Assessment of critical parameters for artificial neural networks based short-term wind generation forecasting

Sewdien, V.N.; Preece, R.; Rueda Torres, J.L.; Rakhshani, E.; van der Meijden, M.

DOI
10.1016/j.renene.2020.07.117

Publication date
2020

Document Version
Final published version

Published in
Renewable Energy

Citation (APA)
Sewdien, V. N., Preece, R., Rueda Torres, J. L., Rakhshani, E., & van der Meijden, M. (2020). Assessment of critical parameters for artificial neural networks based short-term wind generation forecasting. *Renewable Energy*, *161*, 878-892. https://doi.org/10.1016/j.renene.2020.07.117

Renewable Energy 161 (2020) 878–892

Contents lists available at ScienceDirect

Renewable Energy

journal homepage: www.elsevier.com/locate/renene

Assessment of critical parameters for artificial neural networks based short-term wind generation forecasting

V.N. Sewdien [a, b, *], R. Preece [c], J.L. Rueda Torres [a], E. Rakhshani [a], M. van der Meijden [a, b]

[a] Department of Electrical Sustainable Energy, Delft University of Technology (TUD), Delft, the Netherlands
[b] TenneT TSO B.V, Arnhem, the Netherlands
[c] School of Electrical and Electronic Engineering, The University of Manchester, Manchester, UK

ARTICLE INFO

Article history:
Received 16 August 2019
Received in revised form
17 July 2020
Accepted 24 July 2020
Available online 3 August 2020

Keywords:
Artificial neural networks
Forecasting
Loss functions
MIGRATE
Optimizers
Wind

ABSTRACT

Participation of wind energy in the generation portfolio of power systems is increasing, making it more challenging for system operators to adequately maintain system security. It therefore becomes increasingly crucial to accurately predict the wind generation. This work investigates how different parameters influence the performance of forecasting algorithms. Firstly, this work analyzes the combined influence of the input data, batch size, number of neurons and hidden layers, and the training data on the forecast accuracy across forecast horizons of 5, 15, 30 and 60 min. It was found that increasing look ahead times require among others more hidden layers and lower batch sizes. Next, the optimizer and loss function leading to the most accurate forecasts were identified. It was concluded that the Adadelta optimizer and Mean Absolute Error loss function consistently result in the best performing forecasting algorithm. Finally, it was investigated if the most accurate optimizer-loss function combination is influenced by the choice of the performance metric. Whereas the Adadelta-Mean Absolute Error pair remains the most accurate combination irrespective of the evaluation metric, a strong relation was observed between the Root Mean Square Error performance metric and Mean Square Error loss function. Analyses were performed on 12 wind farms.

1. Introduction

With increasing penetration of wind generation, it becomes essential for system operators to accurately predict future wind power injections in the power system, in order to ensure reliable and affordable supply of electricity [1–4]. Imbalances in the power system could drastically increase as a result of forecast inaccuracies and could even lead to frequency stability problems [5]. Wind generation forecasting is performed across different time horizons, as summarized in Table 1.

Depending on the time horizon of interest, forecast models can generally be divided in two categories: statistical models and physical models. Statistical models are preferred for forecast horizons up to 6 h ahead, whereas physical models perform more accurately for longer forecast horizons [6].

1.1. Physical models

Physical models use atmospheric quantities (e.g. wind speed and direction, temperature and pressure), physical properties (e.g. terrain ruggedness index and wind farm layout) and numerical weather predictions (NWP) as inputs for complex meteorological models to forecast future parameters. Historical data are not required for training these forecast model. Physical models are very accurate for forecast horizons exceeding 6 h. However, one of the main challenges with this approach is that it requires specialized equipment for the acquisition and processing of the atmospheric and physical data [7].

1.2. Statistical models

Statistical models are purely mathematical models and mainly use past observed data, sometimes complemented with (NWP) information. For statistical models, machine learning methods are widely used [8], where artificial neural networks (ANN) are among the top used techniques for short-term forecasting [4,7]. The review performed in Ref. [9] even concluded that ANN based forecasting methods are the most efficient ones, provided that the network

* Corresponding author. Department of Electrical Sustainable Energy, Delft University of Technology (TUD), Delft, the Netherlands.
E-mail addresses: vinay.sewdien@tennet.eu, V.N.Sewdien@tudelft.nl (V.N. Sewdien).

https://doi.org/10.1016/j.renene.2020.07.117

Table 1
Forecasting time horizons in operational planning.

Forecasting Time Horizon	Range	Application
Long-term	Days to weeks ahead	• Maintenance schedules of transmission lines during low forecasted renewable energy generation • Maintenance schedules of e.g. wind turbines in order to minimize revenue losses for wind farm owners
Short-term: Day Ahead	24 h ahead	• Operational decision making with regards to the dispatch of renewable energy sources • Dynamic assessment of operating reserves requirements (e.g. for balancing)
Short-term: Near Real Time	Between 24 h and 5 min ahead	• Adjustments of real time dispatch • Dynamic assessment of operating reserves requirements (e.g. for ramp management) • More accurate security analysis

Nomenclature

Abbreviation Description

HDS	Historical data size (amount of historical data)
BS	Batch size
HL	Number of hidden layers
N_{HL}	Number of neurons in each hidden layer
TD	Training Data
FH	Forecast Horizon

configuration is optimized. Spatial correlation models and models based on probabilistic methods are two other types of statistical models.

This research focuses on ANN-based statistical models for short-term forecast horizons of 5, 15, 30, and 60 min. The 5 min forecast horizon (FH 5) is useful for ramp forecasting, which is crucial for power systems with high penetration of wind generation [10,11], an example of which is given in Ref. [12]. FH 15, FH 30 and FH 60 are useful for intraday markets where quarter-hourly and hourly products are traded.

The overall goal of this work is to investigate how the forecast accuracy across different forecast horizons is influenced by changes in the amount of historical data (i.e. historical data size, HDS), batch size (BS), number of hidden layers (HL), number of neurons per hidden layer (N_{HL}), the amount of training data (TD), and the type of optimizer and loss function used in the ANN's algorithm.

Whereas many previous publications have investigated the influence of the amount of historical data on the forecast accuracy, few have analyzed the impact of HDS combined with other aspects of the ANN's structure. In Ref. [13] the influence of the HDS for a single 1 h forecast of wind generation was investigated. The forecasting algorithm contained 1 hidden layer with 3 neurons, with TD 57%. It was found that the optimum HDS is dependent on the learning rate of the algorithm. In Ref. [14] the influence of HDS on the forecast accuracy in terms of root mean square error for FH 30 was investigated. The implemented forecasting algorithm contained 1 hidden layer, whereas HDS was varied from 3 to 8. It was concluded that the highest forecast accuracy is achieved for the ANN with HDS 8. In Ref. [15] the influence of HL and HDS on the wind generation forecast accuracy was investigated. It was found that a simple ANN with HDS 2 and no hidden layers performed the best in terms of forecast accuracy.

The aim of these papers was to identify the ANN with the highest forecast accuracy across one specific forecast horizon for wind generation. Furthermore, the solution space considered in these papers was rather limited, as maximum two ANN parameters were varied. Therefore there are still unresolved questions around the impact of proper tuning of the ANN's parameters on the accuracy and how this differs across different forecast horizons. Thus, the first aim of this work is to address these points by examining the combined influence of the amount of historical data, batch size, number of hidden layers, number of neurons per hidden layer, and the amount of training data on the forecast accuracy for forecast horizons 5, 15, 30, and 60 min ahead. Also, for each of these forecast horizons the impact of properly tuning the ANN's parameters is shown. This impact on the forecast accuracy will be considered by observing the normalized mean absolute error. It should be noted that the focus of this work is not on minimizing the forecast error, but on observing how it is affected by variations in ANN properties across the four different forecast horizons. With these insights it becomes possible to optimize only those parameters that have the biggest influence on the model's performance.

It is acknowledged that practical forecast models will often implement more complex ANNs than are implemented within this study, such as recurrent networks [16,17] or hybrid models [18,19]. However, this study still reveals many insightful aspects and recommendations which are applicable for more complex implementations. For example, if using recurrent networks, the set of parameters required for forecasting will not change from the ones considered here. Alternatively, if using hybrid methods then only the set of input parameters will change (to also include NWP data). By completing the analysis on a more simple yet very effective ANN (as evidenced by the achieved accuracies), the results can be more easily comprehended and generalized.

In the ANN. the optimizer and loss (or cost) function are important parts of the forecast algorithm. The goal of a loss function is to determine the difference between an observed and its forecasted value. The optimizer minimizes the selected loss function by updating a set of weights $\boldsymbol{\theta}$. However, according to the best knowledge of the authors, the influence of different optimizer-loss function pairs on the accuracy of wind power forecasting is not yet examined. A majority of publications do not specify the optimizer and loss function of the implemented forecast algorithm, whereas those publications that do mention them, do not provide any justification for the same. The mean squared error is used as loss function in Refs. [13,15,20–24]. The ANNs in Ref. [22–25] implemented the Levenberg-Marquardt algorithm as optimizer, whereas ADALINE was used in Ref. [15]. Neither the optimizer nor the loss function are given for the forecast model in Ref. [14]. The influence of different permutations of optimizers and loss functions on the forecast accuracy remains unknown. The second aim of this work is to explore how different combinations of optimizers and loss functions influence the error in wind generation forecasting.

Several performance metrics exist in literature for the evaluation of forecast models. The most used ones are the root mean squared error (RMSE, used in Ref. [21,24,26–29]), the mean absolute error (MAE, used in Refs. [21,27,29–31]) and the mean absolute percentage error (MAPE, used in Refs. [21,22,27–29,31]). Usually, multiple metrics are used to evaluate the same forecast model. The third aim of this research is to investigate whether there exists a dependency between the performance metrics and the most accurate optimizer-loss function combination.

The main contribution of this research are as follows:

(1) **Insights are given in how different ANN parameters influence the forecast accuracy.** Increasing look ahead times require more frequent updates of the ANN's weights, reducing the most efficient batch size to 5. For the considered forecast horizons, it is observed that HDS 5 and HDS 10 lead to the most accurate results. The influence of the considered amounts of training data on the forecast accuracy was found to be negligible;
(2) **Based on extensive empirical analysis, the optimizer and loss function leading to the most accurate wind generation forecast were identified.** Adadelta and MAE were found to be the most accurate optimizer and loss function, respectively. This holds true independent of the forecast horizon. This study furthermore gives empirical evidence for (a) the consistent superiority of the MAE as a loss function and (b) the superiority of adaptive optimizers that do not require manual selection of the learning rate (i.e. Adadelta, Adam, Adamax and Nadam) over other optimizers.
(3) **The dependency between performance metrics and most accurate optimizer-loss function was investigated.** Independent of the chosen performance metric, the Adadelta-MAE combination results in the most accurate forecast performance. Furthermore, a strong relation was observed between the nRMSE evaluation metric and the MSE loss function, essentially showing that it may be worth considering using the MSE loss function if (and only if) the goal is to minimize the nRMSE of the forecasts.

