MAchinE Learning for Scalable meTeoROlogy and climate

D1.1 First version of datasets and cost functions to develop machine learning solutions for A1-A6

Peter Dueben & Matthew Chantry

www.maelstrom-eurohpc.eu

D1.1 First version of datasets and cost functions to develop machine learning solutions for A1-A6

Author(s): Peter Dueben (ECMWF), Matthew Chantry (ECMWF)
Thomas Nipen (MetNor),
Greta Denisenko (4cast),
Tal Ben-Nun (ETH)
Bing Gong (FZJ), Michael Langguth (FZJ)

Dissemination Level: Public

Date: 31/08/2021
Version: 1.0
Contractual Delivery Date: 31/08/2021
Work Package/ Task: WP1/ T1.2
Document Owner: ECMWF
Contributors: 4cast, ETH, FZJ, MetNor
Status: Final

MAELSTROM

Machine Learning for Scalable Meteorology and Climate

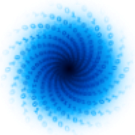
Research and Innovation Action (RIA)
H2020-JTI-EuroHPC-2019-1: Towards Extreme Scale Technologies and Applications

Project Coordinator: Dr Peter Dueben (ECMWF)
Project Start Date: 01/04/2021
Project Duration: 36 months

Published by the MAELSTROM Consortium

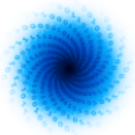
Contact:
ECMWF, Shinfield Park, Reading, RG2 9AX, United Kingdom
Peter.Dueben@ecmwf.int

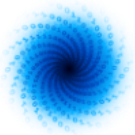
The MAELSTROM project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955513. The JU receives support from the European Union's Horizon 2020 research and innovation programme and United Kingdom, Germany, Italy, Luxembourg, Switzerland, Norway



Contents

Tables



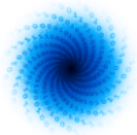


1 Executive Summary

This deliverable provides background information on the first datasets for the six MAELSTROM applications that are now available for download online. For all but one of the datasets, a Jupyter notebook is available that provides a recipe on how to download the data, how to train a vanilla machine learning solution for the given problem, how to evaluate the quality of the solution, and how to plot some of the resulting quantities.

The datasets that are available have also been unified as much as possible with a common framework for data loading -- using the so-called CliMetLab library -- and a common way to store the data in S3 buckets.

While the first version of the datasets is now available at month five of the project, the datasets will still develop further over time and the machine learning solutions will be upgraded as they become more-and-more customised to the needs of the applications.



2 Introduction

2.1 About MAELSTROM

To develop Europe's computer architecture of the future, MAELSTROM will co-design bespoke compute system designs for optimal application performance and energy efficiency, a software framework to optimise usability and training efficiency for machine learning at scale, and large-scale machine learning applications for the domain of weather and climate science.

The MAELSTROM compute system designs will benchmark the applications across a range of computing systems regarding energy consumption, time-to-solution, numerical precision and solution accuracy. Customised compute systems will be designed that are optimised for application needs to strengthen Europe's high-performance computing portfolio and to pull recent hardware developments, driven by general machine learning applications, toward needs of weather and climate applications.

The MAELSTROM software framework will enable scientists to apply and compare machine learning tools and libraries efficiently across a wide range of computer systems. A user interface will link application developers with compute system designers, and automated benchmarking and error detection of machine learning solutions will be performed during the development phase. Tools will be published as open source.

The MAELSTROM machine learning applications will cover all important components of the workflow of weather and climate predictions including the processing of observations, the assimilation of observations to generate initial and reference conditions, model simulations, as well as post-processing of model data and the development of forecast products. For each application, benchmark datasets with up to 10 terabytes of data will be published online for training and machine learning tool-developments at the scale of the fastest supercomputers in the world. MAELSTROM machine learning solutions will serve as blueprint for a wide range of machine learning applications on supercomputers in the future.

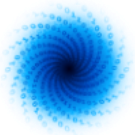
2.2 Scope of this deliverable

2.2.1 Objectives of this deliverable

To establish and publish a first version of the MAELSTROM datasets for the MAELSTROM applications for use in subsequent studies and MAELSTROM performance benchmarks; to provide an easy access point to download and use the datasets via Jupyter notebooks; to unify the datasets as much as possible.

2.2.2 Work performed in this deliverable

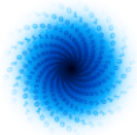
The first versions of the datasets are established and available publicly via an S3 data bucket. A description of the datasets is provided. Furthermore, a simple Jupyter notebook is provided that serves as an example of how the data can be downloaded and loaded into a machine learning



environment. A vanilla machine learning solution is also prepared that serves as an example of how tools should be trained and a cost function for the evaluation of the quality of the solution is given.

2.2.3 Deviations and counter measures

There are no significant deviations from the planned contributions of the deliverable. We have delivered usable versions of five of the MAELSTROM applications that are already based on a common data loading framework (CliMetLab) and available in sufficient size for meaningful performance testing. For application A2 we could, however, not yet collect sufficient data in the form of twitter feeds that are selected around the topic of weather and climate predictions, as we are still waiting on a response from our application to Twitter to receive such data. The dataset will be added as soon as the data is available.



3 A description of the six applications and the datasets

3.1 A1: Blend citizen observations and numerical weather forecast

Motivation and Description:

Public weather forecast providers strive to deliver forecasts that are accurate and tailored to the specific locations of their end users. Although weather models are constantly improving, large errors in models are still common. For example, in mountainous Norwegian terrain, forecasted temperatures can have errors exceeding 10°C, especially in winter-time inversion conditions. Forecast providers, such as MET Norway, therefore rely on machine learning to reduce these errors. The A1 dataset contains high resolution weather data for the Nordic region and forms the basis for the operational forecasts on the weather app Yr (https://www.yr.no).

To produce high resolution forecasts, machine learning models need high resolution target fields for training. An important emerging source of weather measurements is networks of citizen weather stations. These off-the-shelf devices owned and maintained by private individuals typically deliver weather measurements in near real-time and due to their popularity offer high resolution information on current weather conditions. The target field in the A1 dataset is constructed from measurements from Netatmo's network of citizen weather stations. The network density is roughly two orders of magnitude higher than the density of the network of conventional stations operated by MET Norway. The observations have been quality controlled and combined with model data to produce a best estimate of the weather for each hour in the past, to make them easier to use for the training of machine learning models.

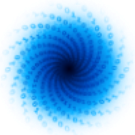
The prediction task is to generate deterministic temperature and/or hourly precipitation forecasts (treated as a median of the forecast distribution) together with a 10-90% confidence interval on a 1x1 km grid. The task could easily be extended to any number of quantiles and even a complete quantile distribution function, but the three chosen quantiles are used in the visualization of temperature and precipitation forecasts and their uncertainty in the Yr app.

