

Postprocessing of NWP Precipitation Forecasts Using Deep Learning

ADRIAN ROJAS-CAMPOS,^a MARTIN WITTENBRINK,^b PASCAL NIETERS,^a ERIK J. SCHAFFERNICHT,^c JAN D. KELLER,^{b,d}
AND GORDON PIPA^a

^a *Institute of Cognitive Science, Osnabrück University, Osnabrück, Germany*

^b *Deutscher Wetterdienst, Offenbach, Germany*

^c *Department of Physics, Imperial College London, London, United Kingdom*

^d *Hans Ertel Centre for Weather Research, Bonn, Germany*

(Manuscript received 20 December 2021, in final form 2 January 2023)

ABSTRACT: This study analyzes the potential of deep learning using probabilistic artificial neural networks (ANNs) for postprocessing ensemble precipitation forecasts at four observation locations. We split the precipitation forecast problem into two tasks: estimating the probability of precipitation and predicting the hourly precipitation. We then compare the performance with classical statistical postprocessing (logistical regression and GLM). ANNs show a higher performance at three of the four stations for estimating the probability of precipitation and at all stations for predicting the hourly precipitation. Further, two more general ANN models are trained using the merged data from all four stations. These general ANNs exhibit an increase in performance compared to the station-specific ANNs at most stations. However, they show a significant decay in performance at one of the stations at estimating the hourly precipitation. The general models seem capable of learning meaningful interactions in the data and generalizing these to improve the performance at other sites, which also causes the loss of local information at one station. Thus, this study indicates the potential of deep learning in weather forecasting workflows.

KEYWORDS: Precipitation; Numerical weather prediction/forecasting; Postprocessing; Deep learning; Neural networks; Rainfall

1. Introduction

Modern-day weather forecasting is a multistep process that generates predictions for the atmospheric state and atmospheric phenomena on various temporal and spatial scales. This process is usually based on the measurement of relevant meteorological parameters, the compilation of a best-estimate initial state using these observations in a data assimilation scheme, and the forward integration of a numerical weather prediction (NWP) model of the physical equations governing the evolution of the atmospheric processes. The latter part of NWP has significantly improved over the last decades (Bauer et al. 2015) at multiple meteorological centers around the world.¹ The increase in performance is based on higher model resolutions, more sophisticated representations of physical processes, and enhanced data assimilation schemes for better initial state estimates.


Despite these developments, inherent shortcomings of these systems in the representation of some processes remain, especially for phenomena of highly nonlinear character, such as precipitation. In this respect, the output of numerical mod-

els is still prone to biases and representation errors, thus, making postprocessing a necessary step in producing weather forecasts. A correction is applied to the model output such that the postprocessed forecast fits the observed meteorological conditions better.

The postprocessing of NWP model output has now been in practice for half a century (Glahn and Lowry 1972) and represents an integral part of the NWP chain. In general, a postprocessing scheme aims to remove systematic biases, incorporate local scale adjustments, correct errors, and produce a finer scale end-use product (Schultz et al. 2021). The process can be understood as estimating a function that maps the model's output onto the observed weather data (Rasp and Lerch 2018). While this mapping would optimally be highly nonlinear in order to capture the nonlinear interactions and dependencies between the model variables and the observations, postprocessing is often implemented based on statistical models [for an overview, the reader is referred to Vannitsem et al. (2021)] using linear dependencies. In recent years, a possible alternative to classical methods has been found in utilizing an artificial neural network (ANN) approach, which is commonly the basis for modern machine learning.

The ANN is a statistical tool inspired by a biological brain capable of approximating any arbitrary function. This approximation is conducted by learning the set of values of parameters that result in the best function describing the dependencies between the input and the output. The learning of the parameters is performed through the exposure of the neural network to the input data and the target result. In that way, the algorithm learns to perform an optimized mapping between the input and the output (Goodfellow et al. 2016).

¹ Current numbers are, for example, available at <http://epsv.kishou.go.jp/EPsv/>.

 Denotes content that is immediately available upon publication as open access.

Corresponding author: Adrian Rojas-Campos, rrojascampos@uos.de

ANNs can learn and capture linear and nonlinear complex relationships in large datasets that are not necessarily obvious. Further, there does not have to be any assumption on the distribution (e.g., Gaussian, gamma) of the variables in the hidden layers of the ANNs and they have overcome classical statistical methods in several fields. They have shown impressive performance in tasks too complex to be solved by rule-based systems, which raises the question of to what extent ANNs can be used to improve NWP forecasts.

In general, the utilization of machine learning in the process of weather forecasting is distributed along three main approaches (Haupt et al. 2021). The most complex approach is to substitute the complete forecast process with a system of neural networks (Sønderby et al. 2020). However, some criticized this approach, arguing that machine learning algorithms are not suited to address the challenges of simulating the processes in the Earth system (Reichstein et al. 2019). A less complex possibility is to replace certain parts of the numerical model, such as computationally expensive and/or parameterized modules (e.g., Brenowitz and Bretherton 2018). The least complex and straightforward approach is to perform a postprocessing step based on ANNs, which is also the focus of this study.

With respect to the latter approach, several implementations of ANNs for the postprocessing of NWP model output have recently been developed for various parameters such as temperature (e.g., Rasp and Lerch 2018; Peng et al. 2020) and wind speed (e.g., Huang and Kuo 2018; Candido et al. 2020; Veldkamp et al. 2021) or for specific applications like renewable energy production (e.g., Sharifian et al. 2018; Theocharides et al. 2020; Haupt et al. 2020).

This study aims to evaluate the potential of ANNs in postprocessing hourly precipitation predictions based on NWP model output. Our approach is to set up, adapt and train the ANNs and assess their added value compared to reference predictions using classical postprocessing techniques such as logistic regression and generalized linear models (GLMs). The precipitation predictions will be performed in a two-part process, first predicting the probability of precipitation as a binary event (yes/no) and then quantitatively predicting the hourly precipitation amount. An ANN approach with a probabilistic output is explored. While there are efforts aimed at precipitation postprocessing with ANNs, the targeted time scale of these studies is quite different from the hourly periods envisioned here, namely, for daily (Ghazvinian et al. 2021; Zhang and Ye 2021) or monthly to seasonal forecasts (Ghamariadyn and Imteaz 2021; Fan et al. 2023; Scheuerer et al. 2020).

The remainder of the paper is structured as follows: the next section describes the data used in the study. Section 3 explains the ANN approach and the classical postprocessing techniques used as a baseline in this paper. In section 4, we present the results of the postprocessing experiments. We end with a discussion of the results and conclusions in section 5.

2. Data

This study uses NWP model output and observational data to set up and train the ANNs and the classical postprocessing

models. Four common locations were selected to perform an initial detailed evaluation of the ANN-based postprocessing.

a. NWP model output

As the input forecasts, we use the NWP model output of the limited area model COSMO-DE-EPS (Peralta et al. 2012) for the period 2011–17, representing the operational ensemble predictions of the German Meteorological Service (DWD) at the time. The COSMO-DE-EPS is a probabilistic model combining multiple instances of COSMO-DE with different starting conditions, comprising 20 ensemble members. It covers Germany, Switzerland, and Austria, as well as parts of the neighboring countries, and it has a resolution of 2.8 km².