The remaining part of this study is organized as follows: Section 2 presents the research method that was used throughout this work. The various parameters of the ANN that are considered in this study are discussed in Section 3, whereas the analysis of the results are given in Section 4. Finally, Section 5 presents the conclusions.

2. Research methodology

The aim of this research is to assess how different parameters of an ANN based forecast model influence its forecast accuracy. To facilitate this goal an ANN based forecast model is developed in Python [32]. This Section presents the research methodology used in this work. First a brief introduction of the ANN concept and its main parameters is given, followed by the implemented simulation approach. The Section concludes with information on the wind power data used in this study.

2.1. Artificial neural networks

An ANN acts as a black box that maps inputs to outputs. In the case of wind power forecasting, it aims to map inputs such as observed wind power values or NWP data to future wind power values. It learns this input-output mapping by training and optimization. A brief summary of the basic form and function of an ANN is provided here (full details can be found in Ref. [33]). Fig. 1 illustrates the general structure of an ANN.

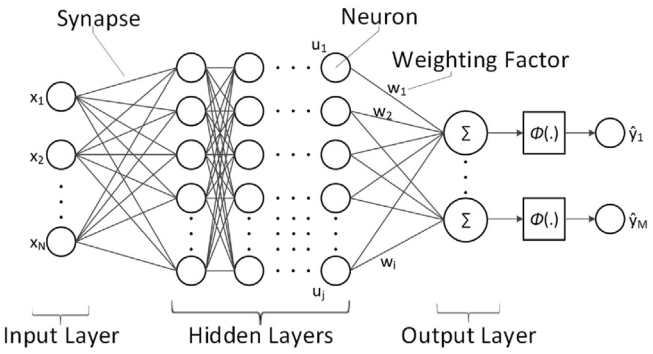

Fig. 1. General architecture of an artificial neural network.

It consists of an input layer, one or more hidden layers, an output layer, and several synapses with their associated weighting factors. Each layer contains a number of neurons. A synapse is the link between two neurons of different layers. With respect to the application of wind power forecasting, the input layer can consist of either previously observed values of the wind power generation or numerical weather prediction data (such as wind speed, pressure, and temperature). Each input variable is assigned to a single neuron in the input layer. The number of neurons in each hidden layer can be chosen arbitrarily. Some sort of optimization is required here, as the number of neurons in the hidden layers influences the forecast performance. An activation function is used to define the output of neurons for the next layer, according to (1). The dimension of the output layer is determined by the number of outputs being forecasted.

$$\widehat{y}_q = \varphi\left(\sum_{i,j=1}^{i,j} w_i u_j\right) \tag{1}$$

The activation function implemented in the ANN for this work is the rectifier function [34] and can mathematically be described as (2). It is widely used due to its low forecast error and high sparsity [35,36].

$$\varphi(\tau) = \max(0, \tau) \tag{2}$$

Based on the objective function of the ANN's optimizer, the weighting factors are updated using the feed forward back propagation (FFBP) technique [37]. The algorithm for the FFBP technique can be decomposed in four steps. In the first step the input data is fed into the ANN, after which a forecasted value is produced, according to (3).[1]

$$\widehat{y}_q = \varphi\left(\sum_{j=1}^{J} w_{j,q}.\varphi\left(\sum_{k}^{K} w_{k,j}.\ldots\ldots\left(\varphi\left(\sum_{n=1}^{N} w_{n,k}.x_n\right)_{HL\ 1}\right)_{HL\ p-1}\right)_{HL\ p}\right)_{output\ layer} \tag{3}$$

[1] The derivation of (3) is given in Appendix B.

$\widehat{y}_q$: forecasted value of the qth neuron in the output layer.

J: number of neurons in hidden layer p
$w_{j,q}$: weighting factor of synapse that connect the jth neuron of hidden layer p to the qth neuron of the output layer
K: number of neurons in hidden layer p-1
N: number of neurons in input layer
HL 1: first hidden layer
P: number of hidden layers

In the second step, the error between the forecasted output, which is a function of weighting factors w_i, and its actual observed value is determined using a loss function, see (4). The error is then back propagated to the output layer. The loss functions are further discussed in Section 3.3.

$$J(\boldsymbol{\theta}) = y - \widehat{y} \tag{4}$$

$\boldsymbol{\theta}$: vector containing all weighting factors w_i

y :observed output
$\widehat{y}$: forecasted output

In the third step, the back propagation continues to the hidden layers. In the final step, the weights are updated, with the aim of minimizing the error. This algorithm stops when a predefined number of epochs (i.e. optimization iterations) has been reached. Usually, the objective function has a form like given in (5). It has the target to minimize the error between the forecasted and observed value. It is calculated as an average of loss functions ε_i for individual training samples i.

$$\varepsilon = \frac{1}{N}\sum_{i=1}^{N}\varepsilon_i \tag{5}$$

At the start of each training process, the weights w_i need to be initialized. A well-known initialization method is the *Xavier* initialization [38]. However, as was shown in Ref. [39], the *Xavier* scheme is not appropriate for the rectifier activation function, because the scheme requires a linear activation function. Therefore the authors of [39] proposed the *He* initialization scheme, which was used throughout this work. The influence of the weight initialization methods on the forecast accuracy is out of the scope of this work.

2.2. Simulation approach

Because of computational limitations, a sequential two step approach is chosen for the parametric investigation (Fig. 2). A total of 12 wind farms (sites) with different geographical characteristics are investigated.

In Step 1, detailed in Fig. 3, the influence of the following parameters on the forecast accuracy is investigated, leading to a total of 108 different permutations per site and FH:

- Number of inputs, i.e. the historical data size (HDS): 5, 10, 20[2]
- Number of hidden layers (HL): 1, 2, 3[3]
- Number of neurons per hidden layer (N_{HL}): 100% (i.e. equal to the number of neurons in the input layer) and 50% (i.e. equal to the average of the neurons in the input and output layer).
- Size of the training data set (TD): 50% and 80% of the test data.

[2] Higher number of inputs did not influence the forecast error positively.
[3] More hidden layers did not influence the forecast error positively.

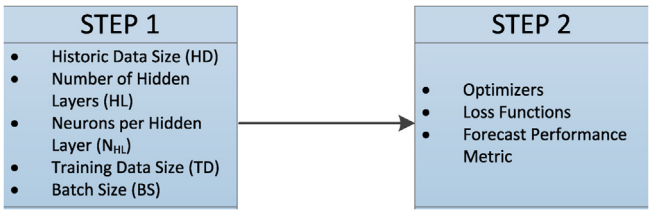

Fig. 2. Two-step simulation approach.

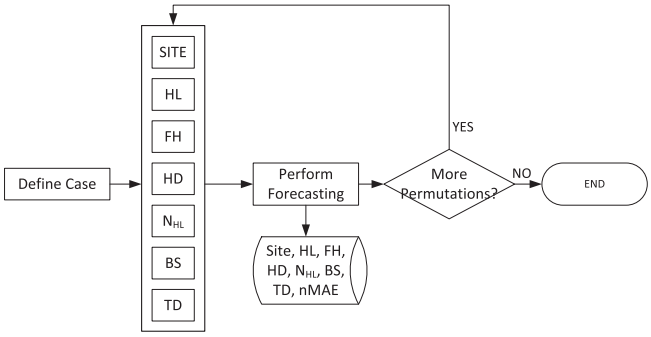

Fig. 3. Implemented simulation approach for Step 1.

- Batch size (BS), i.e. amount of observations after which the weighting factors are updated: 5, 10, 20.

This first step contributes to the first goal of this research and will provide a set of parameters that lead to the most accurate forecast.

In Step 2 (Fig. 4) the influence of the optimizer and loss function on the forecast performance is evaluated. The number of hidden layers, neurons per hidden layer, and training data size are inputs from Step 1.

The optimizers that are investigated in this research are the Stochastic Gradient Descent, RMSprop, Adagrad, Adadelta, Adam, Adamax, and Nadam. The following loss functions are evaluated: mean squared error, mean absolute error, mean absolute percentage error, mean squared logarithmic error, squared hinge, hinge, logcosh, binary crossentropy, kullback leibler divergence, poisson, and cosine proximity. The assessment is carried out for FH 5, FH 15, FH 30 and FH 60.

The influence of the choice of forecast performance metric, i.e. normalized Mean Absolute Error (nMAE), normalized Mean Absolute Percentage Error (nMAPE), or normalized Root Mean Square Error (nRMSE), on the most accurate optimizer-loss function pair is also assessed.

This second step contributes to the second and third goal of this research and will identify the optimizer-loss function pair that results in the most accurate forecast, i.e. lowest nMAE, nMAPE, and nRMSE.

2.3. Data

Depending on the location of a wind farm, the same forecast algorithm can result in different forecast accuracies [40]. To ensure robustness of the results, the analyses in Step 1 and Step 2 are carried out for 12 wind farms, each with different geographical characteristics. By doing so, the results can be considered general enough and can be applied on wind farms with a wide range of geographical characteristics.

The wind generation data used in this work is retrieved from the

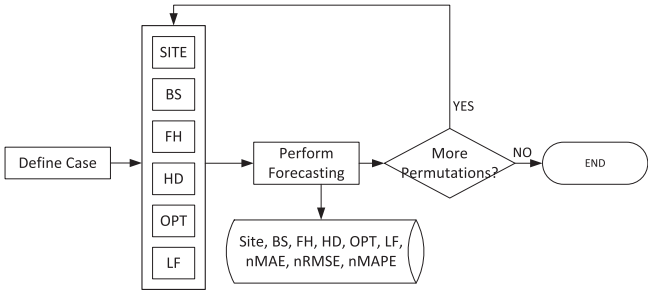

Fig. 4. Implemented simulation approach for Step 2.

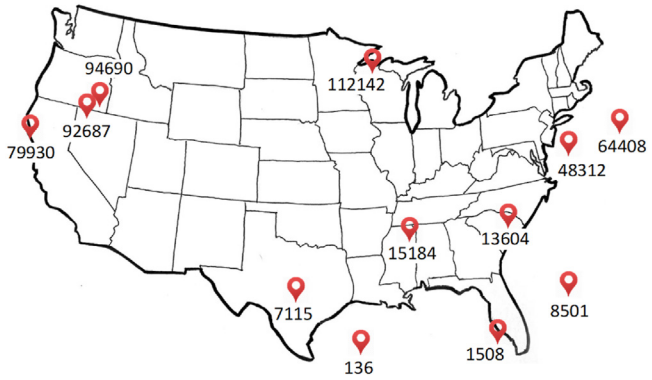

Fig. 5. Location and site IDs of considered wind farms.

WIND Prospector Toolkit of USA's National Renewable Energy Laboratory [41–44], and belongs to 12 different small wind parks of 16 MW each (Fig. 5 and Table 2).

Observed active power generation from the wind turbines is available with a 5 min resolution for the time span 2007–2012. The statistical parametric *t*-test was performed successfully (i.e. rejection of the null hypothesis) on the data sets to determine if all the data belonged to the same population.

It should be mentioned that wind power forecasting is based on the wind farm, geographical and weather information and is independent of the electricity network. Therefore the forecasting results are not influenced by the structure and characteristics of the electricity grid.

