Global Energy Wind Forecasting

Andreea Teodora Bejenariu, s184377
Negin Farhang, s212663

02456 Deep Learning
DTU Compute
January 4, 2022

Abstract

Due to the nature of the wind turbines, an accurate prediction generated wind power is prone to be hard. As such, many different models such as deep learning methods like Convolutional or Recurrent Neural Networks are used to predict wind power. Furthermore, data processing and cleaning are essential when working with real-time datasets. This report tries to present a solution to this problem by processing time-series data, developing the CNN and LSTM model and training them on seven wind farms with different loss functions. Models showed that the LSTM model with MSE loss function gives a more accurate prediction of the wind farm power generation when training on the data sets of wind farms with similar seasonality.

Keywords— Wind power prediction, Convolutional Neural Network , Long short-term memory, Long term forecast of wind power

1 Introduction

Wind Power Forecasting (WPF) represents how much wind power is to be expected in within a future time-frame. Accurate wind power forecasting can be used among other things to maximize the profit by electricity traders and it is a recognized as a major contribution for reliable large-scale wind power integration [1].

This paper seeks to use several deep neural network models in order predict the power of the wind across 7 wind farms within the next 48 hours (h). The performance of the models will be evaluated using the mean absolute error (MAE), mean squared error (MSE) and the root mean squared error (RMSE) between the prediction values and the actual observations. The experiments are carried out using the Kaggle dataset from the Global Energy Forecasting Competition 2012 [2].

The paper will start with section 2 by referencing the related works regarding the prediction success of different WPF models. In section 3 the processing of the Kaggle dataset will be described as well as the reason behind the transformation of the input. The experimental set-ups, as well as the developed models will be thoroughly explained in section 4, following the interpretation of the training results in section 5. Furthermore, the wind power prediction within the next 48h will be assessed and discussed within the same section.

2 Related Work

Wind power generation prediction has always been a challenge in wind energy systems and, as a result, there are a lot of studies around this topic. Studies around this topic are divided into two sections: short and very short-term prediction, which is around a few seconds to six hours ahead, and medium- and long-term predictions, which is around six hours to one week ahead [3].

In short and very short-term prediction, methodology used is mostly Artificial Neural Networks, Markov switching model and adaptive neuro-fuzzy model[3][4][5][6]. In medium and long-term prediction, which this report falls into, the popular methods are Recurrent Artificial Neural networks and adaptive neuro-fuzzy inference systems[3][7][8].
However, also as shown in this report, the prediction error increasing as the prediction time lengthens[8] and seasonality are also something to consider when predicting wind power [9].

Data mining is an important aspect of wind generation prediction since wind power generation works with real-world data, so different parameters are experimented on to find the best result. In most cases, the input data is wind power in earlier time slots and wind speed. Wind direction, air temperature, relative humanity, and rainfall were also considered in many studies.[3]

3 Data processing

The datasets used are the training set of wind power measurements of 7 wind farms (WP1, WP2 ... WP7) and the wind forecasts (WF) given for each farm. The training set was used for timestamp synchronization of the forecast attributes and extracting the actual value $y = wp$. The attributes of the model input were purely extracted from the meteorological forecasts, which are given by the wind speed, wind direction, and its zonal and meridional vector components.

For this experiment, only the training time frame was used to generate the training, validation, and testing sets. Using the weather forecasts attributes and a time-lag of 100h, a two-dimensional matrix input, X_j, was compiled, as shown bellow.

$$X = \begin{array}{cc} \begin{array}{cccc} \text{ws} & \text{wd} & \text{u} & \text{v} \end{array} & \\ \begin{pmatrix} 0.12 & 0.11 & 0.08 & 0.07 \\ 0.13 & 0.14 & 0.06 & 0.04 \\ . & . & . & . \\ 0.22 & 0.24 & 0.12 & 0.11 \end{pmatrix} & \begin{array}{c} t_1 \\ t_2 \\ \\ t_{100} \end{array} \end{array}$$

Prediction was done on this input collected from the $t_1 ... t_{100}$ forecast observations in order to predict the wind power y generated at time t_{101}. Basically, the wind forecast for each farm (WF1, WF2 ... WF7) was used to predict the wind power (WP) generated by each farm

4 Experimental setup

Since prediction was done solely on the meteorological forecasts which are generated at lead times from 12 to 48 hours ahead, models were designed in order to permit a long-term prediction of wind power generation at lead times from 1 to 48 hours. A separate part of the dataset was used to generate 100 time-frame windows of 48h, in order to evaluate the prediction metrics in Section 5.

