



Delft University of Technology

Probabilistic wind power forecasting combining deep learning architectures

Arends, Eric Lacoa; Watson, Simon J.; Basu, Sukanta; Cheneka, Bedassa

DOI

[10.1109/EEM49802.2020.9221929](https://doi.org/10.1109/EEM49802.2020.9221929)

Publication date

2020

Document Version

Final published version

Published in

2020 17th International Conference on the European Energy Market, EEM 2020

Citation (APA)

Arends, E. L., Watson, S. J., Basu, S., & Cheneka, B. (2020). Probabilistic wind power forecasting combining deep learning architectures. In *2020 17th International Conference on the European Energy Market, EEM 2020* Article 9221929 (International Conference on the European Energy Market, EEM; Vol. 2020-September). IEEE. <https://doi.org/10.1109/EEM49802.2020.9221929>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

<https://www.openaccess.nl/en/you-share-we-take-care>

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

Probabilistic wind power forecasting combining deep learning architectures

Eric Lacoa Arends*, Simon J. Watson[†], Member *IEEE*, Sukanta Basu[‡], and Bedassa Cheneka[†]

**Faculty of Electrical Engineering, Mathematics & Computer Science*

Delft University of Technology

Delft, The Netherlands

[†]Faculty of Aerospace Engineering, Wind Energy Section

[‡]Faculty of Civil Engineering and Geosciences

Corresponding author: s.j.watson@tudelft.nl

Abstract—A series of probabilistic models were bench-marked during the European Energy Markets forecasting Competition 2020 to assess their relative accuracy in predicting aggregated Swedish wind power generation using as input historic weather forecasts from a numerical weather prediction model. In this paper, we report the results of one of these models which uses a deep learning approach integrating two architectures: (a) Convolutional Neural Network (CNN) LeNet-5 based architecture; (b) Multi-Layer Perceptron (MLP) architecture –with two hidden layers–. These are concatenated into the Smooth Pinball Neural Network (SPNN) framework for quantile regression. Hyperparameters were optimised to produce the best model for every region. When tuned, the re-forecasts from the model performed favorably compared to other machine learning approaches and showed significant improvement on the original competition results, though failed to fully capture spatial patterns in certain cases when compared to other methods.

Index Terms—wind power forecasting, convolutional neural network, smooth pinball neural network, multilayer perceptron, numerical weather prediction

I. INTRODUCTION

System operators face the challenge of integrating variable wind power into the grid and avoiding possible power imbalances by scheduling other dispatchable generation units and calling on reserve mechanisms [3]. Wind power forecasting serves as a means to facilitate the decision-making of these operators, providing a tool for risk management in electricity markets [4]. This has stimulated research into new methodologies to make best use of Numerical Weather Prediction (NWP) products [5].

Machine learning and Artificial Intelligence (AI) have shown promise in the energy sector to assist data-driven decision making [3]. As the performance of computers improves and algorithms become more efficient, society is shifting to an era of energy digitalisation [6], where the use of Information and Communication Technology (ICT) plays a key role in the energy transition. AI has also become a favored tool to provide probabilistic wind power forecasts [1]. The use of a Convolutional Neural Network (CNN) model is described in [8] for wind power generation forecasting using NWP data,

capturing spatial patterns from relevant meteorological variables. Moreover, a Smooth Pinball Neural Network (SPNN) model is presented in [8], where an alternative is proposed to the traditional quantile deep regression model. Both architectures motivated the development of a deep neural network model described in this paper, developed by *DeepWinds*, Team 18 of the European Energy Markets (EEM) 2020 forecasting competition.

This paper describes the *DeepWinds* model, how it was implemented and its accuracy during the various rounds of the EEM 2020 competition. Furthermore, there is discussion of the challenges in developing the model, including feature selection, application to multiple price regions (climates), and how to apply the model when installed capacity is changing over time.

