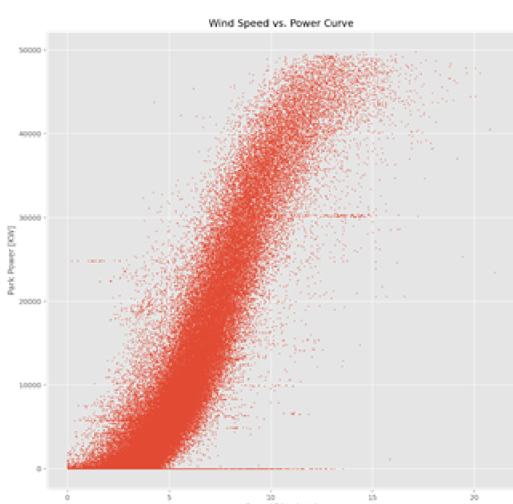


Fig. 2. Speed/Power Curve

In order to have values of the dataset in a common scale, the normalization of the features from 0 to 1 is done by using the following equation:

$$z_i = \frac{x_i - min(x)}{max(x) - min(x)} \tag{1}$$

Therefore, the values of the dataset are now normalized which will improve the performance of the CNN and LSTM neural networks.

Lastly, once the data is processed and cleaned, the final step for preparing the data set is the generation of time-series inputs. The window chosen is predictions within 2 days of data, therefore each of the predictions in the output of the neural networks (i.e. Park Power) will be based on an input with the four attributes aforementioned in the previous paragraphs with a window size of 193 samples for each attribute, thus the input dimension will be 4 *samples* times 193 *time-steps*. This lead to a data set size of 104327 temporal matrices of 4x193 points in each matrix.

3.2. Prediction algorithm

The nature of the prediction leads to look for deep learning algorithms specialized in temporal or time-series resolution. Thus, Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) were the algorithms chosen for this task.

In this section, both CNN and LSTM algorithms will be introduced and the reason why this algorithms were chosen over other will be explained.

3.2.1. Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a deep learning algorithm that is able to classify or predict one output based on the input to the network. The most common inputs, are the ones with high dimensions since this networks are good at identifying and highlighting spatial and temporal information. Thus, the role of the CNN is to reduce the dimensions of an input by keeping the most important information, before applying a prediction/classification network (*dense*).

In the case of this data set, there is a high dimension input (*193 time-steps* X *4 attributes*) with temporal information in each of the attributes. Thus, a CNN fits with the nature of the prediction problem. 1D (considering just temporal information) and 2D (considering temporal and spatial information) convolution structures where tested before defining the final architecture and it was concluded that 1D convolution would perform better as it was expected. This is due to the fact that there is no spatial information, thus a 2D convolution cannot find any spatial information that would improve the

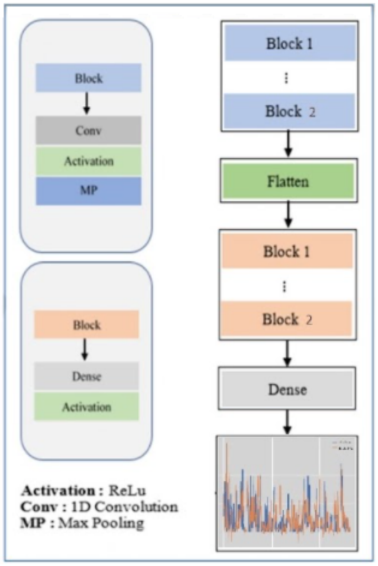


Fig. 3. Convolutional Neural Network architecture.

prediction.

Fig. 3 shows the architecture of the CNN trained to perform the prediction task. As it is shown, it is composed of two convolutional blocks before the prediction network (*dense*). Each of the convolution blocks are composed of:

- **Convolutional layer:** it is the core block of the CNN. This layer consist of a set of learnable filters that are temporally smaller than the inputs, but extend through the full depth of the inputs volume.
- **Activation function:** this block is at the output of the convolutional layers and it is the one that helps to pick the useful information and to fire the rest.
- **Pooling layer:** it is the block that progressively reduces the temporal size of the input to reduce the amount of parameters and computation in the network.

Lastly, a **Fully-connected layer (prediction network)** is implemented connecting all the outputs from the convolutional blocks to predict an output value (i.e. Park Power).

3.2.2. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks are designed to overcome a problem introduced by the Recurrent Neural Network (RNN) with long-term dependencies. LSTMs have a feedback connection that enables them to process entire sequences of data by keeping the useful information of each of the previous data processed in the sequence.