Specifically, for this study, we obtained all forecasts initialized at 0000 UTC with lead times of 3, 4, and 5 h. For each of the four stations used in the study (see below), we extract the COSMO-DE-EPS information of a 5 × 5 grid points (about 14 × 14 km²) neighborhood around the corresponding grid cell. We include all 143 variables of the forecast, with predictions of precipitation, temperature, pressure, etc. Further, for each model variable, we calculate the ensemble mean and standard deviation (i.e., ensemble spread) from the 20 ensemble members, thus resulting in 3575 predictors (143 model parameters at 25 grid points) as input for the postprocessing.

Including all variables of the COSMO-DE-EPS model gives more information to the algorithm about the atmospheric state surrounding the station and allows the ANNs to capture the nonlinear interactions between those variables that help improve the precipitation forecast. However, including all 20 ensemble members as input information could cause a numerical problem known as the curse of dimensionality. The curse of dimensionality is an issue that arises when the input information has too many dimensions. As the number of relevant dimensions of the data increases, the number of configurations of interest grows exponentially and this can render the problem too complex, making it hard or impossible to find a solution to generalize (Goodfellow et al. 2016). To react to this we calculate the mean and standard deviations of the model variables to preserve the essential information from the 20 ensembles, following previous publications on postprocessing using ANNs (Rasp and Lerch 2018).

b. Precipitation observations

As a target, we use the historical hourly precipitation observations from four selected stations of DWD's measurement network: Münster-Osnabrück, Wernigerode, Braunlage, and Redlendorf. The stations represent flatland to midrange mountain topographical situations but avoid more complex (alpine) or forced (i.e., atmospheric forcing: coastal/maritime) characteristics.

The fraction of time steps at which the measurements indicate precipitation is relatively low for each station. In Münster-Osnabrück, the measurements with precipitation represent 10.64% of all observations, 9.48% for Wernigerode, 17.47% for Braunlage, and 17.66% for Redlendorf. As expected, the precipitation amount at each station resembles a gamma distribution, as shown in Fig. 1.

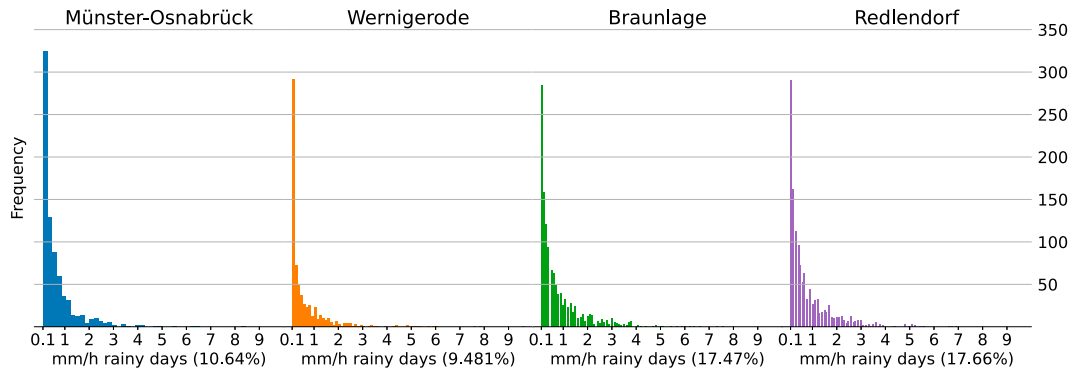


FIG. 1. Distribution of registered precipitations for time steps with precipitation $> 0 \text{ mm h}^{-1}$.

c. Data preprocessing and feature selection

To be used in any of the postprocessing schemes, the data has to be preprocessed. First, the data are compiled by pairing at each time step each observation with the respective COSMO-DE-EPS forecast data. Thus, the time step is discarded if either the observation or the model forecast is missing. Additionally all forecast variables were z-score transformed to reduce numerical problems due to different measurement units.

To reduce the dimensionality of the input data, we employ a Lasso regularization based on L1. This procedure was made for each station to select the most relevant predictors based on the rainy days. The Lasso procedure is a standard regression using an additional error term. This additional error term depends only on the model's parameters and not on the data itself. In the case of Lasso, the additional error term is the sum of absolute weights and, thereby, a penalty for parameters of significant magnitude. Therefore, combining data-driven and only parameter-driven error terms for fitting models prefers models that describe the data well and have small parameters. A particular property of Lasso and L1-based regression is that features that are not important are getting parameters equal to zero and, therefore, can be excluded from the set of used features. The parameter lambda controls the strength of the regularization penalty and, thereby, the number of selected features. The larger the lambda, the smaller the number of used features.

This work performed a full regularization path varying the strength of the L1 penalty. We use cross validation in a randomly selected validation set to identify the best-performing model across different regularization strengths. The selected model regularization strength λ is then chosen at the lowest mean squared error in log space. After feature selection based on the L1 regularization, a linear regression restricted to the

selected features is performed to estimate the regression parameters for these features without a regularization penalty. The results of the feature selection procedures are presented in [appendix A](#), and each station's number of selected features (from 3575) is reported in [Table 1](#).

d. Training, validation, and test dataset

From the complete set of forecasts and observations at each station, 10% was randomly selected for building the test datasets, and the remaining 90% was used for building the training sets. The postprocessing is performed for the 4-h lead time forecast, while the training dataset also comprises the adjacent forecast hours 3 and 5. Therefore, we have four pairs of training and test datasets, one for each station. The final length and dimensions of the training and test datasets used in all following sections are presented in [Table 1](#).

Some implemented procedures like architecture search and cross validation required using a validation dataset. In all cases, the validation set is randomly and dynamically generated as a 10% of the training set and not used for training. In this way, statistical generation is enhanced, and there is no need for an initial extra data split.

3. Methodology

Our approach splits the postprocessing of precipitation into two distinct tasks: the first is to estimate the probability of the hourly precipitation exceeding a threshold of 0.1 mm h^{-1} , and the second is to estimate the hourly precipitation registered by the station (in mm h^{-1}). For each of these tasks, two different postprocessing routines were developed and tested: classical statistical postprocessing techniques (logistic regression

TABLE 1. Final dataset's shapes after feature selection.

Station	Training dataset (lead time 3, 4, and 5 h)		Test dataset (lead time 4 h)	
	Input	Output	Input	Output
Münster-Osnabrück	6909×41	6909×1	246×41	246×1
Wernigerode	6812×28	6812×1	265×28	265×1
Braunlage	6894×117	6894×1	237×117	237×1
Redlendorf	6937×61	6937×1	234×61	234×1

and generalized linear models) on the one hand and probabilistic ANNs on the other.

