A novel approach to multi-horizon wind power forecasting based on deep neural architecture

Dominik Putz[a,b,*], Michael Gumhalter[c], Hans Auer[b]

[a] Siemens AG Austria, Siemensstraße 90, 1210 Vienna, Austria
[b] Energy Economics Group (EEG), Technische Universität Wien, Gusshausstraße 25-29, E370-3, 1040 Vienna, Austria
[c] AEE INTEC, Feldgasse 19, 8200 Gleisdorf, Austria

Abstract

In recent years, renewable energy sources have been installed in large numbers. Wind power in particular, a technology with very high potential, has become a significant source of energy in most power grids. However, wind power generation forecasting and scheduling remain very difficult tasks due to the uncertainty and stochastic behaviour of wind speed. This work provides a novel, powerful tool for wind power forecasting based on neural expansion analysis for time series forecasting (N-BEATS), a deep neural network approach. N-BEATS was designed as an easy-to-implement approach to solving non-linear stochastic time series forecasting problems. Additionally, a loss function is tailored to wind power forecasting to confront the issue of forecast bias. The results are compared with established models, such as statistical and machine learning approaches as well as hybrid models, using the real-world wind power data from 15 different European countries as input. Comprehensive and accurate results are obtained during this work, showing that this methodology can easily compete with other approaches and even outperform them in terms of accuracy in most cases. Additionally, the tailored loss function reduces the error significantly. The N-BEATS architecture is further customized to deliver decomposed components such as trend and seasonality, yielding interpretable outputs. These findings constitute considerable progress and provide support for decision makers.

Keywords: wind power forecasting, neural networks, deep learning, N-BEATS, pinball-sMAPE

1. Introduction

Since the global demand for electrical energy is growing while conventional fossil resources are being depleted, wind energy as a renewable source has developed rapidly and received global attention [1]. In recent years, wind power has been the fastest growing renewable electricity generation technology overall [2], [3]. Despite its many advantages in terms of environment and sustainability, wind power generation exhibits highly volatile behaviour [4]. Therefore, reliable forecasts for effective wind power generation at any time are required. This leads to a very high demand for improving forecasts in terms of accuracy and expanding forecast horizons. The current subject of research is the development of superior and more robust forecast models that are easy to implement. Very short-term wind power forecasting (VSTWPF) is essential for power system operation and planning. Forecast accuracy translates directly into financial performance on the energy market. All these reasons justify interest in new accurate methods for wind power forecasting, especially VSTWPF.

The core objective of this paper is to provide a novel approach to VSTWPF, focusing on wind power generation forecasting over a varying forecast horizon between 15 minutes and 12 hours based on the deep neural architecture N-BEATS[1]. This approach offers numerous advantages, such as being interpretable, fast to train and applicable to a wide array of topics without further specifications being required. This method is further improved by a loss function

*Corresponding author
Email address: `dominik.putz@siemens.com` (Dominik Putz)

[1] N-BEATS: a deep neural architecture based on backward and forward links with a very deep stack of dense layers for univariate time series point forecasting; https://www.elementai.com/

tailored to the N-BEATS approach which elaborates on the progress beyond state of the art. Additionally, N-BEATS is implemented in a configuration that allows the interpretation of the individual forecast components. This method can also be classified as a possible meta-learning approach, which, however, will not be investigated in detail as it is outside the scope of this research.

The paper is structured as follows: Section 2 presents the state of the art regarding very short-term wind power forecasting. Section 3 describes the N-BEATS approach to VSTWPF. Section 4 reports the results of the implemented approach and provides a sensitivity analysis with a focus on different loss functions and forecast horizons. In Section 5, the results are compared with those of other state-of-the-art forecasting approaches and discussed. Finally, a conclusion is provided in Section 6.

2. State of the Art

2.1. Literature review

A recent review of literature on this topic [5] finds that wind power forecasting methods can be divided into two major groups: physical and statistical approaches. Physical methods use physical laws that govern the atmosphere behaviour and rely on extensive meteorological information to estimate the local wind speed and direction [6], [7], [8]. Statistical methods use extensive historical data and optimise model parameters in order to minimise the error between the predicted and the observed values [9]. The statistical approaches have been proven to deliver more accurate results for very short-term prediction [10], [11], as long as overfitting issues are avoided [12].

The statistical approaches can be further categorised into classical models [13], machine learning (ML) models and hybrid models. Classical models are often limited in terms of adaptability. As a result, researchers have become increasingly interested in ML algorithms. The neural networks (NN), which are excellently researched in the field of forecasting, are a prime example thereof. They offer great advantages, such as modeling nonlinear relationships, learning from data and strong parallelisation. A large variety of reliable approaches based on neural networks are shown in [1], [14], [15], [9]. However, due to the considerable success of deep learning in other applications this architecture has also been applied to the forecasting of wind power.

deep learning includes modern NN architectures, which are composed of the combinations of fundamental structures such as multilayer perceptrons, recurrent NNs (RNNs) and convolutional NNs (CNNs). They use sophisticated mechanisms for learning and are therefore are far more complex than simple neural networks. The long short-term memory (LSTM) was proposed [16] to adress the problem of the vanishing or exploding gradient that occurs during the learning process of RNNs. An LSTM consists of a cell and several non-linear gates that control the information inside the cell and choose which data should be kept and propagated to the next time step. The success of LSTMs is evident, including in forecasting. It is shown that they deliver better results than ML models, such as ARIMA, support vector machine and classical NNs [17]. One reason for the big success of LSTMs is that they can be combined impressively well with other methods resulting in so-called hybrid approaches.

Currently, hybrid models are considered as the most promising approaches, further substantiated by the fact that an ES+LSTM (exponential smoothing) approach, which is a hybrid model, won the M4 competition [18], [19]. The M4 competition is the continuation of three previous ones intended to identify the most accurate forecasting method(s) for different types of predictions. Hybrid approaches for wind power prediction that deliver satisfactory results are based on LSTMs and signal decomposition [20], [21], [22]. Independently, other architectures have been proposed, such as the WaveNet architecture [23] for speech synthesis, which uses so-called dilated causal convolutions to learn the long range dependencies.

Another architecture has been introduced, based on the so-called attention mechanism developed for sequence to sequence learning [24], [25]. This approach uses encoder-decoder architectures, where the encoder (RNN) learns a representation of the input while the decoder (RNN) is trained to predict the target sequence one step at a time using the representation learned by the encoder. Inspired by the success of attention models, a so-called Transformer model has been developed [26], that removes RNNs altogether and uses attention, in combination with feed-forward NNs to achieve state-of-the-art results. In addition, this proposal has already been improved for forecasting [27] as well as for natural language processing, such as Bidirectional Encoder Representations from Transformers (BERT) [28].

2.2. Meta-learning

Meta-learning, also known as learning how to learn, has recently emerged as a potential learning paradigm that can absorb information from one task and generalise that information to unseen tasks proficiently [29], [30], [31]. This structure is helpful in real-world applications for the following reasons:

- Sufficiently large datasets may be unavailable or contain gaps with missing information.
- ML paradigms can easily be broken when trying to handle uncommon situations that humans are able to handle comfortably, leading to undesired outcomes.
- It is possible to learn something new without training the model from the beginning due to a certain degree of similarity to the base dataset.

