Deltares

Rainfall downscaling and adjustment methods

Rainfall downscaling and adjustment methods

Author(s)
Ruben Imhoff
Maarten Verbrugge
Henrique Moreno Dumont Goulart

Rainfall downscaling and adjustment methods

Client	Murray-Darling Basin Authority (MDBA)
Contact	Lawrence Lingam
Reference	
Keywords	

Document control	
Version	1.0
Date	03-07-2025
Project nr.	I1000714-048
Document ID	I1000714-048-OA-0001
Pages	42
Classification	
Status	Final

Author(s)		
	Ruben Imhoff Maarten Verbrugge Henrique Moreno Dumont Goulart	

Summary

A crucial component to hydrological forecasting is the presence of high-quality, high-resolution quantitative precipitation estimates (QPE) and quantitative precipitation forecasts (QPF), especially for flooding situations. Current gridded precipitation products from reanalyses, forecasting products and remotely sensed products are a powerful source of information as they provide precipitation quantities in both space and time. Yet, these products often suffer from systematic biases and spatial inaccuracies due to their sometimes large grid cell sizes, which makes it challenging to capture local extremes. While conventional downscaling and adjustment methods, ranging from simple to advanced gauge-based merging in hindcasting mode and statistical methods in forecasting mode, do exist, there is no tool yet that combines all these methods for usage in operational applications.

In this report, we present the open-source pyRainAdjustment Python tool, which integrates various methods for adjustment and downscaling of gridded precipitation products into a single tool compatible with Delft-FEWS. pyRainAdjustment splits the approach in three parts: (1) downscaling, for which it uses a climatology-based downscaling approach that uses a high-resolution base dataset (e.g. WorldClim2) on a monthly basis; (2) adjustment in a hindcasting mode, for which five approaches were implemented (uniform mean field bias adjustment, multiplicative error adjustment, additive error adjustment, mixed error model adjustment and kriging with external drift); and (3) adjustment in a forecasting mode, which consists of the statistical quantile mapping method.

The introduced approaches in pyRainAdjustment were tested for the upper Murray River for MDBA. In hindcasting mode, the results indicate that downscaling and spatial adjustment methods in pyRainAdjustment improve the QPE quality. Although uniform adjustments provide only minimal improvement (8% for ERA5 and slightly more for radar data), spatial techniques such as mixed error modelling and kriging with external drift show better results (up to a maximum of 27% improvement), with mixed error adjustment slightly leading in performance. In forecasting mode, quantile mapping reduces the mean RMSE of ECMWF's IFS forecasts by approximately 10%. The method effectively reduces the frequency of large forecast errors, particularly for moderate to high precipitation events (8–20 mm), while maintaining similar performance in low-error scenarios. Hence, this indicates that the methods in pyRainAdjustment can aid in reducing the errors in both QPE and QPF products.

In addition to the (existing) methods that were brought together in pyRainAdjustment, we introduce the first results of an exploration of machine learning (ML) approaches for gridded precipitation downscaling and adjustment in this report. An advantage of ML methods for this purpose is that they can simultaneously downscale and adjust precipitation data without relying on real-time (gauge) observations. This report describes the current testing setup and considered model architectures and presents the first results using the U-Net architecture on ERA5 reanalysis data with MSWEP QPE as target and reference. The U-Nets show potential in hindcasting mode, with improvements over the original ERA5 QPE in several metrics (with improvement around 10%, but depending on the consulted metric). The model still tends to overpredict moderate precipitation and underpredict extremes, but largely reduces the overall bias of the QPE. These initial results suggest that while U-Net shows potential, further refinement is needed to improve its performance across all precipitation ranges. The research will continue past this project with improving the U-Nets, trying out the other architectures and then moving to the forecasting domain.

Contents

	Summary	**4**
1	**Introduction**	**6**
1.1	Opportunities using machine-learning techniques for adjustment and downscaling.	7
1.2	Structure of this report	8
2	**Rainfall downscaling and adjustment methods**	**9**
2.1	Downscaling using global monthly climatology products	9
2.2	Adjustment methods when rain gauge observations are available	10
2.2.1	Mean field bias adjustment	11
2.2.2	Spatial adjustment methods	12
2.2.3	Kriging with external drift adjustment	13
2.3	Adjustment methods in a forecasting mode	14
2.3.1	Quantile mapping	14
2.4	Data-driven adjustment and downscaling methods	15
2.4.1	U-Nets	15
2.4.2	GAN models	16
2.4.3	Latent diffusion models	16
2.5	Introduction to the pyRainAdjustment package	17
2.5.1	Package setup and contribution	17
2.5.2	Integration and usage in Delft-FEWS	19
2.5.3	Configuration options	24
3	**Case study setup for the MDBA domain**	**26**
3.1	Available observation data and forecasts	26
3.2	Hindcasting test setup	27
3.3	Forecasting test setup	28
3.4	Data-driven model setup and benchmarking	28
3.4.1	Features considered	28
3.4.2	General functioning of machine learning training	28
3.4.3	Preprocessing	29
3.4.4	Performance analysis	29
4	**Case study results**	**30**
4.1	Hindcasting	30
4.2	Forecasting	32
4.3	Data-driven downscaling and corrections	34
5	**Conclusions and recommendations**	**36**
5.1	Recommendations for application and future research	37
	References	**39**

1 Introduction

Timely and spatially accurate quantitative precipitation estimates (QPE) and forecasts (QPF) are generally the most critical inputs into hydrological forecasting models. Although rain gauge observations are the only direct observation (and thus not estimation) of rainfall, they have the disadvantage that they only measure at a point location and cannot provide any forecasting information. This results in a poor spatial representation, especially when the gauge network density is low. Remotely sensed QPE from weather radars or satellites usually offer complete spatial coverage, but can suffer from substantial biases or other errors with respect to gauged observations. This does not only hold for remotely sensed products, as spatial accuracy can be a challenge for many (global) reanalysis and forecasting models. This is due to (1) their large spatial resolution (e.g., ~28 km for the GFS forecasts, ~9 km for the ECMWF IFS forecasts and ~30 km for the ERA5 reanalysis products) and (2) the significant biases these products can have, especially for high (convective) rainfall amounts.

Especially on the smaller spatial scales, where quicker hydrological responses take place and the effect of local intense rainfall events is more pronounced in river discharge and reservoir levels, more precision through higher resolution and a higher accuracy through lower biases of the QPE and QPF products is warranted (Berne et al., 2004). To achieve this with the currently available (global) QPE and QPF products, it is recommended to downscale and adjust the rainfall products. The challenge in doing so, is that there is currently no one-size-fits-all solution, which requires different downscaling and adjustment approaches for each type of QPE and QPF product.

Typically, the approaches for downscaling and adjustment of gridded QPE and QPF products are split into three different domains: (1) downscaling methods, (2) precipitation adjustment methods when ground observations are available and (3) precipitation adjustment in forecasting mode.

Regarding the first domain, downscaling, there are mainly two approaches for those: dynamic downscaling, which is physics-informed but comes at a high computational cost, and statistical downscaling, which uses either mathematical or climatology-based relations between the data to be downscaled and another higher resolution target data.

Regarding the second domain, precipitation adjustment of QPE products when ground observations are available, this is generally the most straightforward approach of the three domains. Since gridded QPE products only provide information about either the past or the current time, rain gauge observations can be used to adjust these products. Combining one or multiple gridded rainfall product(s) with rain gauges generally improves the overall QPE product (e.g., Ochoa-Rodriguez et al., 2019). There is a wide extent of literature on different so-called QPE merging techniques, ranging from simple, uniform adjustment factors for the entire grid (Smith and Krajewski, 1991; Seo et al., 1999), to more advanced spatial mixed error models (e.g., Bronstert et al., 2010), and eventually to full geostatistical merging methods such as kriging with external drift or Bayesian approaches (e.g., Todini, 2001; Goudenhoofdt and Delobbe, 2009). Although the more advanced methods provide better adjusted QPE products, it is worth noting that they require a high density of rain gauges, which is generally challenging if not impossible in real time.

Finally, regarding the third domain, which focuses on QPF adjustment, the true challenge originates from the absence of observations to correct with in forecasting mode. Typically, statistical approaches (often referred to as model output statistics, or MOS in short) are used to adjust the forecast either through direct adjustment or calibration of the forecast output. A popular method in the forecasting domain is quantile mapping, which tries to equate the cumulative distribution function (CDF) of the forecast to that of the observations, by determining the CDFs on a large sample of past data and assuming that the mapping still holds for future forecasts (e.g., Canon et al., 2015). This method is particularly strong for forecasts that have a systematic bias in any part of the CDF, such as for low or high precipitation amounts.

The aforementioned methods within the three main domains for gridded QPE and QPF downscaling and adjustment are all available through literature or existing code, but one tool that can perform downscaling and adjustment on both gridded QPE and QPF products is still lacking. This makes it challenging to get started with these downscaling and adjustment tools in an operational platform when many different products are imported and used. Hence, we see a need to bring all aforementioned options together in one package that can be integrated within Delft-FEWS. In this report, we present pyRainAdjustment, a Python package for downscaling and adjustment of gridded QPE and QPF products that can be fully integrated with Delft-FEWS.

In the two sections below, we describe another running research line on machine learning for downscaling and adjustment, and the overall structure of this report in which pyRainAdjustment will be further introduced and tested.

1.1 Opportunities using machine-learning techniques for adjustment and downscaling.

An issue with the so-called conventional adjustment and downscaling techniques is that their requirements can be limiting in what is feasible for a specific area. For instance, merging approaches between gauges and gridded QPE products are only possible in real time when sufficient automatic rain gauges are present and provide quality-controlled observations, and quantile mapping for gridded QPF adjustment is only successful when a sufficiently long historical dataset is available, including rain gauges. Besides, downscaling and adjustment cannot take place at the same time.

The rapidly developing field of machine learning (ML) for downscaling and correction of meteorological variables can provide an alternative here. The concept behind ML-based downscaling and adjustment it is to train an ML model to learn the relations between a coarse resolution dataset (QPE or QPF) and a high-resolution gridded reference dataset, which can consists of interpolated gauge observations, a quality-controlled local QPE product, a higher-resolution reanalysis or radar dataset, etc. Once the model is correctly trained, it is capable to downscale meteorological information to finer resolutions and also adjust it for biases on the fly. A major advantage of such an approach is twofold: (1) we can now perform both downscaling and adjustment at once, and (2) there is no need for base data, such as rain gauges, in real time anymore, which greatly reduces the dependency on real-time data availability.

Especially for global climate models (GCMs), multiple studies have already successfully applied such an ML approach on various meteorological variables (e.g., Stengl et al., 2020; Tan et al., 2021; Wang and Tian, 2022; Kheir et al., 2023; Lin et al., 2023). Most commonly, ERA5 data is used as reference for training of the ML models, which provides a great reference for larger-scale applications, but is not sufficient for smaller-scale applications, including short to medium-range forecasting. In the field of short to medium-range rainfall forecasting where even higher resolution is warranted, ML methods are now slowly being picked up. As the QPF resolutions are already higher in both space and time than for GCMs, this asks for different approaches and reference datasets. Typically, high-resolution corrected radar or satellite reanalysis datasets (e.g., MSWEP or CHIRPS) are used. Recent studies have shown some potential using U-Nets for wind fields (Höhlein et al., 2020), (VAE-)GANs for QPF (Leinonen et al., 2020; Harris et al., 2022) and latent-diffusion models for wind and temperature fields (Tomasi et al., 2025). These approaches show potential, but it is yet to be explored if they work on rainfall and in particular on both gridded QPE and QPF datasets.

Therefore, besides the introduction of pyRainAdjustment, we have explored the usage of ML models for rainfall QPE and QPF adjustment and downscaling. This report will briefly describe the approaches that we have tested and the pilot that is in progress. This study is part of ongoing research, meaning that this document solely reports on the current progress. The final results will be communicated at a later stage through either a report or scientific publication. Having pyRainAdjustment available (the main goal of this report), makes it possible to have an excellent benchmark for the ML approaches. This makes it possible to go a step further than current literature by also testing how the ML adjustment and downscaling approaches perform compared to so-called conventional methods.

1.2 Structure of this report

This report is structures as follows: chapter 2 discusses the various rainfall downscaling and adjustment methods that have been tested and applied in this study, including the ML exploration (section 2.4) and the setup of pyRainAdjustment and its configuration in Delft-FEWS (section 2.5); chapter 3 describes the case study setup to test pyRainAdjustment and some of the ML tools for the MDBA domain; the results of this setup are presented in chapter 4; followed by some overall conclusions and recommendations in chapter 5.

As mentioned in section 1.1, the machine learning approaches for rainfall downscaling and correction are not (yet) part of the pyRainAdjustment Python package, as this is part of ongoing research. This report merely describes the current status and results of the research for the MDBA domain. The final results will be shared through a different report or scientific publication.

2 Rainfall downscaling and adjustment methods

This chapter describes the different methods that have been tested and implemented in pyRainAdjustment for Delft-FEWS. Starting with section 2.1, we describe the basic climatology-based downscaling functionality that is part of pyRainAdjustment. Section 2.2 describes the adjustment methods that are implemented in a hindcasting setting when rain gauges are available for merging with gridded rainfall products. Section 2.3 describes the quantile mapping adjustment method that has been implemented in a forecasting mode when no rain gauge information is available. Then, section 2.4 describes an alternative approach to 'conventional' downscaling and adjustment methods, by using data-driven (machine-learning) methods for both downscaling and adjustment in hindcasting and forecasting mode. Finally, in section 2.5 we describe the implementation of these methods in the pyRainAdjustment package and the integration of this package and its usage in Delft-FEWS.

2.1 Downscaling using global monthly climatology products

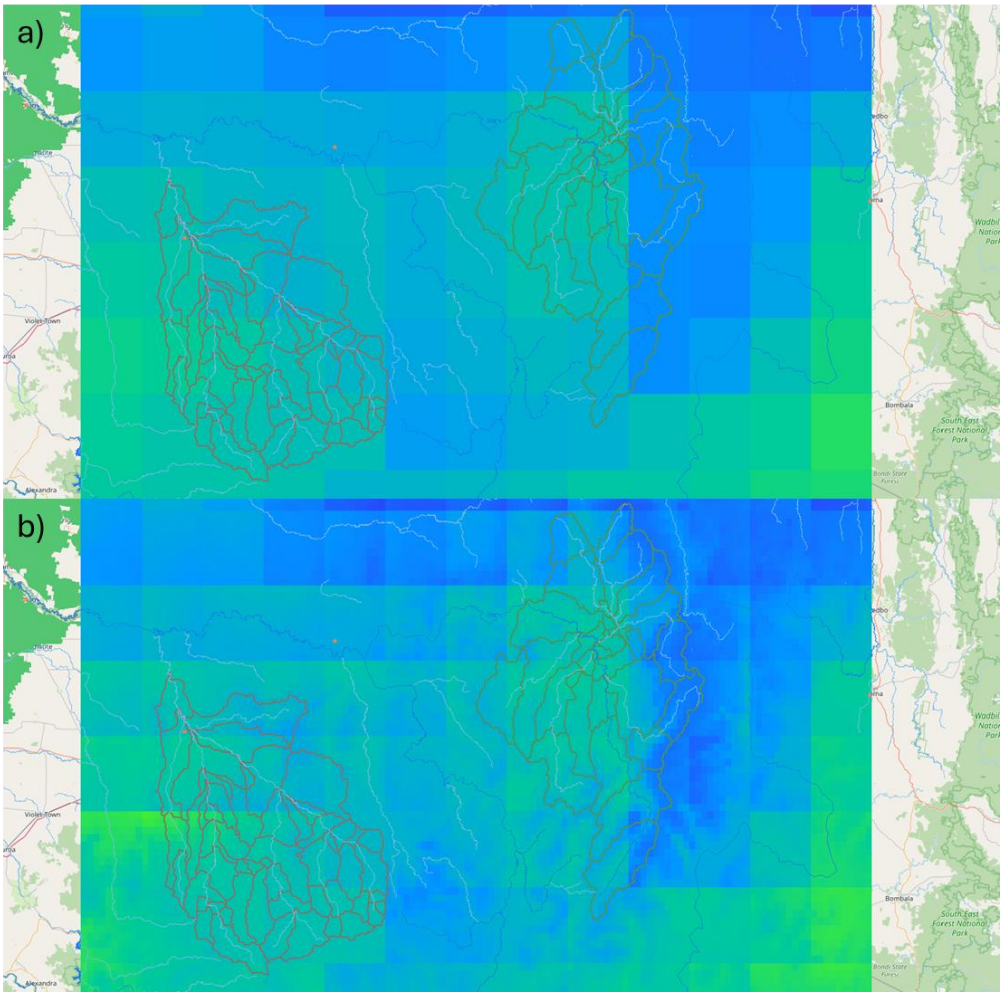

Figure 2-1 Example of climatology-based downscaling of daily ERA5 reanalysis data from (a) the original spatial resolution (approximately 30 km) to (b) a resolution of 3 km on 2024-11-30.