The structure of the remaining part of this paper is as follows: in Section II, the competition is described and an overview of the measured and forecast data is given. The methodology for the deep neural network model is explained in Section III. Section IV presents an analysis of the competition results, highlighting the performance of the *DeepWinds* model with respect to other models in the competition. The conclusions are given in Section V.

II. COMPETITION SETTING

The EEM organizers hosted a day-ahead market forecasting competition, in which teams were asked to predict the aggregated wind power of four price regions in Sweden, using a probabilistic methodology. The competition was divided into six submission rounds with every round focusing on two months of onshore wind power output during 2001 for which day-ahead forecasts were to be produced. The data provided to the competitors in order to make the forecasts consisted of three elements:

- NWP data of seven different meteorological variables.
- Aggregated wind power from the four price regions in Sweden.
- A record of the wind turbines installed in Sweden over the period of interest.

The EEM 2020 edition started on May 5, 2020 and ended on June 9, 2020. The data were released the day after every round submission, giving one week to train the models and produce new results for the following round. The ranking was published three days after the submission deadline for every round.

The data corresponding to the year 2000 were made available in advance of the competition proper, allowing participants to train their initial models and develop their forecast strategy, depending on their approach.

This first part of the data were daily netCDF format NWP model output consisting of 24 hourly values of seven meteorological variables (2m temperature, 10 m zonal and meridional wind speed components, 10 m wind gust speed, mean sea level pressure, relative humidity and total cloud cover) for ten ensemble members on a 71×169 grid covering Sweden with a spatial resolution of $10 \text{ km} \times 10 \text{ km}$. The forecasts were generated by MET Norway and archived by Greenlytics.

The first challenge was how best to utilize the multi-dimensional dataset which contained 83,993 variables per hour, accounting for 8736 hours in year 2000 (May 14, 2000 and September 26, 2000 were missing). Data cleansing and dimensionality reduction strategies were necessary to develop a model which was not computationally prohibitive.

The second part of the data consisted of the aggregated power production for the four price regions in Sweden, defined as: SE1, SE2, SE3 and SE4. The competition required quantile day-ahead forecasts to be produced for these data. Therefore, a further challenge was to derive quantile forecasts from a trained single-value output. Furthermore, a decision was required between training a single model for all price regions or training separate models for each region.

The third part of the data corresponded to a record over time of installed wind turbines in Sweden as the power capacity increased significantly during the training and forecasting time horizon of the competition (2000–2001). Therefore, there was a challenge in capturing the dynamics of the changing wind power installed capacity. Moreover, this record contained 4004 turbines accounting for 8640 MW, while in reality 4099 wind turbines were installed by that time, accounting for 8984 MW. As a consequence, the record did not entirely represent the actual conditions in which the aggregated power output was based for every price region.

The share of the installed capacity by the end of 2001 was 15.4 %, 34.8%, 30.8% and 18.9%, for the four price regions SE1–SE4, respectively. At the same time, the share of the number of turbines was 11.8%, 27.8%, 36.2% and 24.2%, respectively. Furthermore, the average terrain height in each region was 348 m, 476 m, 170 m and 73 m, respectively. It is important to note that the highest terrain location is 1003 m in SE2. This information is relevant to understand the diversity of conditions in every price region which represented a challenge to produce accurate forecasts with the limited data provided.

The accuracy of the models was measured using the pinball

loss function. In this edition of the competition, the prediction output was required to consist of nine deciles from Q10 to Q90. Equation (1) shows the formula for the pinball loss function, ρ_k^i , evaluated for each price region:

$$\rho_k^i(q_k^i, y_k) = \begin{cases} (i/100)(y_k - q_k^i), & y_k \geq q_k^i \\ (1 - i/100)(q_k^i - y_k), & q_k^i < y_k \end{cases} \quad (1)$$

where i represents the percentile to be assessed (between 10 and 90), q_k^i is the predicted power value and y_k is the observed power value at time step k . Note that this formula only gives positive values. One of the purposes of this loss function is to properly penalize over- and under-estimates [8]. The final forecast score is calculated by averaging the pinball loss function over all percentiles, price regions and time steps for the particular two-month period.