3. ANN parameters

This Section gives a short description of the main parameters that are used throughout this work. Section 3.1 discusses the metrics used for evaluating forecast models. Section 3.2 and Section 3.3 briefly introduce the optimizers and loss functions, respectively.

Table 2
Wind farm locations.

Site ID	Longitude	Latitude
136	−93.660828	25.789566
1508	−82.809998	26.368622
7115	−99.497406	30.336601
8501	−77.39856	29.295036
13,604	−88.724579	33.849228
15,184	−80.262238	33.363693
48,312	−73.391205	37.496029
64,408	−70.430237	38.473736
79,930	−123.977585	39.193207
92,687	−118.889999	41.542522
94,690	−118.084106	41.870815
112,142	−90.955688	46.140095

3.1. Forecast evaluation metrics

In order to evaluate the performance of different forecast models, their accuracies are compared with each other. However, this only makes sense when the input data is exactly the same across all models. Evaluating a model's performance using forecast accuracies of the model at different geographic locations does not lead to meaningful conclusions, as the accuracies are influenced by the geographical characteristics of the wind farms under consideration. The following metrics are widely used for quantifying forecast accuracies:

Normalized Mean Absolute Error (nMAE):

$$nMAE = \frac{1}{P_{rated}} \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{6}$$

Normalized Root Mean Square Error (nRMSE):

$$nRMSE = \frac{1}{P_{rated}} \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \tag{7}$$

Normalized Mean Absolute Percentage Error (nMAPE, also known as the bias error)

$$nMAPE = \frac{1}{P_{rated}} \frac{100}{N} \sum_{i=1}^{N} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{8}$$

In (2)–(4) P_{rated} is the nameplate capacity of the wind farm, N is the number of samples in the data set, y_i is the observed value at timestep i, and $\hat{y}_i$ is the forecasted value at timestep i.

The use of nRMSE for the evaluation of model performance is discouraged, due to the fact that nRMSE is more sensitive to outliers (i.e. larger errors are penalized more heavily). In wind speed datasets that inherently have outliers, nRMSE gives a distorted view of the forecast accuracy [45]. The nMAE metric is less susceptible to this as inferred from (6) and is therefore used as much as possible as the performance metric throughout this work. On the other hand, when the goal is to minimize the risks resulting from forecast inaccuracies instead of minimizing the forecast accuracy itself, the nRMSE metric could be more useful. An example of a risk resulting from forecast inaccuracies is the imbalance costs that occur in a power system, due to inaccurate forecasting of the infeed of wind energy [3]. The evaluation of different performance metrics to capture the risks associated with forecast inaccuracies is in itself a research topic that requires a comprehensive study, which is left as future work. For the optimizer-loss function evaluation all the above performance metrics are used, in order to illustrate the impact of the performance metric choice on the most accurate optimizer-loss function pair.

3.2. Optimizers

The goal of many ANN based models is to converge to a set of parameters $\boldsymbol{\theta}$ that comply with an objective function $J(\boldsymbol{\theta})$. Optimizers in ANNs are required for updating the set of weights $\boldsymbol{\theta}$ used for mapping the input to the output. Assume a training data set for a forecasting algorithm containing N samples, each with P dimensions. At the i-th iteration, the following holds true for the parameter set $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i + \Delta\boldsymbol{\theta}_i \tag{9}$$

The convergence of the weights $\boldsymbol{\theta}$, with the aim of optimizing the objective function J($\boldsymbol{\theta}$), is achieved through the gradient descent method. In this method the updated parameter set $\boldsymbol{\theta}_{i+1}$ is achieved by applying small changes $\Delta\boldsymbol{\theta}_i$ which are proportional to the negative of the gradient of the function at the current point ($\boldsymbol{\theta}_i$):

$$\Delta\boldsymbol{\theta}_i = -\alpha g_i \tag{10}$$

α :learning rate.

g_i: gradient of the parameters at the i-th iteration

Several optimizers exist, which are discussed below.

3.3. SGD

The Stochastic Gradient Descent (SGD) method [46] calculates the gradient of a loss function J($\boldsymbol{\theta}$) with regards to the weights $\boldsymbol{\theta}$ after each sample pair (x^i,y^i). A learning rate $\boldsymbol{\alpha}$ needs to be determined manually and remains unchanged throughout the full set of simulations. For the SGD method, $\boldsymbol{\alpha}$ is identical for all input neurons and is therefore defined as a global learning rate. The set of weights $\boldsymbol{\theta}$ are updated N times per epoch:

$$\boldsymbol{\theta}_{i+1} = \theta_i - \boldsymbol{\alpha}\nabla_{\theta} J\left(\theta; x^i, y^i\right) \tag{11}$$

On the contrary, when using the batch gradient descent method, the weights are updated only once per epoch.

3.4. Adagrad

The adaptive gradient descent (Adagrad) algorithm [47] contains a global α. This α, however, is not constant and is updated after each iteration i. To implement this adaptive characteristic, Adagrad introduces an exponentially decaying correction factor for each dimension P and is based on all previous gradients of dimension P:

$$\Delta\theta_i = -\frac{\boldsymbol{\alpha}}{\sqrt{\sum_{n=1}^{i} g_n^2}} g_i \tag{12}$$

$\boldsymbol{\alpha}$:global learning rate.

In this way, each dimension P has its own dynamic $\Delta\theta_i$ which is inversely proportional to the past gradient magnitudes. Because of this characteristic, the algorithm is suitable for data sets with high levels of sparsity [48]. One of the drawbacks of this method is the manual selection of an initial global $\boldsymbol{\alpha}$.

3.5. Adadelta

The Adadelta method [49] dynamically adapts the learning rate over time using, among others, an exponentially decaying average of the previous squared gradients. Furthermore, it eliminates the need for the manual selection of a global $\boldsymbol{\alpha}$[4] as shown in (9):

$$\Delta\theta_i = -\frac{RMS[\Delta\theta]_{i-1}}{RMS[g]_i} g_i \tag{13}$$

One of the advantages of the Adadelta algorithm is that it is robust against different initial values of α, which is not the case with the SGD or Adagrad algorithm. The RMSprop and Adadelta methods are somewhat similar. The main difference is that RMSprop still requires the manual selection of the learning rate.

3.6. Adam

The adaptive moment estimation (Adam) method [50], like Adagrad, determines a unique learning rate α_i for each weight θ_i, which is updated for every sample pair x^i,y^i. It uses two correction factors for the update rule: an exponentially decaying average of the previous squared gradients (like Adagrad and Adadelta) and an exponentially decaying average of the previous updates $\Delta\theta_i$. In Refs. [50] it was shown that Adam has a better performance than other adaptive optimizer algorithms. It is among the most popular optimizers used in ANNs [51]. When Adam is generalized, the Adamax algorithm is achieved.

3.7. Nadam

The Nesterov-accelerated adaptive moment estimation (Nadam) method [51] is developed based on the Adam algorithm. The main difference is that whereas Adam uses a classical momentum for determining the exponentially decaying factor, Nadam uses the Nesterov accelerated gradient.

3.8. Loss functions

A loss function is a mathematical formula that calculates the difference between an observed output and its forecasted value. A very simple loss function in given in (10):

$$J(\boldsymbol{\theta}) = y - \widehat{y}_{\boldsymbol{\theta}} \tag{14}$$

y : observed output.

$\widehat{y}_{\boldsymbol{\theta}}$: forecasted output for weights θ

Several loss functions exist and different loss functions will give different errors for the same set of input data. As will be proven in this work, the choice of the loss function has a significant effect on the performance of the forecast model.

The *mean squared error* (MSE) is widely used in linear regression and uses the ordinary least squares method for minimizing the error. The *mean absolute error* (MAE) is used to measure the distance between an observed value and its forecast. Whereas it is easier to calculate the derivative for the MSE, large errors have a relatively bigger influence on the MSE. In these cases, the MAE is more robust to outliers, since the error is not squared. To overcome this issue, the *mean squared logarithmic error* (MSLE) loss function can be used. One of the advantages of this function is that it does not penalize large differences, given that the observed and forecasted values are also large numbers. However, when the large errors are not related to outliers, MAE is more efficient in minimizing the loss function, as it penalizes the large errors more severely, and therefore forces a faster convergence of the weighting factors $\boldsymbol{\theta}$. The *mean absolute percentage error* (MAPE) is a variant of MAE. One of the drawbacks of the MAPE is that it cannot be used when the observed value is 0 (i.e. division by 0).

The mathematical formulas for the loss functions considered in this research are given below. The mathematical derivation of these loss functions is out of the scope of this work.

$$\text{Mean Squared Error } loss(y, \widehat{y}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y}_i)^2 \tag{15}$$

[4] Full derivation of (9) is available in Ref. [44].

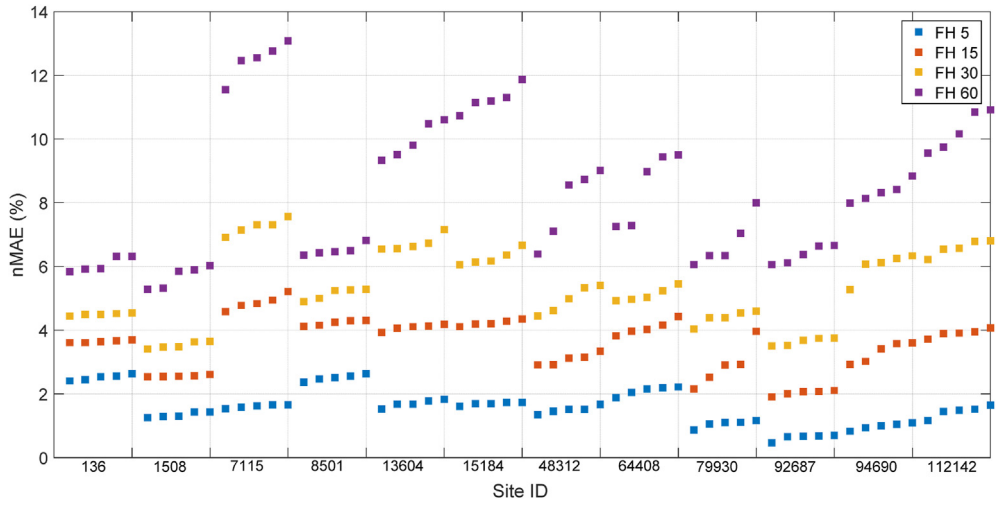

Fig. 6. nMAE for 12 wind farms across four forecast horizons.