Most precipitation downscaling methods are either climatology based or statistics based, with the field of artificial intelligence slowly providing an alternative to these conventional approaches, as described in section 2.1. In pyRainAdjustment, we implemented a relatively simple but robust climatology-based downscaling method.

The method uses monthly climatology data to downscale the original precipitation grid to a predefined higher-resolution grid. It works in a couple of steps:

1 The high-resolution climatology data is reprojected with an averaging resampling scheme to the coarser spatial grid the original precipitation data is on. This way, the high-resolution data gets averaged out over the coarser grid cells.
2 The climatology data on the coarser grid is then downscaled with a nearest neighbour reprojection to the finer desired output grid, which preserves the data on the coarse grid, but presents it on the finer output grid. This product is regarded as the 'coarse grid'.
3 The high-resolution climatology data is subsequently also directly reprojected to the output grid with an averaging resampling scheme in case the output grid is coarser than the original grid resolution of the climatology data. This is regarded as the 'fine grid' and preserves the high-resolution climatology data on the output data as well as possible.
4 On the output grid, a downscaling factor is calculated per grid cell, which is calculated as:
5 $f_{downscaling} = \frac{P_{fine}}{P_{coarse}}$,
6 with P_{fine} the climatology data on the 'fine grid' and P_{coarse} the climatology data on the 'coarse grid'.
7 Finally, the original gridded precipitation data is downscaled to the finer resolution by first reprojecting it with a nearest neighbour transformation to the finer grid and subsequently multiplying it with $f_{downscaling}$ to correct it with the climatology-based downscaling factor.

As a standard climatology dataset, we recommend using the WorldClim2 dataset by Fick & Hijmans (2017), which is a monthly dataset containing global precipitation climatology data on a 1-km resolution. Figure 2-1 illustrate the use of the introduced downscaling method with the WorldClim2 dataset by downscaling daily ERA5 reanalysis data from a 30-km resolution to a 3-km resolution. The result clearly provides more resolution and starts to show the effect of the topography on the precipitation distribution, which is also present in the WorldClim2 dataset. However, the individual original ERA5 grid cells remain somewhat visible, which is something that a relatively simple downscaling method cannot overcome.

2.2 Adjustment methods when rain gauge observations are available

Reaching better resolution through downscaling is not the only step to improve gridded precipitation products for hydrological purposes. As most gridded precipitation products are heavily biased, generally through a combination of systematic and random errors, bias adjustment procedures are a crucial additional step to improve precipitation products for hydrological purposes. When rain gauge observations are available, for instance in hindcasting or state-updating mode in an operational system, bias adjustment is a little more straightforward as gridded precipitation products can be adjusted using the gauge observations. As part of this project, we have implemented three different bias adjustment methods. The methods vary from simple, uniform corrections (mean field bias, see section 2.2.1) to more advanced geospatial adjustment methods that require a much higher rain gauge density in real time (see sections 2.2.2 and 2.2.3).

2.2.1 Mean field bias adjustment

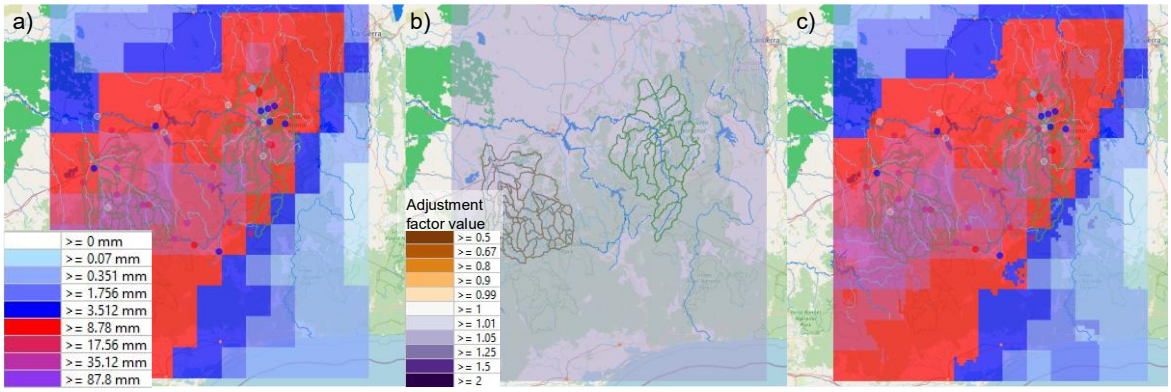

Figure 2-2 Example of mean field bias adjustment of the ERA5 reanlaysis data for 2024-11-28. a) the original QPE from ERA5, b) the adjustment factor and c) the adjusted ERA5 QPE.

The mean field bias (MFB) adjustment method provides a spatially uniform multiplicative adjustment factor that can be applied to adjust the gridded rainfall field (Figure 2-2). It uses all available rain gauges and corresponding grid cells to calculate an adjustment factor for the entire grid as follows:

$$F_{MFB} = \frac{\sum_{n=1}^{N} G(i_n, j_n)}{\sum_{n=1}^{N} P_{orig}(i_n, j_n)}.$$

Here, $G(i_n, j_n)$ the rain gauge sum for gauge *n* at location (i_n, j_n) for the used time interval and $P_{orig}(i_n, j_n)$ the precipitation value at the corresponding grid cell of the original, unadjusted gridded precipitation product. Once calculated, F_{MFB} is multiplied with P_{orig} to get the adjusted output precipitation grid:

$$P_{adj}(m, n) = F_{MFB}(m, n) * P_{orig}(m, n).$$

To ensure F_{MFB} does not blow up when $\sum_{n=1}^{N} P_{orig}(i_n, j_n)$ is either very small or zero, we typically work with a threshold for the entire sum of both the gauges and corresponding grid cells or a threshold per gauge – grid cell couple (see also section 2.5). The advantage of the MFB method is that the uniform adjustment makes the method independent of the number of available rain gauges, offering a robust method in operations. However, the spatially uniform application makes it insensitive to spatial variations in the gridded precipitation product quality, for instance due to the large extent of the domain, specific local weather phenomena or local topography.

2.2.2 Spatial adjustment methods

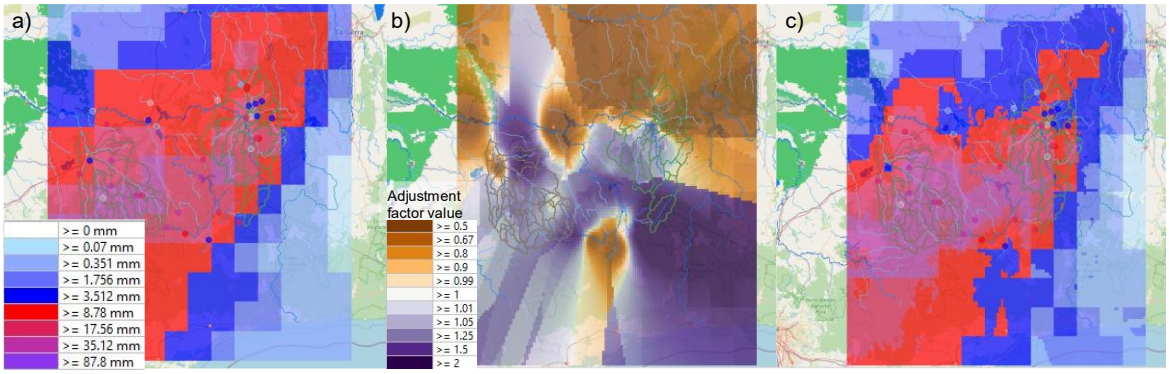

Figure 2-3 Example of a spatial mixed error model to adjust the ERA5 reanalysis data for 2024-11-28. a) the original QPE from ERA5, b) the adjustment factor, interpolated onto the original grid with an inverse-distance weighting approach. and c) the adjusted ERA5 QPE.

To not only adjust in time but also in space, we implemented spatial adjustment methods in pyRainAdjustment as well. With these methods, an adjustment factor per rain – grid cell pair is calculated and subsequently all point based adjustment factors are interpolated back onto the original grid of the gridded precipitation product. To make this interpolation possible, we implemented four different interpolation schemes in pyRainAdjustment: nearest neighbour, linear, inverse-distance weighting and ordinary kriging. The interpolated gridded adjustment factor is then used to adjust the gridded precipitation field in space for that time step (Figure 2-3).

The adjustment method can be calculated through a multiplicative, additive or mixed approach. The multiplicative approach resembles the MFB adjustment, but works per gauge - grid cell pair instead of over the sum of all of them simultaneously:

$$F(i,j) = \frac{G\ (i,j)}{P_{orig}(i,j)}.$$

Subsequently, all local values of $F(i,j)$ are interpolated back onto the original grid and the final adjusted precipitation grid is then calculated as:

$$P_{adj}(m,n) = F(m,n) * P_{orig}(m,n),$$

with (m,n) the grid cell for row *m* and column *n*.

The additive approach calculates an error instead of an adjustment factor between the gauge – grid cell pair:

$$E(i,j) = G(i,j) - P_{orig}(i,j).$$

After interpolating all $E(i,j)$ onto the original grid, the final adjusted precipitation grid is calculated as:

$$P_{adj}(m,n) = \begin{cases} 0.0, & P_{orig}(m,n) + E(m,n) < 0 \\ P_{orig}(m,n) + E(m,n), & P_{orig}(m,n) + E(m,n) \geq 0 \end{cases}.$$

Finally, the mixed error model assumes that both a multiplicative and additive error can be present and tries to correct for that in the following way (Bronstert et al., 2010; Figure 2-3):

$$G(i,j) = P_{orig}(i,j) * (1 + \delta) + \varepsilon,$$

with δ the multiplicative error and ϵ the additive error. To determine both δ and ε, a least-squares estimation is made in which we assume δ and ε to be both independent and normally distributed. With this assumption, ε and δ are calculated as (Bronstert et al., 2010):

$$\varepsilon = \frac{G(i,j) - P_{orig}(i,j)}{P_{orig}(i,j)^2 + 1},$$
$$\delta = \frac{G(i,j,) - \varepsilon}{P_{orig}(i,j)} - 1.$$

Both δ and ε are then interpolated onto the original grid and subsequently, $P_{adj}(m,n)$ is calculated with

$$P_{orig}(m,n) * \big(1 + \delta(m,n)\big) + \varepsilon(m,n).$$

2.2.3 Kriging with external drift adjustment

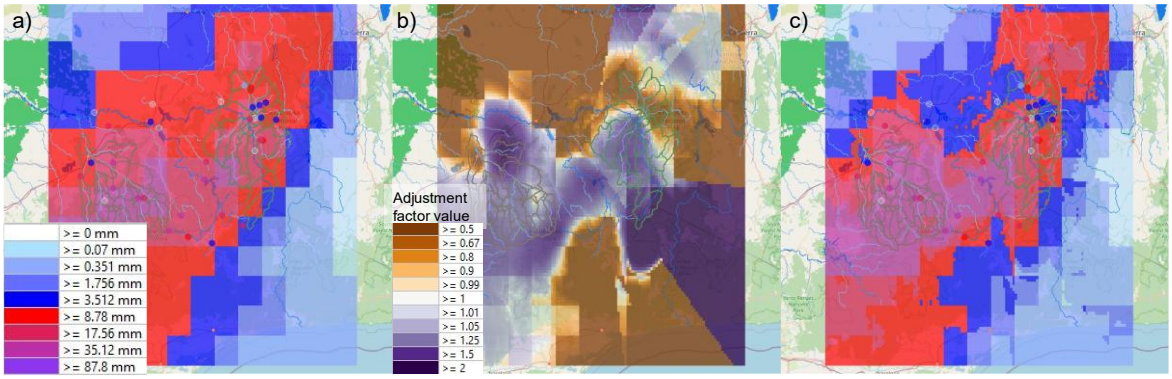

Figure 2-4 Example of kriging with external drift adjustment of the ERA5 reanlaysis data for 2024-11-28. a) the original QPE from ERA5, b) the adjustment factor and c) the adjusted ERA5 QPE.

The most advanced geospatial adjustment technique that we implemented in pyRainAdjustment is kriging with external drift. Ordinary kriging is an interpolation technique in which the prediction at grid cell *i* is based on the weighted sum of all surrounding rain gauge values, as:

$$P(m,n) = \sum_{n=1}^{N} \lambda_n \, G(i_n, j_n),$$

with λ_n the weight for the n^{th} gauge location. In inverse-distance weighting (IDW), this weight is dependent on the distance of the gauge to the grid cell, but in ordinary kriging, this weight is determined with a covariance model in combination with the location of the target grid cell. To estimate the covariance model, a semi-variogram model is used to determine the relationship between the semi variance of the data and the distance between the data points. In pyRainAdjustment, the method can either automatically derive this semi-variogram model, assuming a spherical model, or a standard exponential model is used ($1.0\ e^{10,000} + 0.0$), similar to what is applied in the widely-used wradlib Python package (Heistermann et al, 2013). For more information on kriging, we refer to Wackernagel (2003).

In kriging with external drift (KED) a spatially correlated residual (the external drift term) is added to the ordinary kriging equation, which ensures that the estimated precipitation grid follows the structure of the external drift when interpolating the gauge values to the grid using the kriging approach. The advantage of this approach over the aforementioned adjustment methods is that KED can be seen as a more hybrid approach in which the gauge values are the starting point and the gridded rainfall is assumed to capture the spatial patterns better, which is then used in the interpolation (kriging) step. This results in an end product that can contain values different from those found both in the gauge and gridded precipitation data, with patterns that still mimic the gridded precipitation data but adjusted for the observed values measured by the gauges (Figure 2-4). This method is generally considered one of the best adjustment methods for gridded rainfall product (Goudenhoofdt and Delobbe, 2009), but has as disadvantage that its calculation time is longer and that it needs a high density of rain gauges that are available in real time, which generally limits the operational applicability of this method.

2.3 Adjustment methods in a forecasting mode

In forecasting mode, rain gauges are not available to correct the gridded rainfall product(s), as future forecast rainfall is not observed yet. This means that the adjustment methods in section 2.3 are not possible and that we thus should use a different approach to correct our precipitation forecasts. Quantile mapping, also referred to as quantile regression, is currently one of the most widely used and advanced precipitation forecast correction methods (e.g. Gudmundsson et al. 2012).

2.3.1 Quantile mapping

Quantile mapping is a correction method that considers the variability of bias throughout the distribution of the target variable. Rather than only correcting the mean values of the data, quantile mapping can assess the degree to which lower, higher or extreme values need to be adjusted to match the target distribution of a reference dataset.

There are a variety of quantile mapping methods available (Cannon et al., 2015), each with differing degrees of complexity. There are parametric approaches, that fit distributions on the original and reference datasets. Benefits are that the method can extend into the parts of the distribution that the sample would not cover itself and it can account for trends within the sample, for example the climate signal. However, that also poses challenges with extrapolation and selection the right parametric model to fit the distributions.