Input and output data:

The dataset contains two predictands, 2m temperature and hourly precipitation. Both datasets share most of the same predictors. The datasets contain gridded forecasts on a 1x1 km Lambert conformal grid (Lambert 1772) covering the Nordic countries. The available predictors are:

Predictor name	Unit	Has leadtime?
2m temperature	°C	Yes
Cloud area fraction	1	Yes
Precipitation amount	mm	Yes
10m x wind component	m/s	Yes
10m y wind component	m/s	Yes
2m temperature bias previous day	°C	Yes
2m temperature bias at initialization time	°C	No

Table 1: A1 predictors for 2m temperature and hourly precipitation forecast



The temperature dataset contains all predictors whereas the precipitation dataset contains the first five. The first five predictors are forecast fields from the weather model. The original resolution is 2.5 km, but these have been downscaled using nearest neighbours to a 1 km grid. The "2m temperature bias previous day" predictor is the difference between the weather model's forecasted air temperature and the target field for the previous day at the same time of day. "2m temperature bias at initialization time" is the error of the weather model at initialization time. These predictors have been shown to be important predictors of forecast error in the future (Nipen et al. 2020).

The target field is a gridded best estimate of 2m temperature (units °C) and hourly precipitation (units mm), and is produced by a combination of observation data sources and gridded model data. It is provided on the same grid as the predictors.

Dataset structure:

Data is stored as one forecast run (record) per NetCDF file and each file includes predictors and the target fields. For a given file, the predictor field has 3 dimensions (x, y, and predictor). Dimensions x and y describe the horizontal space in the Lambert conformal projection and have bounds of -11.76 to 41.76 and 52.30 to 73.86 in the longitudinal and latitudinal dimensions respectively. To reasonable approximation this horizontal grid can be considered equispaced and thus suitable for image-based machine learning approaches. More advanced approaches might incorporate more knowledge of the grid structure. The predictor dimension combines predictors and their values for different forecast lead times. With the exception of the "2m temperature bias at initialization time" predictor, all predictors have values for each lead time. The predictor dimension is a concatenation of all predictors for different lead times.

The target field has 3 dimensions (x, y, and leadtime). Additionally, the dataset contains static metadata variables, such as the latitudes, longitudes, and altitudes of the gridpoints of the 1 km grid; the lead times in hours; and the name of the predictor and its units that each index in the predictor dimension corresponds to.

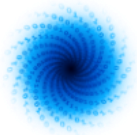
Two tiers will be provided, with the following specifications:

Metric	Tier 1	Tier 2
Data size	5 GB	10 TB
Grid size	128x128	2321x1796
Spatial resolution	1 km	1 km
Lead times	8 (0, 6, 12, … 42)	61 (0, 1, 2, … 60)
Time period	2017-2020	2017-2020
Number of predictors	7	7
Predictor size (time, y, x, predictors)	1457 * 128 * 128 * 49	1457 * 2321 * 1796 * 428
Target size (time, y, x, leadtimes)	1457 * 128 * 128 * 8	1457 * 2321 * 1796 * 61

Table 2: A1 datasets for 2m temperature and hourly precipitation forecast

Tier 1 is a subset of Tier 2, containing a much smaller spatial extent and fewer lead times, and is small enough for testing on a laptop.

Loss function: For evaluating the temperature forecasts, use the quantile score (Koenker and Bassett, 1978; Gneiting, T., and A. E. Raftery, 2007) given by:



$$S_\tau(u) = \begin{cases} u(\tau - 1), & u < 0 \\ u\,\tau\,, & u \geq 0 \end{cases}$$

where τ is quantile level and u is the difference between the quantile forecast and the target. In this application, the quantile score for the three predicted quantiles (10%, 50%, and 90%) are added together to a final score. For precipitation, a new score that better takes into account desired temporal aggregation properties of the scenario will be developed.

Typical machine learning solutions: The current solution for temperature used on Yr is a simple linear regression between a subset of the parameters ("Forecasted 2m temperature", "2m temperature bias previous day", and "2m temperature bias at initialization time"). A simple vanilla solution using all predictors is demonstrated in the MAELSTROM-Yr Jupyter notebook.

3.2 A2: Incorporate social media data into the prediction framework

Motivation and description

This application aims to harvest weather related data from social networks and to process it into a qualitative, geo-localized information stream in near real time.

For this, one can use three types of information:

1. citizen-sensing (passive, indirectly self-reported or unsolicited information)
2. crowd-sensing (active contributions)
3. tweeting sensors (local weather sensors which are using Twitter as a low-cost infrastructure)

The plan for this application is to first develop text mining and processing tools for automated classifications of unstructured text. After that, automated classifiers for detection and severity of predefined climate indices need to be defined.

The information from the three types of sensing described above can be transformed into qualitative maps representing different climate/weather indices while focusing on dominant infrastructures in Europe (e.g. airports). This data can be compared to weather predictions of ECMWF at the same locations but for different lead times using machine learning. Finally, the IFS forecasts would be bias corrected locally to the needs of the end-users.

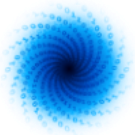
Input and output data

Weather forecast data

For information about the available weather data see the description for application 6 as these two applications have a lot of overlap in their required weather data. To avoid data doubling those two applications will thus use the same data.

Twitter data

As stated in section 2.2.3, we were not able to collect twitter for this deliverable so far, hence no description of the data is possible. It will be added as soon as the data is available.



Loss function and quality measures

As one of the first steps the work for this application will be concerned with the classification of unstructured text data with regard to extreme weather situations. Appropriate loss functions could thus be the Cross Entropy Loss or the Hinge Loss.

Details will be added when data is available, and the example machine learning notebook can be provided.

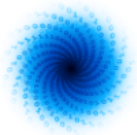
3.3 A3: Build neural network emulators to speed-up weather forecast models and data assimilation

Motivation and description: Due to limited resolution, not every important process can be represented explicitly within weather and climate models. The processes that cannot be resolved are represented by so-called parametrisation schemes that mimic sub-grid-scale dynamics based on the physical fields that are resolved. Parametrisation schemes for models of the atmosphere represent radiation, clouds and convection, turbulence, and gravity wave processes and form a significant part of weather and climate models both in terms of the complexity of the model code and computational cost.

It has been shown that parametrisation schemes can be emulated by neural networks and that these emulators are often much faster when compared to the conventional parametrisation scheme (for example Chevalier et al. 1998, Brenowitz et al. 2018, O'Gorman and Dwyer 2018, Rasp et al. 2018, Chantry et al. 2021). In this task, the neural networks learn a mapping between input and output fields of the conventional schemes. Furthermore, neural network emulators are easier to port to heterogeneous hardware due to the availability of machine learning libraries such as Tensorflow or PyTorch. In contrast, as conventional parametrisation schemes, typically written in Fortran code, are difficult to port to accelerators such as Graphical Processing Units (GPUs). Finally, the neural network emulators can also be used to generate tangent linear and adjoint model code that is required for 4DVar data assimilation -- as used at ECMWF -- but typically laborious to develop for parametrisation schemes (Hatfield et al. 2021).