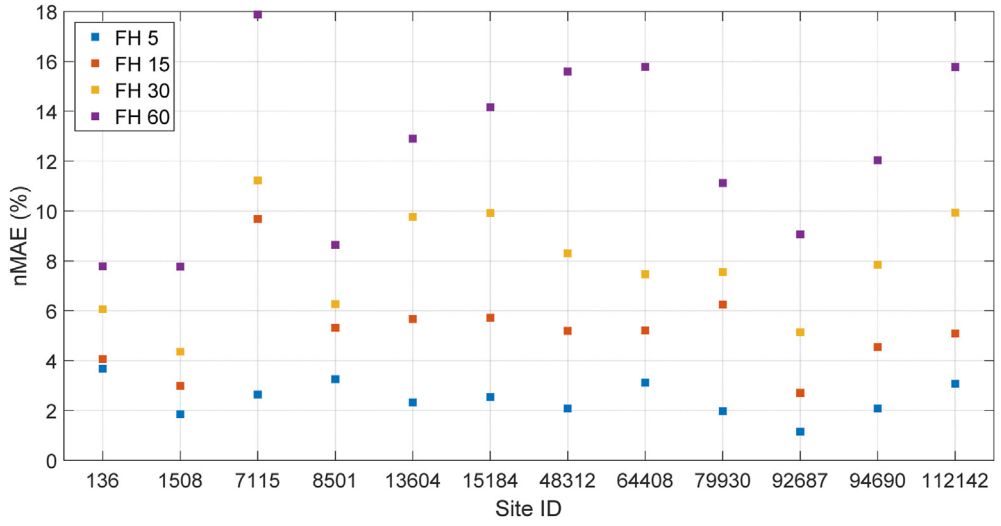

Fig. 7. First Quartile of nMAE for 12 wind farms across four forecast horizons.

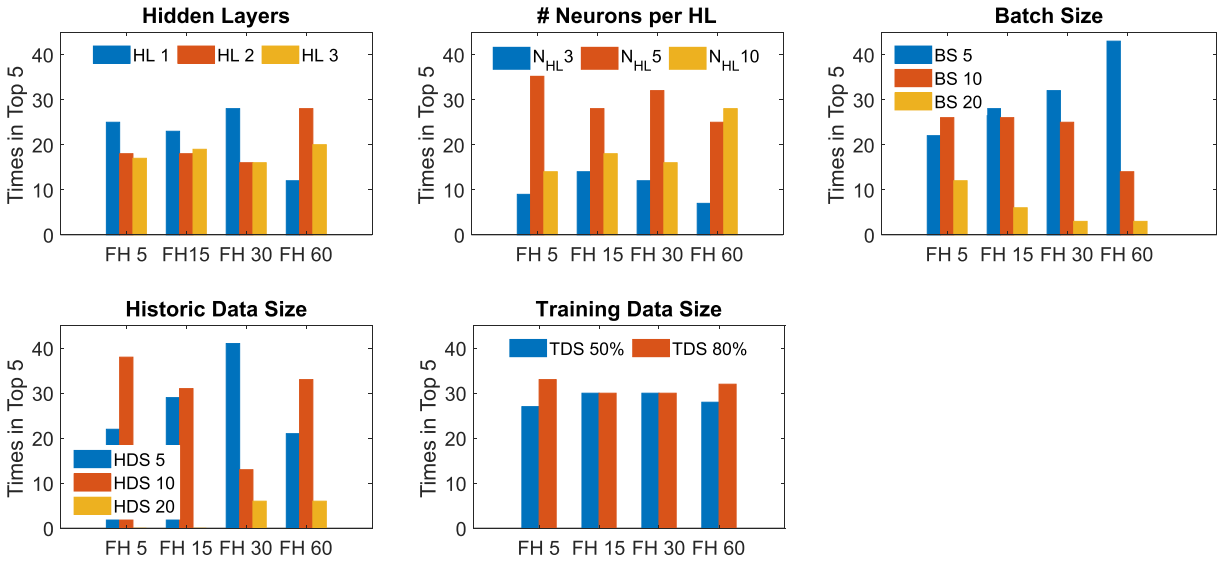

Fig. 8. Parametric evaluation of ANN based forecast model.

$$\text{Mean Absolute Error } loss(y,\hat{y}) = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \quad (16)$$

$$\text{Mean Absolute Percentage Error } loss(y,\hat{y}) = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \quad (17)$$

$$\text{Mean Squared Logarithmic Error } loss(y,\hat{y}) = \frac{1}{N}\sum_{i=1}^{N}(\log(y_i+1) - \log(\hat{y}_i+1))^2 \quad (18)$$

$$\text{Hinge } loss(y,\hat{y}) = \sum_{i=1}^{N}\max\left(0, \frac{1}{2} - y_i\hat{y}_i\right) \quad (19)$$

$$\text{Squared Hinge } loss(y,\hat{y}) = \sum_{i=1}^{N}\max\left(0, \frac{1}{2} - y_i\hat{y}_i\right)^2 \quad (20)$$

$$\text{Log} - \text{Cosh } loss(y,\hat{y}) = \sum_{i=1}^{N}\log(\cosh(\hat{y}_i - y)) \quad (21)$$

$$\text{Binary Cross} - \text{Entropy } loss(y,\hat{y}) = \frac{1}{N}\sum_{i=1}^{N}[y_i.\log(\hat{y}_i) + (1-y_i)log(1-\hat{y}_i)] \quad (22)$$

$$\text{Kullback Leibler Divergence } loss(y,\hat{y}) = \frac{1}{N}\sum_{i=1}^{N}(y_i.\log(y_i)) - \frac{1}{N}\sum_{i=1}^{N}(y_i.\log(\hat{y}_i)) \quad (23)$$

$$\text{Poisson } loss(y,\hat{y}) = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i.\log(\hat{y}_i)) \quad (24)$$

$$\text{Cosine Proximity } loss(y,\hat{y}) = \frac{\sum_{i=1}^{N} y_i.\hat{y}_i}{\sqrt{\sum_{i=1}^{N}(y_i)^2}.\sqrt{\sum_{i=1}^{N}(\hat{y}_i)^2}} \quad (25)$$

All the optimizers and loss functions mentioned in Section 3 have been used for the evaluation of the ANN based forecast model.

4. Results & discussion

This section presents the results of the two previously mentioned simulation steps. Three questions were investigated. First, it was investigated how parameters of an ANN based forecast model influence the model's performance across different forecast horizons. Second, the impact of the ANN's optimizer and loss function selection on the forecast accuracy was investigated. And lastly, the influence of the model's evaluation metric on the ranking of the most efficient optimizer-loss function combination was looked into.

4.1. Influence of ANN model parameters on forecast accuracy

The first aim of this research was to examine how the amount of historical data, batch size, number of hidden layers, number of neurons per hidden layer, and the amount of training data influence the forecast accuracy for forecast horizons of 5, 15, 30, and 60 min. For each of these forecast horizons, several permutations of the

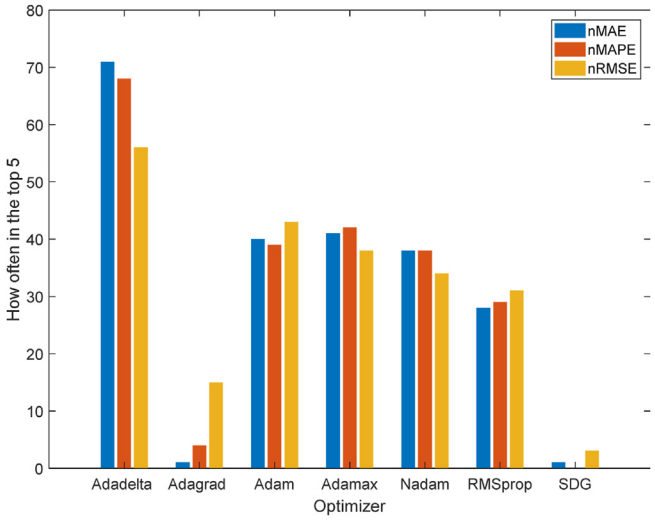

Fig. 9. Ranking of optimizers.

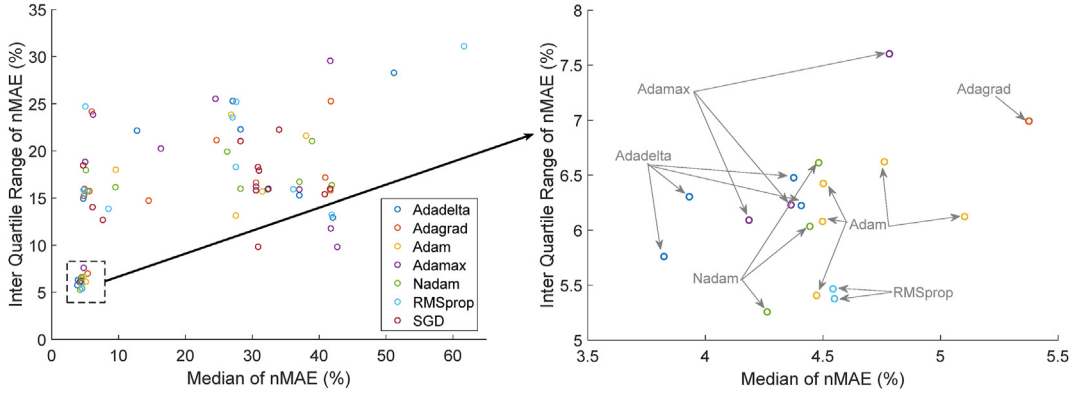

Fig. 10. Median versus Inter Quantile Range for optimizers.

above mentioned parameters were produced for analyzing their impact on the forecast accuracy, measured as nMAE. The analyses were carried out for 12 different wind farms. For each of these 12 sites, Fig. 6 shows the five best nMAEs per forecast horizon. Fig. 7 depicts for each site the first quartile of the nMAE per forecast horizon.

Two interesting observations can be made from Figs. 6 and 7. First, for each of the sites, increasing forecast horizons consistently lead to increasing forecast errors. As the predictability of the state of any highly complex system (such as wind power generation) decreases with increasing look ahead time, the forecast error increases. Similar results were also achieved in Refs. [52–55]. Second, for the same forecast horizon the forecast errors are different across the different sites. When comparing different sites, it is observed that sometimes forecasts with longer look ahead times perform better (both, the top 5 as well as the first quartile) than forecasts with shorter look ahead times. The 60 min forecast of site 136 has a better performance than the 30 min forecast of site 7115, whereas the 30 min forecast of site 1508 and the 15 min forecast of site 92,687 outperform respectively the 15 min forecast of site 8501 and the 5 min forecast of site 8501. This is not an attribute of the forecast model, but is due to the geographical characteristics and ruggedness of the site. The ruggedness is expressed using the ruggedness index (RIX) and higher RIX values lead to increased forecast errors [56]. Similar conclusions regarding the sensitivity of the forecast error to site characteristics (i.e. RIX values) were also achieved in Refs. [54,57,58].

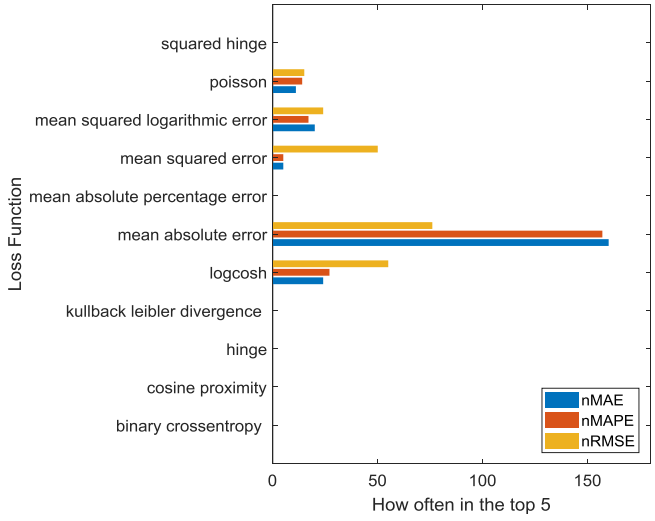

Fig. 11. Ranking of loss functions.

