Deep Learning for Spatiotemporal Nowcasting

Gabriele Franch

ICT International Doctoral School

University of Trento

Advisor:
Cesare Furlanello

Co-Advisor:
Bruno Lepri

Fondazione Bruno Kessler

February 2021

Abstract

Nowcasting – short-term forecasting using current observations – is a key challenge that human activities have to face on a daily basis. We heavily rely on short-term meteorological predictions in domains such as aviation, agriculture, mobility, and energy production. One of the most important and challenging task for meteorology is the nowcasting of extreme events, whose anticipation is highly needed to mitigate risk in terms of social or economic costs and human safety. The goal of this thesis is to contribute with new machine learning methods to improve the spatio-temporal precision of nowcasting of extreme precipitation events. This work relies on recent advances in deep learning for nowcasting, adding methods targeted at improving nowcasting using ensembles and trained on novel original data resources. Indeed, the new curated multi-year radar scan dataset (TAASRAD19) is introduced that contains more than 350.000 labelled precipitation records over 10 years, to provide a baseline benchmark, and foster reproducibility of machine learning modeling. A TrajGRU model is applied to TAASRAD19, and implemented in an operational prototype. The thesis also introduces a novel method for fast analog search based on manifold learning: the tool leverages the entire dataset history in less than 5 seconds and demonstrates the feasibility of predictive ensembles. In the final part of the

thesis, the new deep learning architecture ConvSG based on stacked generalization is presented, introducing novel concepts for deep learning in precipitation nowcasting: ConvSG is specifically designed to improve predictions of extreme precipitation regimes over published methods, and shows a 117% skill improvement on extreme rain regimes over a single member. Moreover, ConvSG shows superior or equal skills compared to Lagrangian Extrapolation models for all rain rates, achieving a 49% average improvement in predictive skill over extrapolation on the higher precipitation regimes.

Keywords: nowcasting, deep learning, machine learning, precipitation, extreme events

Disclamer

The work presented in this PhD thesis has been funded by a scholarship from TIM - Telecom Italia: results might be covered by Intellectual Property Rights and Patents.

Computing resources were partially funded by a Microsoft Azure Grant AI for Earth (2018-19), assigned to C.F.

Contents

List of Figures

Chapter 1

Introduction

The activity of guessing the near future by observing current situations – Nowcasting – is part of the natural human behavior, but the term itself was first defined by Keith Browning of the UK Met Office in 1981 as *the description of the current state of the weather in detail and the prediction of changes that can be expected on a timescale of a few hours.* Since then, the term has found adoption in fields outside the meteorological domain, such as economics [10] and human mobility [12].

The main application of nowcasting in meteorology is the prediction of events characterized by a rapid evolution, such as thunderstorms, lightnings, precipitations and wind. These phenomena have an important economic and social impact on many human activities, in particular agriculture, aviation, civil protection and energy production. Moreover, climate change has shown to increase frequency and magnitude of extreme weather events, an effect that has been consistently described, also with a geographical dependence [58, 69]. A shift towards a more extreme precipitation climate (the so-called "tropicalisation") has been predicted by models [41] and observed across Europe [28, 62].

1.1 Precipitation Nowcasting

Nowcasting of precipitation [105] is a crucial tool for risk mitigation especially if implemented for water-related hazards [104, 23, 4, 57, 48], and in alert infrastructures. For civil protection purposes and emergency management, an accurate prediction assessment is crucial when extreme events are approaching. The use of weather radar reflectivity sequences is the mainstay of very short-time (up to 2 h) precipitation nowcasting systems, as there is a known direct relationship between rain rate and measured radar reflectivity [66]. Raw reflectivity volumes generated at fixed time steps by radars are processed into several products in the form of sequences of reflectivity maps, that are used as input for prediction models.