Probabilistic has become the primary type of postprocessing during the last years (Vannitsem et al. 2021). A probabilistic prediction consists of a probability distribution approximating the data generating process. Probabilistic ANNs learn to map the input with the parameters of an output probability distribution, which allows them to provide a range of forecasts with an associated probability. Additionally, a probabilistic approach requires assuming a type of probability distribution for the phenomenon to predict. The following subsections describe the details of the different algorithms and the metrics to evaluate their respective performance.

a. ANN training and testing

Before the training of an ANN can take place, a model architecture (the number of layers and units per layer) needs to be defined. Here, we have performed a preliminary architecture search for each station using their respective training datasets. This exploration gives, as a result, a specific number of layers and units per layer at each station. The process of the architecture search is detailed in [appendix B](#).

The training of an ANN model is performed by presenting the input data together with the desired output of the neural network in an iterative process until a certain level of loss is obtained. Through this process, the network is able to learn the statistical patterns in the input information, hence allowing the network to map this information on the desired output (Goodfellow et al. 2016). During the training phase, the model is fitted using the training dataset. Various techniques may be implemented to enhance statistical generalizability.

Here, we apply a k -fold cross validation during which the training dataset is randomly split into k subsets in which each subset is randomly split into a training part of size $(k-1)/k$ and a validation part of size $1/k$. Using these subsets, k models are independently initialized and fitted based on the training part and validated against the respective validation part. The model with the best performance in their respective validation set is then selected as the result of the training phase. In this study, we used tenfold cross validation during the training phase of each ANN model.

However, the performance of a single ANN model is still subject to considerable variations due to the random initialization of the weights as these have a direct impact on the outcome of the process. To allow for more robust comparisons of the performance of the ANN models, we repeat the aforementioned procedure 20 times with independently chosen random initializations using the same basic model architecture.

Then, this set of models is evaluated with the unobserved test dataset, i.e., predictions are generated with the obtained models by applying them to the test data. These predictions are then compared with the respective precipitation observations using a distance measure which allows us to quantify the models' performance.

b. Estimating the probability of precipitation

The first postprocessing task consists of estimating the probability of hourly precipitation exceeding a threshold of 0.1 mm h^{-1} . For this, a binary transformation of the target precipitation amount was implemented.

1) LOGISTIC REGRESSION

As a baseline for comparison, a classical logistic regression was performed. A logistic regression maps a linear regression on a sigmoid that represents an estimated probability between 0 and 1. The regression coefficients are fitted by a maximum likelihood approach and a probability estimate can be obtained. To prevent the logistic regression from overfitting, a L2-norm penalty term is added to the maximum likelihood optimization problem. The magnitude of this penalty term is controlled by a generalization parameter, which has to be determined for every classification problem. Optimal generalization parameters were found by performing logistic regression classifications with different generalization parameters (from 0.0001 to 1000). Then, the parameter with the best performance (in terms of classification accuracy) is selected.

2) ANN PROBABILITY ESTIMATION

Probabilistic ANNs map the input information with a Bernoulli distribution that provides the probability of precipitation exceeding the 0.1 mm h^{-1} threshold. Bernoulli distribution is used to model random variables whose outcome can be 0 or 1. For each precipitation station, 20 independent ANN algorithms for estimating the probability of precipitation were developed and trained using each station's respective training dataset. Each ANN model consists of a feed-forward network starting with an input layer, followed by n hidden layers with m units with a rectified linear unit (relu) (which changed according to the station following the architecture search, see [appendix B](#)). The network ends with a final single unit output layer with a Bernoulli distribution. The negative logarithmic likelihood was used as a loss function with the Adam optimizer to perform the weights update with a learning rate of 0.0001. The training was performed during 64 epochs using a batch size of 128, and early stopping was used to prevent overfitting. During the test phase, each station-specific ANN was used to generate the predictions for their specific precipitation station. From the resulting output distributions, the parameter p was obtained as the probability of precipitation.

3) EVALUATION METRICS FOR THE PROBABILITY OF PRECIPITATION

One of the main verification methods for dichotomous or categorical probabilistic forecast is the Brier score (Murphy 1973). The Brier score is used to measure the difference between the predicted probability of an event occurring p_i , and the observation of the event happening or not o_i where 1 equals occurrence and 0 equals no occurrence. The distance is

measured as the mean squared probability error calculated in the following way:

$$BS = \frac{1}{N} \sum_{i=1}^N (p_i - o_i)^2. \quad (1)$$

Using the Brier score, a Brier skill score can be calculated. The Brier skill score refers to a relative skill of the probabilistic forecast over a reference model (COSMO-DE-EPS in this case). A skill score of 1 indicates a perfect forecast, and zero or lower indicates no improvement compared to the reference. The Brier skill score is defined as

$$\text{Brier skill score} = 1 - \frac{\text{Brier}_{\text{ANNs_or_logistic_reg}}}{\text{Brier}_{\text{COSMO-DE-EPS}}}. \quad (2)$$

c. Hourly precipitation estimation

In this task, we want to calculate the hourly precipitation registered by the precipitation station (in mm h^{-1}), assuming the occurrence of precipitation. In this case, we use the information only from data points with hourly precipitation higher or equal to 0.1 mm h^{-1} .

1) GENERALIZED LINEAR MODEL

To provide a reference with respect to the performance of the ANN-based approach, classic postprocessing based on generalized linear models (GLM) is employed. A GLM is a generalization of linear regression models that can handle nonlinear and other than normally distributed data. Depending on the data, a link function between the target data and a linear predictand has to be defined and the probability distribution of errors has to be specified. According to the nature of precipitation data, we tested two GLM approaches: an identity link function and the assumption of normally distributed errors on log-transformed precipitation data and a logarithmic link function and the assumption of gamma-distributed errors for nontransformed precipitation data. Results showed only minor differences between these two approaches, and the former approach was pursued due to slightly better performance. To avoid excessive computational cost, an upper limit of iterations in the fitting process was set to 100 000.

2) ANN PROBABILISTIC PRECIPITATION PREDICTION

Probabilistic ANNs for predicting the hourly precipitation map the input with the parameters of a Gaussian distribution using a log-transformed target variable. The ANN calculates the mean and standard deviation with the highest probability according to the input data. In a similar way than the ANNs employed to estimate the probability of precipitation, 20 independent ANNs were trained and tested using their respective datasets at each station. The architecture search informed the architecture selection (see [appendix B](#)). Relu was used in all hidden units. After the hidden layers, two densely connected layers followed by a single unit normal distribution layer were used as output. The negative logarithmic likelihood was used as a loss function and Adam optimizer (0.0001). The training

was performed during 128 epochs with a batch size of 64, and early stopping was used. The median of the predicted distributions for the test set was obtained as a deterministic forecast to enable evaluation and comparison.

3) EVALUATION METRICS FOR PRECIPITATION PREDICTION

The continuous ranked probability score (CRPS; [Hersbach 2000](#)) is a widely used verification method to compare probabilistic continuous forecasts with deterministic targets. However, the CRPS does not perform well in small datasets where it shows a high numerical instability ([Zamo and Naveau 2018](#)). Given the overall small number of observed precipitation events and the pronounced skewness in the respective distribution, we chose not to utilize the CRPS as a metric to evaluate the performance of our estimate models.