2.3. Most promising forecast approaches

So far, a wide variety of approaches has been applied to wind power forecasting that hybridise or build upon some of the most successful classical methods and have led to the discovery of completely new areas of ML. The following state-of-the-art architectures are currently considered the most promising [32]:

- The expansion of hybrid models and further research thereof with advanced LSTMs as their core component have great potential [32]. For instance, using optimised Wavelet Transformation, feature selection, LSTM and crow search algorithm for forecasting delivers outstanding results [20], and so do similar approaches [22].
- The principle of dilated causal convolutions is used by the WaveNet architecture [23], [33]. It offers very efficient training due to the use of high parallelism. This advantage increases the WaveNet's competitiveness against common RNN architectures.
- The attention mechanism [24] and particularly the transformer [26], where the mechanism is extended to intra- or self-attention to learn where to focus on in order to get good feature representations [27].
- Pure deep learning approaches, such as N-BEATS. It is a deep neural architecture based on backward and forward residual links and a very deep stack of fully connected layers. The architecture has a number of desirable properties, being interpretable, applicable without modification to a wide array of target domains, and fast to train. One conclusion of the M4 was that hybrid statistical models are superior, while pure ML models may offer one or two pleasant surprises but only by a small margin [34]. This was further evidenced by six of the pure ML models submitted to the competition not even meeting the competition benchmark. Nevertheless, a recent study shows that N-BEATS is capable of achieving higher forecast accuracy than the winner of the M4 competition [35].

2.4. Progress beyond state of the art

The progress of this work, which goes beyond the current state of research, is outlined in the following items:

- The N-BEATS architecture is applied on VSTWPF for the first time since the N-BEATS algorithm gained attention due to its remarkable results.
- It is one of the first attempts to model an interpretable time series forecast using deep learning methods in the field of wind power forecasting. The approach is parameterised in such a way that the individual parts of the result like trend and seasonality are interpretable while not having any noticeable impact on the forecast accuracy. Current deep learning approaches often have difficulties in providing interpretability of results. Either this possibility does not exist at all, or it is associated with an increased computational effort or a decrease in accuracy.
- A customized loss function is proposed that is well suited for the use in wind power forecasting. With the implementation of a loss function that is optimally designed for the application, a decisive advantage of deep learning can be exploited. The first-time usage of a so-called pinball sMAPE error metric in a deep learning architecture provides reliable and exceptionally accurate very short-term forecasts results in the short term.

3. Methodology

This section is structured into three parts. Firstly, in Section 3.1, the basics of the N-BEATS approach are explained. This includes the deep learning architecture and how it can be interpreted. In Section 3.2, a detailed mathematical description is provided accompanied by a new loss function for N-BEATS to tackle the forecast bias.

3.1. N-BEATS

The N-BEATS architecture itself does not rely on time-series-specific feature engineering or input scaling. Instead, it uses a small set of key principles. For instance, it does not treat forecasting as a sequence-to-sequence problem, but rather as a non-linear multivariate regression problem. This leads to the basic building block which has a fork architecture and is shown in Figure 1.

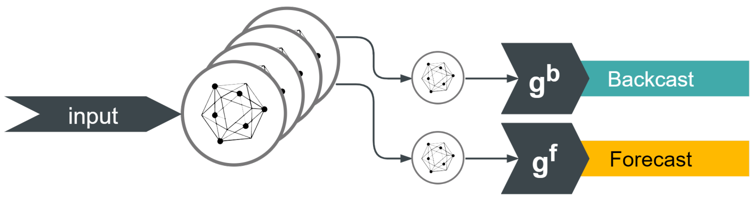

Figure 1: The architecture has two residual branches, one running over backcast prediction of each layer and the other one is running over the forecast branch of each layer. Basically the backcast branch can be understood as sequential analysis of the input time series. The basic block uses a lookback sample as input for the stacked dense layers network with ReLu activation. This network delivers two coefficients as output Θ^b, Θ^f, which are fed into the basis layers following mapping of $g^{f,b}$ to retrieve the forecast and backcast.

The basic block has an input $\mathbf{x}_l$ and two output vectors $\hat{\mathbf{x}}_l$, $\hat{\mathbf{y}}_l$ where the length of the input is a multiple of the forecast horizon. The output vectors describes the block's forward forecast $\hat{\mathbf{y}}$ and the block's best estimate which is the so-called backcast $\hat{\mathbf{x}}$ [35]. The backcast represents the contribution to the decomposition of the input. Thus, it learns the parameters of the context. The input of the l-th block $\mathbf{x}_l$ are residual outputs of the previous blocks. In particular, this network consists of fully-connected (dense) layers with a rectified linear unit (ReLu) [36] regressor shown in Equation 1 with weights $\mathbf{W}_{r,l}$ and bias $\mathbf{b}_{r,l}$, referring to $\mathbf{x}$ as the input of the architecture, using residual blocks and layer superscripts (r and l respectively) and denoting the fully connected layer with weights $\mathbf{W}_{r,l}$ and bias $\mathbf{b}_{r,l}$.

$$\mathbf{h}_{r,l-1} = \textsc{ReLu}(\mathbf{W}_{r,l}\mathbf{x}_{r,l-1} + \mathbf{b}_{r,l}) \tag{1}$$

The output is forked and fed into the basis layer network to retrieve the forecast and the backcast predictors of expansion coefficients Θ_l^f and Θ_l^b, shown in Equation 2.

$$\Theta_{r,l}^{f,b} = \mathbf{W}_{r,l}(\mathbf{h}_{r,l-1}) \tag{2}$$

They are projected on $g^{b,f}$ consisting of the set of basis functions $\mathbf{v}_i^{b,f}$ and summed up to obtain the results $\hat{\mathbf{x}}_l$ and $\hat{\mathbf{y}}_l$ shown in Equation 3 and Equation 4.

$$\hat{\mathbf{x}}_l = \sum_{i=1}^{dim(\Theta_l^b)} \Theta_{l,i}^b \mathbf{v}_i^b \tag{3}$$

$$\hat{\mathbf{y}}_l = \sum_{i=1}^{dim(\Theta_l^f)} \Theta_{l,i}^f \mathbf{v}_i^f \tag{4}$$

The residual principle is used to stack many layers together. Basically, the classical residual architecture adds the input of the stack of layers to its output before passing the result to the next stack which adds the input of the stack of layers to its output [37]. This architecture has already been extended by introducing extra connections from the output of each stack to the input of every other stack that follows it [38]. On the one hand this extension improved the trainability of deep neural network architectures. On the other hand they result in network structures that are difficult to interpret. The proposed architecture was enhanced to provide interpretability, shown in Figure 2 [35]. In general the skip connections facilitate to determine whether the intermediate layer is useful or not. In this architecture the skip connections are modelled in a different way, to make subsequent blocks have an easier job forecasting by removing the backcast outputs from the next block's inputs. It is actually similar to an unrolled LSTM, where the skip connections act like forget gates in an LSTM in order to remove information that is not needed. It passes the processed inputs to the next block, facilitating the preparation of more accurate forecasts. At the same time, each block has a forecast output that is added up with subsequent forecasts in the block to provide a combined forecast. It is possible to stack hundreds of layers and residual blocks effectively using this principle.

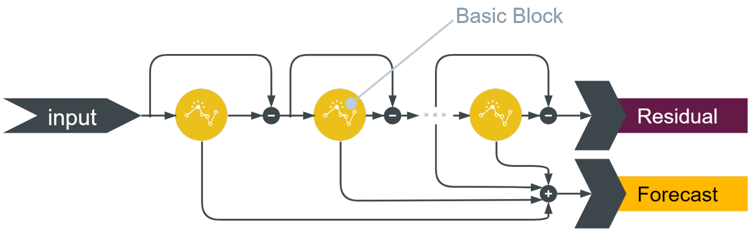

Figure 2: The basic blocks are multi-layer fully connected networks with ReLu activation function. They provide the expansion coefficients $\Theta^{b,f}$ and are connected according doubly residual stacking architecture.