1.2 Deep Learning in precipitation nowcasting

The recent application of machine learning and deep learning techniques in nowcasting has produced several advances in the accuracy of precipitation nowcasting systems. Deep learning (DL) models based on recurrent (RNN) and convolutional neural networks (CNN) have shown substantial improvement over nowcasting methods based on Lagrangian extrapolations [75, 8, 107] for quantitative precipitation forecasting (QPF). Shi et al. [83] introduced the application of the convolutional long short-term memory (Conv-LSTM) network architecture with the specific goal of improving precipitation nowcasting, where LSTM is modified using a convolution operator in the state-to-state and input-to-state transitions. Subsequent works introduced dynamic recurrent connections [84] (TrajGRU), or more complex memory blocks and architectures [100] and increased number of connections between layers [98, 99]. Approaches based on pure CNN architectures have also been presented [7, 1] also exploring prediction of multi-

channel radar products simultaneously [93]. Accurate nowcasting of extreme events will have a crucial impact in the next few years, and major players such as Google have recently appeared on this research playground [1].

1.3 Nowcasting of extreme precipitation events

Here we contribute new materials and methods to improve the prediction of extreme precipitation events using ensemble deep learning and machine learning methods. The monitoring of extreme events, in terms of the intensity, direction, speed and evolution, must be carefully evaluated in order to prevent and contain possible damages to people and environment. The prediction of these events is particularly complex because of their sporadicity: for this reason, deep learning models in literature do not offer high levels of accuracy, and the prediction of these phenomena is incredibly challenging. The problem is also exacerbated by the lack of compelling datasets for the task.

The first contribution of this thesis is dedicated to the curation and release of a new multi-year radar archive dataset, containing a significant collection of intense and extreme precipitation events, specifically aimed at machine learning and deep learning nowcasting research. The dataset is accompanied by the open release of pretrained state-of-the-art model that can be used as baseline, the release of an interactive visualization tool for data exploration, and the publication of code and guidelines for reproducibility.

The second contribution is a new method for fast similarity (analog) search through large precipitation archives leveraging manifold learning techniques for dimensionality reduction and fast euclidean search based on Fast Fourier Transform. The method improves both speed and quality of

analog search over methods proposed in the literature and can be used as building block to implement an analog ensemble predictor.

The third contribution is a novel architecture that leverages deep learning ensembles and stacked generalization to improve predictions of extreme precipitation regimes over published methods. The architecture introduces several new concepts for deep learning in precipitation nowcasting helping to achieve this result.

All the contributions described for precipitation nowcasting are part of a multi-year collaboration effort with the Autonomous Province of Trento civil protection agency, Meteotrentino. The purpose of the collaboration is to provide early warnings to civil protection operators, in order to improve response time and deploy preventive actions when facing extreme precipitation events.

1.4 Structure of the Thesis

In Chapter 2 we present `TAASRAD19`, the first multi-year public dataset specifically aimed at machine and deep learning for precipitation nowcasting tasks. `TAASRAD19` is released along with a tool for interactive data explorations and guidelines for reproducibility.

Chapter 3 introduces current state-of-the-art models for nowcasting via deep learning, ConvLSTM and TrajGRU. `TAASRAD19` is applied to TrajGRU to establish a baseline for performance comparisons.

Chapter 4 presents *MASS-UMAP*, a new method based on the combination of Uniform Manifold Approximation and Projection for dimensionality reduction and fast euclidean search to improve both speed and quality of spatiotemporal analog search in multi-year radar archives, that can be used for fast ensemble retrieval.

Chapter 5 illustrates ConvSG, a new method to improve the nowcasting of extreme precipitation events by combining deep model stacking and orographic features. To achieve this result we introduce several novel concepts. We introduce the concept of Thresholded Rainfall Ensemble (*TRE*), a method to generate an informative ensemble of deep learning models using the same architecture and dataset. Subsequently we introduce *ConvSG*, a novel deep learning stacked generalization model that can be enhanced with orographic features, and that can substantially improve quantitative precipitation forecasting on extreme events.

Chapter 6 summarizes our findings and discuss future directions and outlook, proposing a unifying vision where all the presented advancements can be operationally adopted by meteorological agencies.