III. METHODOLOGY

The deep learning-based (*DeepWinds*) forecasting model was developed in several stages: (a) Data cleansing; (b) Feature engineering; (c) Target engineering; (d) Development of a probabilistic framework; (e) Development of the deep learning framework. These stages are explained below:

A. Data cleansing

Identifying incorrect or missing data may be necessary to avoid any bias when training a model. Using a deep learning approach can facilitate and partly automate this time-consuming task [9]. In the case of the NWP data, the cleansing approach consisted of substituting missing or incorrect values with zeros. Outliers were not filtered out as a deep CNN approach is relatively robust to a small number of such data points. In fact, it was found that only a relatively small number of forecast values needed to be substituted.

B. Feature engineering

Wind power forecasting models developed for one location may not be representative of other locations for a variety of reasons, e.g. the effects of varying terrain height, localised wind speed patterns, differences in local temperature, pressure and humidity, *etc.*, [10]. Forecast bias and accuracy is a function of these and other variables. The challenge is to decide what input variables add value to a probabilistic forecast and how to develop a model which is both accurate and parsimonious.

The first approach was to reduce the dimensionality of the input data. This was done by using the median value of the ten ensembles, as suggested in [11]. The full size grid NWP data was used to model every price region. The aim was thus for the model to use deep learning to output quantile forecasts directly.

According to [12], the accuracy of wind power predictions is seasonally dependent. In order to capture seasonal and diurnal dependencies, a time proxy was used, namely the day of year and time of day, following [14], in the form of periodic functions.

The main feature in deep learning-based models related to power production is wind speed as shown in [13], [14]. This variable is derived from the 10-meter zonal (U10) and meridional (V10) wind speed components. Table I summarizes the correlation between the input forecast variables and the power data for the four price regions.

TABLE I
ABSOLUTE CORRELATION BETWEEN FORECAST VARIABLES AND WIND POWER BY PRICE REGION - HIGHEST CORRELATIONS ARE SHOWN IN BOLD

Feature variable	SE1	SE2	SE3	SE4
Wind speed (WS)	0.523	0.659	0.696	0.560
Wind direction (WD)	0.060	0.060	0.016	0.164
Zonal 10-meter wind (V10M)	0.336	0.453	0.545	0.527
Meridional 10-meter wind (V10M)	0.416	0.512	0.511	0.338
Wind gust	0.521	0.657	0.687	0.548
Mean sea level pressure (MLSP)	0.063	0.240	0.315	0.234
Screen level rel. humidity (RH2M)	0.050	0.017	0.012	0.051
Surface temperature (T2M)	0.034	0.002	0.103	0.199
Total cloud cover (TCC)	0.065	0.149	0.244	0.197

Wind speed resulted to have the greatest correlation with wind power, as expected. On the other hand, wind direction showed almost no correlation. Although wind gust also shows a good correlation with power output, inclusion as an input to the model did not improve the forecast beyond using only the wind speed magnitude as they are already highly correlated between them (>0.97). The relative humidity shows little correlation. Pressure shows a small degree of correlation but this varies significantly by region. Consequently, only the wind speed magnitude was used as a forecast input variable in the model. The wind speed input was scaled by subtracting the mean and dividing by the variance of the training dataset to promote an efficient optimization process of the model [15].

C. Target engineering

As mentioned earlier, installed capacity per region changed over time. In order for the model to adequately cope with this variation, the power values for each price region were divided by the current installed capacity for that respective hour. This has the effect of normalizing the output values in the form of a capacity factor in order to train the model. Predicted capacity factors are then converted back into power production values for producing the final forecasts, assuming that wind turbine availability is 100%.

D. Development of a probabilistic framework

Traditional Bayesian statistics are used when more information about a forecast is required, extending the model from a deterministic to a probabilistic nature, by inferring the distribution over the dataset. Nevertheless, Bayesian methods have a high computational cost when used with large datasets and thus are unsuitable for this application [16].