Instead, a relatively simple and robust method is to apply quantile mapping using the empirical cumulative density function, which in essence finds the values of both forecast and reference data at a regular quantile interval (each percentile, for example) throughout a reference period. Afterwards, it computes adjustment factors for each quantile of the distribution (Figure 2-5). A forecast that falls in between percentiles will find the required correction by linearly interpolating between the adjustment factors. This poses a challenge for the extremes, where a forecast value that was larger than the maximum in the reference period will be extrapolated, which is not always accurate. At the same time, this method assumes stationarity of the distribution in the future, which seldomly holds in environmental forecasting due to the effects of climate change. Even though these concerns exist, the simplicity of the methodology and its robustness are the main reasons for its widespread adoption. These reasons are also our main rationale to implement the empirical cumulative density function-based quantile mapping in pyRainAdjustment.

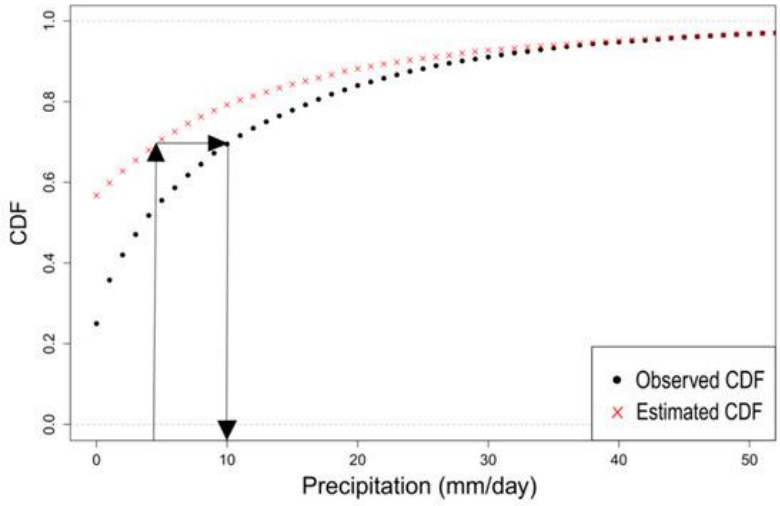

Figure 2-5 Example of empirical quantile mapping (Ringard et al., 2017).

2.4 Data-driven adjustment and downscaling methods

The methods listed in sections 2.2 and 2.3 can be considered ‘conventional’ adjustment methods. Although these are proven methods, they have a couple of drawbacks:

- In real time, the gauge density is generally lower due to the latency in rain gauge reporting (e.g., once a day), outages, maintenance or inadequate quality control. This means that the effectiveness of the adjustment methods is lower in real time than for hindcasting studies.
- In forecasting mode, the quantile mapping requires significant storage capabilities when high resolution and large grids are used in combination with many intervals.
- Downscaling and correction are always two separate steps, although they are dependent on each other.

Recent machine-learning applications show that downscaling and adjustment are also possible through these data-driven approaches. Using a machine-learning algorithm to perform these steps has two advantages: (1) the downscaling and adjustment can be applied simultaneously and (2) although a large (reference) dataset are needed for model training, there is no need for a reference or gauges in real time, lowering the dependency on gauge availability in real time. In the sub sections below, we describe the machine-learning techniques that were considered and explored for the MDBA domain.

2.4.1 U-Nets

The most common type of machine learning architecture used for downscaling of meteorological data is the U-nets. A U-net is a type of convolutional neural network (CNN) characterized by a U-shaped architecture, which consists of a contracting path (encoder) and an expansive path (decoder) (Ronneberger et al., 2015; Figure 2-6). The encoder processes the input image through layers that reduce its size while extracting important features, capturing the overall context (low scale). The decoder reconstructs the image, but incorporating detailed information from various stages of the encoder (high scale). This integration, often called "skip connections," allows the network to combine the broad understanding of the image with fine-grained spatial details, leading to accurate grid-level segmentation. With that, the U-net is capable of classifying each grid cell to identify distinct objects or regions. While originally designed for medical imaging, it has been widely used in other fields, including meteorological studies (Sharifi et al., 2019, Singh et al., 2021, Hess and Boers, 2022, Le et al., 2023, Ascenso et al., 2024).

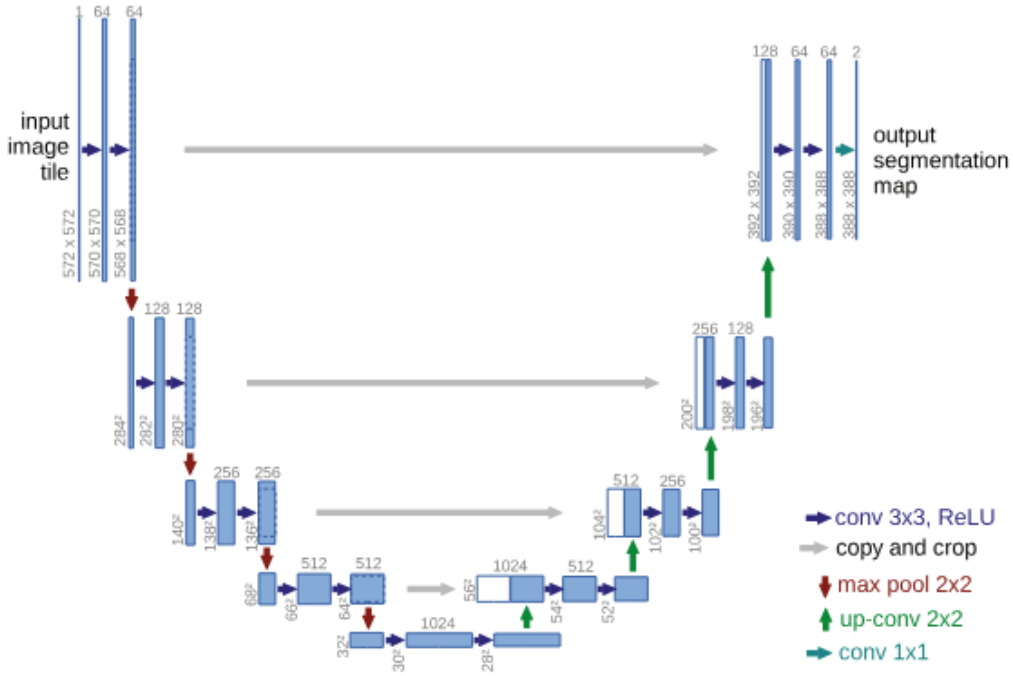

Figure 2-6 U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations. (Figure and legend from Ronneberger et al., 2015).

2.4.2 GAN models

Generative Adversarial Networks (GANs) are used to generate new data, such as images or text, using two competing neural networks: a generator and a discriminator. The generator's role is to produce synthetic data that looks authentic, while the discriminator's role is to distinguish between real data and the data created by the generator. As they train, the generator continuously improves at making believable fakes, and the discriminator becomes more skilled at spotting them. This "adversarial" competition drives both networks to improve, ultimately allowing the GAN to generate high quality data. The model is used in multiple domains, and applications in meteorological downscaling include Stengel et al. (2020) and Harris et al. (2022).

2.4.3 Latent diffusion models

The most recent type of model used for downscaling meteorological data is the latent diffusion model (LDM). LDMs reverses a process of gradual noising of data, gradually going from pure random noise to a clear image. To do so, LDMs first encode the input data into a lower-dimensional latent representation using a perceptual compression module. Then, the core diffusion process, which is iteratively denoising a random latent vector, occurs within this latent space (the compressed representation of the data). Finally, a decoder reconstructs the high-resolution output from the refined latent representation. Despite being more recently used in the meteorological domain, this approach has shown promising results Leinonen et al. (2023), Li et al. (2024), Tomasi et al., 2025.

2.5 Introduction to the pyRainAdjustment package

We have integrated the aforementioned downscaling and adjustment methods in the pyRainAdjustment package, which is described below.

2.5.1 Package setup and contribution

pyRainAdjustment is an open-source Python package that uses the wradlib Python package (Heistermann et al., 2013) as backbone (Figure 2-7). The tool uses a gridded precipitation dataset (hindcast or forecast) in netCDF format as input together with the rain gauge data (netCDF format containing all available gauges; only in hindcasting mode) and a configuration file, which can easily be exported and called through Delft-FEWS. The spatial grid projection (e.g., WGS84 or Lambert, etc.) does not matter, as long as the projection, latitude and longitude values are defined in the exported netCDF (this automatically happens through Delft-FEWS).

Based on the requested configuration, either downscaling, hindcasting corrections (MFB, additive, multiplicative, mixed or KED adjustments) or forecasting (quantile mapping) corrections are applied. In the case of quantile mapping, the statistics can (1) either be derived from a historical dataset or (2) applied when already derived and when a forecast is available in real time. All options and configurations of pyRainAdjustment are described in section 2.5.2, but below we highlight a few additional features that pyRainAdjustment offers. When this information is provided, pyRainAdjustment will apply the downscaling or adjustment and returns the downscaled/adjusted gridded precipitation together with the adjustment factor when adjustment has taken place.

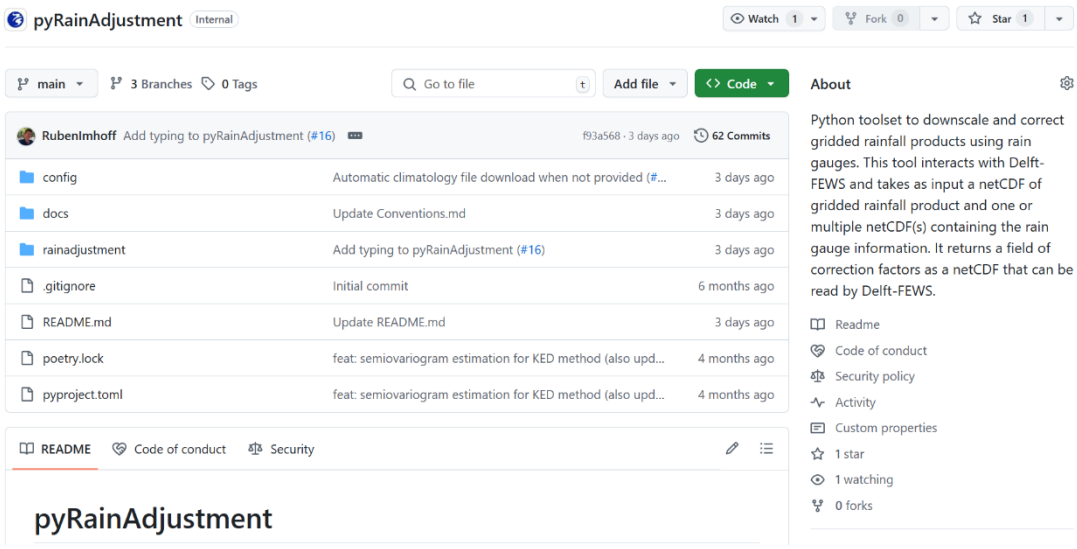

Figure 2-7 Screenshot of the pyRainAdjustment github page.

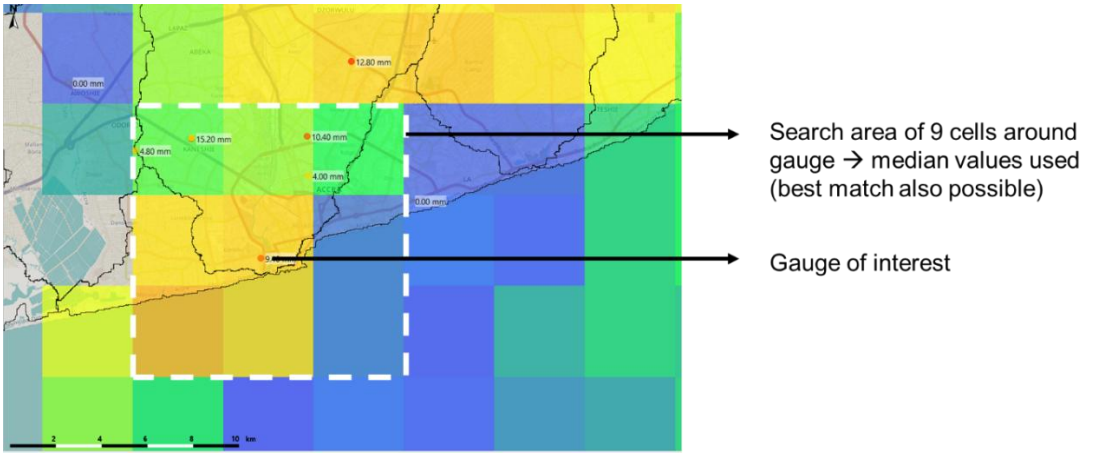

Figure 2-8 Example of the search function to find the corresponding grid cell(s) for a provided gauge location in pyRainAdjustment. This example, tested over Accra, Ghana, uses the median value of the nine grid cells around the gauge of interest.

In hindcasting mode, or when deriving the quantiles for the quantile mapping application, grid cells have to be matched with the provided rain gauges. pyRaindAdjustment uses one of wradlib's tools for that, which searches for the closest grid cell given a rain gauge location, based on the distance between the gauge and the centre of the grid cell (given the latitude and longitude coordinates). Provided that many gridded rainfall products are estimates based on remotely sensed observations (e.g., from radar or satellite), we generally get a gridded rainfall estimate at a specific altitude above ground (typically 1500 m for radar), meaning that what we see on the gridded QPE, is not necessarily what falls on the ground, besides the error that comes with remotely sensed estimates from rainfall anyway. Hence, it is often better to search the corresponding grid cell(s) with a specific radius around the rain gauge. This is possible with a 'search box' in which the closest *n* cells are searched for around the rain gauge. When this box of cells is found, the user can still define whether to use the best matching cell (with the gauge) for the adjustment procedure, or the median, mean, minimum or maximum of all cells (Figure 2-8).

To ensure the adjustment procedure in pyRainAdjustment is stable, a few additional settings can be provided to the tool:

- The minimum number of gauges that should be present in order to perform an adjustment. This ensures that a large grid is never corrected based on just a few gauges (if that functionality is warranted).
- A threshold (in the unit of the data, e.g. mm or mm h^{-1}) can be provided, which assures that adjustment only takes place when both the gauge and corresponding grid cell are above the threshold. This prevents from dividing by zero in multiplicative approaches or from correction towards zero.
- A maximum adjustment factor can be provided, ensuring that the adjustment cannot 'blow up' resulting in unrealistic large or small values.

In downscaling mode, a climatology product can be provided. When no climatology product is available, pyRainAdjustment will automatically download and prepare the WorldClim2 monthly precipitation climatology on 1-km resolution (Fick and Hijmans, 2017). This climatology is placed in the "clim" sub folder of the pyRainAdjustment module in Delft-FEWS, so in principle this step only has to take place once (it takes approximately 10 minutes).

As open-source Python package, pyRainAdjustment is updated and further developed regularly. Contributions to the package, through testing functionalities, adding a new functionality or updating examples and documentation are always appreciated. This can be done directly through the github page and a contributing guide explaining how to do so is present in the ReadMe of the pyRainAdjustment github page.

2.5.2 Integration and usage in Delft-FEWS

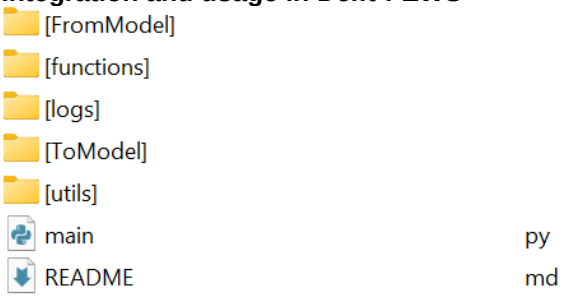

Figure 2-9 Contents of the pyRainAdjustment package.

To connect pyRainAdjustment to Delft-FEWS, it should be zipped and placed as a ModuleDataSetFile in the FEWS configuration. The package consists of one main script, which calls all other functions and tooling placed in the sub folders indicated in Figure 2-9. pyRainAdjustment is then called from Delft-FEWS through a GeneralAdapter run by calling "python main.py --xml_config ToModel/[xml_config_name].xml --requested functionality [functionality]". The requested functionality is either "downscaling", "adjustment" or "qq_mapping", and the xml_config is exported by the GeneralAdapter run to the ./ToModel folder of pyRainAdjustment based on the properties (settings) provided to the run.