Chapter 2

Radar data for Machine Learning

Every good machine learning story begins with a good data story. The first chapter is dedicated to the most time consuming and often scientifically neglected part of every data science endeavour, where every data scientist and machine learning researcher often spends most of the time: collecting, managing, cleaning and analyzing data sources. Our contribution in this area is the open publication of a curated dataset of more than 9 years of radar-based precipitation data, specifically aimed at machine learning and deep learning researchers, that can be used as a benchmark to implement machine and deep learning models for precipitation nowcasting. The release is accompanied by an interactive visualization tool that can be used for data exploration and qualitative assessment.

2.1 Radar data resources and challenges

Significant efforts have been undertaken by the meteorological community to share open source weather radar resources, including software for analysis

and visualization [46, 45, 47, 76, 8, 59] and open data repositories [94]. Open datasets are collected and maintained by international Weather Data institutions across the US and Europe. The main publicly available products are (a) RADOLAN and RADKLIM [77], by the German Weather Service; (b) NEXRAD Level II [5], by the US National Oceanic and Atmospheric Service (NOAA); and (c) the dataset by the Royal Netherlands Meteorological Institute (KNMI) [49]. All these datasets provide radar reflectivity (as well as rain-gauge) data, with 1km spatial resolution, and 5min temporal resolution. While several data sources are available, the provided data often require several post processing steps for being retrieved in bulk format. Moreover, open data repositories are often not accompanied by labeling information for proper classification of the precipitation data present in the archives.

2.2 The TAASRAD19 open data collection

Here we introduce TAASRAD19, a dataset of weather radar maps covering an area of 240km of diameter, collected in the Trentino South Tyrol region, in the center of the Italian Alps. TAASRAD19 features more than 9 years (7/2010-12/2019) of reflectivity product of the radar, at high spatial resolution (0.5km) with 5min temporal updates. TAASRAD19 is the first available resource for sub-kilometer, high-frequency, extended time-span weather data for the Italian Alps. Notably, highly variable orography and environmental complexity make precipitation forecasting exceptionally challenging in the area. The temporal coverage of the dataset (almost 900 thousand time steps over about $1,250$ days with precipitations) is thus a key enabler for developing computational models for precipitation nowcasting and early detection of extreme events, in particular for implementing analog ensemble models and machine learning solutions.

The data included in TAASRAD19 were provided by *Meteotrentino*, the

official Civil Protection Weather Forecasting Agency of the Province of Trento, Italy. The agency operates a single-polarization Doppler C-Band Radar, in collaboration with the Civil Protection of the Province of Bolzano. The latter is responsible for the maintenance, operation and calibration of the receiver, as well as the generation of the products, while Meteotrentino is responsible for all the downstream tasks, that is quality control, rainrate conversion, forecasting and alerting. The radar is located on Mt. Macaion ($1,866$ m.a.s.l.), within a complex orographic environment in the center of the Italian Alps (N 46 29'18", E 11 12'38").

The radar system is an EEC DWSR-2500C and has been in operation since 2001 at the beginning with different operating modes and scan strategies (6 to 10 minutes time-steps). Between 2009 and 2010 the radar analog receiver was upgraded with the installation of an Eldes NDRX digital receiver. The update has improved both the signal quality and the scanning frequency of the radar system. Since the upgrade completed in mid 2010, the radar has been operating with the same scan strategy at a constant time-step of 5 minutes, for a total of 288 time steps per day. Details about the operational parameters and scan strategy are reported in Table 2.1 and Figure 2.1 respectively.