Fig. 8 shows for the four FH how often each of the parameters ended up in the top 5 nMAE. The higher the presence of a parameter in the top 5, the higher its influence on the forecast error.

With the aim of maximizing the system's predictability with increasing forecast horizons, the complexity of forecasting algorithms also increases. Two factors that contribute to the algorithm's complexity are the architecture of the ANN (i.e. the number of hidden layers, the number of neurons per hidden layer and, the historical data size) and the training algorithm (i.e. the batch size and the test data size).

In terms of the size of the training data set, it can be concluded that it has a very small sensitivity for the forecast horizon and that either 50% or 80% of the test data can be used for learning purposes. However, the lower the size of the training data set, the earlier a forecast model can be fully operational. In this research the test data consisted one year of data, which results in a 6 months period before the model could be fully operational.

Up to FH 30 one hidden layer provided sufficient complexity to minimize the forecast error (41.7% of the top 5 cases for FH 5, 38.3% for FH 15 and 46.7% for FH 30 contained one hidden layer). Two hidden layers claimed a share of 46.7% in the top 5 nMAE for FH 60. The complexity is even further increased by assigning 10 neurons per hidden layer for FH 60 (in 46.7% of the top 5 cases), as opposed to 5 neurons for FH 5 (in 61.7% of the top 5 cases). The share of batch size 5 in the top 5 cases consistently increases with increasing forecast horizons (from 36.7% for FH 5–71.7% for FH 60). This means that compared to FH 5 the algorithm needs to update its weights more often for FH 60. This increased complexity, resulting from the increasing hidden layers, neurons per hidden layer, and reduced batch size, confirms that the longer the look ahead time of the forecast model, the more complex the ANN structure and training algorithm will be [59].

Regarding the historical data size, it is observed that HDS 5 and HDS 10 lead to the most accurate results. Furthermore, a steady increase in the share of HDS 5 is observed until FH 30. For FH 60 the share of HDS 5 in the top 5 decreases to 35%, whereas HDS 10 increases to 55%. The reason for this could be the need for increased observed data required for capturing the dynamics associated with FH 60, which is confirmed when examining the variance: data sets associated with FH 60 have a larger variance than datasets associated with FH 5.

The lowest forecast errors and associated ANN parameters for

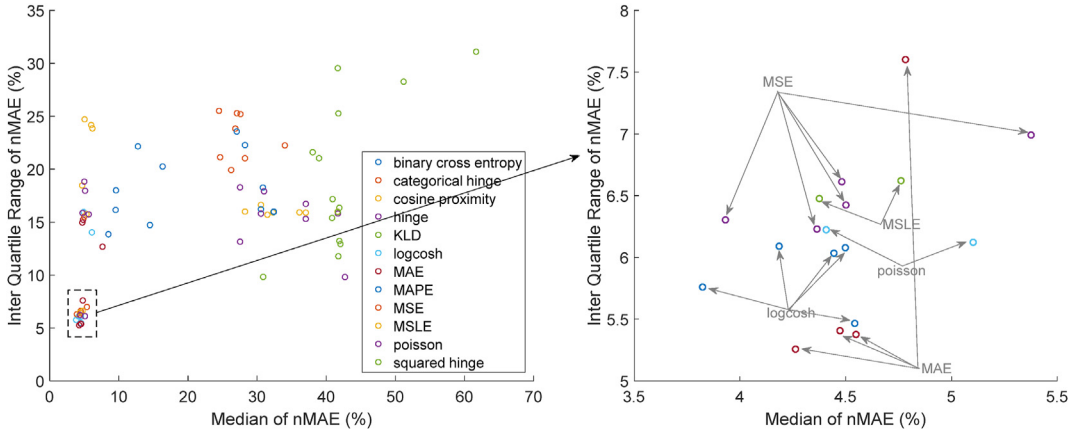

Fig. 12. Median versus Inter Quantile Range for loss functions.

Table 3
Ranking of optimizer-loss function pairs based on nMAE.

	Adadelta (%)	Adagrad (%)	Adam (%)	Adamax (%)	Nadam (%)	RMSprop (%)	SGD (%)
binary cross entropy	0	0	0	0	0	0	0
cosine proximity	0	0	0	0	0	0	0
hinge	0	0	0	0	0	0	0
kullback leibler divergence	0	0	0	0	0	0	0
logcosh	4.5	0	1.4	1.8	2.3	0.9	0
MAE	22.7	0.5	14.1	14.5	12.7	8.2	0
MAPE	0	0	0	0	0	0	0
MSE	0.5	0	0.5	0.5	0.5	0.5	0
MSLE	1.8	0	1.8	0.5	1.8	2.7	0.5
poisson	2.7	0	0.5	1.4	0	0.5	0
squared hinge	0	0	0	0	0	0	0

Table 4
Ranking of optimizers when using nMAE, nMAPE, and nRMSE.

Optimizer	nMAE (%)	nMAPE (%)	nRMSE (%)
Adadelta	32.3	30.9	25.5
Adagrad	0.5	1.8	6.8
Adam	18.2	17.7	19.5
Adamax	18.6	19.1	17.3
Nadam	17.3	17.3	15.5
RMSprop	12.7	13.2	14.1
SGD	0.5	0	1.4

Table 5
Ranking of loss functions when using nMAE, nMAPE, and nRMSE.

Loss Function	nMAE (%)	nMAPE (%)	nRMSE (%)
binary cross entropy	0	0	0
cosine proximity	0	0	0
Hinge	0	0	0
kullback leibler divergence	0	0	0
logcosh	10.9	12.3	25.0
MAE	72.7	71.4	34.5
MAPE	0	0	0
MSE	2.3	2.3	22.7
MSLE	9.1	7.7	10.9
poisson	5.0	6.4	6.8
squared hinge	0	0	0

each site and forecast horizon are given in APPENDIX A Parameters of Best Performing ANNs.

4.2. Optimizer-loss function evaluation

The second goal of this research was to identify the optimizer, loss function and optimizer-loss function combination that resulted in the most accurate forecasting algorithm. A brute force search across all possible combinations of optimizer and loss function was conducted using varying forecast horizons, resulting in thousands of cases. In order to assess which optimizer and loss function is most appropriate, the errors associated with each resulting forecast (across the 12 selected sites and different forecast horizons) were calculated. This provided a ranked list of combinations for each site and FH. Following this, the number of times an optimizer or loss function appeared in the top 5 of the ranked list was calculated. This was performed three times, using different error evaluation methods (nMAE, nMAPE, and nRMSE).

From the results as shown in Fig. 9, it is observed that Adadelta is the most accurate optimizer (32.3% of the top 5 cases), followed by Adamax (18.6%) and Adam (18.2%). Adadelta has the biggest share in the top 5 nMAEs for each forecast horizon: 26.7%, 20%, 31.7%, and 40% for respectively FH 5, FH 15, FH 30 and FH 60.

The best performance of Adadelta can be explained by the fact that it does not require manual selection of the global learning rate α (see (13)), unlike e.g. RMSprop and Adagrad. The influence of the learning rate on the forecast accuracy and required simulation iterations have been thoroughly investigated in different fields [30,60–67]. These studies successfully show the effect of incorrect selection of the manual learning rate on the forecast performance. The current work complements these conclusions by providing a comparison among different optimizers, based on empirical results. A majority of the publications on ANN based wind forecasting do not mention the implemented optimizer. The ANNs in Ref. [22–25] implemented the Levenberg-Marquardt algorithm as optimizer, whereas ADALINE was used in Ref. [15]. However, none of the publications elaborate on the choice of optimizer. This study provides empirical evidence on the superiority of the Adadelta optimizer for short-term wind forecasting.

Fig. 10 plots for each optimizer the median versus the inter quartile range (IQR) of the nMAE. The left graph gives an overview of the performance of all optimizers, whereas the right graph focuses on the optimizers with the lowest median and the lowest IQR. It indeed shows Adadelta as the optimizer with the lowest median. Depending on the trade-off between variability (i.e. IQR) and accuracy (i.e. median), users can chose different optimizers as the

Table 6
Ranking of optimizer-loss function pairs based on nMAPE.

	Adadelta (%)	Adagrad (%)	Adam (%)	Adamax (%)	Nadam (%)	RMSprop (%)	SGD (%)
binary cross entropy	0	0	0	0	0	0	0
cosine proximity	0	0	0	0	0	0	0
hinge	0	0	0	0	0	0	0
kullback leibler divergence	0	0	0	0	0	0	0
logcosh	4.5	0	0.9	2.7	2.3	1.8	0
MAE	21.4	1.8	14.1	13.2	13.2	7.7	0
MAPE	0	0	0	0	0	0	0
MSE	0.5	0	0.5	0.5	0.5	0.5	0
MSLE	1.4	0	1.8	0.5	1.4	2.7	0
poisson	3.2	0	0.5	2.3	0	0.5	0
squared hinge	0	0	0	0	0	0	0

Table 7
Ranking of optimizer-loss function pairs based on nRMSE.

	Adadelta (%)	Adagrad (%)	Adam (%)	Adamax (%)	Nadam (%)	RMSprop (%)	SGD (%)
binary cross entropy	0	0	0	0	0	0	0
cosine proximity	0	0	0	0	0	0	0
hinge	0	0	0	0	0	0	0
kullback leibler divergence	0	0	0	0	0	0	0
logcosh	5.9	2.7	4.1	2.7	5.5	4.1	0
MAE	10	1.4	7.7	7.7	4.5	3.2	0
MAPE	0	0	0	0	0	0	0
MSE	5.5	2.3	4.5	4.1	3.2	3.2	0
MSLE	1.8	0.5	1.4	1.8	1.4	2.7	1.4
poisson	2.3	0	1.8	0.9	0.9	0.9	0
squared hinge	0	0	0	0	0	0	0

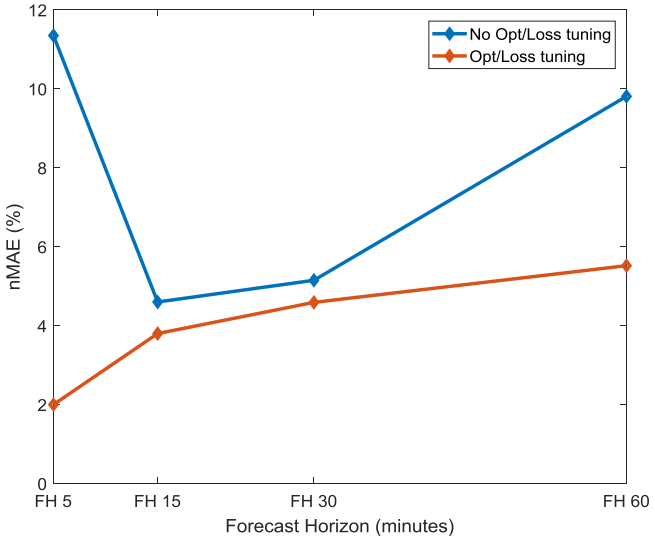

Fig. 13. nMAE comparison with and without optimizer-loss function tuning.

preferred algorithm.