One model uses Convolutional Neural Networks (CNN) and the other one uses a network with a Long Short-Term Memory (LSTM) layer. The first model was used in the experimental Setup I and the second was used in Setup II, of which results will be discussed in Section 5. The models were trained using the Least Absolute Deviations (L1) loss and the Least Squarred Errors (L2) loss, which are described in equations 1 and 2, respectively.

$$L_{1\,Loss} = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{1}$$

$$L_{2\,Loss} = \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{2}$$

N	number of of total wind power observations
y_i	actual measured wind power
$\hat{y}_i$	predicted wind power

4.1 Model

4.1.1 *Input*

1. *Wind speed*: Wind speed is one of the most important features when calculating wind power generation. Wind turbines are designed to work in a specific range of wind speed, called cut-in speed. When the wind speed is in the range of cut-in to rated speed, the generated power is cubically increased with the increase of the wind speed. The relation is as follows:

$$p = \frac{1}{2} C_p A_\rho V^3 \tag{3}$$

Where P is wind wheel output power (kW), $C_p$is wind wheel power factor, A is wind wheel sweep area (m^2), ρ is air density (kg/m^3) and V is wind speed(m/s).[10]
However, as the wind speed increases, the cubical relation no longer follows and results in a more flat relation. [11]

2. *Wind direction*: Wind direction influences the efficiency of the wind turbines and thus is an important factor when predicting the power. Since

in wind farms, many wind turbines can influence the wind direction on the other wind turbines, it is important to factor the wind direction.[10]

3. *Meteorological Wind Direction*: Meteorological wind direction v and u represent the direction of the wind speed as wind direction and wind speed can be derived from v and u and thus they have a direct effect on wind power generation and they have been included as the training data as well.

4.1.2 *Output*

Two classes are considered in wind power generation, short-term prediction, medium and long-term power prediction. The model developed in this report is a long-term prediction, 48 hours ahead of the actual wind power generation.

4.1.3 *CNN*

A CNN is an artificial neural network with each layer consisting of 2-dimensional planes. In the preprocessing stage, a 2D input based on the wind direction, wind speed, u, and v of each wind farm was developed. A CNN model was trained on this input. In the first stage, there is a convolutional layer and the input is convoluted by a filter and added biases. After that the features extracted are put into a max pooling layer and in the end, a fully connected layer, to give a linear function for the features. The structure of the CNN is shown below in Figure 1:

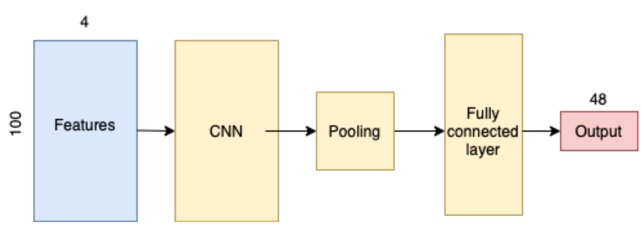

Figure 1: CNN Structure

4.1.4 *LSTM*

Since the input data is in the form of a time series, an LSTM network was also trained on the input. LSTM is a form of recurrent neural network for processing a sequence of data and it has proven to be good at learning to store information over extended time intervals [12]. Forecast attributes in time t_j were concatenated to form a sequence of data and then were trained using an LSTM layer and a fully connected linear layer.

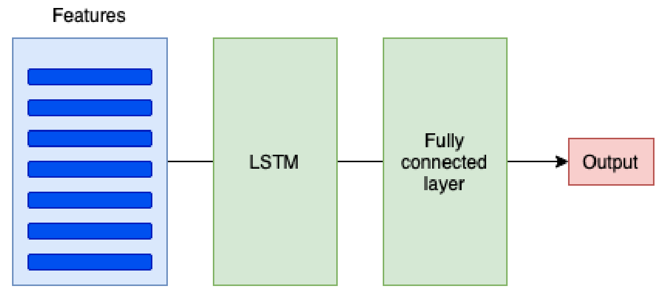

Figure 2: LSTM Structure

Figure 2 shows the structure of this network. The result of each model and with different loss functions are discussed in the next section.