Physical parametrisation schemes are typically solved in a columnar approach, i.e. a column of input data within an atmosphere model is used to predict the impact of the physical process for each vertical level in the column. This makes the problem a one-dimensional problem that is solved for each horizontal grid-point on a global grid. The physical parametrisation schemes provide contributions to the "tendencies'' that are used to update the prognostic model fields.

The first version of the benchmark dataset of A3 will provide input and output data of the ecRad Tripleclouds radiation scheme (Hogan and Bozzo 2018) as it is used for operational weather predictions at ECMWF. The dataset can therefore be used to learn the Tripleclouds radiation scheme. However, as the ecRad scheme is available as open source software (https://github.com/ecmwf/ecrad), the dataset can also be used to generate output data for more costly versions of the ecRad, such as the SPARTACUS scheme that is taking the three dimensional cloud effects into account but is currently too expensive for operational weather predictions (Hogan



et al. 2016, Meyer et al. 2021). For this, ecRad could be used to generate new output fields for training that are based on SPARTACUS rather than Tripleclouds.

Additionally, the dataset also includes input and output pairs of the gravity wave drag parametrisation scheme. The emulation of the gravity wave drag scheme was already investigated in Chantry et al. 2021.

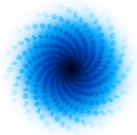
Input and output data: All data is diagnosed from simulations of the weather forecast model of ECMWF -- the so-called Integrated Forecasting System (IFS). We run forecasts with the IFS model regularly saving the global input and output data of the two parametrisation schemes. For the radiation parametrisation this data is computed on a 40km grid (TL511), resulting in 271,360 samples of grid columns for each timestep. Vertical columns can be treated independently but may be batched together at training or inference time. The IFS model is run every 30 days, for 30 days, saving the radiation inputs and outputs every 25 hours (model 125 timesteps). Our training dataset comprises one year (2020), totalling 271,360 (grid points) x 29 (timesteps) x 13 (forecasts) = 102,302,720 training columns. An additional 31,477,760 columns are provided for testing from four forecasts in 2019. For the emulation of the non-orographic gravity wave drag scheme, the details can be found in Chantry et al. 2021.

The input data consists of physical fields that are passed into the radiation or gravity-wave-drag schemes within simulations (see Tables 1 & 2). For radiation, the output fields are fluxes of short and long wave radiation between vertical levels as well as at the top and bottom of the atmosphere given in [W m^{-2}]. Next to the real fluxes between levels, the direct radiation -- that is calculating the radiation in the absence of clouds -- is also calculated. However, the quantity that is most important for the quality of IFS simulations is the heating rate, which is calculated from the fluxes as the difference of incoming and outgoing radiation fluxes per vertical level and at the surface. The heating rate is calculated as the pressure derivative of net heating fluxes,

$$HR = -\frac{g}{c_p}\frac{d(Flux_{up} - Flux_{down})}{dp}$$

where g is the gravitational force and c_p is the specific heat of dry air, taken to be constant at 1004 J kg^{-1} K^{-1}. As such, the heating rate will be weighted by a term that depends on pressure which varies by several orders of magnitude between the top and bottom of the atmospheric column.

Field Name	Location	Unit
Specific humidity	Full level	1
Ozone, CO2, N2O, CH4, O2, CFC11, CFC12, HCFC22, CC14 volume mass mixing ratio	Full level	1
CO2 volume mixing ratio	Full level	1
Cloud fraction	Full level	1
Gridbox-mean liquid water mixing ratio	Full level	1
Gridbox-mean ice water mixing ratio	Full level	1
Ice effective radius	Full level	1
Cloud overlap parameter	Full level	1
Fractional std of cloud optical depth	Full level	1
Inverse of cloud effective horizontal size	Full level	m^{-1}
12 types of aerosol mass mixing ratios	Full level	



Pressure	Half level	Pa
Temperature	Half level	K
Solar irradiance	Surface	W/m^2
Longitude and latitude	Surface	Degrees
Skin temperature	Surface	K
Cosine of solar zenith angle	Surface	1
Six bands of short-wave albedo	Surface	1
Six bands of direct short-wave albedo	Surface	1
Two long-wave emissivity	Surface	1

Table 3: A3 predictors for the radiation challenge. There are 137 full model levels, 138 "half levels" that are defined between the full levels, at the top of the atmosphere, and at the surface, and surface inputs which are scalars.

For the gravity wave drag emulation, the input fields are physical fields from IFS simulations (see Table 4). The output fields are the influence of gravity waves on the tendencies of wind in zonal and meridional direction for each vertical level. Both inputs and outputs have been preprocessed into single columns in the order described by the table and text respectively.

Field Name	Location	Unit
Zonal velocity	Full level	m/s
Meridional velocity	Full level	m/s
Temperature	Full Level	K
Surface pressure	Surface	Pa
Surface geopotential	Surface	m^2s-2

Table 4: A3 predictors for the gravity wave drag challenge.

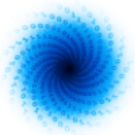
Data structures:

Dataset tiers will be created by providing different time windows of data.

- Tier-1 will be a short time window to show the data structure summing up to ~3GB of data.
- Tier-2 will cover 2 years, allowing for a complete year as training data with a second year for testing/validation data summing up to 3TB of data.

For Tier-1, the data will be provided in NetCDF format. For Tier-2, the data will be provided in NetCDF with a TFRecord version being provided for the radiation dataset. Preliminary tests found the shuffled read performance of TFRecord was vastly superior to that of NetCDF (even when using the dask package; https://dask.org/), however NetCDF maintains greater data structure so might provide added value for data analysis. Both tiers and formats will be easily accessed using CliMetLab. For the NetCDF data, no normalisation is provided. For the TFRecord loader, a mean & standard deviation normalisation across the training data (2020) has been calculated, but users are free to use a different normalisation approach.