Fig. 4 shows how a LSTM block is structured in order to have the property previously introduced. It is composed of 3 gates with different actions and a fourth block that behaves like a memory:

- **Forget gate:** this is the first gate of the LSTM block where it is decided which pieces of long-term memory should be forgotten at each stage. The previous hidden state and the new input data are feed into the neural network. In this gate, it will be decided which information of the *cell state* is useful given the previous hidden state and the new input data.
- **Input gate:** the goal of this gate is to decide which information of the new input data should be added to the long-term memory. As in the previous gate, this decision is also given the previous hidden state and the new input data.
- **Output gate:** it decided the new hidden state. It is using the information from the new updated cell state, the previous hidden stated and the new input data.
- **Cell state:** it carries the long-term memory of the network by encoding the aggregation of useful data from all the previous time-steps that have been processed.

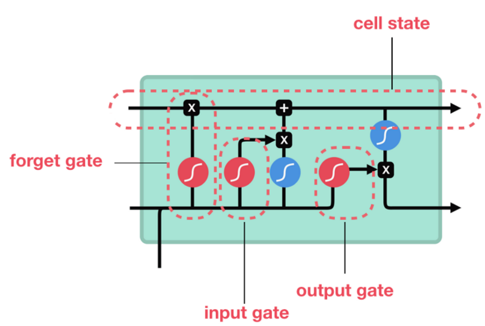


Fig. 4. Long Short-Term Memory block.

4. RESULTS

The forecast was conducted both following a regular generalization model approach and a time series approach. All in all, the aim is to determine the optimal rolling window to train the model and to determine which architecture better translates past trends into future values. It was measured the accuracy based on several metrics, which will be explained later on. Out of all the models, the five CNNs and the five LSTMs, the best results are obtained using a CNN for the second fold. This best fold out of the five for the CNN that it was used to perform the cross validation can be seen below in Figure 5:

About the results, the best forecast is the model of the 2nd

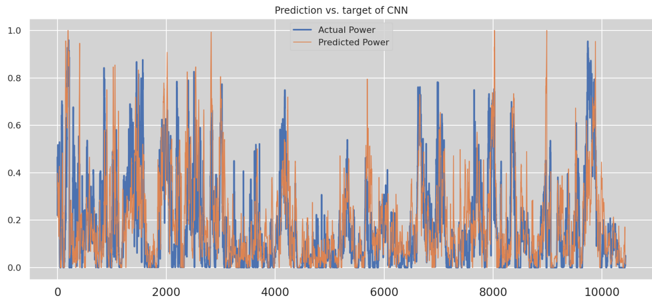


Fig. 5. Forecast of the Cross Validation's 2nd fold for the CNN.

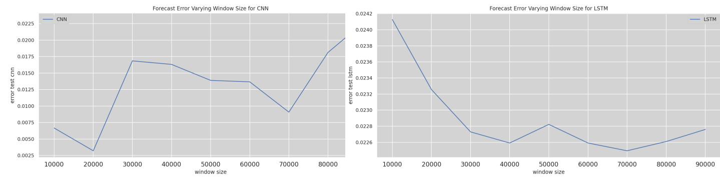
fold for the CNN. It is highly accurate compared to the other configurations, with a difference of one order of magnitude when compared to the LSTM models. To correctly evaluate the performance it was taken into account the Root Mean Squared Error, the Mean Squared Error, the Mean Absolute Error and the Accuracy Rate, a metric that comes from the Chinese electricity market from where the data comes (see equation 2). The use of the different metrics come from acknowledging that the metrics represent different aspects of a good forecast, and taking all of them into account makes for a good forecasting metric. The overall forecasting errors can be seen in the table 1.

$$acc\_rate = \left(1 - \frac{\sqrt{\sum_{i=1}^{n}(P_{mi} - P_{pi})^2}}{Cap \cdot \sqrt{n}}\right) \times 100\% \quad (2)$$

Model	RMSE	MSE	MAE	Accuracy Rate
CNN 1	0.065234	0.004255	0.047008	93.476639
CNN 2	**0.061425**	**0.003773**	**0.044672**	**93.857479**
CNN 3	0.063341	0.004012	0.045737	93.665944
CNN 4	0.063103	0.003982	0.046235	93.689698
CNN 5	0.062996	0.003969	0.044963	93.700367
LSTM 1	0.135245	0.018331	0.093681	86.475544
LSTM 2	0.135775	0.018465	0.093598	86.422473
LSTM 3	0.133345	0.017822	0.092876	86.665483
LSTM 4	0.134010	0.017995	0.092983	86.598954
LSTM 5	0.133188	0.017793	0.092379	86.681238

Table 1. Search of the best model for the 5-fold cross validation.

In order to have a good forecasting model it is interesting to identify the best rolling window, which can be understood as how long back in time is it good to look to in order to make a good forecast. This analysis was done both for the CNN and the LSTM and obtained differing results for both. The results can be seen in the charts below:



(a) Best window for CNN (b) Best window for LSTM.

Fig. 6. Identification of the best rolling window for each neural network architecture.

5. DISCUSSION

As introduced before, it was conducted the forecast following two approaches, a 5-fold random cross-validation, and a temporal validation. This two approaches, the first as a generalizing model procedure and the latter as a time series procedure are complementary bringing valuable insights.