4.2 *Hyper-parameters*

Hyperparameter	**Value**
Setup I: CNN with L2 loss	
Batch Size	64
Epochs	50
Learning Rate	1e-3
Weight Decay	1e-6
CNN layer (x1) kernel size	4
Number of filters	8
Pooling layer (x1) kernel size	2
Dropout layer (x1) drop-out rate	0.2
Fully Connected Layer (x1) hidden size	85
Setup II: LSTM with L2 loss	
Batch Size	64
Epochs	50
Learning Rate	1e-5
Weight Decay	1e-10
LSTM layer (x1) hidden size	120
Fully Connected Layer (x1) hidden size	500
Dropout layer (x1) drop-out rate	0.2

Table 1: Chosen hyperparameters values for the CNN model (Setup I) and the LSTM model (Setup II)

The two models were trained using the same training and validation set, as well as tested on the same 48h time-frame windows. The models were implemented using PyTorch [13] ver. 1.10 and the training was done on the Tesla K80 GPU. The models were initially trained using the data collected from Wind Farm 1 (WF1). Empirically it was found that using the MSE loss gave better training results for both the CNN and LSTM models (see Section 5). Thus, the models using L2 loss (Setup I and Setup II) were selected and trained further using the data from the remaining six wind farms. The hyperparameter values of these setups are given in Table 1.

Apart from the layer moderate complexity, the table also shows the regularization parameters: using dropout and weight decay. The tuning of those values was done empirically, although cross-validation techniques could have been used in order to determine the best combination. The training and evaluation results are discussed in Section 5.

5 Result and discussion

This section discusses the result of training in performance metrics and presents one example plot of the output compared to the baseline prediction

5.1 *Training*

The models were firstly trained with WF1 data. Then the training proceeded with the data from the other wind farms data, which was sequentially added according to similar seasonality trends (see Figure 3). The model that was trained on WF1, WF2, WF5 and WF6 gave the best score overall. The data set that the model was tested on was from WF6. Figure 4 shows the learning curve of the CNN on the data set from WF1 and figure 5 displays the learning curve of the LSTM. These figures are showing the training curve in blue and the validation curve in orange. The training curve for other wind farms showed similar properties. As both figures show, the CNN at first seems to have a higher convergence, whereas the LSTM model tends to overfit the training data. However, the LSTM training loss converges to a lower value than the CNN.

5.2 *Performance metrics*

Table 2 and table 3 show the MSE, RMSE and MAE scores with the MSE loss function for baseline (linear regression), CNN and LSTM for predicting WP1 and WP6. The table show, that as the models are subsequently trained on more WF data, the models become better at predicting the wind power of each farm. However, due to difference in seasonality (see Figure 3), the combination of forecast data which is most suitable and gives close results to the regression model are: WF1, WF2, WF5 and WF6. Furthermore, the linear regression model proves to be a hard baseline model to compete with, as its scores are lower than both neural network models.