In order to build a model that could be generalized for all price regions, a non-parametric approach was followed. No assumptions about the shape of the wind speed distributions were made, as this can vary in time for a given location [17]. However, the non-parametric approach requires the tuning

of additional parameters and consequently, brings additional computational costs.

The Smooth Pinball Neural Network (SPNN) architecture was used to develop a deep learning model for quantile regression that uses a non-parametric approach [18], [19] and is a combination between a Huber loss (smooth L1-loss) and a pinball loss using an objective function S_j for each j^{th} decile ($j = i/10$) with quantile value $\tau=i/100$, of the form:

$$S_j = \tau \cdot u_j + \alpha \cdot \log(1 + \exp(-\frac{u_j}{\alpha})) \quad (2)$$

The difference between the observed and predicted value for each decile, u_j , was smoothed using the parameter, α , to promote a non-convex optimization algorithm, facilitating the convergence of the model. The advantage of this approach was that optimization was based directly on the forecast evaluation metric used in the competition.

Non-parametric deep learning models can have difficulties interpreting the ranked order of quantiles. Hence, another advantage of the framework was dealing with the quantile cross-over problem: it occurs when the prediction output values for higher quantiles is smaller than lower quantiles (e.g. Q20 < Q10). This behaviour becomes common when explanatory variables are heteroscedastic [8]. Eventually, the estimations do not follow the nature of a probability distribution function, reducing the reliability of the forecast. As a consequence, a penalty factor was applied in the objective function such that it stimulates particular local minima, similar to the idea of reinforcement learning [20]. The penalty function P is given by:

$$P = \kappa \cdot \max[0, \epsilon - (q^{<\tau-1>} - q^{<\tau>})]^2 \quad (3)$$

The margin, ϵ , expresses the desired spacing between two consecutive quantile forecast values, q^τ , and $q^{\tau-1}$, while the penalty factor, κ , indicates the severity of the cross-over error. This penalty term was added to the smoothed pinball function in (2), meaning that three additional parameters had to be tuned in the model.

E. Development of the deep learning framework

The final stage consisted of expanding the SPNN framework for quantile regression, concatenating CNN and MLP architectures. Hence, the input layer consists of two branches: (a) the NWP grid data, introduced in the CNN architecture; (b) the time proxy, introduced in the MLP architecture.

Three CNN architectures were considered, namely LeNet-5 [21], AlexNet [22] and the VGG-16, the latter used to win the Imagenet competition in 2014 [23]. The motivation to introduce a CNN was to capture spatial patterns in the NWP data. On the other hand, to simplify the MLP branch, a simple architecture with two hidden layers was used (6 and 26 nodes).

Quantile forecasts were made simultaneously following the methodology in [8]. The concatenation between the CNN and the MLP takes place at the last hidden layer of both architectures. This structure is integrated in the framework of the traditional SPNN, in which the nodes of the output layer

represent the quantiles to be forecasted. Hence, the output layer corresponds to nine nodes, each having a customized optimization function based on their respective decile target (τ), given in (2). The final quantile regression loss function (QRLF) can be expanded to the sum of every unit of the dense output layer representing the nine deciles:

$$QRLF = \sum_{j=1}^9 S_j \quad (4)$$

Figure 1 illustrates the framework of the SPNN, including the concatenation of both branches.

The models were trained separately for each price region. However, the entire NWP grid of data was used to train a single model. As a consequence, the deep learning algorithm was able to capture the relationships between the input data, without introducing a spatial subset of the meteorological variables. Figure 1 shows an schematic representation of the final concatenated model, based on the SPNN quantile regression framework.