Offline calibration of the radar system is performed at least once a year with scheduled maintenance for the calibration of both the transmitter and receiver ends. During normal operations, continuous monitoring on the receiver end is performed by the Built In Test Equipment (BITE), while polar volume quality assessment is performed as part of the regular scan strategy by monitoring variations of the recorded background noise at high elevation (long PTR scan at high elevations) and by solar scans. The polar reflectivity generated by the scan is filtered and corrected from the back-scattering of most of fixed obstacles using a Doppler correction filter [17]. The products used here are calibrated by the data provider Meteobolzano by

Table 2.1: Technical characteristics and operational parameters of the mt. Macaion radar

Parameter	**Value**
Operational range	120 km
Maximum range	480 km
Resolution in range	250 m
Antenna Gain	45.8 db
3 dB beamwidth	0.9°
Wavelength	5.3 cm (5.6 Ghz)
Peak Power	307 kW
Pulse duration	0.8 μs
Clutter to Signal Ratio (CSR)	8.0
Signal Quality Index (SQI)	0.25
Ground clutter correction	Doppler filter

taking into account the corrections computed as a results of the monitoring and calibration.

Although multiple data products can be derived by the radar, in this data descriptor we focus on the 2D maximum reflectivity MAX(Z) generated from the filtered polar volume provided to the authors from Meteobolzano. The MAX(Z) product is computed as the maximum value (expressed in dBZ) for each pixel of a predefined grid, measured on the vertical section in the filtered and corrected polar volume. For the Mt. Macaion radar, the product consists of a bi-dimensional metric grid of 480×480 pixels (projected in UTM 32N coordinate system), with 500m pixel size covering an area of 240km of diameter (27,225 sq km) and centered at the radar site.

Despite the maximum reflectivity is potentially affected by noise, MAX(Z) is still preferred in the case of mountainous environments over alternative features commonly used for quantitative precipitation estimation, e.g. the *Constant Altitude Plain Position Indicators* (CAPPI). In fact, due

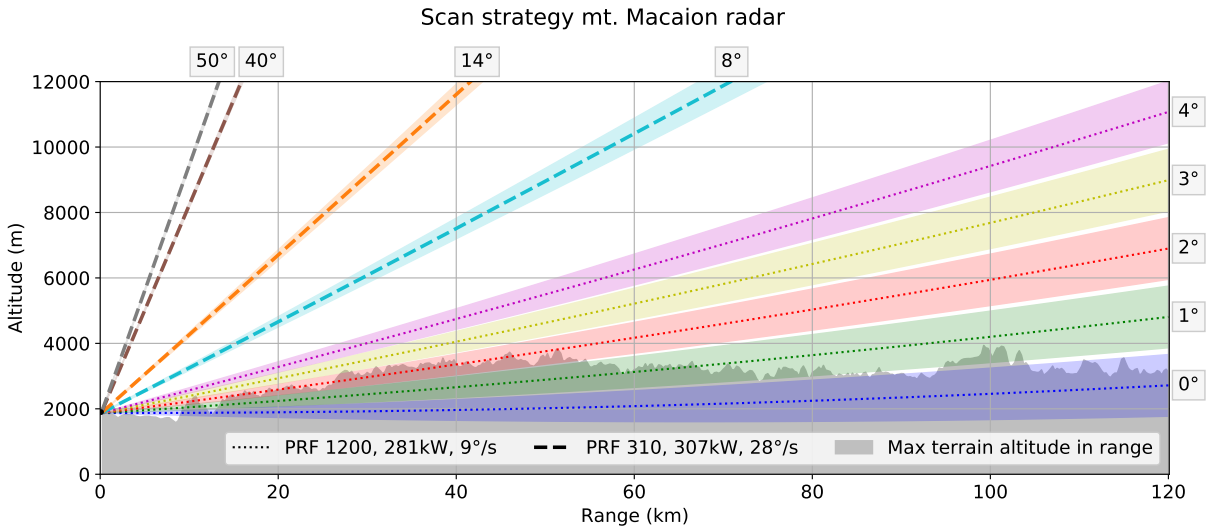


Figure 2.1: Scan strategy and signal characteristics of the mt. Macaion radar. Different Pulse Request Frequency (PRF), Power and rotation speeds are used in low and high elevation scans. Low elevation scans perform slower rotation ($9°/s$), use higher pulse frequency (1200) and lower power (281kW), while high elevation scans perform fast rotation ($28°/s$), use low pulse frequency (310) and higher power (307kW). The maximum terrain altitude for each range bin is reported in gray, showing that substantial beam blocking is encountered at lower elevations (0-2 degrees).