In contrast to classical approaches deep learning approaches for time series forecasting often suffer from lack if interpretability. This is one of the most challenging obstacles when it comes to applying those approaches in practice [39]. N-BEATS can be made interpretable by setting the functions $g^{b,f}$, that can be either learned or instead engineered to account for different effects such as trend and seasonality. By changing the mapping functions $g^{b,f}$ for $\Theta^{b,f}$ to a trend and seasonality form makes the stack outputs interpretable, shown in Figure 3. A typical characteristic of trend is that most of the time it is a monotonic function, or at least a slowly varying function. In order to obtain this behaviour $g^{b,f}$ is set to be a polynomial of small degree, a function slowly varying across the forecast horizon. To model seasonality a cyclical, recurring fluctuation is required. An intuitive choice for a cyclical function is the Fourier series.

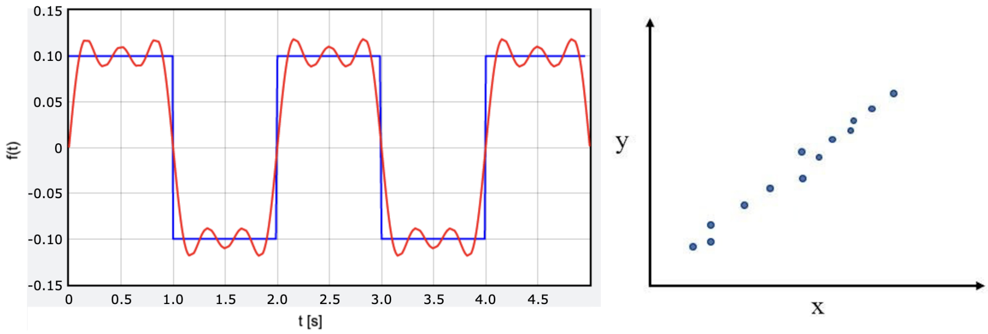

Figure 3: Schematic example for a cyclical or monotonic functions $y(x)$ for $g^{b,f}$.

The output components of the model can be separated and analysed. By knowing the nature of each basis layer, the user can estimate the contribution of each component, since the total global output is a simple sum of the partial outputs of each block. Thus providing interpretability. It was observed that the impact of this change on the error is negligible. It is similar to how the hidden state of an RNN is shared across all time steps. In addition to interpretability and accuracy benefits, as measured on several well-known datasets, the model is very fast to train and easy to apply.

Consequently, N-BEATS uses a dense layer as a multivariate regression block with a ReLu for non-linearity, which gets repeated many times. This architecture is actually very similar to an unrolled LSTM, where skip connections act like forget gates in LSTM to remove unneeded information and pass the processed input to the next block, facilitating the production of better forecasts.

3.2. Loss Function

The most used error metrics for forecasting are the mean absolute percentage error (MAPE) shown in Equation 5 and the symmetric mean absolute percentage error (sMAPE) shown in Equation 6. These were also used in the M4 competition [34].

$$\text{MAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \frac{|y_t - \hat{y}_t|}{|y_t|} \tag{5}$$

$$\text{sMAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \frac{|y_t - \hat{y}_t|}{|y_t + \hat{y}_t|\,/2} \tag{6}$$

Both are similar in that they normalise the absolute difference between prediction and observed values. The approach may produce more accurate results, because training, validation and performance error metric goals are identical and ideally aligned by using MAPE during training as well as for performance evaluation. Nevertheless, there occur two main issues:

- Firstly, the denominator $(y_t + \hat{y}_t)$ can become negative or even 0. In the case of wind power forecasting, 0 can occur and has to be treated separately. In brief, both nominator and denominator become 0, a case that is basically undefined.

- Secondly, the sMAPE treats over- and underprediction unequally. As an example for underprediction, if the observed value is 100 and the predicted value 90, then the sMAPE delivers 5.26%. By contrast, a target value of 100 and predicted value of 110 constitutes an overprediction and delivers a sMAPE of 4.76%. There are modifications of the sMAPE that allow to measuring the direction of the bias, which provides additional information about the quality of the result.

In this work it, is found that during backtests the models tend to have a positive bias. A solution for this is for example the pinball function, shown in Equation 7 [18]. It is an asymmetric function, that penalises actual values that are above and below a certain quantile τ in different ways in order to counteract the bias.

$$L_t = \begin{cases} (y_t - \hat{y}_t)\,\tau & \text{if } y_t \geq \hat{y}_t \\ (\hat{y}_t - y_t)\,(1-\tau) & \text{if } \hat{y}_t > y_t \end{cases} \tag{7}$$

The τ parameter can be adjusted, and it is advised to keep it low to avoid overforecasting. The basic pinball loss is an important loss function on its own; minimizing it produces quantile regression [18]. Setting $\tau \in (0{,}0.5)$ tends to compensate overestimation bias, and setting $\tau \in (0.5{,}1)$ tends to compensate under-estimation bias. In this work, an adaptation of the pinball function (pinball-sMAPE) shown in Equation 8 as a loss function within the N-BEATS is introduced. This is a novel solution for N-BEATS to alleviate the well-known bias problem. A convenient feature of NN-based systems is used: the simplicity of creating a loss function aligned with any business/scientific targets.

$$P_t = \frac{100\%}{n} \sum_{t=1}^{n} \begin{cases} \frac{(y_t - \hat{y}_t)}{(y_t + \hat{y}_t)}\tau & \text{if } y_t \geq \hat{y}_t \\ \frac{(\hat{y}_t - y_t)}{(y_t + \hat{y}_t)}(1-\tau) & \text{if } \hat{y}_t > y_t \end{cases} \tag{8}$$

In the case of the pinball-sMape the denominator becoming 0 could only occur if the actual and predicted values are both 0 at the same timestep, since only non-negative values are allowed. All $y_t = 0$ rows are dropped in order to prevent division-by-zero errors. This approach does not have a noticeable effect on the model because there exist hardly any of such cases in the used datasets. This can be explained by the fact that the datasets show aggregated numbers from several wind farms across a country and an occurence with no generation at all is rare. The majority of zero generation values can be traced back to missing or invalid measurement values.

4. Experiments and results

In this section, the proposed N-BEATS model for STWPF is applied to the real-world datasets described in Section 4.1. Additional models based on classical statistical methods and machine learning methods are implemented to compare them with N-BEATS in terms of accuracy. These models are briefly described in Section 4.2. The results regarding accuracy are shown in Section 4.3.

4.1. Dataset and Training

Real-world open-source[2] wind power datasets from 15 different European countries [40] are used and can be found attached in the Appendix. Each data set represents the aggregated wind power of a country that is used and processed by control area operators. Currently, time series are mainly processed hourly. However, the trend is moving to finer time intervals. Therefore, data sets with a 30-minute and 15-minute resolution have also been examined:

- 15min (01/01/2020 - 30/09/2020): AT, DE, NL
- 30min (01/01/2020 - 30/09/2020): CY (with gaps), GB, IE
- 60min (01/01/2019 - 30/09/2020): DK, ES, FI, FR, GR, IT, NO, PL, RO

The dataset of CY has some gaps in the history, and it is of interest to see how well the models can handle such cases.

The proposed method uses only windpower time series as input since it is a univariate time series forecasting architecture. The input is a time series of consecutive measured wind power values. N-BEATS does not process exogenous factors and influencing quantities such as wind speed. As a result, depending on the configuration, the predicted wind power for the next time step or a whole time series for the next time steps is obtained. In addition to this, further result components such as trend and seasonality are delivered.

Datasets are split into train, validation and test subsets. Table 1 shows the dates where these splits are located within the datasets for 15min, 30min and 60min time sets. In the first step the time series gets filtered to replace missing or NaN entries with 0. After splitting the datasets for each country a model is fitted with training and validated with validation data which leads to 15 different trained models. For performance evaluation the test sets are processed into multi-step time windows consisting of analysis and subsequent forecast time series (measured values). In general, the analysis window has multiple times the length of the forecast time series. The proposed approach delivers the forecast time series dependent on analysis time series. The predicted time series is followingly compared to the actual one to assess accuracy.

time resolution	countries	set	begin
		train	01/01/2020
15 minute	AT, DE, NL	validation	30/06/2020
		test	15/08/2020
		train	01/01/2020
30 minute	CY (with gaps), GB, IE	validation	30/06/2020
		test	15/08/2020
		train	01/01/2019
60 minute	DK, ES, FI, FR, GR, IT, NO, PL, RO	validation	28/02/2020
		test	15/06/2020

Table 1: Split of datasets into training for fitting the model, validation for hyperparameter tuning and test to assess performance.