The training phase consisted of using 10 months of shuffled data from year 2000, while the testing data consisted of the remaining two months. Furthermore, a validation split was performed in the training set to evaluate both bias and variance: indicators of under- and over-fitting. To simplify the tuning process, the SPNN parameters, α and κ , were replicated from [18], and only the margin parameter, ϵ , was used to calibrate the forecasts. Moreover, the kernel random initialization used a normal distribution, while the regularization term, $\lambda = 0.0001$, was used to avoid over-fitting [18]. Regarding the activation function, the Rectified Linear Unit (ReLU) function was used. Finally, an early stopping criterion was employed instead of defining a fixed number of epochs. In this manner, once the validation error showed no further improvement, the model terminated the optimization algorithm to avoid memorizing the training dataset.

IV. RESULTS

A. Model training

The selection of the CNN architecture was performed prior to the first round of the competition. Once the competition started, the deep learning architecture was not altered. Table II shows the results of the different architectures considered comparing the predicted and observed wind power values for the two-month testing period and determining the best mean absolute percentage error (MAPE) after model tuning.

It can be seen that the simple LeNet-5 architecture clearly showed the best results for this application. The more complex architectures (i.e. AlexNet and VGG-16) generalized the data, failing to predict well the power values during the testing period. Therefore, it was decided to concatenate the LeNet-5 architecture with the MLP architecture.

Four elements (hyper-parameters) were tuned to produce the best model for every price region: the type of sub-sampling layer (maximum or average pooling), the degree of spatial dropout, the batch size and the margin to manage quantile

TABLE II
COMPARISON OF RESULTS BETWEEN DIFFERENT CNN ARCHITECTURES;
MAPE = MEAN ABSOLUTE PERCENTAGE ERROR.

Architecture	Overview	Best MAPE
LeNet-5	Conv. layer: [6, 16] Fully-connected layer: [120, 84] Kernel: 3x3 Padding: no; stride: 1	8.5%
VGG-16	Conv. layer: [16, 16, 64, 64, 128, 128] Fully-connected layer: [4096, 4096] Kernel: 3x3 Padding: no; stride: 1	18.7%
AlexNet	Conv. layer: [96, 256, 384, 384, 384] Fully-connected layer: [4096, 4096] Kernel: variable from 3x3 to 11x11 Padding: 1; stride: variable from 1 to 4	16.2%

cross-over. Table IV-A shows the results of the pinball loss metric as a function of four different models with different hyper-parameter settings. The best (lowest) values for each price region are shown in bold.

TABLE III
PINBALL LOSS FUNCTION AS A FUNCTION OF HYPER-PARAMETER VALUES
FOR EACH PRICE REGION. THE BEST CHOICE OF PARAMETERS FOR EACH
PRICE REGION IS SHOWN IN BOLD

Parameters	SE1	SE2	SE3	SE4
Model A a. Max Pooling b. Spatial Dropout = 0.25 c. Batch size = 64. d. $\epsilon = 0.002$	38.9	184.2	117.4	77.3
Model B a. Max Pooling b. Spatial Dropout = 0 c. Batch size = 64. d. $\epsilon = 0.001$	34.6	139.6	92.2	60.5
Model C a. Average Pooling b. Spatial Dropout = 0 c. Batch size = 64. d. $\epsilon = 0.001$	34.9	143.3	92.4	59.6
Model D a. Max Pooling b. Spatial Dropout = 0 c. Batch size = 32. d. $\epsilon = 0.001$	32.2	148.3	106.0	58.2

Based on the sensitivity analysis, two models were considered, namely Model B and Model D. Both models performed best with a margin, $\epsilon = 0.001$, while the best sub-sampling layer approach was Max Pooling to reduce the shape of the input data. Moreover, Spatial Dropout did not improve the score, hence only the λ term was used for regularizing the weights of the kernels. In Model B, a batch size of 64 performed best for price regions SE2 and SE3. In contrast, the best fit for price regions SE1 and SE4 was achieved with a batch size of 32, corresponding to Model D. Finally, to avoid forecast quantile values exceeding the installed capacity factor at each time step, a clipping factor between 65% to 88% was applied based on the historical training data starting from the median quantile.