Based on the ranking of the loss functions, as depicted in Fig. 11, the mean absolute error results in the most accurate forecasting algorithm (72.7% of the top 5 cases). This holds true across all the investigated forecast horizons: its share in the top 5 nMAEs for each forecast horizon is 30%, 51.7%, 100%, and 85% for respectively FH 5, FH 15, FH 30 and FH 60. It is worth noting that the mean absolute error remains the best loss function, independent of the choice of performance metric, i.e. nMAE, nMAPE and nRMSE. This is explained by the fact that the relation between the loss function and any performance metric is unidirectional: the loss function influences the performance of the forecasting algorithm, but not the other way around. The goal of the loss function is to calculate the difference between a forecasted and observed value in the feedforward backpropagation algorithm, which is then used as input for the optimizer. Depending on the performance of the loss function, the weights of the neurons in the ANN are adjusted to facilitate the convergence of the optimizer.

This is an interesting result, considering the fact that most of the wind generation forecast models in literature have implemented the MSE as loss function, e.g. Ref. [13,15,20–24]. This can be explained by the fact that historically, the (R)MSE has been popular, largely because of its theoretical relevance in statistical modelling [68]. However, as these are more sensitive to outliers than the MAE, using (R)MSE results in a slower convergence of the forecasting algorithm, leading to less accurate results for the same number of epochs. This study gives empirical evidence for the consistent superiority of the MAE as a loss function.

As was done for the optimizers, Fig. 12 illustrates the median versus the IQR for each loss function. The figure on the left gives a global overview of all the loss functions, whereas the figure on the right zooms in on the region of interest (i.e. low median and low IQR). One ANN configuration using the logcosh loss function has the lowest median, whereas the mean absolute error performs better on the IQR. As was the case with the optimizers, users might choose different loss functions as the preferred choice, depending on the trade-off between the variability and accuracy.

Table 3 gives for each considered optimizer-loss function pair its share in the top 5 nMAEs. From this Table it is observed that the Adadelta-MAE pair results in the most accurate forecast model, with a share of 22.7% in the top 5 nMAEs. The second best

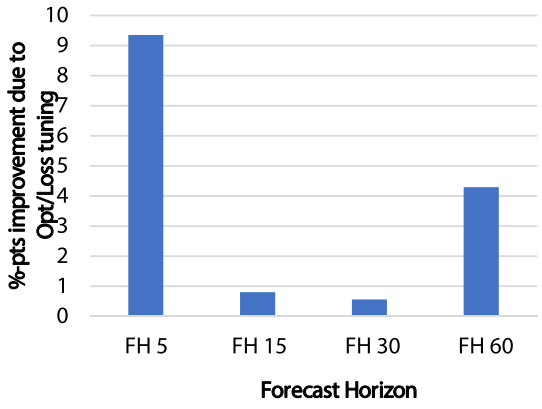

Fig. 14. Additional improvement in nMAE as a result of optimizer-loss function tuning.

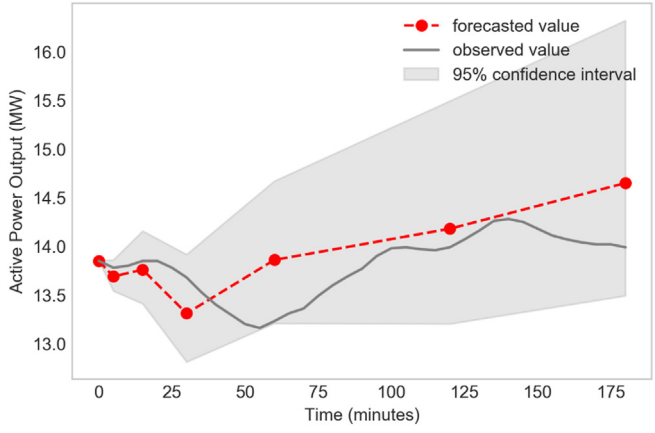

Fig. 15. Observed versus forecasted values of most accurate implemented algorithm.

combination is the Adamax-MAE pair (14.5%), followed by the Adam-MAE pair (14.1%).

4.3. Influence of forecast performance metrics

The third goal of this research was to identify whether there exists a relation between the forecast performance metrics (i.e. nMAE, nMAPE and nRMSE) and the most accurate optimizer, loss function, and optimizer-loss function combination. The most used performance metrics are the RMSE (used in Refs. [21,24,26–29]), the MAE (used in Refs. [21,27,29–31]) and the MAPE (used in Refs. [21,22,27–29,31]).

Table 4 shows for the investigated optimizers their share in the top 5 nMAE, nMAPE and nRMSE. The order of the best optimizer is the same across all performance metrics, with the only difference for the nRMSE: Adam comes second and Adamax comes third (it's the other way around for nMAE and nMAPE). Therefore it is concluded that the forecast performance metric has a negligible influence on the optimizer's ranking. Looking at the share of the optimizers in the top 5, it is clear that the adaptive optimizers that do not require manual selection of the learning rate (i.e. Adadelta, Adam, Adamax and Nadam) clearly outperforms the other optimizers.

For the loss functions, a strong relation is observed between the nRMSE metric and the MSE, as is shown in Fig. 11 and Table 5. Irrespective of this, the results show that it is still more appropriate to use the mean absolute error as the loss function and the results suggest that this is the best all-purpose loss function – particularly if one is concerned with nMAE or nMAPE forecast errors. Table 5 Ranking of loss functions when using nMAE, nMAPE, and nRMSE.

The shares of the different optimizer-loss function combinations in the top 5 when using the nMAE forecast performance metric, is given in Table 3. Table 6 and Table 7 give the shares in the top 5 when using nMAPE, respectively nRMSE.

From these tables it is observed that the Adadelta-MAE pair results in the most accurate forecast model, independent of the metric used for evaluating the model's performance. Adam-MAE and Adamax-MAE are the next best pairs. The results suggest that Adadelta is the most appropriate optimizer regardless of the error that is being minimized. However, the shares differ significantly when nRMSE is used. In this case, optimizer-loss function pairs with MSE as the loss function have a major increase in their top 5 shares.

4.4. Value of increased forecast accuracy

The plots in Fig. 13 and Fig. 14 illustrate the achieved improvements in forecast accuracies when optimizers and loss functions are properly selected (i.e. the best possible selections are made) compared to when they are not. For the 5 min and 60 min horizons, improvements of 9%-pts, respectively 4%-pts can be gained, highlighting the importance of correct ANN design.

Placing this in perspective: for a 100 MW wind farm in the UK, a 1.2%-pts improvement in the nMAE could result in an increased estimated yearly revenue of 177,000 EUR [69]. Similar analysis were carried out for Ireland [70], Spain [71] and the IEEE 118-bus test system [3], where the decrease in system operational costs and increase in the revenue of wind farm owners as the result of improvements in the forecast accuracy were presented. The benefits of improved forecast accuracies are not only limited to wind generation. In Ref. [2] the decrease in costs for ramping, curtailment and system operation due to improved solar power forecasting are presented.

Due to smoothing effects, the forecast accuracy may even further decrease with increasing geographical area. When compared to the forecast of a single wind farm (as is the case in this research), forecast errors on control area level are up to 63% lower [72].

Based on the results of this work, a forecasting algorithm was developed with parameters tuned to lead to the most accurate results. An overview of the observed and forecasted values (including 95% confidence interval) for one instance are given in Fig. 15.

Summarizing, the following key findings result from the presented work:

- Increasing look ahead times require more frequent updates of the ANN's weights, reducing the most efficient batch size to 5;
- For the historical data size, it is observed that HDS 5 and HDS 10 lead to the most accurate results;
- It was found that the influence of the considered amount of training data, i.e. 6 months (50%) or 9.6 months (80%), is rather limited;
- Adadelta was found to be the most accurate optimizer, as it does not require manual selection of a global learning rate. The superiority of adaptive optimizers that do not require manual selection of the learning rate over other optimizers was proven;
- The MAE loss function leads by far to the most accurate forecasts, in contrast to the MSE, which is commonly used in literature;
- A strong relation was observed between the nRMSE evaluation metric and the MSE loss function, essentially showing that it may be worth considering using the MSE loss function if (and only if) the goal is to minimize the nRMSE of the forecasts.

The analysis in this work focused on two categories of parameters. The results obtained for the ANN-structure related parameters (e.g. number of hidden layers, neurons) are also applicable for hybrid models, as they reveal which parameters have a large influence (e.g. batch size) on the forecast accuracy and which do not (e.g. training data). For this category of parameters, any considered forecasting method would need retuning of the suggested parameters.

The conclusions regarding the ANN-algorithm related parameters (i.e. optimizer and loss function) would remain valid for hybrid and complex ANN models. Hybrid and complex ANN models changes the ANN's structure (e.g. smaller historic data size, larger batch size, etc.) and it was found that the most superior optimizer and loss function are not dependent on the ANN-structure related parameters (more so for the loss function than for the optimizer). Therefore, these results are valuable and transferable to other forecasting methods utilizing ANNs.

5. Conclusions

The share of cheap, volatile wind energy in the generation portfolio of power systems is rapidly increasing across the world. Its intermittent nature, however, poses new challenges for system operators, with balancing being one of them. Therefore accurate forecasting of wind generation becomes crucial for secure and efficient system operation.

This research focused on forecasting wind generation for near real time and operational planning purposes across forecast horizons of 5, 15, 30, and 60 min. The main contributions of this research are as follows. First, it was investigated how different parameters of an ANN based forecasting algorithm influence the forecast accuracy. It was found that increasing look ahead times require more complex ANNs. For up to 30 min ahead, the highest accuracy is achieved if the ANN has 1 hidden layer and 5 neurons per hidden layer. For 60 min ahead, 2 hidden layers and 10 neurons

per hidden layer are required. The increasing complexity, related with the increasing look ahead times, also requires more frequent updates of the weights, reducing the most efficient batch size to 5. For the historical data size, it is observed that HDS 5 and HDS 10 lead to the most accurate results. Furthermore, it was shown that the influence of the amount of training data, i.e. 6 months (50%) or 9.6 months (80%), is rather limited. The models were capable of achieving the same accuracies with 6 months of data, which could result in earlier deployment of such forecast models.

Second, the optimizer, loss function, and optimizer-loss function pair that lead to the most accurate forecasts were identified. Adadelta was found to be the most accurate optimizer, as it does not require manual selection of a global learning rate. This study furthermore gave empirical evidence for the superiority of adaptive optimizers that do not require manual selection of the learning rate (i.e. Adadelta, Adam, Adamax and Nadam) over other optimizers. Whereas the RMSE was found to be the preferred loss function in literature, the results obtained in this study do reveal that the MAE by far leads to the most accurate forecasts. The Adadelta-MAE pair was also identified as the most accurate optimizer-loss function combination.