N-BEATS is implemented in Python[3] with *tensorflow* [41] as well as in *PyTorchForecasting* [42]. The learning progress and results are visualised via *TensorBoard* [41]. Table 2 lists the configuration of the model.

[2]https://open-power-system-data.org/
[3]https://www.python.org/about/

parameter	value
optimizer	Adam
tensorflow	v2.6
PyTorchForecasting	v0.7
learning rate	optimised by PyTorch Lightning
max epochs	50
batch size	128
early stopping	true
reduce on plateau patience	1000
share stacks	true
stack types	trend + seasonality
weight decay	0.01
max. lookback horizon	variable - 24 time steps (6h-48h)
forecast horizon	variable - 4 time steps (15minute-12h)
shuffling of samples	true
hidden dense layers	512
layers in residual block	4
loss function	pinball sMAPE

Table 2: Overview of the parameters for the N-BEATS approach.

4.2. Models

The models that are used for comparision are outlined below.

- ARIMA - Autoregressive Integrated Moving Average $ARIMA(p, d, q)(P, D, Q)_m$ model implemented via *statsmodels.tsa.arima.model.ARIMA* from *statsmodel* in Python. A seasonal ARIMA model is used where m refers to the number of periods in each season and P,D,Q refer to the autoregressive, differencing, and moving average terms for the seasonal part of the ARIMA model.

- MLP - multilayer perceptron, which is a feed forward NN with a single hidden layer. In general, this is the most commonly used NN with an activation function. MLP utilises a supervised learning technique called backpropagation for training. For activation, the commonly used sigmoidal function is employed. The implementation is chosen through *tensorflow* in Python [43].

- LSTM - a long-short-term memory, which can be classified as an RNN in the DL sector, implemented via *tensorflow* in Python. In contrast to standard MLP architecture, the LSTM has feedback connections for enhancement and avoids the vanishing of the gradient. The cell has the ability to forget part of its previously stored memory and replace it with part of the new information. In general, an LSTM consists of a cell, input gate, output gate and forget gate. The cell remembers information and all the other gates control the flow of information into and out of the cell. LSTM became very popular for time series forecasting due to its robust results. It is widely used and researched for VSTWPF.

- WT-LSTM - wavelet transformation with LSTM as hybrid model implemented via *pywt* and *tensorflow* in Python. This hybrid approach delivers significantly more accurate results compared to conventional models. In addition, the M4 competition stated that hybrid approaches will be more frequently used in the future due to their great potential. A prime example thereof is the WT-LSTM, where the Wavelet transformation is used to examine the stochastic nature of wind power. This leads to a decomposition where breakpoints and discontinuities are provided by the WT. Additional techniques, such as feature selection are used to further improve the accuracy [20].

- LSTM-MSNet - LSTM with classical decomposition and multiple seasonal patterns (MSNet) implemented via *tensorflow* in Python [44]. Its superiority lies in the fact that it is a globally trained LSTM, which means that a single prediction model is built across all the available time series to retrieve the so-called cross series knowledge of related time series. This can be further improved by including multi-seasonal decomposition.

- ES-RNN - exponential smoothing with an RNN, which is a multivariate hybrid DL algorithm is implemented via *tensorflow* in Python [18]. The ES decomposes the time series into level, trend and seasonality components. The RNN is trained with all series, has shared parameters and is used to learn common local trends among the series while the ES parameters are specific to each time series. The models are combined by including the output of the RNN as the local trend component in the ES model.

4.3. Results

Samples of forecasts with different forecast horizons are shown in Figure 4. Table 3 provides an overview of the forecasting metrics for Germany. The mean absolute percentage error (MAPE), symmetric mean absolute percentage error (sMAPE), mean percentage error (MPE), R2 score and mean average absolute error (MAE) are used as metrics.

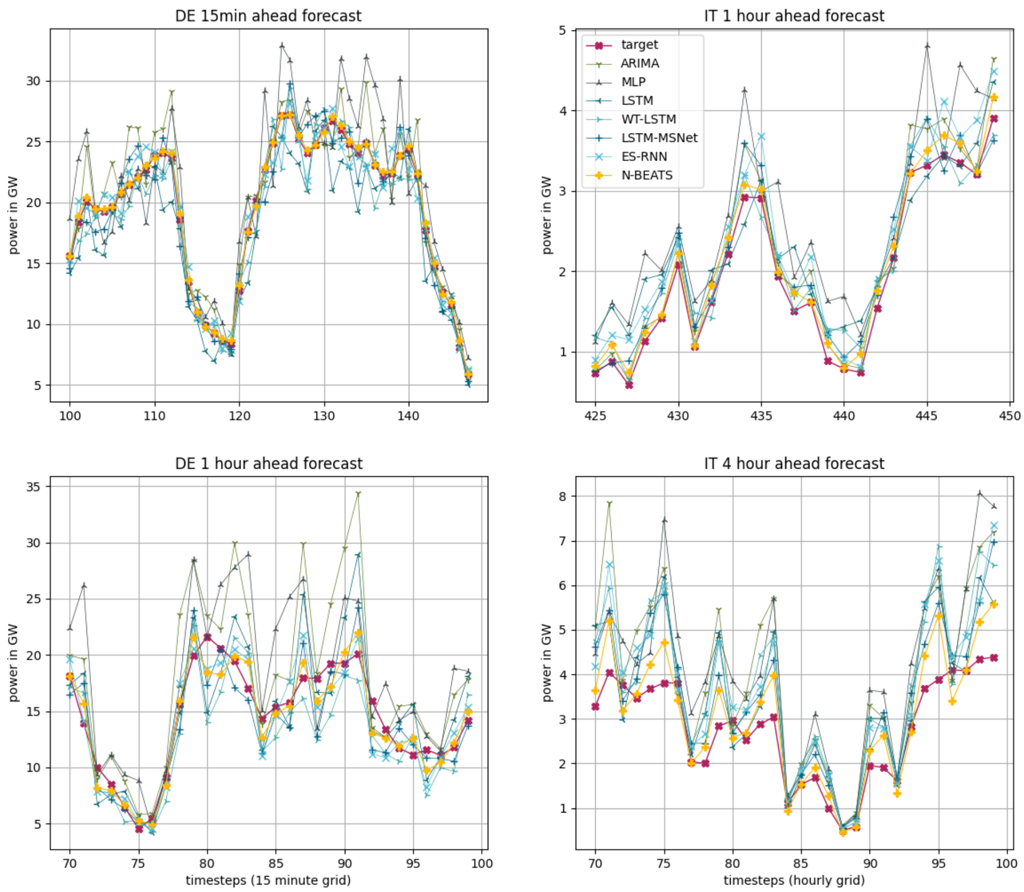

Figure 4: Top left figure shows a sample of a 15 minutes ahead forecast (Dataset with 15 minutes time resolution). Bottom left figure shows a sample of a 1 hour ahead forecast (Dataset with 15 minutes time resolution). Top right figure shows a sample of a 1 hour ahead forecast (Dataset with 1 hour time resolution). Bottom right figure shows a sample of a 4 hour ahead forecast (Dataset with 1 hour time resolution).

model	MAPE in %	sMAPE in %	MPE in %	R2 score
ARIMA	7.83	5.25	-2.22	0.965
MLP	15.32	9.37	-2.87	0.934
LSTM	12.11	7.21	-3.66	0.957
WT-LSTM	4.71	4.12	-1.26	0.982
LSTM-MSNet	4.22	3.89	-1.09	0.986
ES-RNN	4.04	3.67	-0.99	0.991
N-BEATS	**3.98**	**3.34**	**-0.56**	**0.998**

Table 3: Overview of the forecasting metrics for German dataset with a forecast horizon of 15 minutes. The N-BEATS results are highlighted.