Instead, we use the median of the predicted distributions as a deterministic forecast and evaluate its performance using the skill score of the linear error in probability space (LEPS) ([Ward and Folland 1991](#)). LEPS is the mean absolute difference between the values that the forecast and observation take in the observations' climatological cumulative distribution function (CDF), i.e., LEPS is analog to the RMSE but in CDF space. This is an especially appropriate metric for the evaluation of precipitation as it does not overstate the significance of potentially larger errors at higher precipitation amounts.

The term skill score indicates the degree of improvement of the postprocessed LEPS compared to the LEPS of the original COSMO-DE-EPS forecasts as reference. The forecast is perfect when the skill score equals 1, while a skill score of zero and below means no improvement or less skill in the postprocessed estimates compared to the reference.

Considering f_i as a distinct forecast, o_i as the respective observation and $\text{CDF}_o(\cdot)$ as the CDF of the observations determined by an appropriate climatology, LEPS is defined as

$$\text{LEPS} = \frac{1}{N} \sum_{i=1}^N \left| \text{CDF}_o(f_i) - \text{CDF}_o(o_i) \right|, \quad (3)$$

with the LEPS based skill score for the postprocessing algorithms being calculated as

$$\text{LEPS skill score} = 1 - \frac{\text{LEPS}_{\text{ANNs_or_GLM}}}{\text{LEPS}_{\text{COSMO-DE-EPS}}}. \quad (4)$$

d. Generalized postprocessing approaches

The approach mentioned above defines a specific ANN model for each station individually, which exhibits one significant limitation: it requires an architecture search, feature selection, and model training for each station, which could become computationally expensive for many stations. To tackle this limitation, we train 20 additional general ANN models using the merged datasets from all stations for each task. In this case, we selected a deeper and wider architecture (8 hidden layers of 64 units each) in order to account for the potentially increased complexity. All other hyperparameters

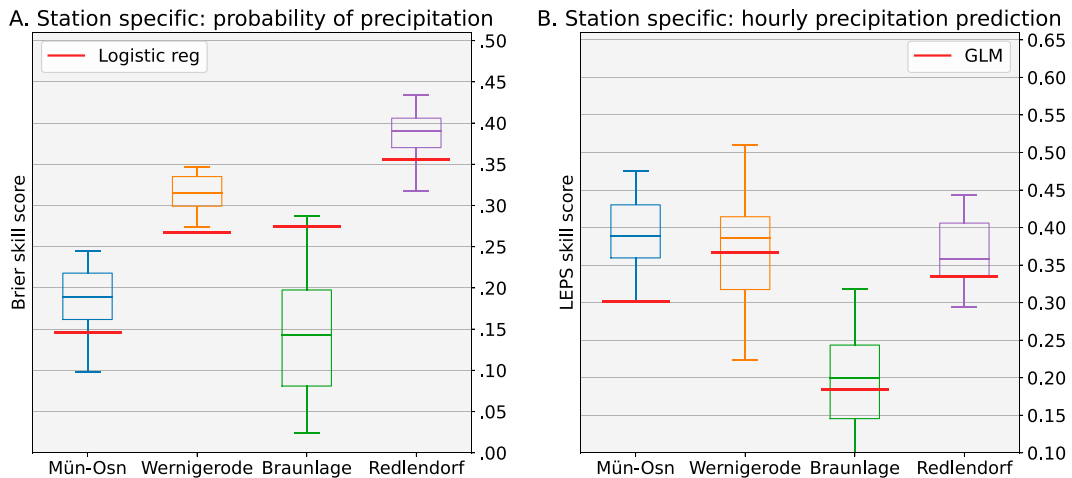


FIG. 2. (a) Brier skill scores of 20 ANN models trained to estimate the probability of precipitation. (b) LEPS skill score of 20 ANN models trained for precipitation amount prediction task. Statistical baseline performance (logistic regression and GLM) is included as a reference with a red line. Both skill scores range from $-\infty$ to +1, with negative values indicating worst performance than the reference model, positive values indicating improvement, and a perfect forecast with +1.

were similar to the station-specific ANNs. Each of these 20 general ANNs were used to generate predictions for each station, and their performance was evaluated individually at each location.

e. Implementation

The management and analysis of the data and the training and testing of the different algorithms has been developed on the JUWELS Supercomputer at the Jülich Supercomputer Center (JSC). The implementation is based on Python 3.8.3 (Van Rossum and Drake 1995) together with the Scipy Ecosystem (Virtanen et al. 2020). TensorFlow 2.3.1 (Abadi et al. 2015) and TensorFlow Probability (Dillon et al. 2017) were utilized for the implementation of the deep learning algorithms. The mpi4py library (Dalcin et al. 2019) is used to distribute tasks on the HPC. The logistic regression and the GLM are implemented using the python package scikit-learn (Pedregosa et al. 2011). Finally, we use Jupyter Laboratory for data preprocessing and plotting (Kluyver et al. 2016).

4. Results

Our methodology splits the postprocessing problem into two separated tasks: estimating the probability of precipitation and predicting the hourly precipitation. For each task, we trained 20 station-specific ANNs using the training data from each station and 20 general ANNs using the merged training set. The performance of the ANNs is compared against the result of classical logistic regression and a GLM-based postprocessing scheme using the same datasets for training and testing. This section presents the results of applying an ANN algorithm to postprocess precipitation predictions at four exemplary observation sites in Germany.

a. Station-specific postprocessing

We first train and test the ANNs using the datasets from the respective station exclusively. Figure 2a summarizes the skill scores for the probability of precipitation estimation and Fig. 2b is for the precipitation amount prediction. Each box-and-whisker plot represents 20 ANN runs using random weight initializations, and the statistical baselines are provided as a reference indicated by the red line.

1) STATION-SPECIFIC: PROBABILITY OF PRECIPITATION

First, we analyze the station-specific ANNs' performance to estimate the probability of the binary event "precipitation yes/no" at each station. The results are presented in Table 2.

As shown in Fig. 2a, in 3 out of 4 stations, the ANN postprocessing produced better predictions than the logistic regression, potentially due to the flexibility and nonlinear interactions considered by the ANNs. Specifically, at Münster-Osnabrück, Wernigerode, and Redlendorf, the ANNs' median values (see results for specific ANNs) are higher than those obtained by the logistic regression (LR).

Only Braunlage's median Brier skill score of the station-specific ANNs is below the references. This might be connected to the

TABLE 2. Brier skill scores of the specific and general ANNs as well as the Brier skill score for the logistic regression models for the four exemplary stations.

	Baseline	Specific ANNs		General ANNs	
	LR	Median	Max	Median	Max
Münster-Osnabrück	0.146	0.189	0.245	0.235	0.344
Wernigerode	0.267	0.315	0.347	0.306	0.392
Braunlage	0.270	0.143	0.287	0.287	0.377
Redlendorf	0.356	0.390	0.434	0.403	0.471

TABLE 3. LEPS skill scores of the specific and general ANNs as well as the LEPS skill score for the GLM for the four exemplary stations.