Finally, the relation between the identified optimizer-loss function pair in the previous step and the choice of evaluation metric (nMAE, nMAPE, nRMSE) was investigated. Whereas the Adadelta-MAE pair remains the most accurate combination independent of the evaluation metric, a strong relation was observed between the nRMSE evaluation metric and the MSE loss function. This is essentially showing that it may be worth considering using the mean squared error loss function if (and only if) the goal is to minimize the nRMSE of the forecasts.

The studies conducted in this work were based on data retrieved from 12 different wind farms, each with their own geographical characteristics, and should therefore increase the applicability of the obtained results.

The analyses in this research focused on forecasts for short-term operational planning up to 60 min ahead. Future research could be directed towards the same exercise for hybrid forecast models, containing ANNs and NWP models. This would enable longer forecast look ahead times.

CRediT authorship contribution statement

V.N. Sewdien: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing - original draft, Writing - review & editing. **R. Preece:** Conceptualization, Formal analysis, Methodology, Supervision, Validation, Visualization, Writing - original draft, Writing - review & editing. **J.L. Rueda Torres:** Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Visualization, Writing - review & editing. **E. Rakhshani:** Methodology, Resources, Software, Validation, Visualization, Writing - review & editing. **M. van der Meijden:** Funding acquisition, Methodology, Resources, Supervision, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was carried out as part of the MIGRATE project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 691800. This paper reflects only the authors' views and the European Commission is not responsible for any use that may be made of the information it contains.

APPENDIX A. Parameters of Best Performing ANNs

The lowest forecast errors and associated ANN parameters for each investigated site and forecast horizon are given in the tables below.

Appendix B. Mathematical Expression of Feedforward Algorithm

This Appendix derives the mathematical formula for the feedforward forecast algorithm. Consider Fig. A1 The ANN has 1 input

Table 8
Best ANN parameters for FH 5

Site	FH	HL	N_{HL}	BS	HDS	TD	nMAE (%)
136	5	2	5	10	10	0.8	2.405
1508	5	1	5	5	10	0.8	1.252
7115	5	3	5	5	10	0.5	1.534
8501	5	1	10	5	10	0.5	2.363
13,604	5	1	10	5	10	0.5	1.526
15,184	5	1	5	10	5	0.5	1.613
48,312	5	2	5	10	5	0.5	1.340
64,408	5	3	5	5	5	0.5	1.874
79,930	5	1	10	5	10	0.5	0.871
92,687	5	2	5	5	5	0.5	0.468
94,690	5	1	3	5	5	0.5	0.828
112,142	5	3	3	5	5	0.5	1.160

Table 9
Best ANN parameters for FH 15

Site	FH	HL	N_{HL}	BS	HDS	TD	nMAE (%)
136	15	2	3	5	5	0.8	3.606
1508	15	3	3	20	5	0.5	2.537
7115	15	1	10	10	10	0.8	4.582
8501	15	1	10	20	10	0.5	4.116
13,604	15	3	10	5	10	0.8	3.921
15,184	15	1	3	10	5	0.5	4.108
48,312	15	1	5	5	5	0.8	2.910
64,408	15	2	5	10	10	0.8	3.819
79,930	15	3	5	5	10	0.8	2.151
92,687	15	2	3	5	5	0.5	1.906
94,690	15	3	10	5	10	0.8	2.926
112,142	15	2	5	5	5	0.5	3.718

Table 10
Best ANN parameters for FH 30

Site	FH	HL	N_{HL}	BS	HDS	TD	nMAE (%)
136	30	1	10	10	10	0.5	4.442
1508	30	3	5	10	5	0.5	3.409
7115	30	2	3	5	5	0.5	6.911
8501	30	3	5	5	5	0.5	4.893
13,604	30	1	5	10	5	0.8	6.547
15,184	30	2	5	5	5	0.8	6.049
48,312	30	3	5	5	5	0.5	4.447
64,408	30	1	10	10	10	0.8	4.930
79,930	30	3	20	5	20	0.2	4.035
92,687	30	3	5	5	5	0.8	3.503
94,690	30	1	5	5	5	0.5	5.272
112,142	30	1	5	10	5	0.5	6.215

Table 11
Best ANN parameters for FH 60

Site	FH	HL	N_{HL}	BS	HDS	TD	nMAE (%)
136	60	2	5	5	10	0.8	5.840
1508	60	2	10	5	10	0.8	5.288
7115	60	3	10	5	10	0.5	11.542
8501	60	3	5	5	5	0.5	6.347
13,604	60	3	5	5	5	0.5	9.331
15,184	60	1	3	5	5	0.5	10.737
48,312	60	2	10	5	10	0.8	6.389
64,408	60	3	10	5	10	0.8	7.257
79,930	60	2	5	5	5	0.5	6.053
92,687	60	2	10	5	10	0.8	6.053
94,690	60	3	10	5	10	0.5	7.985
112,142	60	3	5	5	10	0.8	9.554

layer with three input neurons, two hidden layers with four neurons each and one output layer with two neurons. Weight $w_{i,j}$ represents the weighting factor of the synapse connecting neuron *i* to neuron *j*. For simplicity reasons, the activation function $\Phi(.)$ is shown only for the output layer Fig. A1. However, all neurons, except those in the input layer, have an activation function in the same structure as is shown for the output layer.

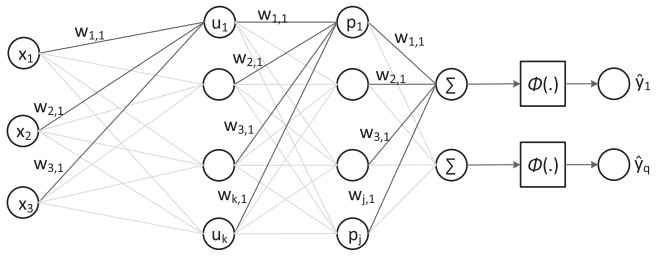

Fig. A1. Artificial neural network..

The value of the neurons in the first hidden layer (i.e. u_1, u_2, u_3, u_4) are calculated as given in (A1).

$$\begin{cases} u_1 = \varphi\left(x_1.w_{1,1} + x_2.w_{2,1} + x_3.w_{3,1}\right) \\ u_2 = \varphi\left(x_1.w_{1,2} + x_2.w_{2,2} + x_3.w_{3,2}\right) \\ u_3 = \varphi\left(x_1.w_{1,3} + x_2.w_{2,3} + x_3.w_{3,3}\right) \\ u_4 = \varphi\left(x_1.w_{1,4} + x_2.w_{2,4} + x_3.w_{3,4}\right) \end{cases} \quad (A1)$$

With N neurons in the input layer and K neurons in the first hidden layer, (A1) can be generalized to (A2).

$$u_k = \varphi\left(\sum_{n=1}^{N} w_{n,k}.x_n\right) \quad (A2)$$

Similarly, the value of the neurons in the second hidden layer are calculated using (A3).

$$\begin{cases} p_1 = \varphi\left(u_1.w_{1,1} + u_2.w_{2,1} + u_3.w_{3,1} + u_4.w_{4,1}\right) \\ p_2 = \varphi\left(u_1.w_{1,2} + u_2.w_{2,2} + u_3.w_{3,2} + u_4.w_{4,2}\right) \\ p_3 = \varphi\left(u_1.w_{1,3} + u_2.w_{2,3} + u_3.w_{3,3} + u_4.w_{4,3}\right) \\ p_4 = \varphi\left(u_1.w_{1,4} + u_2.w_{2,4} + u_3.w_{3,4} + u_4.w_{4,4}\right) \end{cases} \quad (A3)$$

With K neurons in the first hidden layer and J neurons in the second hidden layer, (A3) can also be generalized:

$$p_j = \varphi\left(\sum_{k=1}^{K} w_{k,j}.u_k\right) \quad (A4)$$

The value of the neurons in the second hidden layer can be expressed in terms of the neurons of the input layer by substituting (A2) in (A4):

$$p_j = \varphi\left(\sum_{k=1}^{K} w_{k,j}.\varphi\left(\sum_{n=1}^{N} w_{n,k}.x_n\right)\right) \quad (A5)$$

The values of the neurons in the output layer are calculated using (A6) and can be generalized to (A7).

$$\begin{cases} y_1 = \varphi\left(p_1.w_{1,1} + p_2.w_{2,1} + p_3.w_{3,1} + p_4.w_{4,1}\right) \\ y_2 = \varphi\left(p_1.w_{1,2} + p_2.w_{2,2} + p_3.w_{3,2} + p_4.w_{4,2}\right) \end{cases} \quad (A6)$$

$$y_q = \varphi\left(\sum_{j=1}^{J} w_{j,q}.p_j\right) \quad (A7)$$

Finally, when substituting (A5) in (A7), the generalized mathematical relation of the input neurons and the output neurons in obtained:

$$\hat{y}_q = \varphi\left(\sum_{j=1}^{J} w_{j,q}.\varphi\left(\sum_{k=1}^{K} w_{k,j}.\varphi\left(\sum_{n=1}^{N} w_{n,k}.x_n\right)\right)\right) \quad (A8)$$

Equation (A8) describes the concept of the feedforward forecasting algorithm.

References

[1] K. Bruninx, K. Van Den Bergh, E. Delarue, W. D'haseleer, Optimization and allocation of spinning reserves in a low carbon framework, IEEE Trans. Power Syst. Syst. 31 (2) (2016) 872–882.
[2] C. Brancucci Martinez-Anido, et al., The value of day-ahead solar power forecasting improvement, Sol. Energy 129 (2016) 192–203.
[3] Q. Wang, H. Wu, A.R. Florita, C. Brancucci Martinez-Anido, B.M. Hodge, The value of improved wind power forecasting: grid flexibility quantification, ramp capability analysis, and impacts of electricity market operation timescales, Appl. Energy 184 (2016) 696–713.
[4] Q. Chen, K.A. Folly, Wind power forecasting 51 (28) (2018) 414–419. IFAC-PapersOnLine.
[5] K. Das, M. Litong-Palima, P. Maule, P.E. Sorensen, Adequacy of operating reserves for power systems in future european wind power scenarios, IEEE Power Energy Soc. Gen. Meet. 2015 (2015).
[6] G. Giebel, R. Brownsword, G. Kariniotakis, The state of the art in short-term prediction of wind power, ANEMOS.plus (2011) 1–110.
[7] A. Ahmed, M. Khalid, A review on the selected applications of forecasting models in renewable power systems, Renew. Sustain. Energy Rev. 100 (2019) 9–21. September 2018.
[8] H.S. Dhiman, D. Deb, J.M. Guerrero, Hybrid machine intelligent SVR variants for wind forecasting and ramp events, Renew. Sustain. Energy Rev. 108 (April) (2019) 369–379.
[9] M. Sharifzadeh, A. Sikinioti-Lock, N. Shah, Machine-learning methods for integrated renewable power generation: a comparative study of artificial neural networks, support vector regression, and Gaussian Process Regression, Renew. Sustain. Energy Rev. 108 (2019) 513–538. July 2018.
[10] J. Zhang, M. Cui, B.M. Hodge, A. Florita, J. Freedman, Ramp forecasting performance from improved short-term wind power forecasting over multiple spatial and temporal scales, Energy 122 (2017) 528–541.
[11] V.N. Sewdien, et al., Effects of increasing power electronics based technology on power system stability: performance and operations, CIGRE Sci. Eng. J. 11 (2018) 5–17.
[12] E. Ela, B. Kirby, ERCOT Event on February 26 , 2008: Lessons Learned, 2008.
[13] G. Li, J. Shi, On comparing three artificial neural networks for wind speed forecasting, Appl. Energy 87 (7) (2010) 2313–2320.
[14] M. Monfared, H. Rastegar, H.M. Kojabadi, A new strategy for wind speed forecasting using artificial intelligent methods, Renew. Energy 34 (3) (2009) 845–848.
[15] E. Cadenas, W. Rivera, Short term wind speed forecasting in La Venta, Oaxaca, México, using artificial neural networks, Renew. Energy 34 (1) (2009) 274–278.
[16] T.G. Barbounis, J.B. Theocharis, M.C. Alexiadis, P.S. Dokopoulos, Long-term wind speed and power forecasting using local recurrent neural network models, IEEE Trans. Energy Convers. 21 (1) (2006) 273–284.
[17] Q. Cao, B.T. Ewing, M.A. Thompson, Forecasting wind speed with recurrent neural networks, Eur. J. Oper. Res. 221 (1) (2012) 148–154.
[18] J. Zhou, X. Yu, B. Jin, Short-term wind power forecasting: a new hybrid model