The MPE is a metric to evaluate over- and underprediction while the MAPE is a metric for overall accuracy. A positive bias means underprediction and vice versa. The most remarkable result to emerge from the data is that N-BEATS outperforms all other used models in terms of accuracy with a MAPE of 3.98%. Generally, a MAPE below 4% is considered as major improvement. The hybrid model approaches deliver similar accuracy with ES-RNN as the second most accurate model with a MAPE of 4.04%. N-BEATS also delivers the lowest bias with an MPE of -0.56. In Section 4.4.1 other loss functions for N-BEATS are examined and it is shown that the pinball sMAPE as the selected loss function overall improves the approach. It has been observed that a τ of 0.375 delivers the most accurate results accross all datasets.

Figure 5 displays the MAPE for each country. The table shows that N-BEATS delivers stable and accurate results for most countries and that it is most accurate approach for 10 out of 15 countries. Despite CY having some gaps in its history, there is no significant impact on the forecast accuracy since the error metrics are in the same range as for the other countries.

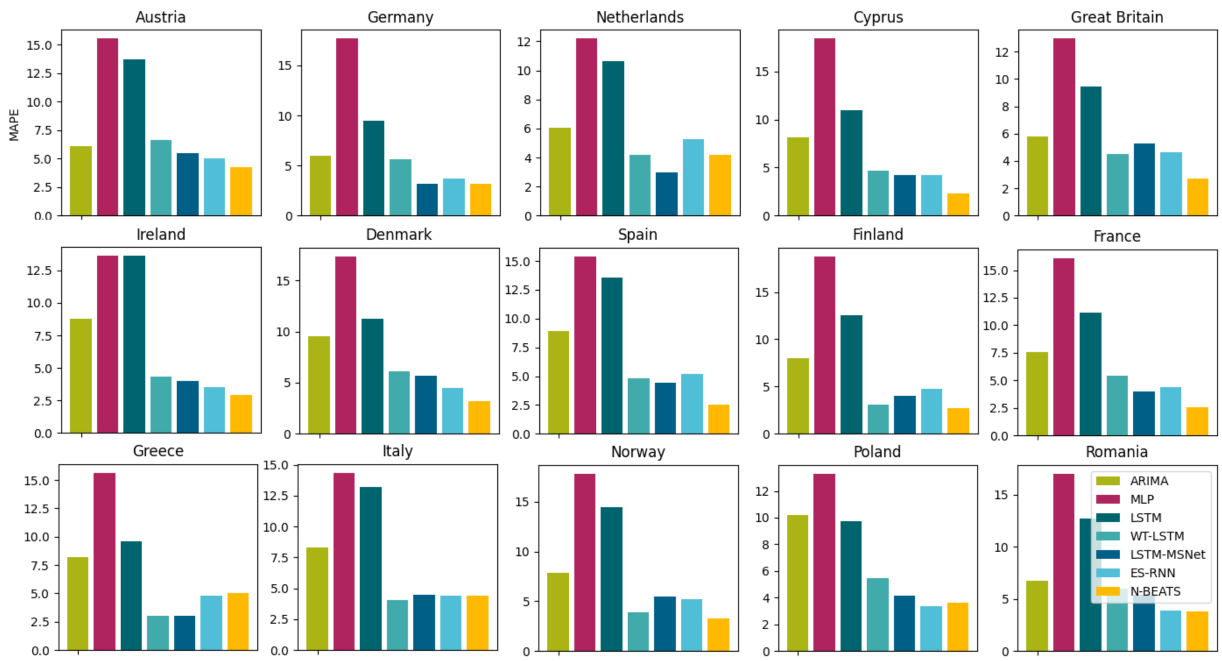

Figure 5: MAPE for each country.

The forecast error varies throughout the year and hour of day as shown in Figure 6. During spring and autumn the forecast inaccuracy peaks. This is because the wind often fluctuates the most during these periods. The fact that the wind is most discontinuous during these seasons obviously makes forecasting more difficult. This behavior is highly dependent on location. Similar behavior is observed by examining the dependence of the forecast error on time of day. Generally, stronger winds do not occur until the afternoon, after the sun has warmed the ground and warmer air masses rise. This results in more turbulence, which increases the difficulty of forecasting. Overall the approach delivers robust results with minor variation since the error fluctuations are within the range of approximately 1% MAPE.

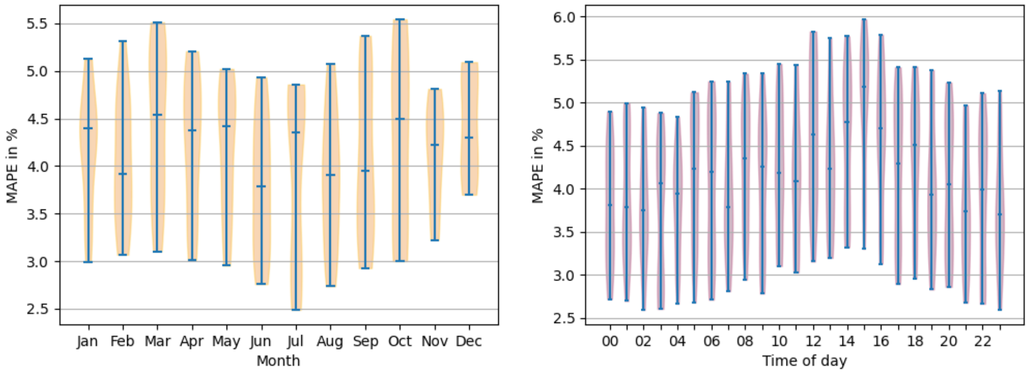

Figure 6: Forecasting error in relation to time of the year (month) and time of day (hour).

4.4. *Sensitivity analysis*

This section examines the impact of varying some model parameters, such as different loss functions and time resolutions of datasets on the result in terms of accuracy.

4.4.1. *Different Loss Functions*

Different loss functions also provide different results in terms of accuracy. Table 4 shows the MAPE for the N-BEATS model for different loss functions. The result shows that the pinball sMAPE function significantly improves the accuracy.

loss function	MAPE
MAE	7.72
MAPE	9.18
RMSE	12.25
sMAPE	8.78
pinball sMAPE, $\tau = 0.25$	9.62
pinball sMAPE, $\tau = 0.375$	**3.98**
pinball sMAPE, $\tau = 0.5$	8.78

Table 4: Sensitivity analysis of the loss function for N-BEATS. The analysis is carried out with the Germany dataset and a forecast horizon of 15 minutes.

4.4.2. *Time Resolution*

In general, historical time series occur in different resolutions. Often, an intermediate step exists to interpolate the time series to the desired resolution. The most commonly used time resolutions are 15 minutes, 30 minutes and 60 minutes. Table 5 summarises the errors at different time resolutions and forecast horizons.