Loss function and quality measures: As a regression problem, the mean-squared-error is the main loss function that will be used. The quality can also be assessed by visual inspection of the vertical flux or heating rate profiles, in particular when focussing on the worst profile during testing. To define acceptable accuracy is challenging. The ultimate test is to check whether the emulator is causing any degradation to the accuracy of the whole IFS forecast coupled to the emulator, but this



is only possible to assess in online testing that are expensive as they require to run the full IFS. Previous modelling attempts have found that smoothness of the heating rate profiles is important for stability when connected to the IFS model. As a result, we recommend users use the minimal outputs approach, and predict the heating rates and three boundary fluxes for each wavelength (an example of this is provided in the example notebook). The alternate approach, to predict the columnar fluxes and use the equation above to derive the heating rate would have to be extremely careful to incorporate this smoothness into the loss function. In a recent paper attempting to learn the “3D” effects of radiative heating the authors established the mean differences between the TripleClouds scheme and Spartacus schemes of ecRad (Meyer et al 2021). They found a mean absolute difference of O(1 W/m^2) for the boundary fluxes and O(10^{-7}K/s) for the heating rates. These can act as a guideline for researchers looking to establish acceptable errors in any machine learning emulator.

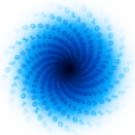
Typical machine learning solutions: Previous studies have often used very simple architectures, e.g. fully-connected neural networks, sometimes with bespoke final layers to ensure conservation properties (e.g. a sum over the column of outputs). Our current solution for gravity wave drag uses five hidden layers with O(10^5) degrees of freedom in the system, but other pieces of physics might require more complex solutions. There is certainly still room for more intelligent network design to improve efficiency. The parametrisation emulation problem tries to find a network that is as accurate as possible while generating the lowest computational cost possible. It is useful to understand how errors of the neural network emulators reduce if the cost of the neural networks increases. Once this has been established, a subset of the "best" models, i.e. optima in the cost/loss space, can be tested in online simulations that are coupled to the IFS model.

Inference timing benchmark: The cost for inference of the existing scheme can be measured, but is currently run on a CPU architecture, making fair comparisons to the machine learning emulator challenging (Chantry et al. 2021). For the radiation dataset, 16960 columns take approximately 60 seconds on a single core of an Intel(R) Core(TM) i7-9700 CPU @ 3.00GHz, equivalent to 3.5ms per profile.

3.4 A4: Improve ensemble predictions in forecast post-processing

Motivation and description: To be useful, weather forecasts are not only required to provide the most likely future scenario, but are equally required to produce the probability of specific weather events. To get estimates of probability distributions for predictions, weather forecast centres typically run ensemble simulations that perform a number of simulations in parallel, with each simulation being perturbed by either a change in initial conditions and/or by adding a stochastic forcing to the model (Berner et al. 2017).

The ENS10 dataset is designed to help the development of machine learning tools to improve ensemble predictions via post-processing. It consists of the model output data of ECMWF "hindcast" experiments. These are ensemble forecasts with 10 ensemble members that are spread over 20 years (1998-2017) with two forecasts per week.



Input and output data: The dataset contains all 10 ensemble members on a 0.5-degree latitude/longitude grids at forecast lead times 0/24/48 hours. Parameters in the dataset are listed below.

Field Name	Location	Unit
Sea surface temperature	Surface	K
Total Column Water	Surface	kg/m^2
Total column water vapour	Surface	kg/m^2
Convective precipitation	Surface	m
Mean sea level pressure	Surface	Pa
Total cloud cover	Surface	1
10m U wind component	Surface	m/s
10m V wind component	Surface	m/s
2m temperature	Surface	K
Total precipitation	Surface	m
Skin temperature at the surface	Surface	K
U wind component	10/50/100/200/300/400/500/700/850/925/1000 hPa	m/s
V wind component	10/50/100/200/300/400/500/700/850/925/1000 hPa	m/s
Geopotential	10/50/100/200/300/400/500/700/850/925/1000 hPa	m^2/s^2
Temperature	10/50/100/200/300/400/500/700/850/925/1000 hPa	K
Specific humidity	10/50/100/200/300/400/500/700/850/925/1000 hPa	1
Vertical velocity	10/50/100/200/300/400/500/700/850/925/1000 hPa	Pa/s
Divergence	10/50/100/200/300/400/500/700/850/925/1000 hPa	1/s

Table 5: A4 predictors and predictands.

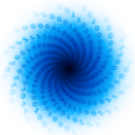
A tier-1 version of the dataset (ENS5-mini) is additionally provided. It is a 9.5 GB subset of ENS10, which only contains the pressure level data and parameters (i.e., surface data is omitted). This dataset is spatially cropped over Europe (40-60° N, 0-40° E), temporally cropped to the first ten years (1998-2007), contains only 0h/24h forecast lead times, and comprises a subset of 5 ensemble members.

Dataset structure: The dataset is grouped by day-of-year (i.e., a single date contains all 20 years of predictions), where each date contains three steps: 0, 24 and 48 hour lead time. Thus, files contain three days at a time.

In every file, there are 6 dimensions of data (in this order): ensemble member, time (year offset from 1998), forecast lead time (0h, 24h, 48h), pressure level, latitude, and longitude.

Loss functions and quality measures: To gain an understanding of the forecast skill of the combined ensemble predictions, the Continuous Ranked Probability Score (CRPS) measure (Hersbach 2000) is used. CRPS is the integral of the square of the difference between the Cumulative Distribution Function of the probabilistic predictions F and the ground truth y, as shown in the following formula:

$$\mathrm{CRPS}(F, y) = \int_{-\infty}^{\infty} [F(x) - 1_{x>y}]^2 dx$$



Other loss functions that can be used are Mean Squared Error (MSE), latitude-weighted MSE variants, or Structural Similarity (SSIM, Wang et al. 2004).

Typical machine learning solutions: Methods such as Ensemble Model Output Statistics (EMOS) (Gneiting et al. 2005) and Bayesian Model Averaging (BMA) (Raftery et al. 2005) currently allow for improvements of the raw ensemble forecast skill. Hamill and Whitaker (2007) show initial explorations of those techniques on re-forecast datasets, also used in our work, for temperature at 850 hPa (T850) and geopotential at 500 hPa (Z500). Advances in neural networks have only recently reached the field of ensemble models in weather forecasting, focusing on its application to specific weather stations (Rasp & Lerch 2018). We expand on these works by applying DNNs on improving the forecast skill for global predictions, specifically extreme weather forecasts, while reducing their computational cost.

3.5 A5: Improve local weather predictions in forecast post-processing

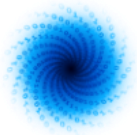
Motivation and Description:

The benefits of accurate weather predictions are far-reaching and range from direct economic revenues, e.g. in agriculture or the renewable energy sector, to the prevention of socioeconomic losses due to high-impact weather. While these benefits outweigh by far the expense to sustain a global observation network of the atmosphere and to run numerical weather prediction (NWP) models (Lazo et al., 2009 and Bauer et al., 2015), there is still a gap between the capability of contemporary NWP models and the economic requirements on the spatio-temporal resolution of such forecasts. Global and regional models operate nowadays with grid spacings in the $\mathcal{O}$(1-10 km). At the same time accurate predictions are demanded at even finer spatial scales ($\mathcal{O}$(100m-1km)).