On the one hand, the Regression-like shows how good the models tried out translate atmospheric features into forecasted power. By randomizing the forecasting we ensure there is no overfitting in the predictions, which is shown when checking the training and validation errors for a large number of epochs for the best model, CNN2. It can be seen in Figure 7.

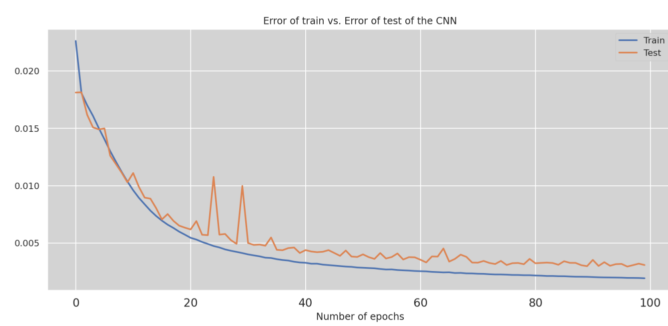


Fig. 7. Training and Validation error of CNN 2.

On the other hand, the time series approach brings clarity on how long back should a model look back in order to make an accurate forecast, this is, the rolling window size. The charts in Figure 6 show precisely this. Looking at these charts, we want to achieve a compromise between the number of input samples taken, which translates into computing power; and the explainability, the accuracy of the model.
Looking at the LSTM´s chart, we identify a neck in 30.000 time steps. This means that the "ideal" size of the test set

is 30.000 samples. It is important to note that this window size has been checked with a separation of 10.000 time steps, which might be too big for a big dataset. If studied outside of the academic context of a graduate course, the window size would have been checked for smaller steps, for instance 3000 time steps (blocks of 15 minutes). Nonetheless, it is clear that the LSTM architecture predicts better with bigger datasets since the bigger datasets incentivize better accuracy due to the with the forget / don´t forget structure.
If we look at the CNN´s chart, it is clearly seen that a bigger dataset is penalized by the convolutional architecture. The "ideal" window size is 20.000 samples. This is a reasonable result since the CNNs apply filters and kernels to the original data and this can be too distortionate for data too back in the past.

As per the forecasting results, the clear winner is the CNN architecture, with a best RMSE value of 0.061425. Putting it in industry standards, which is the ultimate goal of the dataset´s provider, using the defined CNN model provides a 93,857% accuracy rate. This rate takes into account the estimated total production of the plant relative to the total capacity, which makes a more straightforward metric to measure total power production, in other words, benefits for operating the wind farm.

6. CONCLUSION

All in all, the project suffices its objective of defining an accurate 48-hour forecasting model for wind power forecasting. It does so by looking at two traditional ways of forecasting, one inherited from time series models and one inherited from generalization models. The first one maintains the data in its original sequence and tunes the parameters to identify the best window to conduct the forecasts. The latter, on the contrary, searches for the best model by performing a 5-fold cross-validation, which in turn reduces, if existing, the overfitting.

To conduct the project, we have delved into current benchmarks for wind power forecasting but also for prevalent new techniques that are forcing new advancements in the field, such as the deep learning-based papers mentioned in the Section 2.

We have used two different neural net architectures to conduct the forecast, the LSTM and the CNN. While the LSTM is widely recognised for time series problems, the CNN was a bold try to better understand how 1-dimensional kernels work, and it ended up bringing the best results to the project.

We have tried the two approaches with the two architectures, and evaluated using several metrics not to overweight the characteristics that each of the metrics are skewed towards individually.

As for the next steps, it would be our main objective to further develop the project tuning the hyperparameters of the models and trying out greater epoch sizes to reinforce our udnerstanding of the results obtained.

7. REFERENCES

[1] Yan-fei Li Hui Liu, Xi-wei, "Wind speed forecasting method based on deep learning strategy using empirical wavelet transform, long short term memory neural network and elman neural network," in *Energy Conversion and Management*. Central South Unversity (China), 2017, pp. 498–514.

[2] Kuo-Ming Chao Hsiao-Chung Lin Zong-Yu Yang Wen-Hui Lin, Ping Wang and Yu-Huang Lai, "Wind power forecasting with deep learning networks: Time series forecasting," Khun Shan University (Taiwan), 2021.

[3] Sameeh Ullah Aisha Fatima-Nadeem Javaid Sana Mujeeb, Turki Ali Alghamdi and Tanzila Saba, "Exploiting deep learning for wind power forecasting on big data analytics," 2019.

[4] Mei Yu Wenhuan Lu Tianyi Xu Mankun Zhao-Jie Zhang Ruixuan Zhang Ruiguo Yu, Jie Gao and Zhuo Zhang, "Lstm-efg for wind power forecasting based on sequential correlation features," in *Future Generation Computer Systems*, 2019, vol. 93, pp. 33–42.