	Baseline	Specific ANNs		General ANNs	
	GLM	Median	Max	Median	Max
Münster-Osnabrück	0.302	0.389	0.475	0.267	0.344
Wernigerode	0.367	0.385	0.510	0.502	0.602
Braunlage	0.184	0.200	0.318	0.292	0.350
Redlendorf	0.335	0.359	0.443	0.388	0.424

geographical characteristics of Braunlage and the corresponding orographical amplification or attenuation of precipitation processes. The city is nested in the highland area of the Harz Mountains, with its main ridge lying to the West and Northwest, i.e., the directions of main upstream. Thus, the underlying patterns in the predictors might be more linear, and the logistic regression approach can provide a significant gain in performance. However, the best of the 20 ANN models is able to outperform the logistic regression in Braunlage.

2) STATION-SPECIFIC: HOURLY PRECIPITATION PREDICTION

Second, we look at the performance of station-specific ANNs to estimate the precipitation amount. Figure 2b shows the LEPS skill score of the ANNs in relation to the original COSMO-DE-EPS model output. The results are presented in Table 3.

In this case, the ANNs outperform the classical postprocessing approach at all stations. The median of the ANNs' scores is higher (see results for specific ANNs) than the reference GLM (GLM). Once more, we observe a significantly lower postprocessing performance at Braunlage compared to

other stations. However, in this occasion, the median ANN performance is above the statistical baseline.

b. Postprocessing over all stations

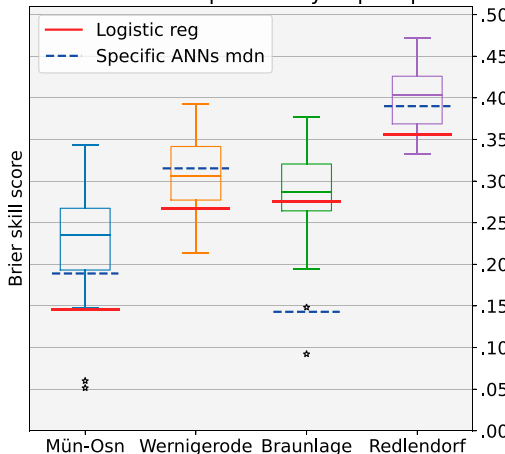
The results presented above show the potential of ANNs for postprocessing NWP precipitation forecasts compared to the statistical methods. As described in section 3, we also trained another version of the postprocessing models using the merged data from all four stations (named "general" ANNs) that we use to generate a prediction for each specific station. In the following, we compare the performance of the general ANN models to that of the station-specific ANNs and statistical baselines.

Figure 3a summarizes the Brier skill score for the general ANNs at each station and Fig. 3b illustrates the performance of the general ANNs using the LEPS skill score for the hourly precipitation forecast. The box-and-whisker plots for each station represent the performance at each individual station of 20 general ANNs using random initializations. The median performance of the station-specific ANNs are presented with a blue dashed line and the statistical baseline with a red line.

1) GENERAL ANNS: PROBABILITY OF PRECIPITATION

First, we analyze the general ANNs' performance for estimating the probability of precipitation for each station. We find that at 3 out of 4 stations, the performance of the ANNs increases compared to the station-specific ANNs when the model is trained with information from all stations (see general ANNs in Table 2). The only exception is Wernigerode, where the median performance slightly decreases but remains above the baseline. This difference in performance is within a range of 10% between the two types of models and therefore, the loss of station-specific information only leads to a slight

A. General model: probability of precipitation



B. General model: hourly precipitation prediction

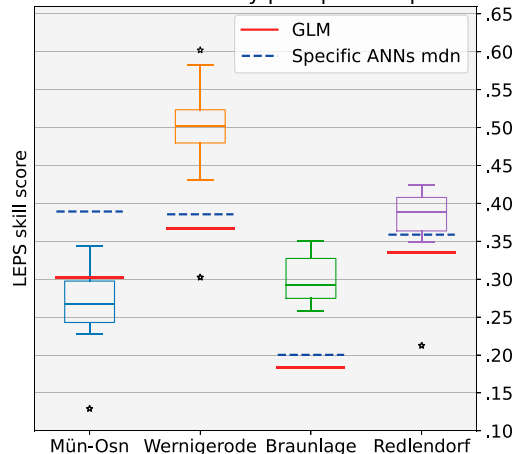


FIG. 3. (a) Brier skill scores of 20 ANN models trained with information from all stations to estimate the probability of precipitation for each test set. (b) LEPS skill score of 20 ANN models trained with information from all stations to calculate the precipitation amount for each test set. The median performance of the station-specific ANN is illustrated with a blue line, and the statistical baseline scores (logistic regression and GLM) are included as a reference with a red line. Both skill scores range from $-\infty$ to +1, with negative values indicating worst performance than the reference model, positive values indicating improvement, and a perfect forecast with +1.

decrease in performance. On the other hand, we observe a small increase in performance at Redlendorf, a moderate one at Münster-Osnabrück, and a very significant one at Braunlage, where the general ANN model overcomes the statistical baseline, which was not achieved by the station-specific ANNs.

2) GENERAL ANNS: HOURLY PRECIPITATION PREDICTION

Last, we look at the general ANNs' performance for estimating the hourly precipitation at each station. In this case, we also found an increase in performance at 3 out of the 4 stations, where the general ANNs obtained a higher median skill score than the station-specific ANNs. We observe a considerable boost in performance at Wernigerode and Braunlage, and a smaller one in Redlendorf (see general ANNs in Table 3). However, we also observe an important decay at Münster-Osnabrück, where the median of the general is below the specific ANNs and the statistical baseline.

The general ANNs results show the benefits of training a general model with all stations due to the ANNs' ability to extrapolate the learned interactions between stations. We also see the risks caused by the possible loss of local information at Münster-Osnabrück. Considerations derived from these results are discussed in the next section.

5. Discussion and conclusions

In this study, we investigate the application of ANNs as a probabilistic postprocessing method for convection-permitting NWP precipitation forecasts. Our results show that ANNs can significantly improve the quality of precipitation predictions compared to the original model output. Further, ANNs obtain higher median performance at most stations than the tested classical postprocessing techniques for estimating the probability of precipitation and hourly precipitation. Using ANNs as a postprocessing tool for weather forecasts can increase the final prediction quality by considering nonlinear and complex interactions between the different forecast variables in different locations around the station. However, as in any deep learning application, the performance is highly dependent on the proper selection of hyperparameters such as architecture, epoch number, and batch size.

One of the main contributions of this study is the comparison of ANNs concerning the level of cross-station generalization. We find that including information from multiple observing sites improves the ability of the postprocessing step at most of the stations. Using information from all stations allows the general models to generalize the learned relationships between stations, which impacts the performance of the ANNs. This can be clearly observed with the increase in performance at Braunlage for the probability of precipitation, and Wernigerode and Braunlage for the hourly precipitation prediction. The inclusion of samples from other stations allows the ANN to overcome the limitations of the station-specific models.