To circumvent the computational burden and shortcomings of running numerical models at higher and higher spatial resolution (e.g. applicability of parameterization schemes), statistical downscaling methods have been developed in the meteorological domain over the last three decades. These methods map the model output depicting the atmospheric state at larger scales to a tailor-made prediction at local scale. Following Wilby and Wigley, 1997, classical statistical downscaling methods can be categorized into regression methods, weather typing schemes and stochastic weather generators. Extensive reviews on these are provided e.g. in Wilby et al., 2004, Maraun et al., 2010, Maraun and Widmann, 2018.
Meanwhile, the application of neural networks (e.g. Liu et al, 2008) and of multi-objective genetic programming (Zerenner, 2017) have also proven success in the domain of statistical downscaling.

Over the last few years, the meteorological domain has started to exploit sophisticated deep learning techniques to enhance the spatio-temporal resolution of weather predictions. Inspired by the success of sophisticated neural networks for generating super-resolution images in computer vision (e.g. Mahapatra et al. 2019, Wang et al. 2019), first studies try to transfer these techniques to the meteorological domain.

In AP5, deep neural network architectures for super-resolution from computer vision are adopted and fine-tuned for statistical downscaling in meteorology. As a starting point, downscaling of 2m temperature which inhibits a high spatio-temporal variability on different scales is targeted. With



increasing complexity of the developed network architecture, downscaling of other meteorological variables will be probed as well.

Input and output data:

At the first stage, a real downscaling application is imitated by coarsening the forecasts of an operational NWP model and training the neural network to recover the fine-scaled information which gets lost during the coarsening procedure. Here, we choose the IFS HRES model that is run operationally at ECMWF. To minimize the inclusion of model forecast errors, analysis data (00 and 12 UTC) and data from the near-short term forecast range (up to 12 hours) will be used allowing us to retrieve hourly gridded atmospheric data for the whole day.

The model data is given onto a regular latitude-longitude grid with $\Delta x = 0.1°$ which constitutes the objective of the downscaling application at hand. Since we focus on downscaling over complex, heterogeneous terrain, we choose Central Europe as the target region and slice the data to a domain consisting of 128x96 grid points in zonal and meridional direction, respectively. With this, a domain between 4°E to 16.7°E and 45°N to 54.5°N is covered.

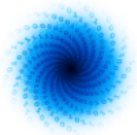
The above-mentioned coarsening of the model data is then undertaken by performing conservative remapping onto a 0.8°-grid corresponding to a resolution reduction factor of eight. While this step removes the fine-scaled information from the data, the model architecture chosen here (see below) requires that input and output data are provided on the same grid. Thus, the coarsened model data is remapped back on the target resolution ($\Delta x = 0.1°$) via bi-linear interpolation. It is noted that the final step to produce the input data does not recover any fine-scaled information at spatial resolution smaller than 0.8°.

All remapping steps are thereby undertaken with special consideration on the meteorological quantity at hand. In our case, the application targets downscaling of the 2m temperature $T_{2m} = T(z_1 = 2m)$. To ensure energetic consistency during the remapping procedure for this exemplary quantity, the dry static energy s is computed beforehand with the help of

$$s = c_{pd} T(z_1 = 2m) + g(z_{sfc} + z_1)$$

where c_{pd} denotes the specific heat capacity of dry air, $g$is the gravitational constant and z_{sfc}stands for the surface elevation.
The dry static energy is approximately conserved for dry-adiabatic processes and therefore ensures that no extra energy is added due to the remapping. Provided that the surface elevation z_{sfc} is also remapped, T_{2m} can be computed from the remapped dry static energy by rearranging the above equation.

At a later stage, the ERA5-reanalysis data (grid spacing 0.3°) will be used as input data to approach a real downscaling approach. Since the underlying numerical model constitutes a frozen version of the IFS model run at coarser resolution ($\Delta x \simeq 25$km), this data is considered to be a proper candidate. Besides, other consistent modeling systems such as the NWP forecast chain operating at the German Weather Service DWD may be considered as well.



Similar to the other applications in MAELSTROM, A5 will provide a two tier datasets. Tier-1 dataset is limited in the number of variables and time steps offered and thus is designed to be fed into a rather simple DL model architecture.

In particular, the 2m temperature and surface elevation obtained from the IFS HRES model at its analysis time steps (00 and 12 UTC) between 2016 and 2020 are used. Since no further information on the atmospheric state is provided, a seasonal filtering for the summer half of the year defined between April and September is applied. Thus, the total dataset comprises 915 samples for 00 UTC and 12 UTC, respectively. Note, that the near-surface layer of the planetary boundary layer is typically stably stratified (well mixed) at 00 UTC (12 UTC), easing the downscaling task for T2m when separated models are trained for analysis time step.

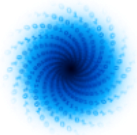
Tier-2 dataset will complement the first data collection by adding the forecasted fields for the first 11 hours and by including more informative meteorological variables. While a detailed selection of predictors is not available yet, proper examples are the 850hPa temperature, the cloud cover and the sensible and latent enthalpy fluxes at the surface. A more complete overview on the potential predictors is provided in Table 6.

Field Name	Location	Unit
2m temperature (input/output)	2m	K
Surface elevation (input/[output])	surface	m
Temperature on pressure levels (input)	850 hPa, 925 hPa	K
Geopotential height (input)	500hPa, 850 hPa, 925 hPa	m^2/s^2
(u,v)-wind components (input)	10m, 850 hPa, 925 hPa	m/s
Total cloud cover (input)	-	1
Low, middle and high cloud cover (input)	-	1
Boundary layer height (input)	-	m
Large-scale precipitation (input)	surface	m
Convective precipitation (input)	surface	m
Sensible heat flux (input)	surface	J/m^2
Latent heat flux (input)	surface	J/m^2
Volumetric soil water (input)	Ground layer 1+2	m^3/m^3
Soil type (input)	surface	-
Leaf area index high vegetation (input)	surface	m^2/m^2
Leaf area index low vegetation (input)	surface	m^2/m^2

Table 6: A5 input fields for the statistical downscaling of 2m temperature.
Data variables used in the tier-1 dataset are given in black, while potential predictors of the tier-2 dataset are indicated in grey. Note that the predictors can be obtained from the IFS HRES and from the ERA5-reanalysis dataset. The 2m temperature also serves as the objective variable (predictand). Similar to Sha et al., 2020a, the surface elevation may serve as an objective to enable transfer learning to other target regions (see exemplary application of the tier-1 dataset).