IdMapFiles

PyRainAdjustment assumes that the provided parameter is called "P", but can generally also cope with other provided parameter names, as long as the provided parameter is the only parameter in the exported dataset. After completing successfully, pyRainAdjustment will return the adjusted gridded precipitation with parameter name "P", which is either downscaled or corrected (Figure 2-10). If the adjustment procedure was used, pyRainAdjustment will also return the adjustment factor (parameter name "adjustment_factor"). This adjustment factor is only a multiplicative adjustment factor, calculated by $\frac{P_{adj}}{P_{orig}}$, and is meant for visualization and testing purposes.

parameter (3)

	= internal	= external
1	F_calc	adjustment_factor
2	P_obs	P
3	P_fcst	P

Figure 2-10 Example of IdMapFiles for pyRainAdjustment.

ModuleConfigFiles

A few examples of ModuleConfigFiles to configure the GeneralAdapter runs using pyRainAdjustment are shown in Figure 2-11 for the hindcasting adjustment, Figure 2-12 for the downscaling and Figure 2-13 for the quantile mapping adjustment. The GeneralAdapter runs generally export two netCDFs containing the gridded rainfall that should be adjusted (gridded_rainfall.nc) and, if present, the gauge observations (precip_gauges.nc). In addition, the module exports an xml file (e.g., "adjustment_settings.json" in Figure 2-11), which contains all properties for the run as set in the WorkflowDescriptors (or alternatively in the WorkflowFiles).

Subsequently, pyRainAdjustment is called with the aforementioned Python call and finally, the resulting netCDF(s) is (are) imported into Delft-FEWS. All configuration options of pyRainAdjustment are described in section 2.5.3, but below we already provide some more information on the quantile mapping approach and its options.

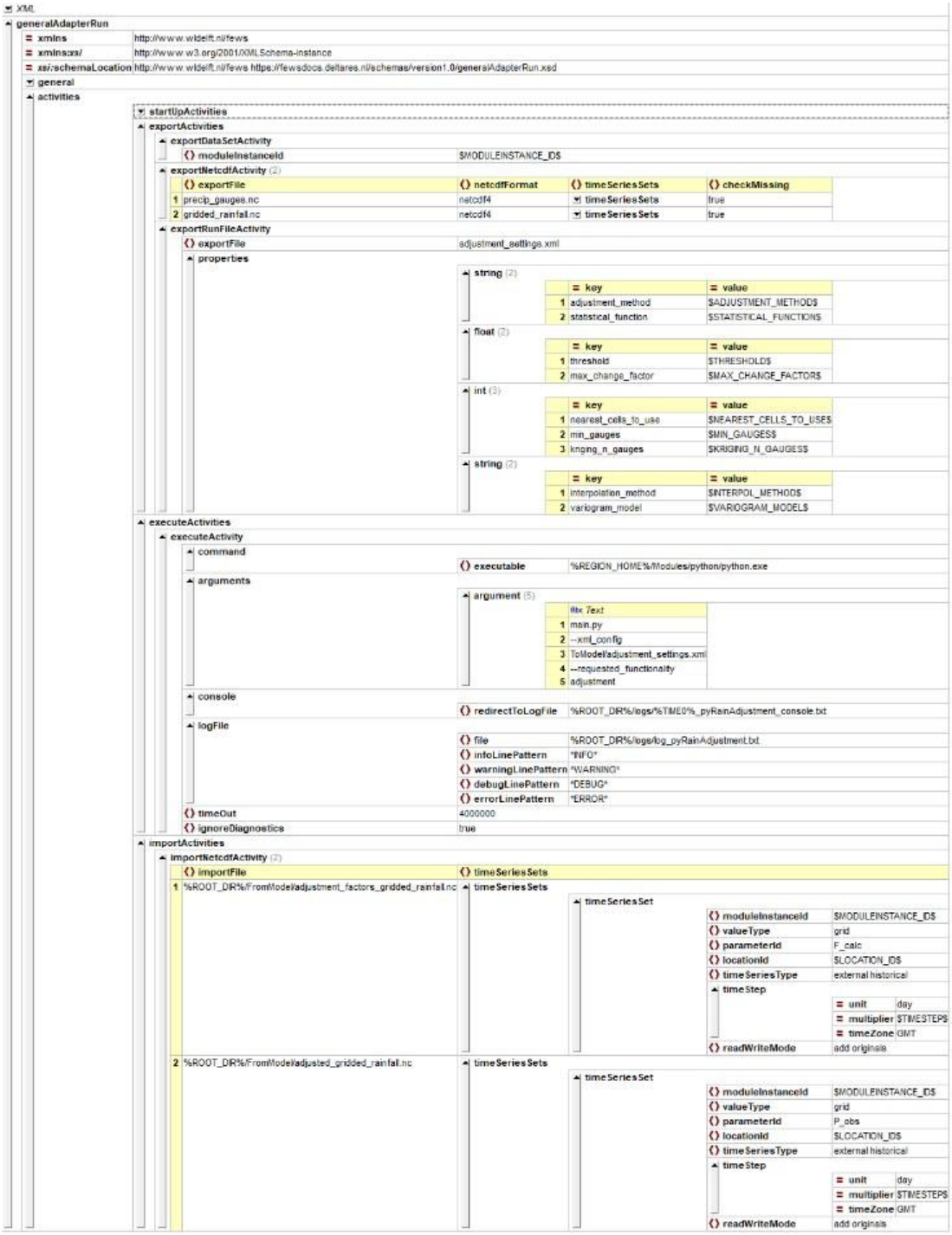

Figure 2-11 Example ModuleConfigFile to run the adjustment procedure with pyRainAdjustment. Note that the exported datasets (input to pyRainAdjustment) should be called "Precip_Gauges.nc" for gauge observations "gridded_rainfal.nc" for the gridded QPE.

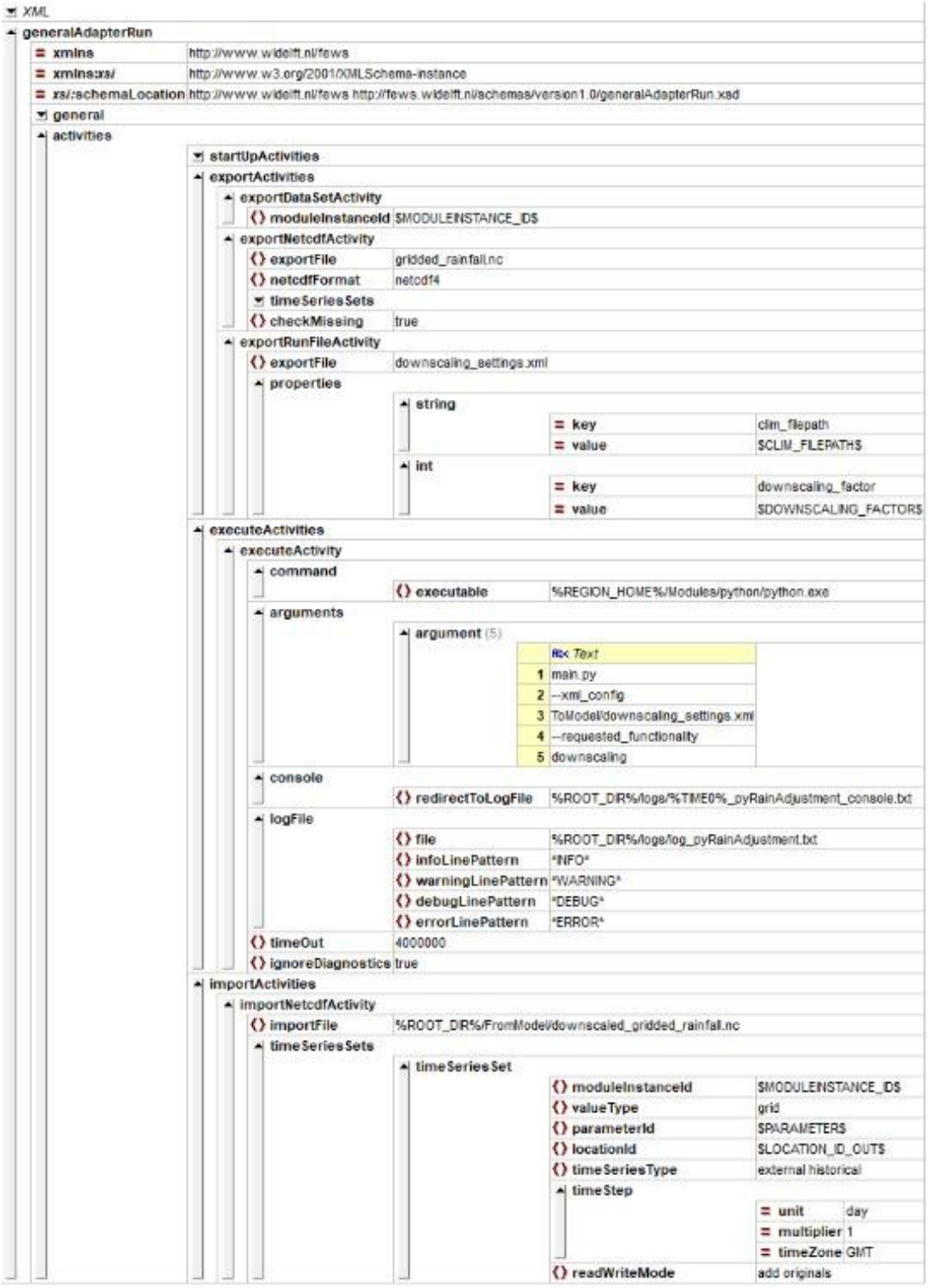

Figure 2-12 Example ModuleConfigFile to run the downscaling procedure with pyRainAdjustment. Note that the exported dataset (input to pyRainAdjustment) should be called "gridded_rainfal.nc" for the gridded QPE.

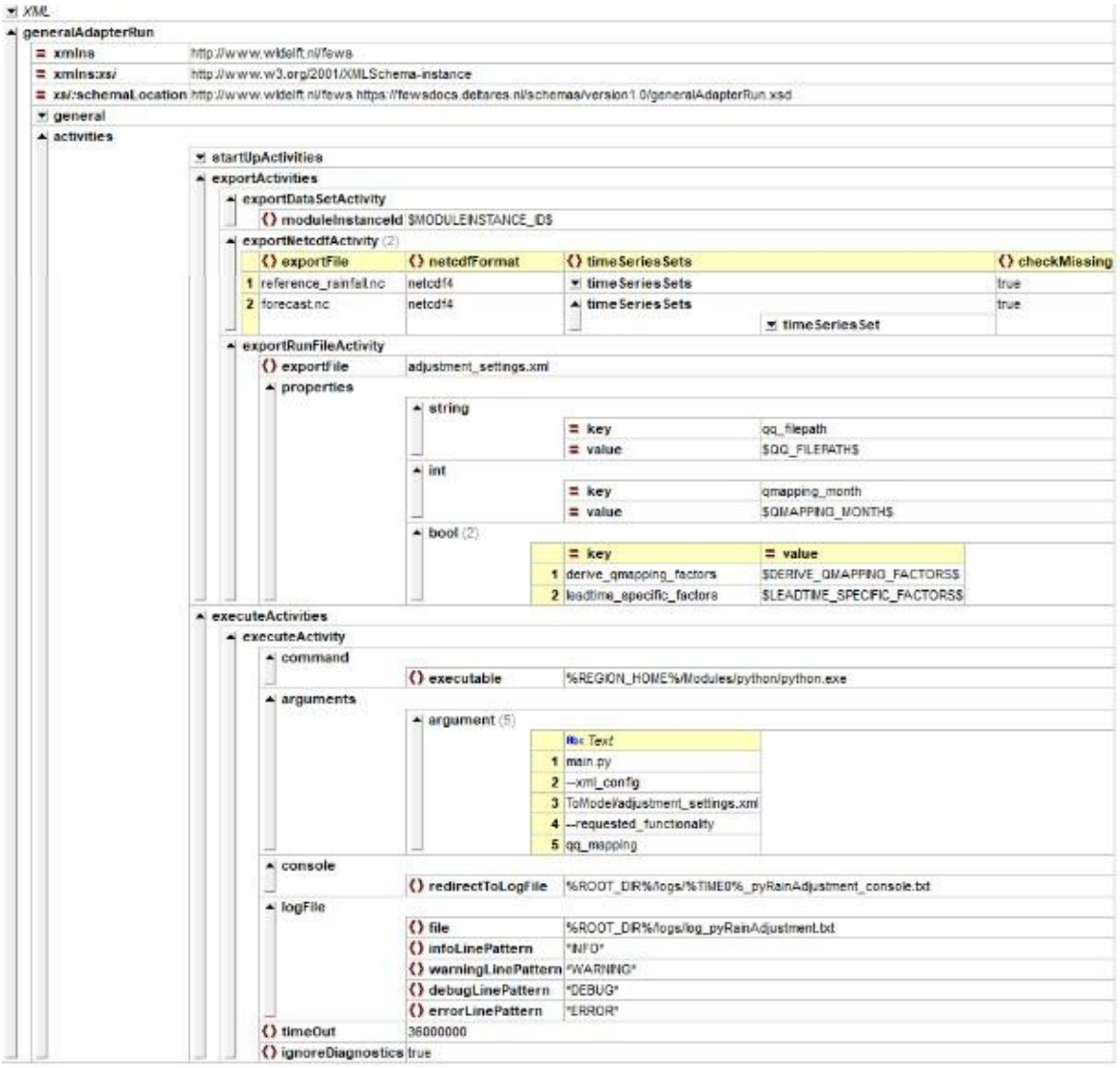

Figure 2-13 Example ModuleConfigFile to run the quantile mapping procedure with pyRainAdjustment. This example derives the quantile mapping correction factors. The ModuleConfigFile for applying the quantile mapping factors to a forecast is nearly identical, only the export of the reference rainfall is not present and instead the module imports the corrected forecast and adjustment factors at the end of the GeneralAdapter run. Note that the exported datasets (input to pyRainAdjustment) should be called "reference_rainfall.nc" for gridded reference rainfall or "precip_gauges.nc" for gauge observations as reference, when determining the adjustment factors. When applying the quantile mapping to a forecast, the forecast should be exported under the name "gridded_rainfall_forecast.nc".

PyRainAdjustment approaches quantile mapping with a few standard options. It will group all historic forecasts and reference data based on the forecast month. This means that all forecasts with an issue time (t_0) in, for instance, March will be corrected based on the distributions of historic forecasts and reference data from March. Thus, a forecast that starts in March and extends into April will only be using the March correction factors.

The method requires a few arguments to be able to run. First of all, the argument "derive_qmapping_factors" needs to be passed to indicate whether new correction files, which form the basis of the method in forecasting mode, need to be computed. If that is the case, pay attention that the forecast time is right at the end of your historic reference period, otherwise parts of the reference period will be missing or data outside of the period might accidentally be included (also note that the exported forecasts should match the time span of the reference data, because pyRainAdjustment does currently not check for that).

Standard, pyRainAdjustment will store correction factors for all lead times together, meaning that it will provide one correction factor that is the same for all lead times, but that varies based on the location in the grid and the month of the issue time. However, the argument “leadtime_specific” (by default ‘False’) can be overwritten to compute correction factors for each leadtime within the provided data. Depending on the provided data, this might be prone to overfitting, which is why it is set to ‘False’ by default.

Once the quantile mapping correction factors have been derived, they are stored in a separate folder, which can be defined by the user or, otherwise, they will be automatically placed in a folder called “qq_correction_factors” in the pyRainAdjustment module folder. From here on, the quantile mapping workflow can be called with "derive_qmapping_factors" set to False and by providing a forecast. Based on that, pyRainAdjustment will search for the month corresponding to the issue time of the forecast and the right lead times (if “leadtime_specific” is True). The gridded rainfall forecast is then corrected based on the quantile the forecast for each grid cell falls in.