Figure 7 reports the coefficient of determination for Germany for each approach. It was noted that some approaches (ARIMA, MLP) tend to overpredict more than others (LSTM, WT-LSTM, LSTM-MSNet). The developed architecture, however, is in most cases only accompanied by a relatively small overprediction, which depends on the

resolution	15min	30min	1h	2h	4h	6h	8h	10h	12h
15 min	3.78	5.99	7.98	13.89	17.23	22.51	27.47	32.88	36.33
30 min	-	4.04	6.48	11.72	14.37	19.94	26.92	31.11	34.11
60 min	-	-	4.12	9.27	12.76	18.34	24.83	30.72	33.88

Table 5: Sensitivity analysis of the time resolution for N-BEATS. The forecast horizon varies from 15 minutes to 12 hours. For the time resolutions of 15 and 30 minutes, only the corresponding data sets were examined. For the others, all data sets were examined and the result values are calculated by averaging them. Results are displayed in MAPE percentages.

data set. For the selected example forecast in Figure 4, it can be seen that N-BEATS also tends to overpredict for Italy dataset. In contrast, it was observed that for some other data sets this issue is negligible. Overprediction can be dealt with to a large extent by a suitable selection of τ. However, this parameter has to be tuned for each model and cannot be determined in general.

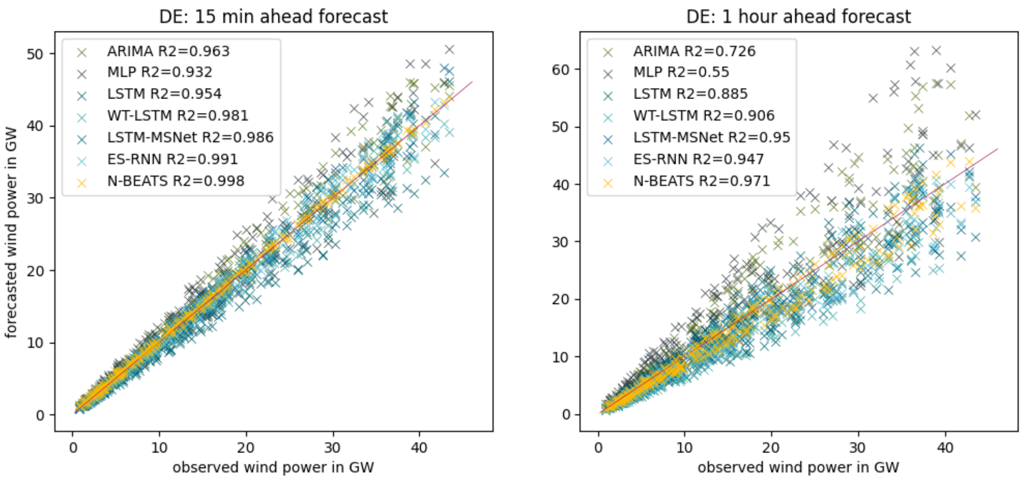

Figure 7: Scatter plot of forecasted vs observed wind power for all implemented models. Left figure displays the coefficient of determination for forecast horizon of 15 minutes for Germany. Right figure displays the coefficient of determination for forecast horizon of 1 hours for Germany.

Figure 8 shows the forecast error distributions of all results by varying the forecast horizon from 15 minutes up to 12 hours ahead. The analysis horizon is set as a multiple of the forecast horizon. Several tests have shown that an analysis period of 4 to 6 times the forecast horizon delivers the best results. After comparing the results with similar publications in this field, it can be concluded that the accuracy of the results of the proposed architecture is exceptionally good for very short-term results, in the range of 4 hours or shorter [1]. Moreover, it was observed that the error varies greatly for longer forecast horizons and is highly dependent on the dataset.

5. Discussion and synthesis of results

The evidence in this work demonstrates that N-BEATS is a new, valuable and pure DL approach for STWPF. It can compete and outperform statistical and classical ML as well as hybrid models. This work tailors the N-BEATS approach by customising a pinball loss function which is a cutting-edge solution to the forecast bias.

Considerable progress has been made with regard to interpretability. One of the most common criticisms of deep learning methods for time series is that they are a black box and the inner processes are not intuitively interpretable. Thus, it is not possible to understand how the result is obtained, in contrast to classical models such as ARIMA, the N-BEATS forecast is discomposed into distinct, human-interpretable outputs. These outputs can be used by utilities

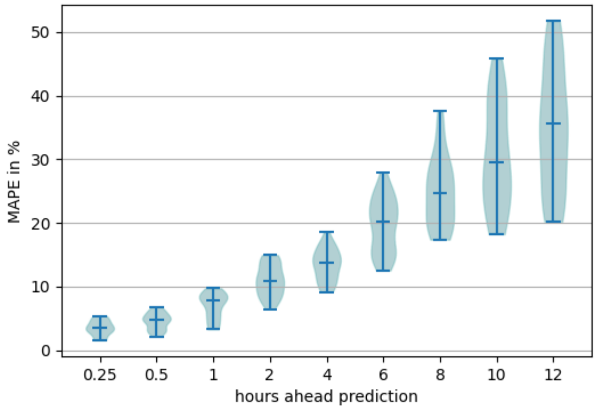

Figure 8: The MAPEs for all countries are depicted as distribution for the corresponding forecasting horizon to be predicted as well as the median and extremas for the 15-minutes, 30-minutes and hourly sample rates.

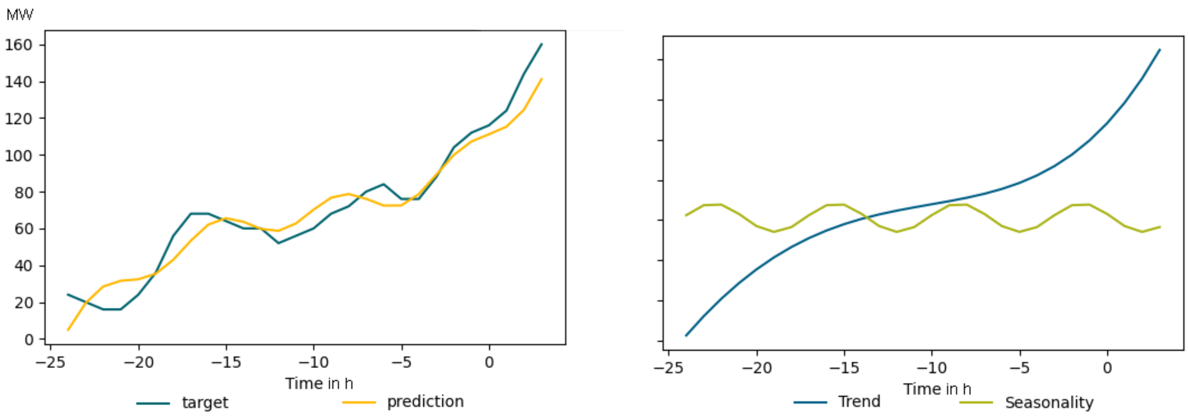

Figure 9: Constraining N-BEATS by adapting $g(\theta)$ to a monotonic and cyclical graph produces an interpretable output. The resulting components, i.e., trend and seasonality, are extracted and may be considered in further processes. A sample output for Austria and a forecast window of 24 time steps which is equivalent to 6 hours is shown.

or system operators to facilitate their decision making, as highlighted in Figure 9. Therefore, any developed model that is interpretable, or at least being interpretable, is beneficial.

Regarding meta-learning, the learning process can be decomposed into an inner and outer training loop [45]. The inner training loop focuses on task-specific knowledge while the outer loop focuses on across-task knowledge. This can be analogised to N-BEATS, where Θ is learnt inside the blocks and makes use of the parameters that are learnt from the outer loop, where gradient descent trains the weight matrices that Θ depends on. As the input passes through the blocks, Θ is slowly updated, and as the backcast is residually stacked with the input, it conditions the learning of Θ as the data feeds through the blocks.

Taken together, these findings confirm that a pure DNN model can deliver competitive forecast results, in contrast to the conclusion of the M4 competition. Moreover, during the implementation of the other models it was found that N-BEATS needs less time to be implemented. It does not require any decomposition and hardly any data pre-processing which is an essential and time-consuming part of the modeling process. Many ML or statistical approaches require additional preliminary steps, such as deseasonalisation or differencing, since they do not deal with non-stationary or non-linear relationships between input and output. In fact, working with raw historic data and using built-in mechanisms, such as residual links, backcast, and the aggregation of partial forecasts, leads to accurate and reliable forecasts.