Data structures:

As mentioned above, dataset tiers will be created by providing different time windows of data and different input variables



- Tier-1 will use only two input variables from the analysis times of the IFS HRES model (00 UTC and 12 UTC). The data is limited to the summer season (April-September) and is provided in netCDF-files, each carrying one month of data. One data file is 24 MB large, summing up to 687 MB in total.
- Tier-2 will cover the whole year and the whole day at an hourly frequency. Since the number of predictor variables will be increased, this dataset will be at least larger by a factor 100 compared to Tier-1. Similar to Tier-1, data will be stored in netCDF-files, but as daily files. Additionally, TFrecords will be provided. At a later stage, other model data such as ERA-5 will be added increasing the data volume to terabyte scale.

Both datasets will be stored and provided via the ECMWF cloud. While Tier-1 allows for a calculation of the normalization parameters with all data in memory, these parameters will be shipped along with the Tier-2 dataset. As a default, z-score normalization will be applied, but other normalization techniques such as min-max scaling will be enabled (and tested) as well. Tier-2 is also distributed in TFrecords-format for performance reasons since this format allows for optimized data streaming into DL model architectures built with TensorFlow.

Typical ML solutions:

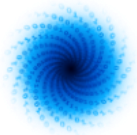
A first comprehensive comparison study by Baño-Medina et al., 2020 examined basic convolutional neural networks for a few meteorological fields. While they conclude that deep learning techniques constitute a promising tool for statistical downscaling, more sophisticated deep convolutional networks have already been applied as well.

The U-net architecture was originally proposed for biomedical image segmentation (Ronneberger et al., 2015). However, since it is fully based on convolutional layers, it is capable of extracting spatial features. Therefore, it has become a popular choice in super resolution tasks of computer vision (see Yao et al., 2018, Lu and Chen, 2019, and Wang, 2019).

The applicability of the U-net architecture for statistical downscaling of meteorological data has been demonstrated in Sha et al., 2020(a,b). Since their particular architecture for daily maximum/ minimum temperature comprises 3.5 million trainable parameters, it constitutes a relatively light-weighted neural network. Therefore, this architecture is chosen as the starting point in AP5 and provided as an example architecture in scope of the tier-1 dataset.

At later stages, the U-net will be further developed by incorporating more model variables. Instead of predicting the diurnal minima and maxima of 2m temperature, an hourly downscaling product will be targeted requiring the depiction of the state of the planetary boundary layer. Thus, the size of the deployed U-net will grow, while the standard L1-loss will still be a reasonable choice for optimization.

However, with increasing intricacy of the downscaling task, it is expected that the U-net will also suffer from shortcomings due to the applied pixel-wise loss function. To circumvent this, it will be complemented with a Generative Adversarial Network architecture, abbreviated by U-net-GAN (see e.g. Wang, 2021). The adversarial loss applied with GANs encourages the model to learn the underlying statistical properties of the data. However, stabilizing the training iteration and



preventing the model from mode collapse are common issues met with GAN. To mitigate the latter, the first Wasserstein distance as an additional loss term is an appealing candidate (Arjovsky, 2017).

Since the near-surface temperature exhibits a clear diurnal cycle (see Zhang et al., 2004, Holtslag et al., 2013), recurrent units will be added to the model architecture in the future as well. With this, the hourly 2m temperature over one day is interpreted as a sequence where recurrent cells such as Long-Short-Term Memory cells are considered to enhance the temporal coherence of the downscaled product.

Note that more sophisticated DL model architectures also enable us to test the downscaling approach on other, more complex meteorological fields such as precipitation. Besides, the GAN part enables the generation of ensembles.

Typical training times:

The basic U-net architecture shipped along with the tier-1 dataset can be trained within 5 seconds per epoch using a Nvidia P100 GPU. Using 30 epochs to optimize the model parameters, the total training time is well below three minutes. However, especially the introduction of recurrent layers increases the computational costs, so that a total training time of several hours to days is expected at later stages.

3.6 A6: Provide bespoke weather forecasts to support energy production in Europe

Motivation and description

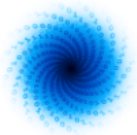
One of the most important challenges to mitigate climate change is to increase the generation of renewable energies. In times of weather situations with low wind and/or solar radiation the production of wind and solar power production has to be complemented by energy production from e.g. biogas plants and storage capacities. An optimal efficiency throughout all energy providers is important here in order to allow for a large market share of renewable energy. This requires accurate forecasts for energy generation that rely heavily on weather predictions, local measurements, and real-time production data from wind turbines and solar panels. However, for excellent forecasts, weather predictions need to be available for the exact location of solar panels or wind turbines. This is not possible to achieve with conventional weather prediction models, even with exascale HPC infrastructure, since the resolution is not high enough to picture local conditions such as topography or land-use.

This application aims to significantly improve predictions of power production from renewable energy sources in order to optimise the usability of renewable energy. Possible users of the improved predictions are power producers, trading companies, grid operators and even whole countries.

Machine learning will be used to fuse the information of local conditions (measurements of local energy production and weather) and numerical weather predictions to learn to predict the energy production at local sites in the future.

Input and output data

D1.1 First version of datasets and cost functions to develop machine learning solutions for A1-A6



Weather forecast data

The used weather forecast data has been derived from the ECMWF-IFS weather forecasting model.

It is available for the area of Europe (see Table 7 for the precise definition) with a spatial resolution of 0.1° in both horizontal directions in space. The temporal resolution is 1 hour with a new model initialised twice a day (at 00:00 and 12:00 UTC). The model simulations are run for several days but the data that is provided here has a maximum lead time of 48 hours.

Submitted for this deliverable will be a time period of six months (January to June) in 2019, however, the aim for the tier-2 dataset will be to gather at least 4.5 years of data (January 2017 to June 2021).

The data is stored in NetCDF format with each file containing one model run for the whole area.

Covered area	35°N to 70°N latitude 25°W to 30°E longitude
Horizontal spatial resolution	0.1 ° x 0.1 °
Temporal resolution	1 h
Frequency of model runs	Twice a day (00:00 and 12:00 UTC)
Length of runs	48 h
Time period	January to June 2019 (Tier-1) January 2017 to June 2021 (Tier-2)

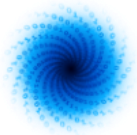
Table 7: A6 data description.

In the vertical direction – apart from the surface parameters -- the meteorological fields are available both on model levels and on pressure levels.

Available model levels are the lowest ten levels (128 – 137) of the ECMWF-IFS weather forecasting model. For pressure levels, there are 500hPa, 800hPa, 925hPa, 950hPa and 1000hPa available.

The fields per level type can be found in Table 8. Fields labelled “accumulated” are provided as accumulated sums over the length of the forecast, e.g. direct solar radiation at hour 12 of the forecast will include the solar radiation for the first 12 hours. Hourly values can be derived by subtracting neighbouring time slices, e.g. hour 12 minus hour 11.