RegionConfigFiles

```
<workflowDescriptor id="Meteo_Rain_Gauge_Adjustment_ERA5" name="Bias correct ERA5" visible="true" forecast="false" autoApprove="true" allowApprove="true
">
  <properties>
    <string key="MODULEINSTANCE_ID" value="pyRainAdjustment"/>
    <string key="PRECIP_ID_TO_ADJUST" value="ImportERA5"/>
    <string key="ID_METEO_INPUT" value="ImportObservedPrecipSynops"/>
    <string key="LOCATION_ID" value="ERA5"/>
    <string key="ADJUSTMENT_METHOD" value="MFB"/>
    <string key="STATISTICAL_FUNCTION" value="median"/>
    <float key="THRESHOLD" value="0.01"/>
    <float key="MAX_CHANGE_FACTOR" value="2.0"/>
    <int key="NEAREST_CELLS_TO_USE" value="1"/>
    <int key="MIN_GAUGES" value="2"/>
    <int key="TIMESTEP" value="1"/>
    <int key="KRIGING_N_GAUGES" value="12"/>
    <string key="INTERPOL_METHOD" value="Idw"/>
    <string key="VARIOGRAM_MODEL" value="auto_derive"/>
  </properties>
</workflowDescriptor>
<workflowDescriptor id="Meteo_Rain_Downscaling_ERA5" name="Downscale ERA5" visible="true" forecast="false" autoApprove="true" allowApprove="true">
  <properties>
    <string key="PRECIP_ID_TO_DOWNSCALE" value="ImportERA5"/>
    <string key="MODULEINSTANCE_ID" value="pyRainAdjustment"/>
    <string key="LOCATION_ID_IN" value="ERA5"/>
    <string key="LOCATION_ID_OUT" value="ERA5_highres"/>
    <string key="PARAMETER" value="P_obs"/>
    <string key="CLIM_FILEPATH" value="%REGION_HOME%/Modules/pyRainAdjustment/clim/wc2.1_30s_prec.nc"/>
    <int key="DOWNSCALING_FACTOR" value="10"/>
  </properties>
</workflowDescriptor>
```

Figure 2-14 Example of the pyRainAdjustment run properties for both the adjustment and downscaling procedures using ERA5 in the WorkflowDescriptors.

In the Delft-FEWS application ROWS, properties for workflows are typically set in the WorkflowDescriptors. This also holds for the pyRainAdjustment GeneralAdapter run properties, see Figure 2-14 and Figure 2-15. All property configuration options for pyRainAdjustment are listed in section 2.5.3. The example in Figure 2-14 shows (1) an adjustment run using the MFB adjustment method, using: the option to search for the nearest grid cell (so without using a search window), a minimum threshold of 0.01 mm, a maximum change factor of 2.0, a minimum number of 2 gauges, and IDW interpolation (which is only used when a spatial adjustment method is selected); (2) a downscaling run using a downscaling factor of 10 and a predefined location where monthly precipitation climatology is stored. The quantile mapping example in Figure 2-15 shows both a derivation and application of the quantile mapping correction factors focusing on the month of March (month 3) and no lead time specific factors, thus resulting in correction factors that hold for each lead time in the IFS forecasts.

```
<workflowDescriptor forecast="true" id="Process_Determine_QuantileMapping" name="Bias correction fitting quantile mapping" visible="true" autoApprove="true" allowApprove="true">
  <properties>
    <string key="MODULEINSTANCE_ID" value="pyRainAdjustment"/>
    <string key="PRECIP_ID_TO_ADJUST" value="ImportIFS"/>
    <string key="ID_METEO_INPUT" value="ImportERA5Land"/>
    <string key="LOCATION_ID" value="IFS"/>
    <string key="QQ_FILEPATH" value="%REGION_HOME%/Modules/pyRainAdjustment/qq_correction_factors"/>
    <int key="QMAPPING_MONTH" value="3"/>
    <bool key="DERIVE_QMAPPING_FACTORS" value="true"/>
    <bool key="LEADTIME_SPECIFIC_FACTORS" value="false"/>
  </properties>
</workflowDescriptor>
<workflowDescriptor forecast="true" id="Process_Apply_QuantileMapping" name="Apply quantile mapping" visible="true" autoApprove="true" allowApprove="true">
  <properties>
    <string key="MODULEINSTANCE_ID" value="pyRainAdjustment"/>
    <string key="PRECIP_ID_TO_ADJUST" value="ImportIFS"/>
    <string key="LOCATION_ID" value="IFS"/>
    <string key="QQ_FILEPATH" value="%REGION_HOME%/Modules/pyRainAdjustment/qq_correction_factors"/>
    <int key="QMAPPING_MONTH" value="3"/>
    <bool key="DERIVE_QMAPPING_FACTORS" value="false"/>
    <bool key="LEADTIME_SPECIFIC_FACTORS" value="false"/>
  </properties>
</workflowDescriptor>
```

Figure 2-15 Example of the pyRainAdjustment run properties for the quantile mapping procedure using ECMWF IFS forecasts in the WorkflowDescriptors.

WorkflowFiles

Since all configuration and properties are set by the ModuleConfigFiles and WorkflowDescriptors, the actual workflow in the WorkflowFiles to run pyRainAdjustment can be rather short. Figure 2-16 shows an example of a workflow to run the climatology-based downscaling of the gridded precipitation dataset (ERA5 here), followed by an adjustment of the rainfall volumes using the local rain gauges.

XML

workflow

xmlns	http://www.wldelft.nl/fews
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLoca...	http://www.wldelft.nl/fews https://fewsdocs.deltares.nl/schemas/version1.0/workflow.xsd
version	1.1

activity (2)

	runIndependent	moduleInstance...	moduleConfigFileName
1	false	$MODULEINSTANCE_ID$	ProcessRainDownscaling
2	false	$MODULEINSTANCE_ID$	ProcessRainGaugeAdjustment

Figure 2-16 Example workflow where first climatology-based downscaling of the gridded precipitation dataset takes place, followed by an adjustment of the rainfall volumes.

2.5.3 Configuration options

pyRainAdjustment can be fully configured with a list of options that are displayed in Table 2-1. These configuration options are passed to pyRainAdjustment through an xml file that is exported by Delft-FEWS. These properties can be configured through a combination of the ModuleConfigFile (Figure 2-11, Figure 2-12 and Figure 2-13) and the WorkflowDescriptors (Figure 2-14 and Figure 2-15).

Table 2-1 List of configuration options in pyRainAdjustment.

Configuration parameter	Functionality this option is applied to	Explanation
clim_filepath	Downscaling	The file path to where the climatology file(s) is (are) stored.
downscaling_factor	Downscaling	The factor with which the original grid resolution is downscaled. “2” would indicate that the resolution would become two times higher. This value is always an integer.
adjustment_method	Adjustment	The used adjustment method (provided as a string). This can be: MFB, additive, multiplicative, mixed, KED and quantile_mapping.

Configuration parameter	Functionality this option is applied to	Explanation
statistical_function	Adjustment	The statistical operator that is used to find the best matching cell/value for the provided gauge location (provided as a string). Options are: mean, median, best, min and max.
threshold	Adjustment	Threshold (float; in the unit of the data, e.g. mm or mm h^{-1}), which assures that adjustment only takes place when both the gauge and corresponding grid cell/value are above the threshold.
max_change_factor	Adjustment	Maximum adjustment factor (float) that can be applied to increase/decrease the grid cell values. Everything above this value will be capped to the maximum adjustment value.
nearest_cells_to_use	Adjustment	The size of the search grid box to find the corresponding grid cells for a given rain gauge location (integer).
min_gauges	Adjustment	The minimum number of gauges required to perform the adjustment procedure (integer).
kriging_n_gauges	KED adjustment and kriging interpolation	The minimum number of gauges required to perform either the KED adjustment or kriging interpolation (integer).
interpolation_method	Adjustment	The interpolation method used to map the adjustment factors back onto the original grid (string).
variogram_model	KED adjustment	The variogram model used for the KED adjustment (string). Options are: standard and auto_derive.
derive_qmapping_factors	Quantile mapping	Setting (Boolean) to either derive the quantile mapping factors (when set to True) or to apply the quantile mapping correction (when set to False).
qmapping_month	Quantile mapping	If specified (integer value), pyRainAdjustment will only derive the quantile mapping factors for the indicated month. If set to None (the default), it will derive the factors for all months in the data.
leadtime_specific_factors	Quantile mapping	Setting to derive lead-time specific correction factors (when True) or one factor for all leadtimes (when False). Defaults to False.
qq_filepath	Quantile mapping	The file path where the quantile mapping factors should be / are stored.
gridded_reference_product	Quantile mapping	Boolean setting to indicate whether the reference rainfall for the quantile mapping procedure is a gridded product (True) or contains point observations from rain gauges (False).

3 Case study setup for the MDBA domain

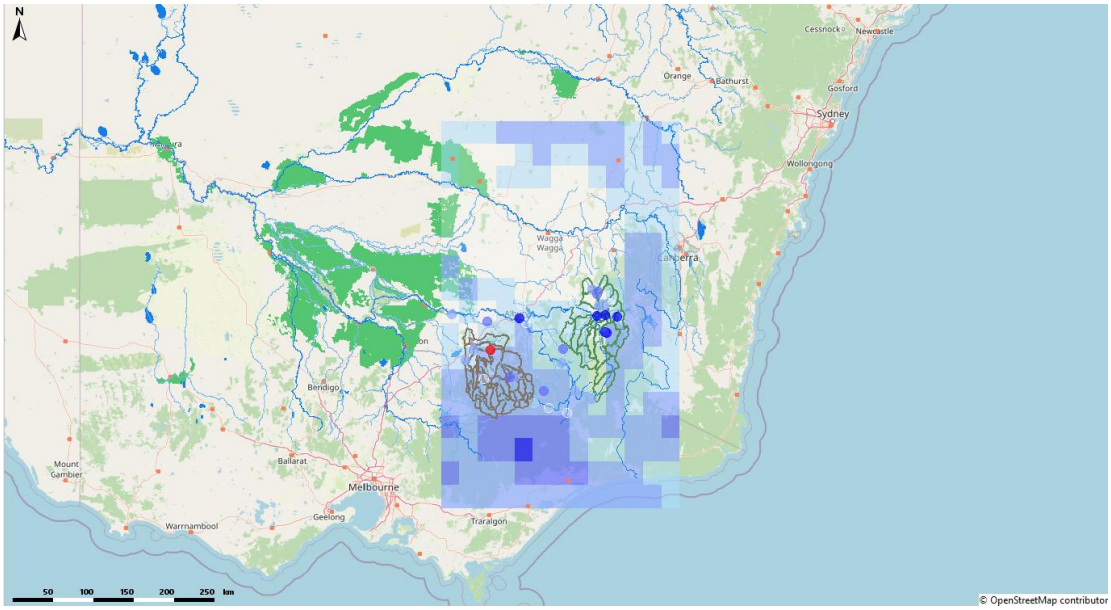

Figure 3-1 The study area, indicated by the extent of the ERA5 daily rainfall estimates (grid cells) and the available rain gauges indicated as dots.

PyRainAdjustment and the machine-learning based downscaling and adjustment has been tested for the MDBA domain, in particular for the upstream reservoir area, see Figure 3-1. In the paragraphs below, we briefly describe the available data and forecasts, and the test setup that we used to analyse the added value of the introduced downscaling and adjustment methods for the MDBA domain.

3.1 Available observation data and forecasts

Rain gauges
Data from 83 hourly rain gauges was available from 2010 onward. The data was also aggregated to daily to allow for testing the adjustment procedures on both hourly and daily time steps.

ERA5 and ERA5 Land Reanalysis
We used ERA5 data in many of our experiments. It combines an array of historical observations (from satellites, weather stations, etc.) with numerical weather models to create a consistent hourly record of the entire globe. This dataset spans from 1940 to the near present, with a resolution of approximately 31 km horizontally and 137 vertical levels up to 80 km altitude. We used precipitation estimates from ERA5 on a 0.25 degree grid resolution and daily time step to test both the downscaling and bias adjustments in a hindcasting mode. ERA5 is a good candidate for testing out the pyRainAdjustment setup, as it provides a long record (we used data from 2010 onward) and has ample room for improvement through both downscaling and correction. ERA5-Land was used to match the grid to the hourly forecasts of ECMWF IFS for the assessment of the quantile mapping method. ERA5-Land is a dataset on an hourly timestep with a 0.1 degree resolution, based on the ERA5 data, and provides high resolution information of surface variables It is available from 1950 to 5 days before the current date.

Radar
Radar data from the Bureau of Meteorology (BoM) for the year 2024 was used as an additional test for the hindcasting adjustment tests. This data already has a 2-km spatial resolution and thus does not need to be downscaled, but adjustment is always needed with radar QPE. The BoM radar data has a 5-min time step and already has undergone a simple real-time bias adjustment using the BoM gauges. This dataset is thus used (on both a daily and hourly time step) to test how much the adjustment methods in hindcasting mode can still add to such a dataset.

ECMWF IFS forecasts
For the forecasting mode of pyRainAdjustment (using the quantile mapping method), we considered the forecasts issued by ECMWF IFS throughout the period 2012-2024. 2012-2023 was taken as the reference period, and 2024 was used to assess the accuracy. Initial analysis has only been carried out on the forecasts of March to limit the quantile mapping derivation time to approximately an hour for the indicated years.

MSWEP
We also include the Multi-Source Weighted-Ensemble Precipitation (MSWEP) data for the validation of our experiments. The MSWEP dataset is a global, high-resolution precipitation product. It merges precipitation estimates from various sources, ground-based rain gauges, satellite observations, and atmospheric reanalysis models, to achieve a 10km spatial resolution from 1979 to present day.

3.2 Hindcasting test setup

In hindcasting mode, we can test the downscaling and adjustment that are possible for reanalysis, state updating and in real time when rain gauge observations are available. For the MDBA domain, we tested the available methods in pyRainAdjustment (sections 2.1 and 2.2) on both the ERA5 reanalysis dataset and the BoM radar data, using the MDBA rain gauges for the adjustment procedure.

The tests with the ERA5 reanalysis dataset took place in two steps. In the first step, the ERA5 dataset was downscaled from an approximately 0.25-degree (approximately 30-km degree) resolution to 0.025 degrees (3 km) using the monthly precipitation climatology of the WorldClim2 dataset (Fick and Hijmans, 2017). Subsequently, the downscaled ERA5 data was bias adjusted. For this step, we tested three different methods: MFB, a spatial mixed error adjustment and kriging with external drift. We tested these steps on the daily ERA5 data from 2010 until 2024 in a leave-one-out fashion, meaning that for each iteration one MDBA rain gauge was not used for adjustment, but only for validation to ensure an independent validation. In the pyRainAdjustment configuration, we configured that the closest 9 grid cells to the gauge should be regarded of which the median value was selected and a maximum change factor of 2.0 was set.

To also test the approaches on a high-resolution dataset, we conducted tests with the BoM radar data on both an hourly and daily basis for the full year of 2024. Because of the high resolution of the radar data, downscaling was not necessary, so the adjustment methods were tested directly on the radar dataset in a similar way as explained for the ERA5 dataset. In the pyRainAdjustment configuration, we configured that the closest 9 grid cells to the gauge should be regarded of which the best-matching value was selected and a maximum change factor of 3.0 was set. This configuration is slightly different from the ERA5 setup and was chosen to test out the effect of a different configuration. Especially a higher maximum change factor could help in better correcting the rainfall values when the bias is high, but could also blow up the correction.

3.3 Forecasting test setup

The tests for the forecasting set up assessed the quantile mapping methodology (section 2.3), where we adjusted the historic IFS forecasts based on the ERA5-Land re-analysis product. The forecasts were corrected disregarding lead time dependency by setting the parameter 'leadtime_specific' to 'False'.