combined extreme-point symmetric mode decomposition, extreme learning machine and particle swarm optimization, Sustainability 10 (9) (2018).
[19] P. Jiang, Y. Wang, J. Wang, Short-term wind speed forecasting using a hybrid model, Energy 119 (2017) 561–577.
[20] F. Fazelpour, N. Tarashkar, M.A. Rosen, Short-term wind speed forecasting using artificial neural networks for Tehran, Iran, Int. J. Energy Environ. Eng. 7 (4) (2016) 377–390.
[21] A. Tesfaye, J.H. Zhang, D.H. Zheng, D. Shiferaw, Short-term wind power forecasting using artificial neural networks for Resource Scheduling in Microgrids, Int. J. Sci. Eng. Appl. 5 (3) (2016) 144–151.
[22] H. Masrur, M. Nimol, M. Faisal, G. Mostafa, "Short Term Wind Speed Forecasting Using Artificial Neural Network : A Case Study," in 2016 International Conference On Innovations In Science, Engineering And Technology, ICISET, 2016, pp. 1–5.
[23] J.P.S. Catalão, H.M.I. Pousinho, V.M.F. Mendes, Short-term wind power forecasting in Portugal by neural networks and wavelet transform, Renew. Energy 36 (4) (2011) 1245–1251.
[24] P. Kumar, N. Singh, M. Ansari, Solar radiation forecasting using artificial neural network with different meteorological variables, in: 2017 Innovations In Power And Advanced Computing Technologies (I-Pact), 2017, pp. 487–491.
[25] J. Mahmoudi, M. Jamil, H. Balaghi, Short and mid-term wind power plants forecasting with ANN, in: 2012 Second Iranian Conference on Renewable Energy and Distributed Generation, 2012, pp. 167–171.
[26] S.H. Chen, et al., Application of bias corrections to improve hub-height ensemble wind forecasts over the Tehachapi Wind Resource Area, Renew. Energy 140 (2019) 281–291.
[27] A. Ahmed, M. Khalid, Multi-step ahead wind forecasting using nonlinear autoregressive neural networks, Energy Procedia 134 (2017) 192–204.
[28] C. Wu, J. Wang, X. Chen, P. Du, W. Yang, A novel hybrid system based on multi-objective optimization for wind speed forecasting, in: Renew. Energy, 2019.
[29] P.J. Zucatelli, et al., Short-term wind speed forecasting in Uruguay using computational intelligence, Heliyon 5 (5) (2019), e01664.
[30] A. Senior, G. Heigold, M. Ranzato, K. Yang, An empirical study of learning rates in deep neural networks for speech recognition, ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc. (2013) 6724–6728.
[31] C. Li, Z. Zhu, H. Yang, R. Li, An innovative hybrid system for wind speed forecasting based on fuzzy preprocessing scheme and multi-objective optimization, Energy 174 (2019) 1219–1237.
[32] Python Language Reference, Version 3.6.3." Python Software Foundation, Available at: https://www.python.org/.
[33] S. Agatonovic-Kustrin, R. Beresford, Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research, J. Pharmaceut. Biomed. Anal. 22 (5) (2000) 717–727.
[34] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, AISTATS '11 Proc. 14th Int. Conf. Artif. Intell. Stat. 15 (2011) 315–323.
[35] V. Nair, G.E. Hinton, Rectified Linear Units Improve Restricted Boltzmann Machines, 2010.
[36] W. Shang, K. Sohn, D. Almeida, H. Lee, Understanding and Improving Convolutional Neural Networks via Concatenated Rectified Linear Units, 2016.
[37] M.A. Nielsen, Neural Networks and Deep Learning, Determination Press, 2015.
[38] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings Of the Thirteenth International Conference On Artificial Intelligence And Statistics (AISTATS-10) 9, 2010, pp. 249–256.
[39] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE International Conference on Computer Vision 2015, 2015, pp. 1026–1034. Inter.
[40] G. Giebel, R. Brownsword, G. Kariniotakis, M. Denhard, C. Draxi, Report: the state of the art in short-term prediction of wind power 110 (2011).
[41] C. Draxl, A.C. Nrel, A Guide to Using the WIND Toolkit Validation Code, 2014.
[42] C. Draxl, A. Clifton, B.M. Hodge, J. McCaa, The wind integration national dataset (WIND) Toolkit,, Appl. Energy 151 (2015) 355–366.
[43] C. Draxl, B. Hodge, A. Clifton, Overview and Meteorological Validation of the Wind Integration National Dataset Toolkit, 2015.
[44] J. King, A. Clifton, B.-M. Hodge, Validation of Power Output for the WIND Toolkit, 2014.
[45] T. Chai, R.R. Draxler, Root mean square error (RMSE) or mean absolute error (MAE)? – arguments against avoiding RMSE in the literature, Geosci. Model Dev. (GMD) 7 (2014) 1247–1250.
[46] H. Robbins, S. Monro, A stochastic approximation method, Ann. Math. Stat. 22 (3) (1951) 400–407.
[47] J. Duchi, E. Hazan, Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, J. Mach. Learn. Res. 12 (2011) 2121–2159.
[48] S. Ruder, An Overview of Gradient Descent Optimization Algorithms, 2016.
[49] M.D. Zeiler, ADADELTA: An Adaptive Learning Rate Method, 2012.
[50] D.P. Kingma, J. Ba, "Adam: A Method for Stochastic Optimization," in International Conference On Learning Representations, 2014, pp. 1–15.
[51] T. Dozat, Incorporating Nesterov momentum into Adam, ICLR Work 1 (2016) 2013–2016.
[52] H. Madsen, P. Pinson, G. Kariniotakis, H.A. Nielsen, T.S. Nielsen, Standardizing the performance evaluation of short- term wind power prediction models, Wind Eng. 29 (6) (2005) 475–489.
[53] H. Holttinen, J. Miettinen, S. Sillanpää, Wind Power Forecasting Accuracy and Uncertainty in Finland, 2013. Finland.
[54] G. Kariniotakis, et al., What performance can be expected by short-term wind power prediction models depending on site characteristics ? Forecast (2004) (November 2016) 22–25.
[55] D. Zastrau, M. Schlaak, T. Bruns, R. Elsner, O. Herzog, Differences in wind forecast accuracy in the German north and baltic seas, Int. J. Environ. Sustain Dev. 5 (6) (2014) 575–580.
[56] Global Wind Atlas, Global wind atlas [Online]. Available, https://globalwindatlas.info/about/purpose. (Accessed 7 July 2018).
[57] A. Mouez Khatab, H. Olivares Espinosa Examiner, J. Nørkaer Sørensen, Performance Analysis of Operating Wind Farms, September, 2017.
[58] N.G. Mortensen, A. Tindal, L. Landberg, Field validation of the delta-rix performance indicator for flow in complex terrain, Terrain 1 (1) (2006), 916–916.
[59] G. Giebel, The State of the Art in Short-Term Prediction of Wind Power, Roskilde, 2011.
[60] L. Bottou, Stochastic gradient descent tricks, in: G. Montavon, G. Orr, K. Muller (Eds.), In Neural Networks: Tricks Of the Trade, Springer, 2012, pp. 421–436.
[61] L. Zhao, et al., Optimization of an artificial neural network system for the prediction of failure analysis success, Microelectron. Reliab. 92 (2019) 136–142. September 2018.
[62] H. Zhao, F. Liu, H. Zhang, Z. Liang, Research on a learning rate with energy index in deep learning, Neural Network. 110 (2019) 225–231.
[63] D. She, M. Jia, Wear indicator construction of rolling bearings based on multi-channel deep convolutional neural network with exponentially decaying learning rate, Meas. J. Int. Meas. Confed. 135 (2019) 368–375.
[64] Y. Bengio, Practical recommendations for gradient-based training of deep architectures, Lect. Notes Comput. Sci. 7700 (2012) 437–478. LECTU.
[65] H.L. Hsieh, M.M. Shanechi, Optimizing the Learning Rate for Adaptive Estimation of Neural Encoding Models, 14, 2018, 5.
[66] J. Shi, J. Song, B. Song, W.F. Lu, Multi-Objective Optimization Design through Machine Learning for Drop-On-Demand Bioprinting, Engineering, 2019, 0–7.
[67] Y. Xu, H. Wang, X. Liu, W. Sun's, An improved multi-branch residual network based on random multiplier and adaptive cosine learning rate method, J. Vis. Commun. Image Represent. 59 (2019) 363–370.
[68] R.J. Hyndman, A.B. Koehler, Another look at measures of forecast accuracy, Int. J. Forecast. 22 (4) (2006) 679–688.
[69] J. Collins, J. Parkes, A. Tindal, Short term forecasting for utility-scale wind farms - the power model challenge, Wind Eng. 33 (3) (2009) 247–257.
[70] E.V. Mc Garrigle, P.G. Leahy, Quantifying the value of improved wind energy forecasts in a pool-based electricity market, Renew. Energy 80 (2015) 517–524.
[71] I. González-Aparicio, A. Zucker, Impact of wind power uncertainty forecasting on the market integration of wind energy in Spain, Appl. Energy 159 (2015) 334–349.
[72] U. Focken, M. Lange, K. Mönnich, H.P. Waldl, H.G. Beyer, A. Luig, Short-term prediction of the aggregated power output of wind farms - a statistical analysis of the reduction of the prediction error by spatial smoothing effects, J. Wind Eng. Ind. Aerod. 90 (3) (2002) 231–246.