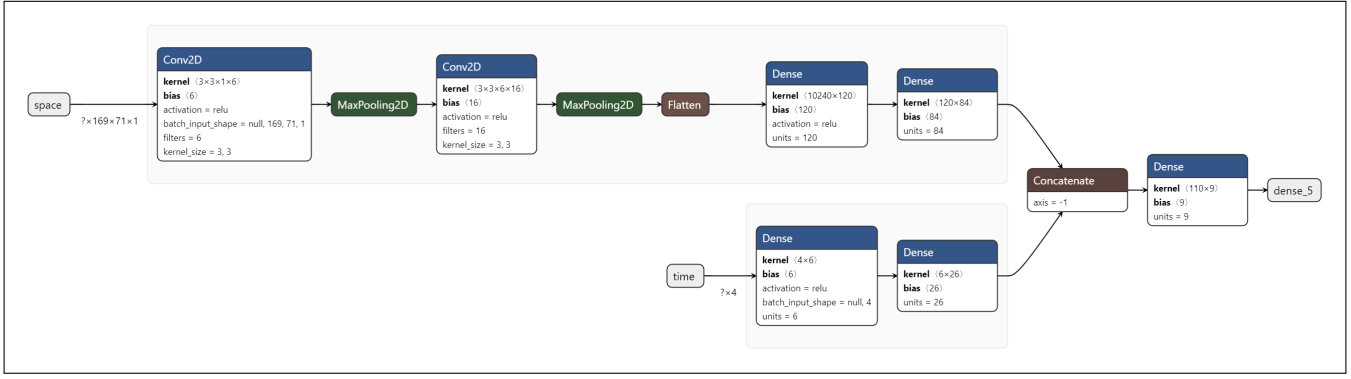


Fig. 1. A visual representation of the final model; (a) a LeNet-5 based CNN architecture for the NWP data; (b) a simple MLP for the time proxy. Both architectures are concatenated in their final hidden layer to compute the output quantiles.

B. Competition and re-forecast results

The pinball score results for the top four teams and the *DeepWinds* model are shown by round in Figure 2. For comparison, the *DeepWinds* model scores are also shown for re-forecasts using the final tuned version of the model in Round 6 and post-competition. Note that the model architecture and the feature engineering have *not* been changed.

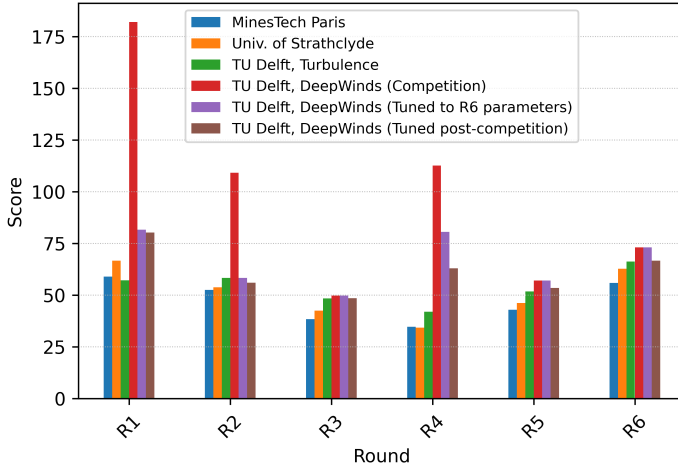


Fig. 2. Pinball scores by round for the top teams in the forecasting competition, compared with the *DeepWinds* model in three cases; (a) Official competition; (b) Tuned until round 6 parameters; (c) Tuned post-competition.

As the final score of the teams was determined by the best five submission rounds, the standings at the end of the competition are shown in Table IV, along with model type.

The *DeepWinds* model re-forecasts show competitive results with respect to the top performing teams and are a significant improvement on the model used during Rounds 1–5. However, the model was not able to capture the spatial patterns sufficiently well in rounds 1 and 4 when compared with other models by using the full size NWP grid data. Note that a separate team from TU Delft (*Turbulence*), which came third, also used a model incorporating a CNN and MLP, but the architecture was quite different to the *DeepWinds* model, and incorporated additional feature engineering.