The correction factors were fitted on all March forecasts in the reference period 2012-2023 (forecast were available once every 12 hours, so 744 forecast in total). March was selected to only focus on one month in this test and to ensure a month is used in which sufficient rainfall was present for validation. Afterwards, these factors were applied on the March 2024 forecasts. The added value of this method was tested by comparing it with the original forecast in terms of the change in root mean square error (RMSE) values. This is a metric that places emphasis on the larger values of the distribution, especially interesting in the context of extreme event analysis.

3.4 Data-driven model setup and benchmarking

3.4.1 Features considered

The objective of the experiment is to explore the performance of data-driven models in downscaling precipitation data in the MDBA region. Based on scientific literature recommendations, we adopt ERA5 as the base dataset for meteorological data to be downscaled (the training data, X). The variables considered are divided into two types: single-level values and variables at specific vertical pressure levels. For single-level values, the variables are Total Precipitation and Total Column Water Vapour. For the vertical level of 850 hPa, the variables included are Temperature, U-component of wind, V-component of wind, and Relative Humidity. In addition to ERA5 data as input, we also consider elevation data based on scientific literature recommendation and on the diverse topography of the MDBA domain. The dataset used is the General Bathymetric Chart of the Oceans (GEBCO).

For the target data (y), the data that the data-driven models would try to learn and replicate, we consider two datasets: the BoM radar data, and the MSWEP data. While the BoM radar data is the highest spatial and temporal resolution data for the MDBA region, being a radar product is has some limitations on temporal continuity and spatial extension. Additionally, the big difference in resolution between BoM radar data (2km) and ERA5 (31km) could pose challenges for the data-driven model to learn. The MSWEP has a smaller resolution gap to ERA5, and longer time periods and spatial extents can be used to train the data-driven models. In the case of MSWEP, we use 3h hourly data from 1990 to 2024 for both datasets.

3.4.2 General functioning of machine learning training

A simplified illustration of how most general supervised machine learning models work is shown in the two equations below:

$$\hat{y} = model(X),$$
and
$$\mathcal{L}(\hat{y}, y).$$

Here, X are the input features (i.e., ERA5 data), y is the reference or target data (i.e., the high resolution data) and $\hat{y}$ the predicted output.

The first equations describes the **prediction phase**. Here, the model predicts $\hat{y}$ values of based on X. The second equation represents the **loss function**. This function quantifies the discrepancy between the model's predicted output ($\hat{y}$) and the true, observed target (y). The loss function is the guide for the model to learn, and provides a numerical measure of the model's error.

During the training of a supervised machine learning model, these two components work in conjunction. The model iteratively adjusts its internal parameters by using the loss function's output as feedback. The objective is to minimize this loss, thereby enabling the model to generate predictions ($\hat{y}$) that closely align with the true observations (y) when presented with new, unseen input data (X). During our experiments, we have explored different loss functions. All models are trained on the Deltares high performance cluster using Graphical Processing Units (GPUs) and the python package *Pytorch*.

3.4.3 Preprocessing

Our training of machine learning models requires that every grid cell and time step has both input and target data. Therefore, we preprocess the ERA5 data to resample it to match the corresponding time and spatial resolution of the target data grid. The resampling and regridding process is a nearest neighbour downscaling given its simplicity and the fact that this method does not create new data (and thus not altering the experiment).

3.4.4 Performance analysis

The performance of the outputs produced by the data-driven models is shown with respect to the target data, and also compared against the pre-processed input data for benchmarking, that is, the downscaled ERA5 data with the nearest neighbour resampling technique. The following metrics are used to compare the performances: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Average Bias, Average Spatial Correlation, Normalized Symmetric Mean Absolute Percentage Error (nSMAPE), and Fractions Skill Score (FSS) at 1.0mm, 5.0mm, and 10.0mm thresholds.

4 Case study results

4.1 Hindcasting

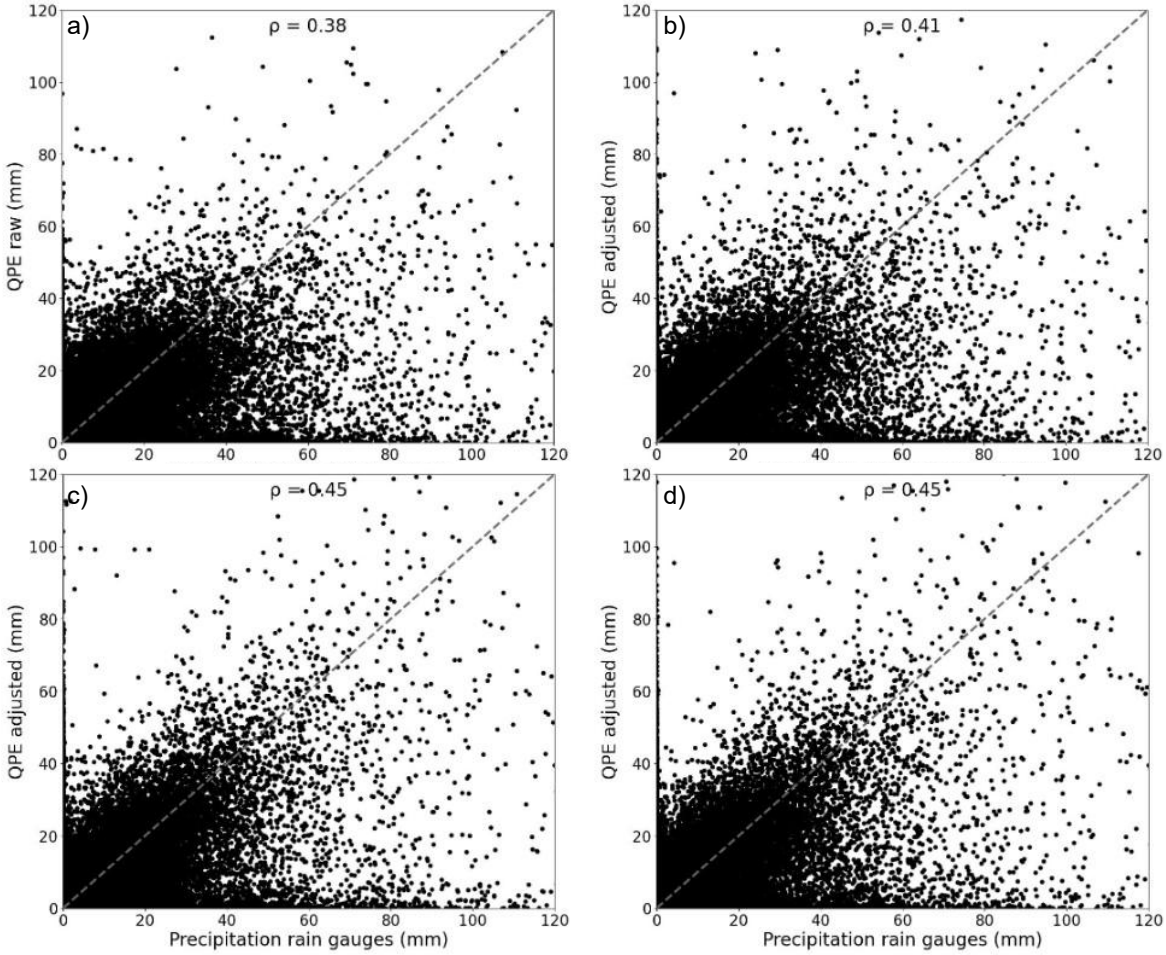

Figure 4-1 Scatter between the daily rainfall amounts as observed by the MDBA rain gauges and corresponding ERA5 daily QPE values for 2010 – 2024. Shown are: (a) The raw ERA5 QPE, (b) the MFB-adjusted ERA5 QPE, (c) the mixed error model adjusted ERA5 QPE, and (d) the KED-adjusted ERA5 QPE.

In hindcasting mode, when gauge observations are present, we can adjust the gridded rainfall estimates using the methods listed in section 2.2. From the results of the tests for the period 2010 – 2024, it becomes directly clear that the ERA5 data needs adjustment, as the ERA5 daily QPE is not well correlated with the observed rainfall data by the rain gauges (Figure 4-1a). The downscaling (from 30 to 3 km) and adjustment steps using pyRainAdjustment generally improve the ERA5 QPE, although sometimes marginally. The MFB adjustment, which can only correct uniformly over the grid, only marginally improves the ERA5 data. This is probably to be expected, since the ERA5 reanalysis contains quality-controlled observations in its model runs, which should already minimize systematic biases. Hence, the QPE likely benefits more from spatial adjustments over uniform (systematic) bias corrections.

Bot the mixed error model and KED, seem to perform better and narrow the wide cloud of points in the scatter plot closer to the 1-on-1 line. However, the correlation is still relatively low with 0.45 for both methods (Figure 4-1c and d).

The narrower spread indicates that the spatial adjustment are advantageous, but the persistently low correlation is at the same time an indication that the match between the very coarse ERA5 data and the local rain gauges is hard to make, as ERA5 estimates rainfall on a 30x30 km^2 resolution, whereas the rain gauges only provide point estimates. This means that correcting ERA5 with only point-scale rain gauges may not be sufficient. The machine-learning approach in section 4.3 may provide a solution. In addition, the restriction to a maximum adjustment factor of 2.0 may play a role here, too, as the discrepancy between ERA5 and the gauges often exceeds a factor 2.0.

To test if the adjustment methods work better on a higher-resolution dataset, we also tested the approach with radar QPE from the BoM (Figure 4-2). As this data has already undergone a gauge adjustment in real time using the BoM rain gauges, the original QPE product is already better correlated with the rain gauges (Figure 4-2a). The simple, but robust MFB adjustment is further improving the QPE (ρ increases from 0.49 to 0.55). Even better improvement can be attained with the spatial adjustment (mixed error and KEG), which can increase the correlation between radar QPE and the gauge up to 0.62, with a much narrower band of points around the 1-on-1 line. Overall, the mixed error adjustment (Figure 4-2c) seems to slightly outperform the KED approach (Figure 4-2d). Since the mixed error adjustment is a more robust approach in real time, as it requires the presence of fewer gauge that are available in real time, this could be the preferred way to go for the MDBA domain.

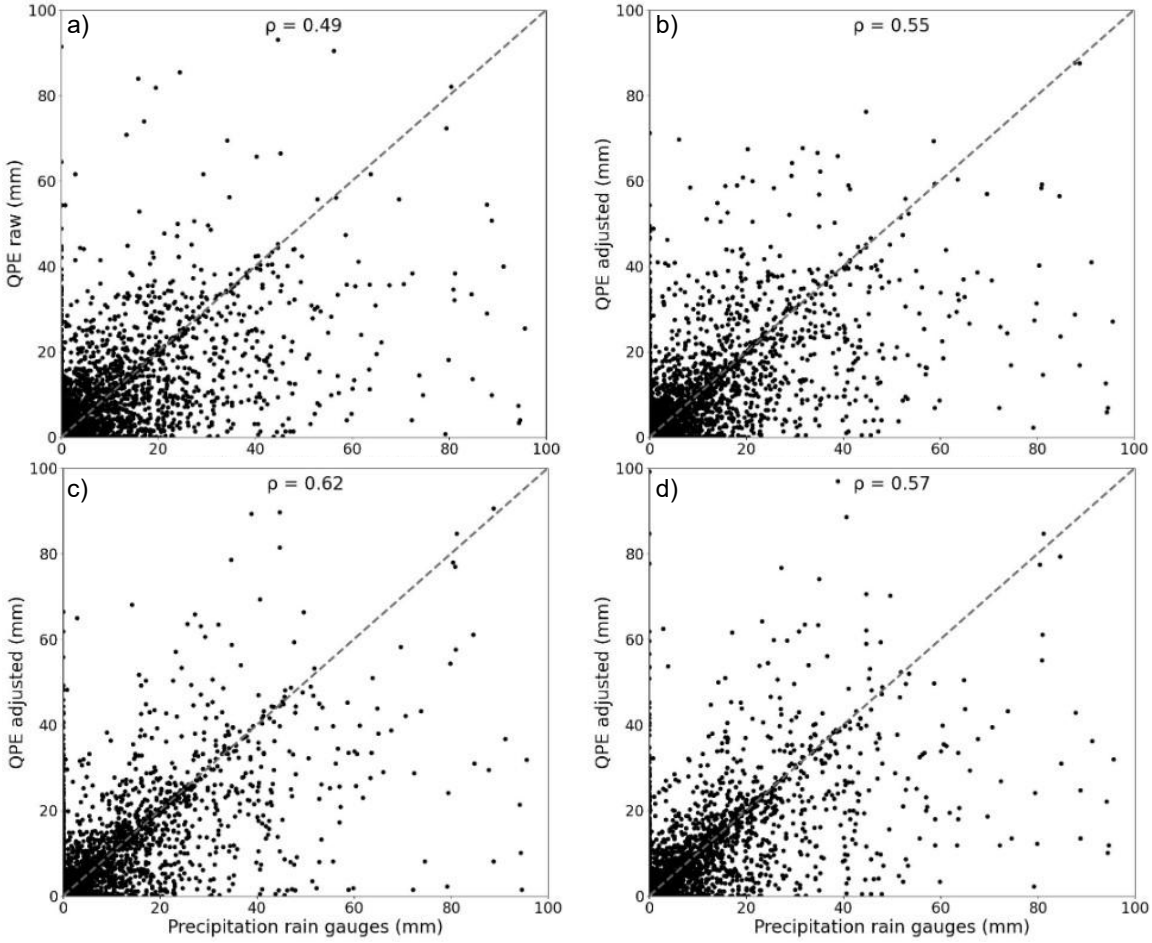

Figure 4-2 Scatter between the daily rainfall amounts as observed by the MDBA rain gauges and corresponding BoM daily radar QPE values for 2024 (based on the leave-on-out analysis, so one point per gauge per day is shown). Shown are: (a) The operational BoM radar product QPE, (b) the MFB-adjusted QPE, (c) the mixed error model adjusted QPE, and (d) the KED-adjusted QPE.

4.2 Forecasting

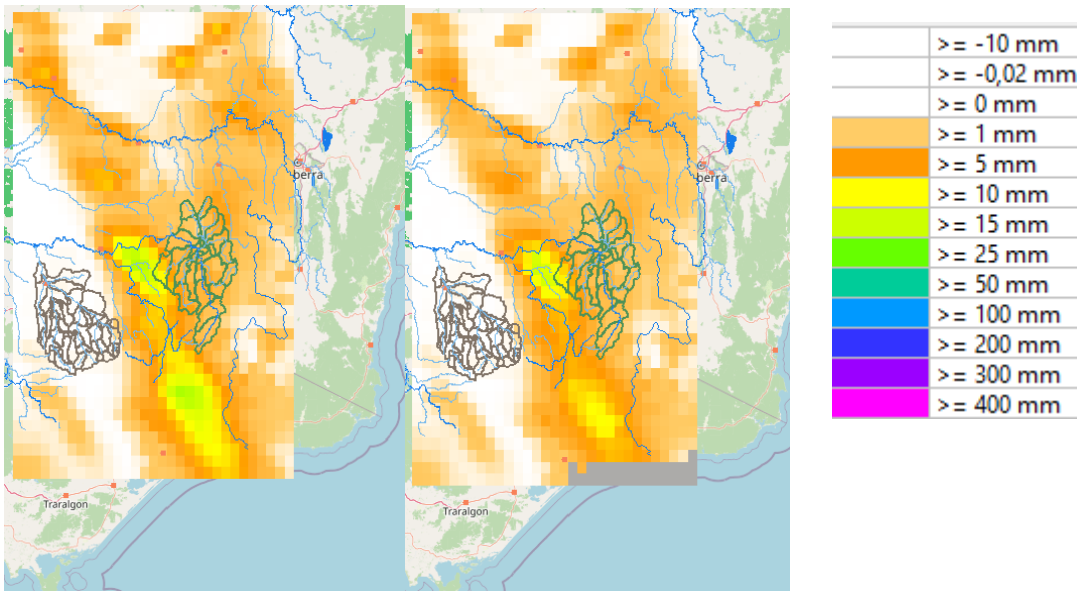

Figure 4-3 Original forecast (left) and corrected forecast (right) issued on 2022-03-18 11:00. Shown are the forecasts for 2024-03-20 01:00.