Level type	Field name	Accumulated
Model Levels	Temperature [K] Specific Humidity [kg/kg] U-wind [m/s] V-wind [m/s]	
Pressure Levels	Temperature [K] Relative Humidity [%] Geopotential [m²/s²] U-wind [m/s] V-wind [m/s]	
Surface Level	Surface solar radiation downwards [J/m²] Direct solar radiation [J/m²] Total sky direct solar radiation at surface [J/m²] Geopotential [m²/s²]	Yes Yes Yes



	Surface pressure [Pa] Height of convective cloud top [m] Total cloud cover [1] Low cloud cover [1] Medium cloud cover [1] High cloud cover [1] Visibility [m] Total precipitation [m] Precipitation type [1] Averaged total lightning flash density in the last hour [1/km²d]	Yes

Table 8: A6 weather model predictors.

Power production data

The second type of input data is power production data from 45 wind energy plants in Germany. They are provided like the weather data as NetCDF with one file per plant. The available fields are listed in Table 9 and a short description of the data is provided.

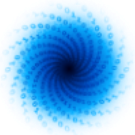
Field name	Short description
production	Mean power production in kW
production_min	Minimal power production in kW
production_max	Maximal power production in kW
wind_sped	Mean wind speed in m/s
wind_speed_min	Minimal wind speed in m/s
wind_speed_max	Maximal wind speed in m/s
rotor	Mean rotor speed in rounds per minute
rotor_min	Minimal rotor speed in rounds per minute
rotor_max	Maximal rotor speed in rounds per minute
errornumber	Error number of the turbine
eisman_regulation_edia	Einspeisemanagement[1] inferred from the power provider E.DIS
eisman_regulation_logs	Einspeisemanagement[1] inferred from the turbine logs
status	Status of the turbine (1: running, 0: stopped, unknown)

Table 9: A6 wind turbine fields. All values apply to the last 10 minutes as this is the temporal resolution.

All information that is saved in the headers is listed in the following:

- Convention (CF 1.6)
- Title
- Summary (short description of the data)
- Institution (4Cast)
- Source (on-site measurements)
- Comment (data provided by the company Notus Energy)

[1] artificial reduction of produced power to ensure net stability



- Name of the wind plant
- Power rating/nominal power
- Hub height
- Night regulation (only if the plant is regulated at night)

Also, information about longitude, latitude and height above sea level are provided for each time series.

The temporal resolution of these data is 10 min. The data availability varies across the sites, with a couple of plants providing data across the entire 4.5 year timeframe but most producing less (the shortest being six months).

The field “production” is used as the target of the machine learning application with the possibility to remove implausible values using the other fields, e.g. “eisman_regulation_edis” and “status”.

Loss function and quality measures

There are a multitude of possible loss functions for this regression problem. We mainly use the normalized mean absolute error (nMAE), where the normalization is done according to the nominal power of the respective wind turbine. The MAE is computed as follows:

$$MAE = \frac{1}{n}\sum_{n=1}^{n} |\hat{Y}_i - Y_i|$$

with n being the number of predicted values, $\hat{Y}_i$the predicted values and $Y_i$the measured values. The nMAE is then:

$$nMAE = \frac{MAE}{Y_{nom}},$$

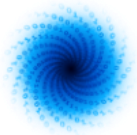
Y_{nom}being the nominal (maximal possible) power.

Additionally, measures to quantify variations of the error (like the variance of nMAE) might become interesting with regard to the quality of the forecast during different conditions (e.g. large weather situation).

Machine learning solution

The plan for this application is to use a multitude of machine learning approaches like neural networks and explainable machine learning but also their combination. Different methods of feature engineering will have a great effect on the quality of prediction as well, e.g. one could just use the weather prediction of one specific grid cell of the numerical weather model or use a greater area around a wind turbine and study the effect on the forecast. Another approach would be to develop a classification of weather for the particular country or for the whole Europe and use this as a feature for the machine learning problem.

The dataset provided in this deliverable will serve as a means to tackle all these different approaches and possibly more.



4 Data access and Jupyter notebooks

Data access through python is provided using CliMetLab plugins. Example jupyter notebooks have also been prepared to explore the data and demonstrate simple machine learning solutions for each problem.

4.1 CliMetLab

CliMetLab manages the downloading and loading of data, for a variety of datasets, dubbed plugins. Plugins have been created for each of the applications, providing data to the user within minimal lines of code. e.g.

```
!pip install climetlab climetlab-maelstrom-radiation
import climetlab as cml
cmlds = cml.load_dataset('maelstrom-radiation')
ds = cmlds.to_xarray()
```

First, using the python package installer tool, pip, we install CliMetLab and the plugin for the A3 radiation dataset. Then we import CliMetLab and load the dataset, which downloads the data on first use. Finally, we can transform the data into an xarray dataset ready for exploration and training. In the table below are the names of the pip repositories and dataset names for each of the applications which can be used in place of the radiation example above.

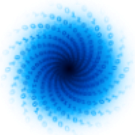
Application	Pip package name	CML dataset name
A1: Postprocessing	climetlab-maelstrom-yr	'maelstrom-yr'
A3: Radiation	climetlab-maelstrom-radiation	'maelstrom-radiation'
NOGWD	climetlab-maelstrom-nogwd	'maelstrom-nogwd'
A4: ENS10	climetlab-maelstrom-ens10	'maelstrom-ens10'
A5: Downscaling	climetlab-maelstrom-downscaling	'maelstrom-downscaling'
A6: Power production	climetlab-maelstrom-power-production	'maelstrom-constants-a-b' 'maelstrom-power-production' 'maelstrom-weather-model-level' 'maelstrom-weather-pressure-level' 'maelstrom-weather-surface-level'

Table 10: MAELSTROM CliMetLab plugins

4.2 Jupyter notebooks

Jupyter notebooks have been created to explore the datasets and demonstrate simple machine learning solutions to act as first benchmarks. These can be accessed through the links in the table below. Each will begin with using CliMetLab to download and load the data and is the recommended place to begin exploring each application.

Application	Jupyter notebook link
A1: Postprocessing	https://github.com/metno/maelstrom-yr/blob/main/notebooks/demo_yr.ipynb
A3: Radiation	https://git.ecmwf.int/projects/MLFET/repos/maelstrom-radiation/browse/notebooks/demo_radiation.ipynb
NOGWD	https://git.ecmwf.int/projects/MLFET/repos/maelstrom-nogwd/browse/notebooks/demo_nogwd.ipynb



A4: ENS10	https://github.com/spcl/climetlab-maelstrom-ens10/blob/main/notebooks/demo_ens10.ipynb https://github.com/spcl/climetlab-maelstrom-ens10/blob/main/notebooks/demo_toy.ipynb
A5: Downscaling	https://git.ecmwf.int/projects/MLFET/repos/maelstrom-downscaling-ap5/browse/notebooks/demo_downscaling_dataset.ipynb
A6: Power production	https://github.com/faemmi/climetlab-plugin-a6/blob/main/notebooks/demo_maelstrom_production_forecasts.ipynb

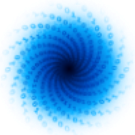
Table 11: Example Jupyter notebooks.