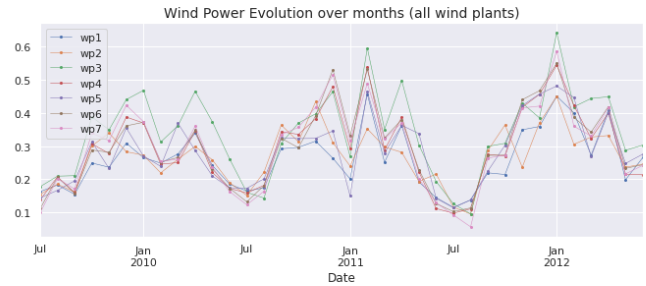

Figure 3: Seasonality for seven wind farms

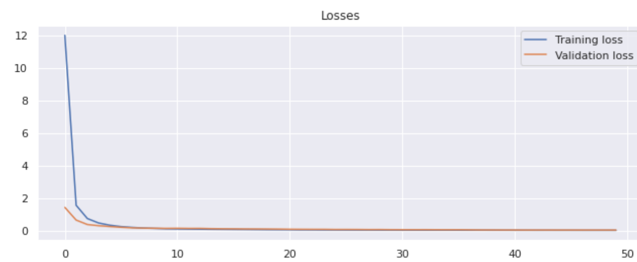

Figure 4: CNN training curve for WF1

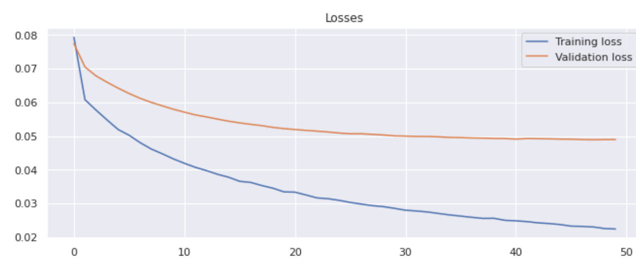

Figure 5: LSTM training curve for WF1

Code for this project is available at `https://github.com/piepear/DeepLearningproject02456`

Metric	Base line	CNN	LSTM
MSE	0.0289623	0.048691995	0.037152655
RMSE	0.1701834	0.22066262	0.19275025
MAE	0.0289623	0.1882019	0.14870939

Table 2: Result metrics for predicting 48h timeframes of WP1 (models were trained only on the WF1 dataset)

5.3 *Example - 48h ahead prediction*

The model from Setup II was subsequently trained on data accumulated from 4 wind farms (WF1+WF2+WF5+WF6) and gave the best results when predicting WP6, which is the power generated by wind farm 6. The prediction was visualized on a single 48h window, lasting from May 1, 2010 to 3 May 2010.

Metric	Base line	CNN	LSTM
MSE	0.0290785	0.036761418	0.03457796
RMSE	0.1705244	0.19173267	0.1859515
MAE	0.1265448	0.14345883	0.13681622

Table 3: Result metrics for predicting 48h time-frames of WP6 (models were trained on WF1, WF2, WF5 and WF6 datasets)

Figure 6 at the bottom shows the LSTM predictions plotted together with the actual wind power values WP6 and the predictions generated from the baseline regression model. From a first glance it seems that the LSTM is a bit better than the baseline at predicting peaks and valleys in the time-series data, but the values are a bit off in terms of magnitude, compared to the baseline predictions. However, it seems that neither the baseline nor the LSTM predictions overlap the actual WP6 values, which means a bit more could be improved in terms of predicting WP6.

The residual plot in Figure 7 shows the predicted values from Setup II (LSTM) of WP6 versus the standardized residuals. The residuals seem symmetrically distributed mostly between the lower and upper bound of ± 0.4 apart of two residuals around -0.6. The residuals are also small, meaning the predictions are not that far-off from the actual values.

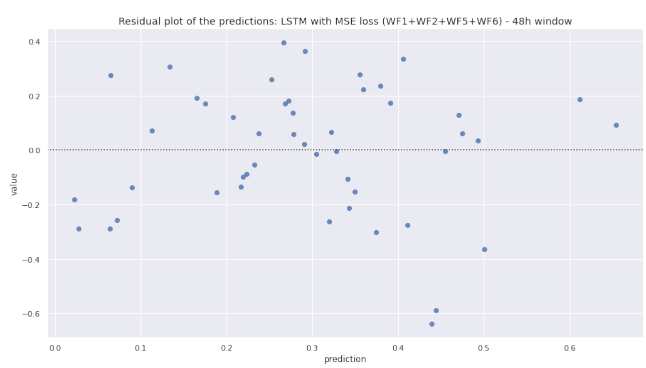

Figure 6: Residual plot of LSTM predictions vs. standardized residuals of WP6 - 48h ahead prediction (time-frame: 1 May 2010 00:00 - 3 May 2010 00:00)

Overall, based on Figure 6 and the residual plot in Figure 6, both the LSTM model from Setup II and the baseline regression model generated accurate predictions for this 48h time-frame window lasting from 1 May 2010 00:00 to 3 May 2010 00:00, which are not extremely far off from the actual values.