TABLE IV
FINAL RESULTS OF THE TOP FOUR TEAMS IN THE FORECASTING COMPETITION COMPARED WITH THE *DeepWinds* MODEL; QRF = QUANTILE REGRESSION FOREST; QGAM = QUANTILE GENERALIZED ADDITIVE MODEL; GBM = GRADIENT BOOSTING MACHINE; HCM = HYBRID CNN-MLP

Rank	Team	Final Score	Model Type
1	MinesTech Paris	44.92	QRF
2	Univ. of Strathclyde	47.93	QGAM & GBM
3	TU Delft, Turbulence	51.52	HCM
-	TU Delft, DeepWinds (Post-competition)	57.55	SPNN, CNN & MLP
-	TU Delft, DeepWinds (until R6 parameters)	63.78	SPNN, CNN & MLP
-	TU Delft, DeepWinds (Competition)	80.38	SPNN, CNN & MLP

The post-competition tuned version of the model followed a re-evaluation of Model B and Model D. This showed that the accuracy of the models displayed a seasonal dependence despite the time proxy, i.e. Model D performed better in winter for price regions SE1 and SE4 but also performed better than Model B in summer for regions SE2 and SE3, and vice versa for Model B. As a consequence, better performance was achieved by having models tuned with hyper-parameters appropriate for both price region and season.

V. CONCLUSIONS

This paper has summarized the approach followed by the *DeepWinds* model to predict wind power production in Sweden using a probabilistic framework for the EEM 20 forecasting competition. The model was based on a deep learning method using the novel Smooth Pinball Neural Network (SPNN), concatenating CNN and MLP architectures.

The resulting framework provided a simple way to output quantile values from NWP input data. The non-parametric approach allowed a generalization of the model to different datasets, allowing it to be trained for different price regions separately. Moreover, ways to reduce data dimensionality and changes in installed capacity were proposed. The application of this deep learning model is suitable for mid- and long-term forecasting and can be used as a benchmark tool for other similar models.

ACKNOWLEDGMENT

The authors would like to thank the team at KTH, Sweden for organising the forecasting competition and providing help in interpreting the data. We would also like to thank Robert Eggermont for making the High-Performance Computing (HPC) cluster from TU Delft available, and by providing his technical support.