6. Conclusions

This work has revealed a new, empirically validated methodology for STWPF. It shows that it is possible to build a pure deep-learning model for time series predictions that takes long-term trends and seasonality into consideration and surpasses the accuracy of existing models that combine ML and statistical approaches when applied to the same datasets.

Although it seems tempting to apply the approach to other areas, the findings might not be transferable since energy related problems often require domain knowledge, which ML has no ability to tackle. Nevertheless, this approach, which is particularly suitable for STWPF specifically, can be a powerful addition to the repertoire of every forecaster. Results so far have been very promising, and the approach could eventually be implemented in real-world forecasting applications in order to assist decision makers.

In further research, it is planned to examine how N-BEATS competes with other recently developed approaches, e.g., successful attention-based models such as BERT and transfer-learning or continual-learning models. Future work will concentrate on the systematic meta-learning understanding of how N-BEATS delivers its accurate results as a function of data and configuration. Beyond these developments, new NN approaches will be developed in other contexts and will help to improve STWPF overall. In addition, this method will be applied in other energy-related areas, such as renewables and load forecasting.

References

[1] I. Okumus, A. Dinler, Current status of wind energy forecasting and a hybrid method for hourly predictions, Energy Conversion and Management 123 (2016) 362 – 371.

[2] F. G. Montoya, F. Manzano-Agugliaro, S. López-Márquez, Q. Hernández-Escobedo, C. Gil, Wind turbine selection for wind farm layout using multi-objective evolutionary algorithms, Expert Systems with Applications 41 (2014) 6585 – 6595.

[3] F. Manzano-Agugliaro, A. Alcayde, F. Montoya, A. Zapata-Sierra, C. Gil, Scientific production of renewable energies worldwide: An overview, Renewable and Sustainable Energy Reviews 18 (2013) 134 – 143.

[4] Q. Hernandez-Escobedo, F. Manzano-Agugliaro, J. A. Gazquez-Parra, A. Zapata-Sierra, Is the wind a periodical phenomenon? the case of mexico, Renewable and Sustainable Energy Reviews 15 (2011) 721 – 728.

[5] P. Zhao, J. Wang, J. Xia, Y. Dai, Y. Sheng, J. Yue, Performance evaluation and accuracy enhancement of a day-ahead wind power forecasting system in china, Renewable Energy 43 (2012) 234 – 241.

[6] M. Khalid, A. V. Savkin, A method for short-term wind power prediction with multiple observation points, IEEE Transactions on Power Systems 27 (2012) 579–586.

[7] T. Nielsen, H. Madsen, Wppt - a tool for wind power prediction, in: Prediction.Procedings of Wind Power for the 21st Century, Kassel, Germany, 25-27 September 2000. Prediction.Procedings of Wind Power for the 21st Century, 25-27 September ; Conference date: 01-01-2000.

[8] I. Sanchez, Short-term prediction of wind energy production, International Journal of Forecasting 22 (2006) 43–56.

[9] E. Taslimi Renani, M. F. M. Elias, N. A. Rahim, Using data-driven approach for wind power prediction: A comparative study, Energy Conversion and Management 118 (2016) 193 – 203.

[10] R. D. Prasad, R. C. Bansal, M. Sauturaga, Some of the design and methodology considerations in wind resource assessment, IET Renewable Power Generation 3 (2009) 53–64.

[11] W. Zhang, Z. Qu, K. Zhang, W. Mao, Y. Ma, X. Fan, A combined model based on ceemdan and modified flower pollination algorithm for wind speed forecasting, Energy Conversion and Management 136 (2017) 439 – 451.

[12] C. Croonenbroeck, D. Ambach, A selection of time series models for short- to medium-term wind power forecasting, Journal of Wind Engineering and Industrial Aerodynamics 136 (2015) 201 – 210.

[13] M. Lydia, S. Suresh Kumar, A. Immanuel Selvakumar, G. Edwin Prem Kumar, Linear and non-linear autoregressive models for short-term wind speed forecasting, Energy Conversion and Management 112 (2016) 115 – 124.

[14] A. N. Celik, M. Kolhe, Generalized feed-forward based method for wind energy prediction, Applied Energy 101 (2013) 582 – 588. Sustainable Development of Energy, Water and Environment Systems.

[15] H. Chitsaz, N. Amjady, H. Zareipour, Wind power forecast using wavelet neural network trained by improved clonal selection algorithm, Energy Conversion and Management 89 (2015) 588 – 598.

[16] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–80.

[17] K. Yan, X. Wang, Y. Du, N. Jin, H. Huang, H. Zhou, Multi-step short-term power consumption forecasting with a hybrid deep learning strategy, Energies 11 (2018) 3089.

[18] S. Smyl, A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting, International Journal of Forecasting 36 (2020) 75 – 85. M4 Competition.

[19] H. Zheng, J. Yuan, L. Chen, Short-term load forecasting using emd-lstm neural networks with a xgboost algorithm for feature importance evaluation, Energies 10 (2017) 1168.
[20] G. Memarzadeh, F. Keynia, A new short-term wind speed forecasting method based on fine-tuned lstm neural network and optimal input sets, Energy Conversion and Management 213 (2020) 112824.
[21] H. Liu, X. Mi, Y. Li, Smart multi-step deep learning model for wind speed forecasting based on variational mode decomposition, singular spectrum analysis, lstm network and elm, Energy Conversion and Management 159 (2018) 54 – 64.
[22] J.-Z. Wang, Y. Wang, P. Jiang, The study and application of a novel hybrid forecasting model – a case study of wind speed forecasting in china, Applied Energy 143 (2015) 472 – 488.
[23] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, Wavenet: A generative model for raw audio, 2016.
[24] J. Chorowski, D. Bahdanau, K. Cho, Y. Bengio, End-to-end continuous speech recognition using attention-based recurrent nn: First results, 2014.
[25] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, Y. Bengio, Attention-based models for speech recognition, Advances in Neural Information Processing Systems 2015-January (2015) 577–585. 29th Annual Conference on Neural Information Processing Systems, NIPS 2015 ; Conference date: 07-12-2015 Through 12-12-2015.
[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need.
[27] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, X. Yan, Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting, 2020.
[28] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.
[29] J. Hu, J. Heng, J. Tang, M. Guo, Research and application of a hybrid model based on meta learning strategy for wind power deterministic and probabilistic forecasting, Energy Conversion and Management 173 (2018) 197 – 209.
[30] Z. Ma, S. Guo, G. Xu, S. Aziz, Meta learning-based hybrid ensemble approach for short-term wind speed forecasting, IEEE Access 8 (2020) 172859–172868.
[31] H. Zang, L. Cheng, T. Ding, K. W. Cheung, Z. Wei, G. Sun, Day-ahead photovoltaic power forecasting approach based on deep convolutional neural networks and meta learning, International Journal of Electrical Power and Energy Systems 118 (2020) 105790.
[32] K. Benidis, S. S. Rangapuram, V. Flunkert, B. Wang, D. Maddix, C. Turkmen, J. Gasthaus, M. Bohlke-Schneider, D. Salinas, L. Stella, L. Callot, T. Januschowski, Neural forecasting: Introduction and literature overview, 2020.
[33] A. Alexandrov, K. Benidis, M. Bohlke-Schneider, V. Flunkert, J. Gasthaus, T. Januschowski, D. C. Maddix, S. Rangapuram, D. Salinas, J. Schulz, L. Stella, A. C. Türkmen, Y. Wang, Gluonts: Probabilistic time series models in python, 2019.
[34] S. Makridakis, E. Spiliotis, V. Assimakopoulos, The m4 competition: Results, findings, conclusion and way forward, International Journal of Forecasting 34 (2018).
[35] B. N. Oreshkin, D. Carpov, N. Chapados, Y. Bengio, N-beats: Neural basis expansion analysis for interpretable time series forecasting, 2020.
[36] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10, Omnipress, Madison, WI, USA, 2010, p. 807–814.
[37] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2015.
[38] G. Huang, Z. Liu, L. van der Maaten, K. Q. Weinberger, Densely connected convolutional networks, 2018.
[39] A. A. Ismail, M. Gunady, H. C. Bravo, S. Feizi, Benchmarking deep learning interpretability in time series predictions, 2020.
[40] Open power system data platform, 2020. https://data.open-power-system-data.org/time_series/2020-10-06/ accessed: 2020-10-30.
[41] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, Tensorflow: Large-scale machine learning on heterogeneous distributed systems, 2016.
[42] J. Beitner, Pytorch forecasting - pytorch forecasting aims to ease timeseries forecasting with neural networks for real-world cases and research alike., 2020. Https://pytorch-forecasting.readthedocs.io/en/latest/ accessed: 2020-11-10.
[43] P. Pełka, G. Dudek, Pattern-based forecasting monthly electricity demand using multilayer perceptron, in: L. Rutkowski, R. Scherer, M. Korytkowski, W. Pedrycz, R. Tadeusiewicz, J. M. Zurada (Eds.), Artificial Intelligence and Soft Computing, Springer International Publishing, Cham, 2019, pp. 663–672.
[44] K. Bandara, C. Bergmeir, H. Hewamalage, Lstm-msnet: Leveraging forecasts on sets of related time series with multiple seasonal patterns, IEEE Transactions on Neural Networks and Learning Systems (2020) 1–14.
[45] A. Antoniou, H. Edwards, A. Storkey, How to train your maml, 2019.