For the forecasting period (March 2024), we applied the quantile mapping (see Figure 4-3 for an example) and computed the RMSE values of each individual forecast for each cell in the domain (-34:-38 latitude, 146-149 longitude; 0.1° resolution).

Plotting the RMSE distributions of the March 2024 forecasts shows an improvement of the mean RMSE value from 0.152 to 0.134, approximately 10% (Figure 4-4). We see that mostly the large errors are reduced in frequency, while the corrected forecasts do have a bit higher frequency of medium errors (0.5-1.0 RMSE).

We can also study how the RMSE values are distributed across the precipitation magnitude of the reference forecast. We can plot the density of the combinations of precipitation magnitude and forecast RMSE in a contour plot, where the outer contours contain a larger part of all data points, and the inner contours show the highest density of the data distribution. Looking at Figure 4-5, we find that indeed the largest reduction in RMSE values are found in the forecasts where the reference precipitation is between 8 and 20mm across the forecast. Here we see that the original forecasts more frequently contained large errors, which are reduced by the quantile mapping method. The lower bound of the contours are roughly equal, indicating that there is no significant reduction in the already lower error situations.

Looking into the most extreme situations (forecast >15 mm), we find that the orange contour diverges from the blue contour on the bottom side (lower RMSE values) but also drops for the higher RMSE values. This indicates that the corrected data generally has fewer high RMSE values for high precipitation forecasts, both for forecasts that were very inaccurate as well as forecasts that were less erroneous. Hence, this indicates that the quantile mapping method is quite advantageous, especially for improving the forecasting of higher-intensity rainfall.

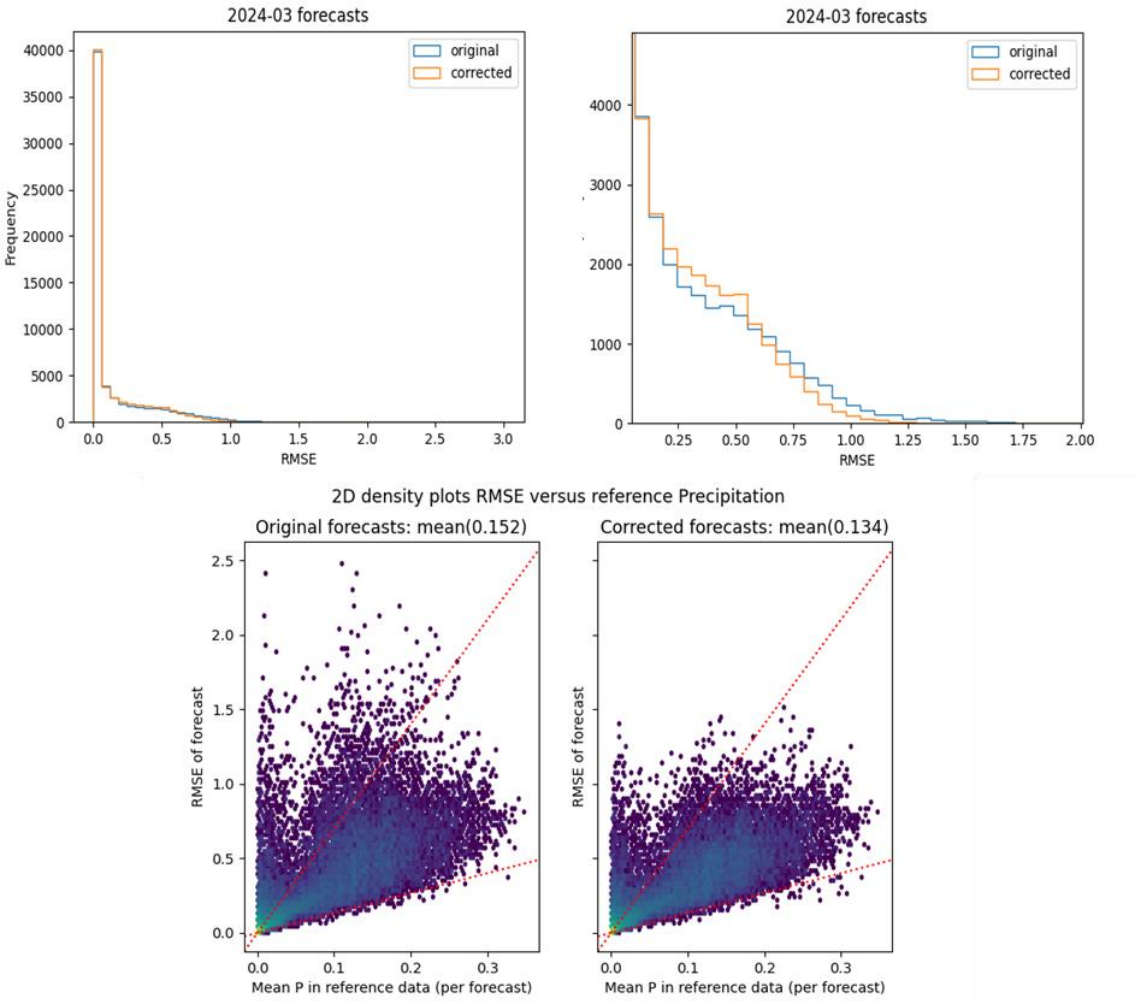

Figure 4-4 (Top) Distributions of the RMSE values for the IFS forecasts in March 2024. (Bottom) 2D density plots of RMSE versus reference precipitation. Colours indicate low (purple) or high (lime green) density. The red dotted lines are for visual aid to more clearly see the difference in the two distributions.

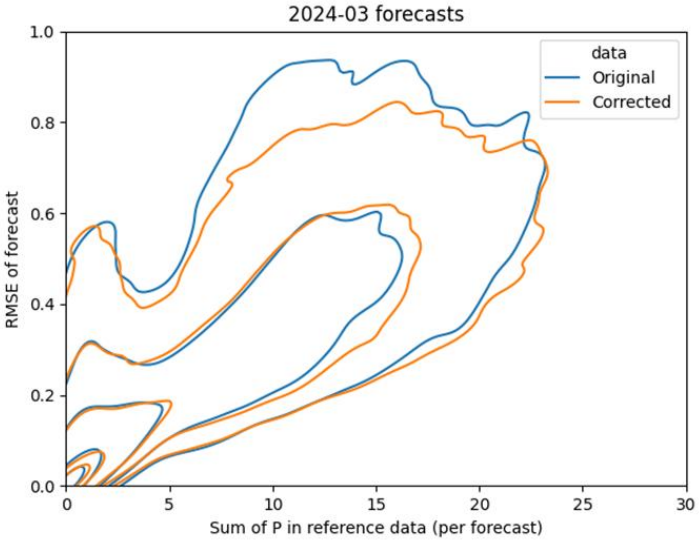

Figure 4-5 2D density contour plot of the precipitation sum per forecast versus the RMSE for the original forecast and the corrected forecasts. Outer contours contain a larger part of the distribution, inner circles indicate highest density.

4.3 Data-driven downscaling and corrections

In this section we demonstrate the most recent results of the ongoing research in which we test the different ML models within the MDBA domain. So far, the focus has been on training the U-Net in a hindcasting mode, as the U-Nets currently seem to provide most potential. The U-Net model was trained and applied on ERA5 data with MSWEP QPE as target reference (see section 3.4). The U-Net model shows varied performance compared to the ERA5 baseline across the metrics (Table 4-1). Root Mean Square Error (RMSE) for the U-Net is 10.7% better than ERA5's, indicating reduced overall error magnitude. Mean absolute error (MAE) shows a smaller improvement of 3.5%. Bias correction shows the highest percentual improvement, +61%, but that is over an already very small margin. For predicting high precipitation thresholds, the Fraction Skill Score (FSS) at 5.0mm and 10.0mm shows modest improvements of +3.23% and +3.11% respectively, indicating enhanced skill in predicting the spatial distribution of heavier rainfall events. A small decrease in performance is observed for the Critical Succes Index (CSI) of events over 10mm, where the U-Net is 3% worse. All other metrics present minimal differences between the ML model and the ERA5 target.

Table 4-1 Summary of the performance of the U-Net model compared to the ERA5 baseline, using target high-resolution precipitation data as ground truth.

Metric	ERA5 vs. Target	U-Net vs. Target	Improvement
MAE	0.2898	0.2796	+3.52%
RMSE	1.0508	0.9385	+10.69%
Bias	-0.0068	-0.0026	+61.48%
Avg Spatial Corr	0.7151	0.7187	+0.50%
nSMAPE	0.4655	0.4623	+0.69%
FSS 1.0mm	0.7413	0.7412	-0.01%
FSS 5.0mm	0.5423	0.5599	+3.23%
FSS 10.0mm	0.6150	0.6341	+3.11%
CSI >10mm	0.1956	0.1892	-3.27%
R95p Correlation	0.9956	0.9958	+0.02%

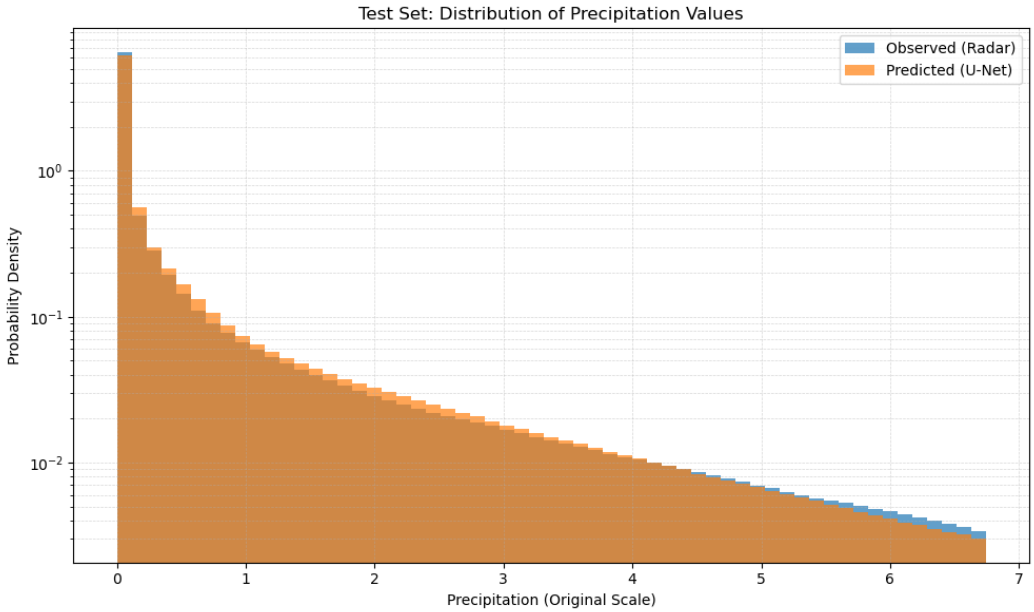

Figure 4-6 Distribution of Precipitation Values. Probability density of observed (MSWEP) and predicted (U-Net) precipitation values across the test set.

The U-Net predicted distribution closely aligns with the observed distribution for lower precipitation values (Figure 4-6). However, for higher precipitation values (e.g., above 3 mm), the U-Net tends to slightly overpredict the frequency compared to the observed data. From >5mm, however, the model starts to slightly underpredict the observations. This is clearly observed in Figure 4-7, where the most extreme time steps are not adequately depicted by the U-net. While the main aspects of large precipitation events are well preserved by the ML model, the output is still very similar to the ERA5 input data.

Given the preliminary nature of these results, we expect to find improvements on subsequent experiments, including optimisation of the hyperparameters of the U-net, including different loss functions tailored to precipitation data, and implementing the other two architectures: GAN and LDM.

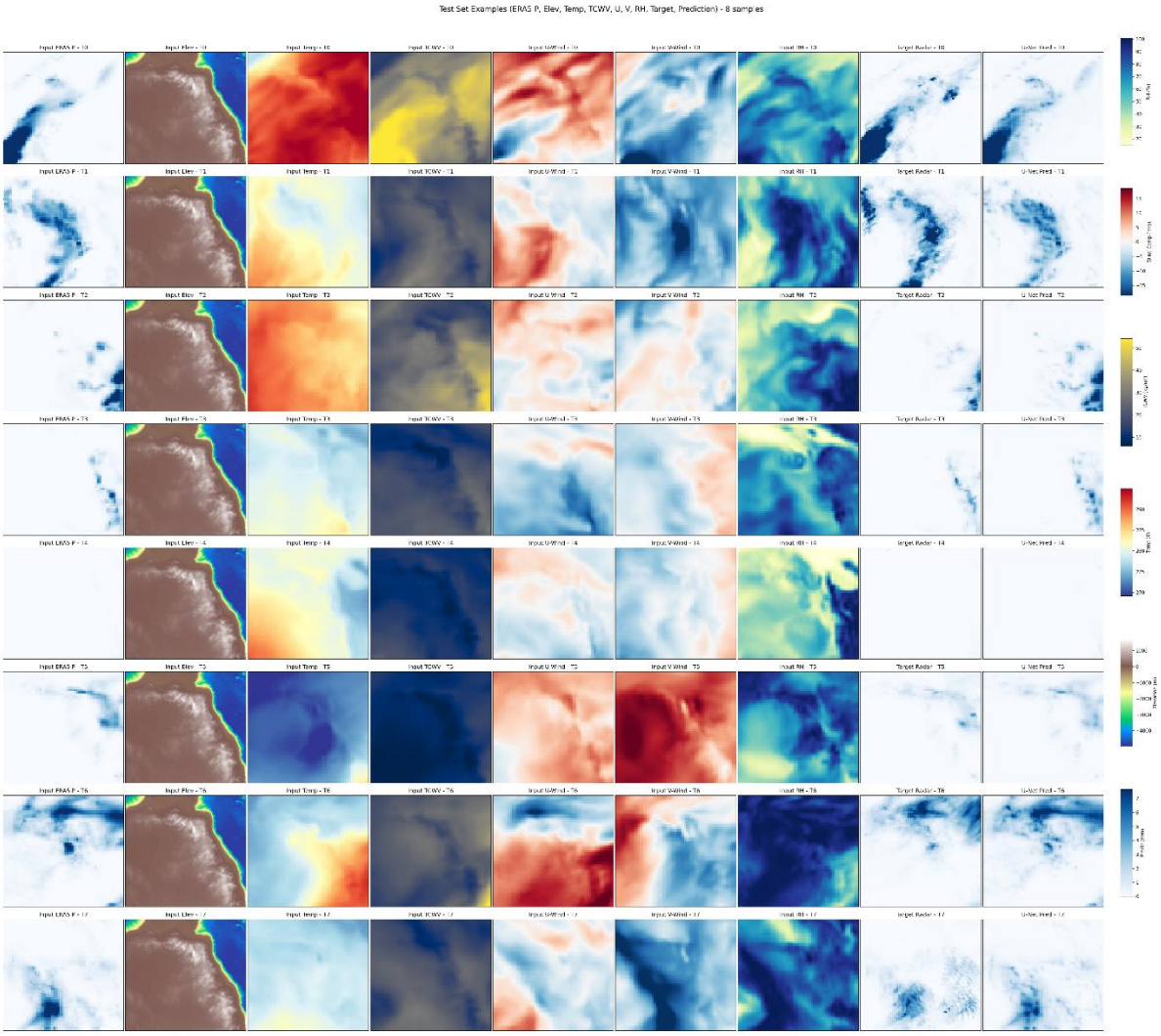

Figure 4-7 Comparison of individual events from the test set (rows). Each column indicates one of the features used in the model, including observed precipitation (MSWEP, second to last column), predicted precipitation (U-Net, last column), and relevant meteorological variables such as total column water vapour (TCWV), wind fields, and relative humidity. The final column indicates the predicted rainfall with the U-Net architecture.