However, we also find that the using a single model for all stations can lead to a partial decrease in performance at specific sites. Although the station-specific ANNs perform above

the baseline in Münster-Osnabrück, this changes when we include information from all stations. This is an undesired effect of using a single model at all stations. The inclusion of more and more diverse samples for training causes the specific interactions learned from a station might be forgotten in the network (Goodfellow et al. 2016), generating a loss of valuable local information and therefore causing decay in performance. Specifically, we observe that the rainy days of Münster-Osnabrück has the precipitation with the lowest mean and highest standard deviation ($M = 0.66$, $SD = 1.42$) between all stations (Wernigerode: $M = 0.70$, $SD = 1.24$; Braunlage: $M = 0.79$, $SD = 0.98$; Redlendorf: $M = 0.79$, $SD = 0.97$). Using a model that learned the relationships from all stations could have generated a tendency to over forecast precipitation at this particular station leading to larger errors in the predictions. Possible approaches to reduce the decrease in performance are the inclusion of additional information about the stations and their surroundings, such as orographical details or land use. This would allow the network to learn the relationship between the specific characteristics of the station and the forecast errors in precipitation.

Additionally, it is essential to consider that general models obtain a comparable or better performance than the station-specific ANNs without requiring a specific architecture search and implementation, which makes them more resource efficient than the station-specific models.

We are aware that this study also has limitations. First, we lack a probabilistic metric to evaluate the precipitation amount predictions. As we discussed earlier, the CRPS becomes unstable in small datasets (Zamo and Naveau 2018). The small number of rainy days and the considerable skewness of the precipitation distribution, thus, made the CRPS an inappropriate metric in our study. Although the probabilistic ANNs provide a complete distribution for hourly precipitation, we are only able to evaluate the median of the distribution as a deterministic forecast, thus, losing potentially important information. Future work could either employ a probabilistic metric adapted to the specific precipitation properties or use longer dataset.

Another limitation of this work is the restricted selection of precipitation stations used. Given that this is a first approximation of the use of ANNs for postprocessing precipitation, we consider that a small selection of stations (avoiding complex or forced topographies) was needed as a first step to the inclusion of deep learning in the meteorological workflow for hourly precipitation. Including precipitation stations with highly contrasting underlying physics would imply a level of complexity out of the scope of this paper, and this limits the scope of our findings. Future applications need to involve more complex and forced locations.

A further well-known limitation in precipitation postprocessing is the need to split the problem into two independent tasks and, therefore, the need to implement two independent solutions. The lack of precipitation in most records and the hardly gamma-shaped distribution of the registered precipitation amounts make its postprocessing a challenging problem. This splitting increases the potential sources of error for the predictions and limits the amount of data available for the

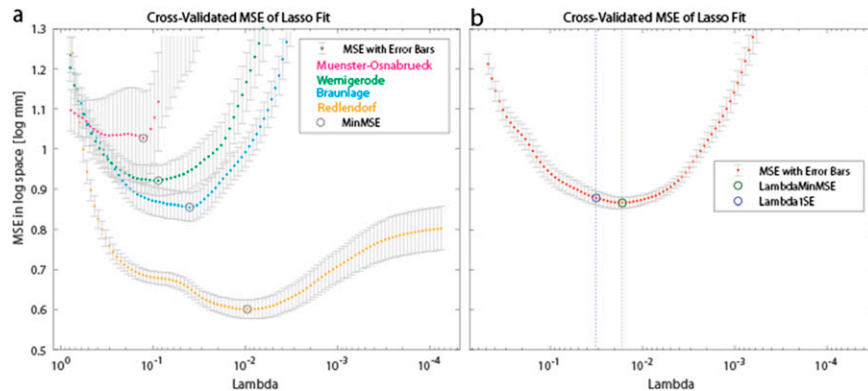


FIG. A1. Mean squared error of the log space precipitation data as a function of the regularization strength lambda. Error bars correspond to one standard error of the MSE evaluated on test data using a tenfold cross-validation scheme. (a) The regularization path for each station and (b) the regularization path for one model over all four stations. The open circles indicate the smallest observed MSE error in log space.

regression problem. Future deep learning solutions should avoid the tasks split between and thus be enabled to learn from all information available to provide a more accurate precipitation estimate.

In summary, our results provide initial evidence in favor of using ANNs for postprocessing precipitation from NWP, as previous works have done it for temperature and wind speed (Rasp and Lerch 2018; Peng et al. 2020; Huang and Kuo 2018; Candido et al. 2020; Veldkamp et al. 2021). According to our results, a possible orientation for future research and practical applications could be to develop general ANN algorithms trained with multiple and diverse stations and including information about the location and topography. Including information about the location and the orography would allow a general ANN to adapt their predictions according to its properties. However, this could affect the performance at locations with very specific topographies. One alternative idea is to develop topography-specific models using stations with shared similar characteristics. This half-way approach would require the development of multiple models, but it significantly reduces the required resources compared to the station-based approach.

An additional aspect that needs to be considered by future research is the unexplored relationship between the loss functions and the metrics to evaluate the predictions. ANNs are trained to minimize a defined loss function. Although we can infer a possible negative correlation between the loss function and the performance metrics from our results, the detailed interaction between both is still an open question. Future research needs to provide an adequate loss function for precipitation predictions that can be used in deep learning algorithms.

From a broader perspective, the results of this study provide evidence in favor of integrating deep learning in weather forecasting workflows. As mentioned by Haupt et al. (2021), or Schultz et al. (2021), partial integration of deep learning techniques has the potential to enhance forecasts and remove biases in simulating the atmospheric state. Especially the ability to learn and extrapolate significant relationships between stations suggests that future postprocessing of NWP forecasts

employing ANNs could benefit by using only a reduced number of models instead of the commonly used station-specific approaches. The nonlinear dependencies learned by the ANNs and the ability to transfer knowledge across stations are the key to the performance of the models.

Acknowledgments. This work was possible thanks to the DeepRain project funded by the Bundesministerium für Bildung und Forschung (BMBF) under Grant 01 IS18047A. The authors gratefully acknowledge Jülich Supercomputing Centre (JSC) for the computing time and support to develop this project. The authors have no conflicts of interest to declare.

Data availability statement. The code used in this study has been made public using the repository https://github.com/DeepRainProject/post_processing_precipitation and the datasets can be downloaded under the direction <https://b2share.eudat.eu/records/c765674ad42c4a46bc3b0fa780f3329b>.

APPENDIX A

Feature Selection Procedure

In Fig. A1, we show the different MSE values for each station with different regularization penalties (lambda values) for specific (Fig. A1a) and general models (Fig. A1b). Table A1 displays the number of relevant features obtained from the Lasso regularization and the final MSE of rainfall after applying regularization.

TABLE A1. Number of selected features and MSE of rainfall.