Appendix

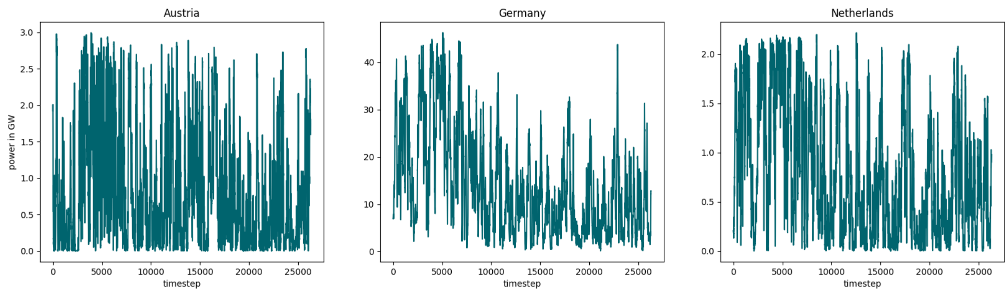

Figure 10: Aggregated wind power production in GW for AT, DE, NL in 15-minute time resolution between 01/01/2020 and 30/09/2020.

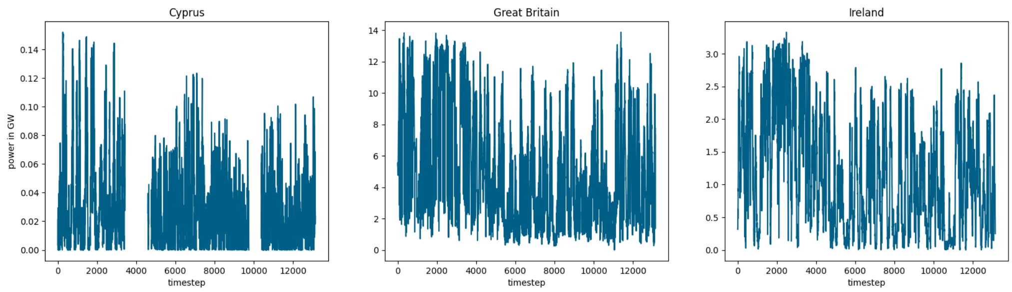

Figure 11: Aggregated wind power production in GW for CY, GB, IE in 30-minute time resolution between 01/01/2020 and 30/09/2020. Cyprus has some gaps in its history.

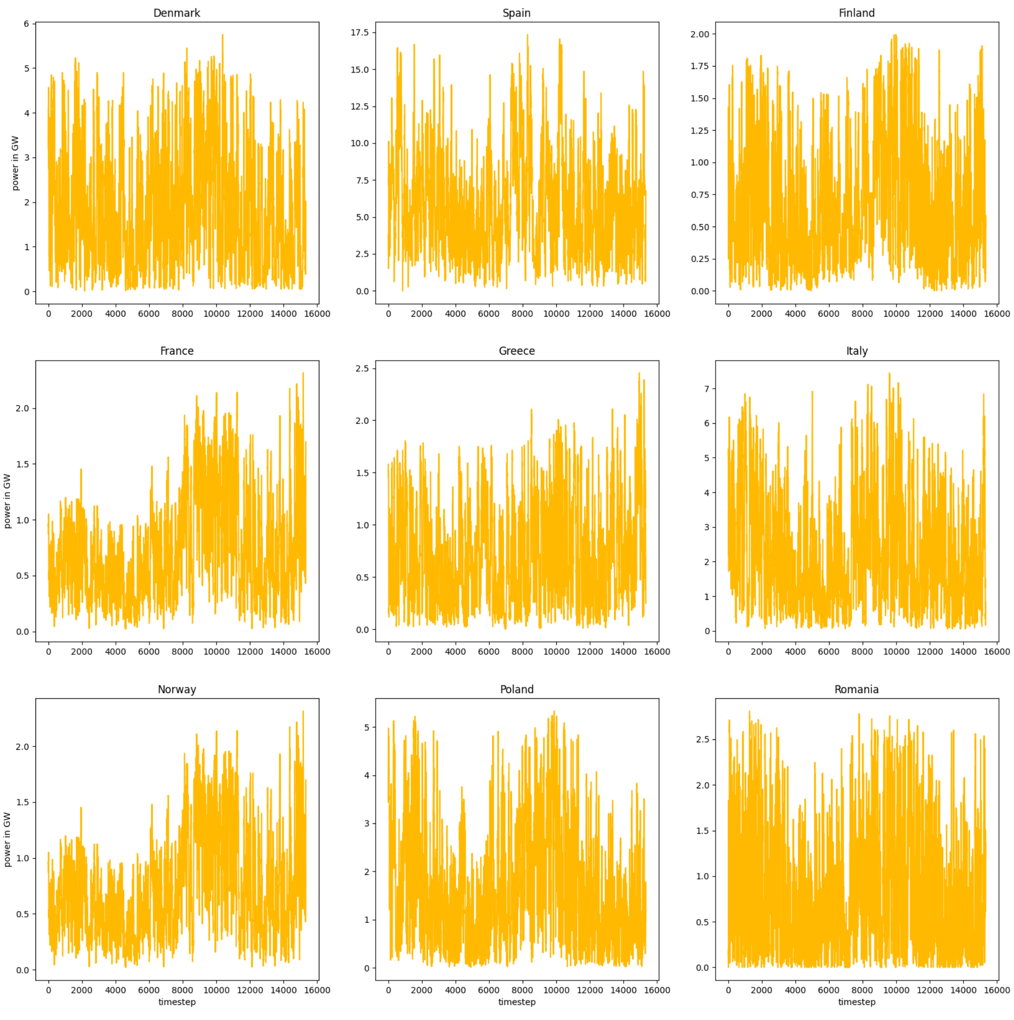

Figure 12: Aggregated wind power production in GW for DK, ES, FI, FR, GR, IT, NO, PL, RO in hourly time resolution between 01/01/2019 and 30/09/2020.