5 Conclusions and recommendations

Accurate and high-resolution quantitative precipitation estimates (QPE) and quantitative precipitation forecasts (QPF) from reanalyses, forecasting products and remotely sensed products such as radar and satellite are essential for effective hydrological forecasting. Yet, current gridded QPE and QPF products often suffer from systematic biases and spatial inaccuracies due to their sometimes large grid cell sizes, which makes it challenging to capture local extremes. While conventional downscaling and adjustment methods exist, ranging from simple to advanced gauge-based merging in hindcasting mode and statistical methods in forecasting mode, there is no tool yet that combines all these methods for usage in operational applications.

In this report, we presented the pyRainAdjustment Python tool, which integrates various methods for adjustment and downscaling of gridded precipitation into a single tool compatible with Delft-FEWS. pyRainAdjustment splits the approach in three parts: (1) downscaling, for which it uses a climatology-based downscaling approach that uses a high-resolution base dataset (e.g., WorldClim2) on a monthly basis; (2) adjustment in a hindcasting mode, for which five approaches were implemented (uniform mean field bias adjustment, multiplicative and additive error adjustment, a mixed error model adjustment and kriging with external drift); and (3) adjustment in a forecasting mode, which consists of the statistical quantile mapping method.

The introduced approaches in pyRainAdjustment were tested for the upper Murray River for MDBA. In hindcasting mode, pyRainAdjustment was applied to both ERA5 reanalysis (on 0.25-degree resolution) and BoM radar QPE (on 2-km resolution), demonstrating its capabilities across varying resolutions. ERA5 data was first downscaled with a factor 10 (to 0.025 degrees) using WorldClim2 climatology and then bias-adjusted with daily rain gauge observations using three methods: mean field bias (MFB), mixed error adjustment and kriging with external drift (KED). Validation took place through a leave-one-out approach for the period 2010 – 2024. For the high-resolution BoM radar data, only adjustment was needed, and validation took place through a leave-one-out approach as well on both the hourly and daily dataset for the year 2024. The results reveal that ERA5 QPE can be quite different from local gauge observations (correlation was 0.38). While downscaling and spatial adjustment methods in pyRainAdjustment improve performance, uniform adjustment methods like MFB have limited impact (improvement in correlation with gauges was around 8%). Spatial techniques such as mixed error modelling and KED show better results (up to 18% improvement), though the coarse resolution of ERA5 and the fixed maximum adjustment factor constrain improvements. In contrast, the higher-resolution BoM radar data, already partially adjusted, respond more effectively to further correction. Spatial methods again outperform MFB, with mixed error adjustment slightly leading in performance (maximum improvement in correlation with rain gauges was around 27%) and offering a more robust option for real-time applications in the MDBA domain than KED, as it requires fewer gauges in real time.

In forecasting mode, quantile mapping was applied to ECMWF's IFS forecasts using ERA5-Land as a reference. The quantile mapping adjustment factors were derived for all March months in the period 2012 – 2023 and applied to March 2024. Applying quantile mapping results in improvement in forecast accuracy, reducing the mean RMSE by approximately 10%. The method effectively reduce the frequency of large forecast errors, particularly for moderate to high precipitation events (8–20 mm), while maintaining similar performance in low-error scenarios. For more extreme rainfall forecasts (>15 mm, i.e. the higher percentiles), quantile mapping clearly decrease both high and moderate RMSE values, highlighting its strength in enhancing the reliability of high-intensity rainfall predictions. Hence, this indicates that the methods in pyRainAdjustment can effectively reduce errors in both hindcasting and forecasting of precipitation.

While conventional adjustment methods have been widely used for meteorological data correction and downscaling, they face limitations in real-time applications due to gauge availability and storage demands. Therefore, this report also explored the potential of machine learning (ML) approaches, which can simultaneously downscale and adjust precipitation data without relying on real-time (gauge) observations. Although this research is part of a larger ongoing effort, here we started with testing three different ML architectures: U-Nets, GANs and Latent Diffusion Models. This report presents the first results of the U-Net architecture using ERA5 reanalysis data as input and MSWEP QPE as target and reference. Current results highlight the potential of using U-Nets in hindcasting mode, with improvements over the original ERA5 QPE in several metrics, including RMSE (10.7%), the overall bias (61%), nSMAPE (3.5%), and FSS at higher precipitation thresholds (3.1%–3.2%), indicating better overall accuracy and skill in predicting heavier rainfall. However, some metrics show degraded performance, with especially a worse CSI score compared to the original ERA5 data, indicating that for higher thresholds, the U-Nets do not seem to capture all values well. The model tends to overpredict moderate precipitation and underpredict extremes, contributing to the observed bias. These mixed results suggest that while U-Net shows potential, further refinement is needed to improve its performance across all precipitation ranges. In the coming months, our research will continue with improving the U-Nets, trying out the other architectures and then moving to the forecasting domain.

5.1 Recommendations for application and future research

Given the current status of pyRainAdjustment, the ongoing machine-learning improvements for downscaling and adjustment, and, overall, to enhance the accuracy and operational usability of gridded precipitation datasets and forecasting products in hydrological forecasting, we list some implementation options and possible future research steps below.

Operational testing and implementation options

- In this report, we have tested the approaches on ERA5, IFS and radar data, because of its long record and availability. However, it is worth exploring the tested methods on satellite-based QPE products as alternative to radar and reanalysis data, especially in gauge-sparse regions.
- Test and integrate pyRainAdjustment into FEWS ROWS to enable streamlined downscaling and adjustment workflows on different gridded QPE and QPF products. This could include applying quantile mapping in real time to correct gridded QPF products, with a focus on improving high-intensity rainfall predictions. Validation of the products, by benchmarking it with the uncorrected QPE and QPF products, would be recommended.
- Regarding the quantile mapping, real-time testing of the method could expand the current analysis for March with additional forecast months. In addition, we recommend focusing on assessing the impacts of the quantile mapping approach on the local water balance and hydrological simulations.

- Operational pilot using ML techniques for real-time rainfall downscaling and bias adjustment of gridded QPE and QPF products, leveraging their ability to perform both tasks simultaneously without reliance on real-time gauge data.

Quantile mapping enhancements

- In the current setup of the quantile mapping approach, we have only tested adjustment factors that are the same for all lead times. This gave more consistent factors over time (instead of fluctuations per hour) and thus better results. However, we think it is worth investigating lead-time-specific fitting procedures (this is already possible in pyRainAdjustment) to improve forecast correction across different horizons, which become particularly valuable for (seasonal) forecasts with longer lead times, such as longer-range to seasonal forecasts.

Research directions

- Once the ML downscaling and adjustment capabilities have sufficient quality and are robust and reliable, we recommend finding a way to integrate these methods into pyRainAdjustment. This step requires a (relatively) generalizable ML setup, which is to be further explored. If we have ML methods in pyRainAdjustment, it will enable seamless usage alongside conventional downscaling and adjustment methods within operational platforms.
- An additional research focus for downscaling and adjustment of gridded rainfall with ML methods is the transferability of these models across different regions and datasets to support broader applicability. This is still to be explored.

References

Berne, A, G Delrieu, J Creutin, and C Obled. 'Temporal and Spatial Resolution of Rainfall Measurements Required for Urban Hydrology'. *Journal of Hydrology* 299, no. 3–4 (1 December 2004): 166–79. https://doi.org/10.1016/S0022-1694(04)00363-4.

Bronstert, A., E. Zehe, and S. Itzerott. 'Operationelle Abfluss- Und Hochwasservorhersage in Quellgebieten - OPAQUE : Abschlussbericht ; Laufzeit Des Vorhabens: 01.06.2006 Bis 31.03.2010', 2010. https://doi.org/10.2314/GBV:65175206X.

Cannon, Alex J., Stephen R. Sobie, and Trevor Q. Murdock. 'Bias Correction of GCM Precipitation by Quantile Mapping: How Well Do Methods Preserve Changes in Quantiles and Extremes?', 1 September 2015. https://doi.org/10.1175/JCLI-D-14-00754.1.

Fick, Stephen E., and Robert J. Hijmans. 'WorldClim 2: New 1-Km Spatial Resolution Climate Surfaces for Global Land Areas'. *International Journal of Climatology* 37, no. 12 (2017): 4302–15. https://doi.org/10.1002/joc.5086.

Goudenhoofdt, E., and L. Delobbe. 'Evaluation of Radar-Gauge Merging Methods for Quantitative Precipitation Estimates'. Hydrology and Earth System Sciences 13, no. 2 (18 February 2009): 195–203. https://doi.org/10.5194/hess-13-195-2009.

Gudmundsson, L., J. B. Bremnes, J. E. Haugen, and T. Engen-Skaugen. 'Technical Note: Downscaling RCM Precipitation to the Station Scale Using Statistical Transformations – a Comparison of Methods'. *Hydrology and Earth System Sciences* 16, no. 9 (21 September 2012): 3383–90. https://doi.org/10.5194/hess-16-3383-2012.

Harris, Lucy, Andrew T. T. McRae, Matthew Chantry, Peter D. Dueben, and Tim N. Palmer. 'A Generative Deep Learning Approach to Stochastic Downscaling of Precipitation Forecasts'. *Journal of Advances in Modeling Earth Systems* 14, no. 10 (2022): e2022MS003120. https://doi.org/10.1029/2022MS003120.

Heistermann, M., S. Jacobi, and T. Pfaff. 'Technical Note: An Open Source Library for Processing Weather Radar Data (*Wradlib*)'. *Hydrology and Earth System Sciences* 17, no. 2 (28 February 2013): 863–71. https://doi.org/10.5194/hess-17-863-2013.

Hess, Philipp, and Niklas Boers. 'Deep Learning for Improving Numerical Weather Prediction of Heavy Rainfall'. *Journal of Advances in Modeling Earth Systems* 14, no. 3 (2022): e2021MS002765. https://doi.org/10.1029/2021MS002765.

Höhlein, Kevin, Michael Kern, Timothy Hewson, and Rüdiger Westermann. 'A Comparative Study of Convolutional Neural Network Models for Wind Field Downscaling'. *Meteorological Applications* 27, no. 6 (2020): e1961. https://doi.org/10.1002/met.1961.

Kheir, Ahmed M. S., Abdelrazek Elnashar, Alaa Mosad, and Ajit Govind. 'An Improved Deep Learning Procedure for Statistical Downscaling of Climate Data'. *Heliyon* 9, no. 7 (1 July 2023). https://doi.org/10.1016/j.heliyon.2023.e18200.

Le, Xuan-Hien, Linh Nguyen Van, Duc Hai Nguyen, Giang V. Nguyen, Sungho Jung, and Giha Lee. 'Comparison of Bias-Corrected Multisatellite Precipitation Products by Deep Learning Framework'. *International Journal of Applied Earth Observation and Geoinformation* 116 (1 February 2023): 103177. https://doi.org/10.1016/j.jag.2022.103177.

Leinonen, Jussi, Daniele Nerini, and Alexis Berne. 'Stochastic Super-Resolution for Downscaling Time-Evolving Atmospheric Fields With a Generative Adversarial Network'. *IEEE Transactions on Geoscience and Remote Sensing* 59, no. 9 (September 2021): 7211–23. https://doi.org/10.1109/TGRS.2020.3032790.

Li, Lizao, Robert Carver, Ignacio Lopez-Gomez, Fei Sha, and John Anderson. 'Generative Emulation of Weather Forecast Ensembles with Diffusion Models'. *Science Advances* 10, no. 13 (29 March 2024): eadk4489. https://doi.org/10.1126/sciadv.adk4489.

Lin, Hai, Jianping Tang, Shuyu Wang, Shuguang Wang, and Guangtao Dong. 'Deep Learning Downscaled High-Resolution Daily near Surface Meteorological Datasets over East Asia'. *Scientific Data* 10, no. 1 (12 December 2023): 890. https://doi.org/10.1038/s41597-023-02805-9.

Ochoa-Rodriguez, S., L.-P. Wang, P. Willems, and C. Onof. 'A Review of Radar-Rain Gauge Data Merging Methods and Their Potential for Urban Hydrological Applications'. *Water Resources Research* 55, no. 8 (2019): 6356–91. https://doi.org/10.1029/2018WR023332.

Ringard, Justine, Frederique Seyler, and Laurent Linguet. 'A Quantile Mapping Bias Correction Method Based on Hydroclimatic Classification of the Guiana Shield'. *Sensors* 17, no. 6 (June 2017): 1413. https://doi.org/10.3390/s17061413.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. 'U-Net: Convolutional Networks for Biomedical Image Segmentation'. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, edited by Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, 234–41. Cham: Springer International Publishing, 2015. https://doi.org/10.1007/978-3-319-24574-4_28.

Seo, D. -J, J. P Breidenbach, and E. R Johnson. 'Real-Time Estimation of Mean Field Bias in Radar Rainfall Data'. *Journal of Hydrology* 223, no. 3 (1 October 1999): 131–47. https://doi.org/10.1016/S0022-1694(99)00106-7.

Sharifi, Ehsan, Josef Eitzinger, and Wouter Dorigo. 'Performance of the State-Of-The-Art Gridded Precipitation Products over Mountainous Terrain: A Regional Study over Austria'. *Remote Sensing* 11, no. 17 (January 2019): 2018. https://doi.org/10.3390/rs11172018.

Singh, Manmeet, Bipin Kumar, Suryachandra Rao, Sukhpal Singh Gill, Rajib Chattopadhyay, Ravi S. Nanjundiah, and Dev Niyogi. 'Deep Learning for Improved Global Precipitation in Numerical Weather Prediction Systems'. arXiv.org, 20 June 2021. https://arxiv.org/abs/2106.12045v2.

Smith, James A., and Witold F. Krajewski. 'Estimation of the Mean Field Bias of Radar Rainfall Estimates'. *Journal of Applied Meteorology and Climatology* 30, no. 4 (1 April 1991): 397–412. https://doi.org/10.1175/1520-0450(1991)030<0397:EOTMFB>2.0.CO;2.

Stengel, Karen, Andrew Glaws, Dylan Hettinger, and Ryan N. King. 'Adversarial Super-Resolution of Climatological Wind and Solar Data'. *Proceedings of the National Academy of Sciences* 117, no. 29 (21 July 2020): 16805–15. https://doi.org/10.1073/pnas.1918964117.

Tan, Jinkai, Sheng Chen, Chia-Ying Lee, Guangtao Dong, Wenyan Hu, and Jun Wang. 'Projected Changes of Typhoon Intensity in a Regional Climate Model: Development of a Machine Learning Bias Correction Scheme'. *International Journal of Climatology* 41, no. 4 (2021): 2749–64. https://doi.org/10.1002/joc.6987.

Todini, E. 'A Bayesian Technique for Conditioning Radar Precipitation Estimates to Rain-Gauge Measurements'. *Hydrology and Earth System Sciences* 5, no. 2 (30 June 2001): 187–99. https://doi.org/10.5194/hess-5-187-2001.

Tomasi, Elena, Gabriele Franch, and Marco Cristoforetti. 'Can AI Be Enabled to Perform Dynamical Downscaling? A Latent Diffusion Model to Mimic Kilometer-Scale COSMO5.0_CLM9 Simulations'. *Geoscientific Model Development* 18, no. 6 (1 April 2025): 2051–78. https://doi.org/10.5194/gmd-18-2051-2025.

Wackernagel, Hans. 'Ordinary Kriging'. In Multivariate Geostatistics: An Introduction with Applications, edited by Hans Wackernagel, 79–88. Berlin, Heidelberg: Springer, 2003. https://doi.org/10.1007/978-3-662-05294-5_11.

Wang, Fang, and Di Tian. 'On Deep Learning-Based Bias Correction and Downscaling of Multiple Climate Models Simulations'. *Climate Dynamics* 59, no. 11 (1 December 2022): 3451–68. https://doi.org/10.1007/s00382-022-06277-2.

Wang, Fang, Di Tian, and Mark Carroll. 'Customized Deep Learning for Precipitation Bias Correction and Downscaling'. *Geoscientific Model Development* 16, no. 2 (25 January 2023): 535–56. https://doi.org/10.5194/gmd-16-535-2023.

Deltares is an independent institute for applied research in the field of water and subsurface. Throughout the world, we work on smart solutions for people, environment and society.

Deltares

www.deltares.nl