6 Conclusion

Overall, both the CNN and the LSTM models with MSE training loss (Setup I and II, respectively) gave satisfactory results when making predictions of the wind power on the Kaggle dataset from the Global Energy Forecasting Competition 2012. Due to the wind forecasts (WFs) being generated ahead of time, long-term predictions could be made of 48 hours ahead.

In terms of the performance metrics (MSE, RMSE and MAE) the LSTM from Setup II performed much better than the CNN from Setup I (see tables 2 and 3). This is somewhat expected since the data accumulated from the wind farms is in fact a time-series data. Unfortunately, none of the neural network architectures could beat the baseline model, which is a Linear Regression model. However, considering the similarities in the scores, both neural networks gave satisfactory results. It is highly likely that the neural network models could beat a simpler baseline, which could be a moving average prediction model, but more investigation needs to be done.

The time-frame window which was gave as an example in Section 5.3 shows that the predicted values are close to the actual wind power measurements. In terms of predicting peaks and valleys of the wind power fluctuations throughout time, the LSTM model seems more promising than the baseline (see Figure 7), but the model needs to be improved in order to predict closer to the actual values. The symmetrically distributed residual plot in Figure 6 shows that both the baseline and the LSTMs gave a satisfying accuracy.

Finally, when it comes to selecting the best model so far for the 48h ahead wind power prediction, one would recommend choosing the Linear Regression model due to its simplicity and good metric scores. However, the LSTM seems more pormising in terms of predicting the actual trends of the wind power evolution throughout time, as it might be able to detect peaks and valleys in the time-series data. Thus, more investigation needs to be done within optimizing the LSTM architecture or its hyper-parameters.

Figure 7: LSTM predictions plotted alongside linear regression predictions (green) and the actual values (gray) of WP6 - 48h ahead prediction (time-frame: 1 May 2010 00:00 - 3 May 2010 00:00)

References

[1] Xiaochen Wang, Peng Guo, and Xiaobin Huang, "A review of wind power forecasting models," *Energy procedia*, vol. 12, pp. 770–778, 2011.

[2] Dr. Tao Hong, SAS Institute, USA, "Global energy forecasting competition 2012 - wind forecasting," `https://www.kaggle.com/c/GEF2012-wind-forecasting/`, Accessed: 2021-12-22.

[3] Ilhami Colak, Seref Sagiroglu, and Mehmet Yesilbudak, "Data mining and wind power prediction: A literature review," *Renewable Energy*, vol. 46, pp. 241–247, 2012.

[4] Muttaqi KM Johnson PL, Negnevitsky M, "Short term wind power forecasting using adaptive neuro-fuzzy inference systems," pp. 1–6, 2007.

[5] Madsen H. Pinson P, "Probabilistic forecasting of wind power at the minute time-scale with markov-switching autoregressive models," pp. 1–8, 2008.

[6] OHair E Li S, Wunsch DC and Giesselmann MG, "Comperative analysis of regres- sion and artificial neural network models for wind turbine power curve estimation," vol. 123, pp. 327–332, 2001.

[7] Tsikalakis AG Katsigiannis YA, Georgilakis PS, and Hatziargyriou ND, "Improved wind power forecasting using a combined neuro-fuzzy and artificial neural network model," vol. 3955, pp. 105–115, 2006.

[8] Yona A Senjyu T, Urasaki N, and Funabashi T, "Application of recurrent neural network to long-term-ahead generating power forecasting for wind power generator," pp. 1260–1265, 2006.

[9] Song Z Kusiak A, Zheng H, "Wind farm power prediction: a data mining approach," vol. 12, pp. 275–93, 2009.

[10] Jingning Shan Gang Chen and Et al, "Research on wind power prediction method based on convolutional neural network and genetic algorithm," pp. 3573–3578, 2019.

[11] R. Wolfson, New York: Norton, 2nd ed. edition, 2012.

[12] Schmidhuber J. Hochreiter S, "Long short-term memory," pp. 1735–1780, 1997.

[13] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," 2019.