REFERENCES

- [1] Maldonado-Correa, Jorge and Solano, J. C. and Rojas-Moncayo, Marco. "Wind power forecasting: A systematic literature review", *Wind Engineering*, Early Access, pp 1–14, 2019, doi: 10.1177/0309524X1989167.
- [2] "Global energy review 2020 – analysis", International Energy Agency, apr. 2020. <https://www.iea.org/reports/global-energy-review-2020> (consulted jul. 15, 2020).
- [3] S. Mujeeb, T. A. Alghamdi, S. Ullah, A. Fatima, N. Javaid, and T. Saba, "Exploiting Deep Learning for Wind Power Forecasting Based on Big Data Analytics", *Applied Sciences*, vol. 9, nr. 20, p. 4417, oct. 2019, doi: 10.3390/app9204417.
- [4] Morales, Juan M., Antonio J. Conejo, Henrik Madsen, Pierre Pinson, and Marco Zugno. "Integrating renewables in electricity markets: operational problems", *Springer Science & Business Media*, vol. 205, 2013.
- [5] G. Kariniotakis and P. Pinson, "Data science for renewable energy prediction", presented by Smart4RES, Online, jun. 04, 2020, consulted: jun. 04, 2020. [Online]. Available in: <https://www.slideshare.net/sustenergy/smart4res-data-science-for-renewable-energy-prediction>.
- [6] S. Kr. Jha, J. Bilalovic, A. Jha, N. Patel, and H. Zhang, "Renewable energy: Present research and future scope of Artificial Intelligence", *Renewable and Sustainable Energy Reviews*, vol. 77, pp. 297–317, sep. 2017, doi: 10.1016/j.rser.2017.04.018.
- [7] "Forecasting Competition", *European Energy Markets 2020*. <https://eem20.eu/forecasting-competition/> (consulted jul. 16, 2020).
- [8] F. Rodrigues en F. C. Pereira, "Beyond Expectation: Deep Joint Mean and Quantile Regression for Spatiotemporal Problems", *IEEE Trans. Neural Netw. Learning Syst.*, pp. 1–13, 2020, doi: 10.1109/TNNLS.2020.2966745.
- [9] T. Rozario, T. Long, M. Chen, W. Lu, en S. Jiang, "Towards automated patient data cleaning using deep learning: A feasibility study on the standardization of organ labeling", *arXiv:1801.00096 [physics]*, dec. 2017, Consulted: jul. 18, 2020. [Online]. Available from: <http://arxiv.org/abs/1801.00096>.
- [10] S. S. Soman, H. Zareipour, O. Malik and P. Mandal, "A review of wind power and wind speed forecasting methods with different time horizons," *North American Power Symposium 2010*, Arlington, TX, 2010, pp. 1-8, doi: 10.1109/NAPS.2010.5619586.
- [11] Wallach, D. "When and why to predict using the mean or median of a crop multi-model ensemble." *FACCE MACSUR Reports 10.S* (2017): 37.
- [12] Foley, A. M., P. G. Leahy, and E. J. McKeogh. "Wind power forecasting & prediction methods." *2010 9th International Conference on Environment and Electrical Engineering*. IEEE, 2010.
- [13] Pinson, Pierre, and Henrik Madsen. "Ensemble-based probabilistic forecasting at Horns Rev." *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology* 12.2 (2009): 137-155.
- [14] Torres, J. M., R. M. Aguilar, and K. V. Zuñiga-Meneses. "Deep learning to predict the generation of a wind farm." *Journal of Renewable and Sustainable Energy* 10.1 (2018): 013305.
- [15] Ng, Andrew. "Machine learning. coursera." *Stanford University*, [Online]. Available: <https://www.coursera.org/learn/machine-learning> [Accessed 15 February 2020] (2016).
- [16] S. Abeywardana, "Deep Quantile Regression", *Medium*, mar. 20, 2019. <https://towardsdatascience.com/deep-quantile-regression-c85481548b5a> (consulted jul. 27, 2020).
- [17] Pinson, Pierre, and George Kariniotakis. "Conditional prediction intervals of wind power generation." *IEEE Transactions on Power Systems* 25.4 (2010): 1845-1856.
- [18] K. Hatalis, A. J. Lamadrid, K. Scheinberg, en S. Kishore, "Smooth Pinball Neural Network for Probabilistic Forecasting of Wind Power", *arXiv:1710.01720 [stat]*, oct. 2017, Consulted: jul. 18, 2020. [Online]. Available from: <http://arxiv.org/abs/1710.01720>.
- [19] K. Hatalis, A. J. Lamadrid, K. Scheinberg, en S. Kishore, "A Novel Smoothed Loss and Penalty Function for Noncrossing Composite Quantile Estimation via Deep Neural Networks", *arXiv:1909.12122 [cs, eess]*, sep. 2019, Consulted: jul. 18, 2020. [Online]. Available from: <http://arxiv.org/abs/1909.12122>.
- [20] Sutton, Richard S., and Andrew G. Barto. "Reinforcement learning: An introduction." MIT press, 2018.
- [21] LeCun, Yann. "LeNet-5, convolutional neural networks." URL: <http://yann.lecun.com/exdb/lenet> 20.5 (2015): 14.
- [22] Krizhevsky, A., I. Sutskever, and G. E. Hinton. "2012 AlexNet." *Adv. Neural Inf. Process. Syst.* (2012): 1-9.
- [23] R. Thakur, "Step by step VGG16 implementation in Keras for beginners", *Medium*, aug. 20, 2019. <https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c> (consulted jul. 18, 2020).