4.3 Code Repositories

The repositories for these plugins and notebooks can be found at the links below.

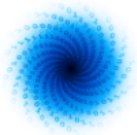
Application	URL
A1: Postprocessing	https://github.com/metno/maelstrom-yr
A3: Radiation NOGWD	https://git.ecmwf.int/projects/MLFET/repos/maelstrom-radiation https://git.ecmwf.int/projects/MLFET/repos/maelstrom-nogwd
A4: ENS10	https://github.com/spcl/climetlab-maelstrom-ens10
A5: Downscaling	https://git.ecmwf.int/projects/MLFET/repos/maelstrom-downscaling-ap5
A6: Power production	https://github.com/faemmi/climetlab-plugin-a6

Table 12: Plugin code repositories.



5 Conclusion

MAELSTROM deliverable 1.1 provides initial datasets for the six applications in the project. Crucially, it also provides easy and consistent methods to download and load each of the datasets into Python. The datasets that are published together with this deliverable will form the basis for the software and hardware benchmarks of MAELSTROM and the interactions between work package 1, 2, and 3 within the project. However, the datasets will also enable external machine learners to train meaningful machine learning tools that are based on large datasets and to publish results such that they can benefit the application developments within MAELSTROM.



6 References

Brenowitz, N. D., & Bretherton, C. S. (2018). Prognostic validation of a neural network unified physics parameterization. Geophysical Research Letters, 45 (12), 6289–6298.

Chantry M, Hatfield S, Dueben P, Polichtchouk I, Palmer T. Machine learning emulation of gravity wave drag in numerical weather forecasting. Journal of Advances in Modeling Earth Systems. 2020:e2021MS002477.

Chevallier, F., Chéruy, F., Scott, N., & Chédin, A. (1998). A neural network approach for a fast and accurate computation of a longwave radiative budget. Journal of Applied Meteorology, 37 (11), 1385–1397.

Hatfield, Samuel Edward; Chantry, Matthew; Dueben, Peter Dominik; Lopez, Philippe; Geer, Alan Jon: Building tangent-linear and adjoint models for data assimilation with neural networks, DOI:10.1002/essoar.10506310.1, 2021.

Hogan, R. J., & Bozzo, A. (2018). A Flexible and Efficient Radiation Scheme for the ECMWF Model. Journal of Advances in Modeling Earth Systems, 10(8), 1990–2008. https://doi.org/10.1029/2018MS001364

Hogan, R. J., Schäfer, S. A. K., Klinger, C., Chiu, J. C., & Mayer, B. (2016). Representing 3-D cloud radiation effects in two-stream schemes: 2. Matrix formulation and broadband evaluation. Journal of Geophysical Research: Atmospheres, 121(14), 8583–8599. https://doi.org/10.1002/2016JD024875

Lambert, Johann Heinrich. *Notes and Comments on the Composition of Terrestrial and Celestial Maps (1772)*. No. 8. Department of Geography, University of Michigan, 1972.

Meyer, D, Hogan, R. J., Dueben P. D., Shannon L. Mason S. L. (2021) Mason Machine Learning Emulation of 3D Cloud Radiative Effects, https://arxiv.org/abs/2103.11919.

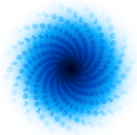
O'Gorman, P. A., & Dwyer, J. G. (2018). Using machine learning to parameterize moist convection: Potential for modeling of climate, climate change, and extreme events. Journal of Advances in Modeling Earth Systems, 10 (10), 2548–2563.

Rasp, S., Pritchard, M. S., & Gentine, P. (2018). Deep learning to represent subgrid processes in climate models. Proceedings of the National Academy of Sciences, 115 (39), 9684–9689.

Berner, J., Achatz, U., Batte, L., Bengtsson, L., De La Camara, A., Christensen, H. M., ... & Yano, J. I. (2017). Stochastic parameterization: Toward a new view of weather and climate models. Bulletin of the American Meteorological Society, 98(3), 565-588.

T. Gneiting, A. E. Raftery, A. H. Westveld, and T. Goldman, "Calibrated probabilistic forecasting using ensemble model output statistics and minimum crps estimation," Monthly Weather Review, vol. 133, no. 5, 2005. [Online]. Available: https://doi.org/10.1175/MWR2904.

A. E. Raftery, T. Gneiting, F. Balabdaoui, and M. Polakowski, "Using bayesian model averaging to calibrate forecast ensembles," Monthly Weather Review, vol. 133, no. 5, 2005. [Online]. Available: https://doi.org/10.1175/MWR2906.1



T. M. Hamill and J. S. Whitaker, “Ensemble calibration of 500-hPa geopotential height and 850-hPa and 2-M temperatures using reforecasts,” Monthly Weather Review, vol. 135, no. 9, 2007. [Online]. Available: https://doi.org/10.1175/MWR3468.1

S. Rasp and S. Lerch, “Neural Networks for Postprocessing Ensemble Weather Forecasts,” Monthly Weather Review, vol. 146, no. 11, Nov 2018.

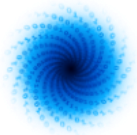
Hersbach, H. (2000). Decomposition of the continuous ranked probability score for ensemble prediction systems. Weather and Forecasting, 15(5), 559-570.

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4), 600-612.

Koenker, R., and J. Bassett, 1978: Regression quantiles. Econometrica, 46, 33–50, https://doi.org/10.2307/1913643.

Nipen, T. N., I. A. Seierstad, C. Lussana, J. Kristiansen, and Ø. Hov, 2020: Adopting Citizen Observations in Operational Weather Prediction. Bull. Amer. Meteor. Soc., 101, E43–E57.

Gneiting, T., and A. E. Raftery, 2007: Strictly proper scoring rules, prediction, and estimation. J. Amer. Stat. Assoc., 102, 359–378, https://doi.org/10.1198/016214506000001437



Document History

Version	Author(s)	Date	Changes
0.1	Peter Dueben and Matthew Chantry (ECMWF)	12/08/2021	Version for review
1.0	Peter Dueben and Matthew Chantry (ECMWF)	31/08/2021	Post internal review

Internal Review History

Internal Reviewers	Date	Comments
Bing Gong (Forschungszentrum Juelich)	27/08/2021	Minor changes
John Bjørnar Bremnes (Norwegian Meterological Institute)	27/08/2021	Minor changes

Estimated Effort Contribution per Partner

Partner	Effort
ECMWF	4 PMs
MetNor	3 PMs
4cast	1.5 PMs
FZJ	2 PMs
Total	**10.5**