Precipitation station	No. of selected features	MSE in test (mm)
Münster-Osnabrück	41	0.643
Wernigerode	28	0.560
Braunlage	117	0.591
Redlendorf	61	0.530

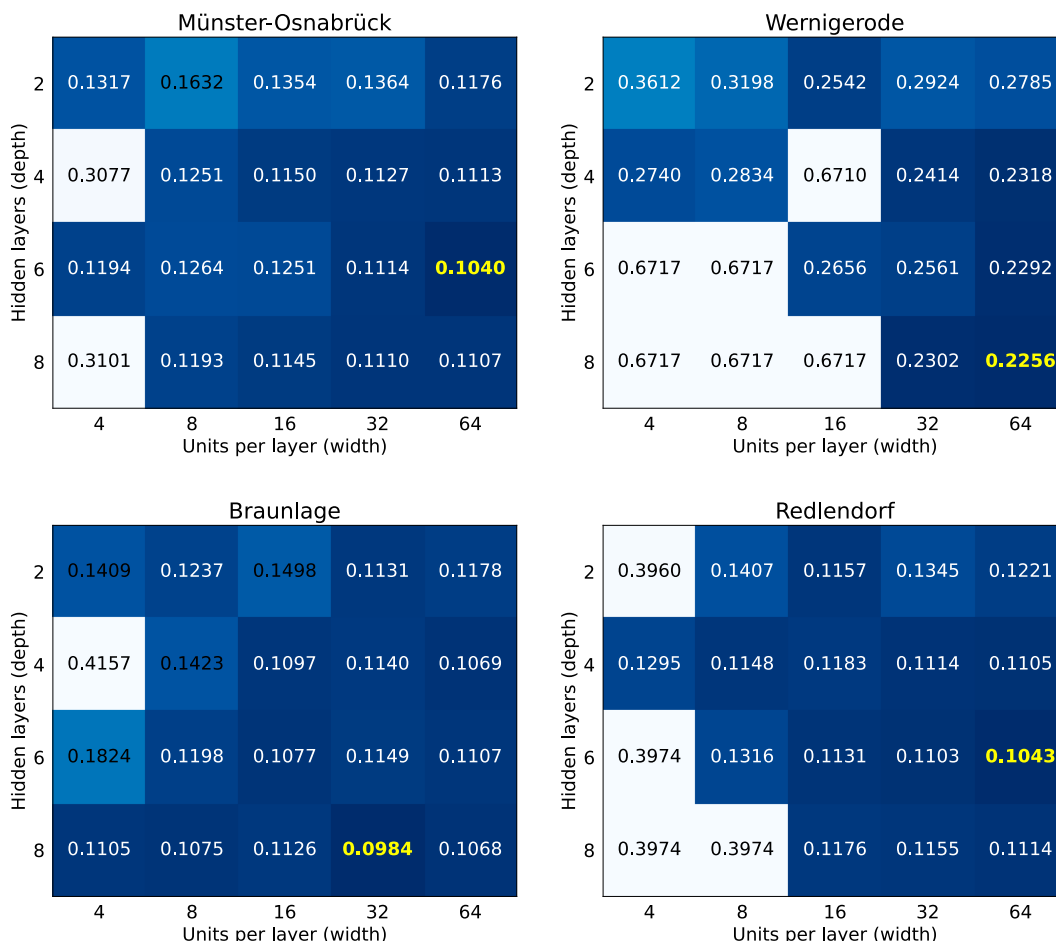


FIG. B1. Results of architecture exploration. Median skill score of exploration runs from the respective depth and width configuration, for the regression task. The highest performing architecture is shown in yellow.

APPENDIX B

ANN Architecture Search

The main advantage of an ANN is its ability to learn any arbitrary function from a specific problem. This capacity relies heavily on the right selection of hyper-parameters, such as training epochs, batch size, and network architecture. The network architecture corresponds to the way the different neurons are connected to each other, and it is configured by two main components: the number of hidden layers between input and output (depth) and number of neurons of these layers (width). The search for the right architecture for a specific problem is usually driven by experimentation guided by monitoring the validation set error (Goodfellow et al. 2016).

In this study, an initial architecture exploration is performed with the goal to find the best architecture for each of the four locations. This was required in order to develop each of the following steps. For each of the stations, depths of 2, 4, 6, and 8 together with widths between 4, 8, 16, 32, and 64 are tested by running five pilot models and estimating the median performance for the regression task in the validation

set. The complete results of this exploration are presented in Fig. B1.

As a result from the exploration, the architecture with the highest median performance is chosen for each of station, specifically:

- Münster-Osnabrück: 6 hidden layers of 64 neurons each
- Wernigerode: 8 hidden layers of 64 neurons each
- Braunlage: 8 hidden layers of 32 neurons each
- Redlendorf: 6 hidden layers of 64 neurons each.

REFERENCES

- Abadi, M., and Coauthors, 2015: TensorFlow: Large-scale machine learning on heterogeneous systems. Google Research, 19 pp., <https://storage.googleapis.com/pub-tools-public-publication-data/pdf/45166.pdf>.
- Bauer, P., A. Thorpe, and G. Brunet, 2015: The quite revolution of numerical weather prediction. *Nature*, **525**, 47–55, <https://doi.org/10.1038/nature14956>.
- Brenowitz, N. D., and C. S. Bretherton, 2018: Prognostic validation of a neural network unified physics parameterization.

- Geophys. Res. Lett.*, **45**, 6289–6298, <https://doi.org/10.1029/2018GL078510>.
- Candido, S., A. Singh, and L. Delle Monache, 2020: Improving wind forecasts in the lower stratosphere by distilling an analog ensemble into a deep neural network. *Geophys. Res. Lett.*, **47**, e2020GL089098, <https://doi.org/10.1029/2020GL089098>.
- Dalcin, L., M. Mortensen, and D. E. Keyes, 2019: Fast parallel multidimensional FFT using advanced MPI. *J. Parallel Distrib. Comput.*, **128**, 137–150, <https://doi.org/10.1016/j.jpdc.2019.02.006>.
- Dillon, J. V., and Coauthors, 2017: TensorFlow distributions. arXiv, 1711.10604v1, <https://doi.org/10.48550/ARXIV.1711.10604>.
- Fan, Y., V. Krasnopolsky, H. van den Dool, C.-Y. Wu, and J. Gottschalk, 2023: Using artificial neural networks to improve CFS week 3–4 precipitation and 2-meter air temperature forecasts. *Wea. Forecasting*, <https://doi.org/10.1175/WAF-D-20-0014.1>, in press.
- Ghamariadyn, M., and M. A. Imteaz, 2021: Prediction of seasonal rainfall with one-year lead time using climate indices: A wavelet neural network scheme. *Water Resour. Manage.*, **35**, 5347–5365, <https://doi.org/10.1007/s11269-021-03007-x>.
- Ghazvinian, M., Y. Zhang, D.-J. Seo, M. He, and N. Fernando, 2021: A novel hybrid artificial neural network-parametric scheme for postprocessing medium-range precipitation forecasts. *Adv. Water Resour.*, **151**, 103907, <https://doi.org/10.1016/j.advwatres.2021.103907>.
- Glahn, H. R., and D. A. Lowry, 1972: The use of model output statistics (MOS) in objective weather forecasting. *J. Appl. Meteor.*, **11**, 1203–1211, [https://doi.org/10.1175/1520-0450\(1972\)011<1203:TUOMOS>2.0.CO;2](https://doi.org/10.1175/1520-0450(1972)011<1203:TUOMOS>2.0.CO;2).
- Goodfellow, I., Y. Bengio, and A. Courville, 2016: *Deep Learning*. MIT Press, 296 pp.
- Haupt, S. E., and Coauthors, 2020: Combining artificial intelligence with physics-based methods for probabilistic renewable energy forecasting. *Energies*, **13**, 1979, <https://doi.org/10.3390/en13081979>.
- , W. Chapman, S. V. Adams, C. Kirkwood, J. S. Hosking, N. H. Robinson, S. Lerch, and A. C. Subramanian, 2021: Towards implementing artificial intelligence postprocessing in weather and climate: Proposed actions from the Oxford 2019 workshop. *Philos. Trans. Roy. Soc.*, **A379**, 20200091, <http://doi.org/10.1098/rsta.2020.0091>.
- Hersbach, H., 2000: Decomposition of the continuous ranked probability score for ensemble prediction systems. *Wea. Forecasting*, **15**, 559–570, [https://doi.org/10.1175/1520-0434\(2000\)015<0559:DOTCRP>2.0.CO;2](https://doi.org/10.1175/1520-0434(2000)015<0559:DOTCRP>2.0.CO;2).
- Huang, C.-J., and P.-H. Kuo, 2018: A short-term wind speed forecasting model by using artificial neural networks with stochastic optimization for renewable energy systems. *Energies*, **11**, 2777, <https://doi.org/10.3390/en11102777>.
- Kluyver, T., and Coauthors, 2016: Jupyter notebooks—A publishing format for reproducible computational workflows. *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, F. Loizides and B. Schmidt, Eds., IOS Press, 87–90.
- Murphy, A. H., 1973: A new vector partition of the probability score. *J. Appl. Meteor.*, **12**, 595–600, [https://doi.org/10.1175/1520-0450\(1973\)012<0595:ANVPOT>2.0.CO;2](https://doi.org/10.1175/1520-0450(1973)012<0595:ANVPOT>2.0.CO;2).
- Pedregosa, F., and Coauthors, 2011: Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.*, **12**, 2825–2830, <https://doi.org/10.48550/arXiv.1201.0490>.
- Peng, T., X. Zhi, Y. Ji, L. Ji, and Y. Tian, 2020: Prediction skill of extended range 2-m maximum air temperature probabilistic forecasts using machine learning postprocessing methods. *Atmosphere*, **11**, 823, <https://doi.org/10.3390/atmos11080823>.
- Peralta, C., Z. Ben Bouall'egue, S. E. Theis, C. Gebhardt, and M. Buchhold, 2012: Accounting for initial condition uncertainties in COSMO-DE-EPS. *J. Geophys. Res.*, **117**, D07108, <https://doi.org/10.1029/2011JD016581>.
- Rasp, S., and S. Lerch, 2018: Neural networks for postprocessing ensemble weather forecasts. *Mon. Wea. Rev.*, **146**, 3885–3900, <https://doi.org/10.1175/MWR-D-18-0187.1>.
- Reichstein, M., G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, and Prabhat, 2019: Deep learning and process understanding for data-driven earth system science. *Nature*, **566**, 195–204, <https://doi.org/10.1038/s41586-019-0912-1>.
- Scheuerer, M., M. B. Switanek, R. P. Worsnop, and T. M. Hamill, 2020: Using artificial neural networks for generating probabilistic subseasonal precipitation forecasts over California. *Mon. Wea. Rev.*, **148**, 3489–3506, <https://doi.org/10.1175/MWR-D-20-0096.1>.
- Schultz, M. G., C. Betancourt, B. Gong, F. Kleinert, M. Langguth, L. H. Leufen, A. Mozaffari, and S. Stadler, 2021: Can deep learning beat numerical weather prediction? *Philos. Trans. Roy. Soc.*, **A379**, 20200097, <https://doi.org/10.1098/rsta.2020.0097>.
- Sharifian, A., M. J. Ghadi, S. Ghavidel, L. Li, and J. Zhang, 2018: A new method based on type-2 fuzzy neural network for accurate wind power forecasting under uncertain data. *Renew. Energy*, **120**, 220–230, <https://doi.org/10.1016/j.renene.2017.12.023>.
- Sønderby, C. K., and Coauthors, 2020: MetNet: A neural weather model for precipitation forecasting. arXiv, 2003.12140v2, <https://doi.org/10.48550/arXiv.2003.12140>.
- Theocharides, S., G. Makrides, A. Livera, M. Theristis, P. Kaimakis, and G. E. Georgiou, 2020: Day-ahead photovoltaic power production forecasting methodology based on machine learning and statistical postprocessing. *Appl. Energy*, **268**, 115023, <https://doi.org/10.1016/j.apenergy.2020.115023>.
- Vannitsem, S., and Coauthors, 2021: Statistical postprocessing for weather forecasts: Review, challenges, and avenues in a big data world. *Bull. Amer. Meteor. Soc.*, **102**, E681–E699, <https://doi.org/10.1175/BAMS-D-19-0308.1>.
- Van Rossum, G., and F. L. Drake Jr., 1995: *Python Reference Manual*. Centrum voor Wiskunde en Informatica, 283–303.
- Veldkamp, S., K. Whan, S. Dirksen, and M. Schmeits, 2021: Statistical postprocessing of wind speed forecasts using convolutional neural networks. *Mon. Wea. Rev.*, **149**, 1141–1152, <https://doi.org/10.1175/MWR-D-20-0219.1>.
- Virtanen, P., and Coauthors, 2020: SciPy1.0: Fundamental algorithms for scientific computing in Python. *Nat. Methods*, **17**, 261–272, <https://doi.org/10.1038/s41592-019-0686-2>.
- Ward, M. N., and C. K. Folland, 1991: Prediction of seasonal rainfall in the north nordeste of Brazil using eigenvectors of sea-surface temperature. *Int. J. Climatol.*, **11**, 711–743, <https://doi.org/10.1002/joc.3370110703>.
- Zamo, M., and P. Naveau, 2018: Estimation of the continuous ranked probability score with limited information and applications to ensemble weather forecasts. *Math. Geosci.*, **50**, 209–234, <https://doi.org/10.1007/s11004-017-9709-7>.
- Zhang, Y., and A. Ye, 2021: Machine learning for precipitation forecasts postprocessing: Multimodel comparison and experimental investigation. *J. Hydrometeorol.*, **22**, 3065–3085, <https://doi.org/10.1175/JHM-D-21-0096.1>.