to the high operating altitude of the receiver, CAPPI and similar products can miss precipitation events at altitudes lower than the radar location. Moreover, the MAX(Z) products helps alleviating the severe beam blockage that the radar experiences by the nearby mountains at lower elevation scans. Computation of the occlusion experienced by the radar at the first three elevations (0, 1 and 2 degrees) of the volume is displayed in Figure 2.2, along with an example cross section. Note that MAX(Z) has been recently adopted as the standard by the pan-european radar composite for real time assessment purposes [79].

The MAX(Z) product is thresholded at a lower bound of 0 dBZ: while the receiver can indeed observe a certain amount of drizzle in the negative range of MAX(Z), its detection is exceptionally uneven for the mt. Macaion radar, being drizzle mainly a low altitude phenomena. The high altitude of

the receiver (1870 m.a.s.l.) and the severe beam blockage at low elevations allow the receiver to register drizzle almost exclusively in the underlying Adige valley, situated below the radar (200 m.a.s.l.). Even there, most of the drizzle is observed several hundred meters above the ground, and as such, it evaporates before reaching the ground. Given all these considerations and the specificity of the MAX(Z), it has always been standard practice for Meteotrentino to threshold the product at 0 dBZ.

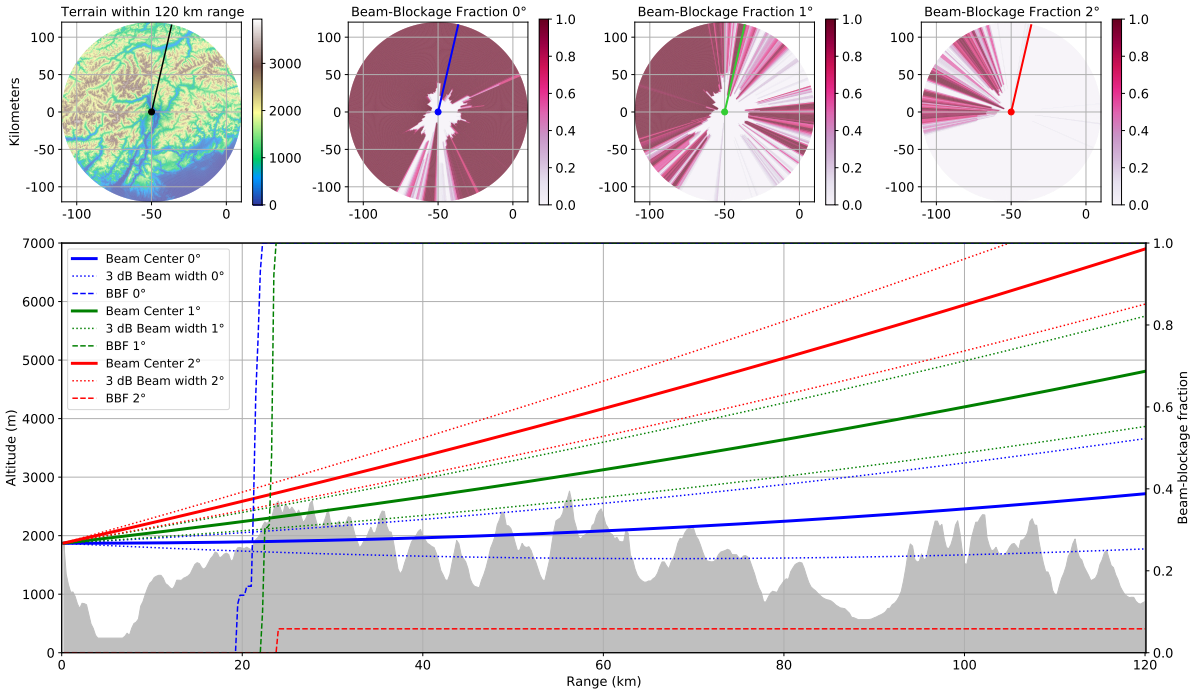


Figure 2.2: Beam Blockage Fraction (BBF) at different range and altitude. The upper row shows the digital terrain model, and the beam blockage percentage for the scans at 0, 1 and 2 degrees of elevation. The blue, green and red segments at 13 degrees azimuth on the maps (above) correspond to the cross section lines shown in the lower plot. The dashed lines represent the Beam Blockage Fraction (BBF) for each elevation at a given range.

An example of MAX(Z) product overlaid on the digital elevation model (gray colour map), main idrography (in blue), and administrative borders (red line) is displayed in Figure 2.3.

In order to standardize the development of nowcasting models from

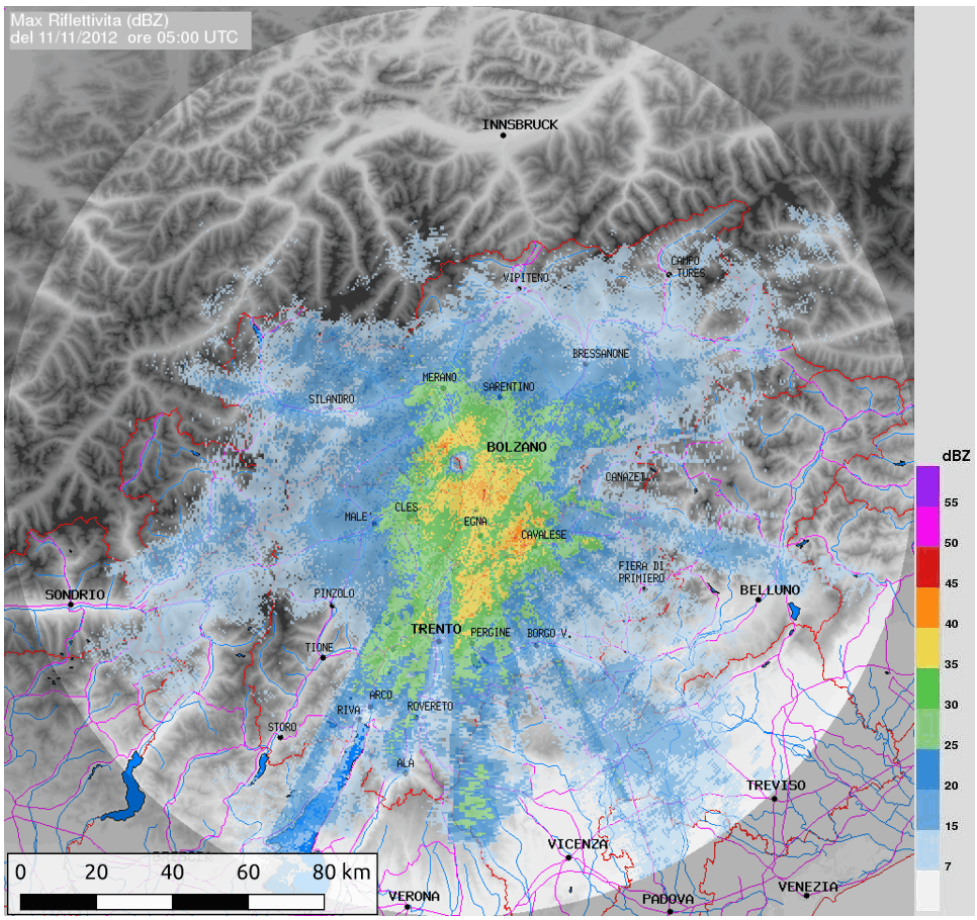


Figure 2.3: An example of observed radar reflectivity scan (MAX(Z) product) available in the TAASRAD19 dataset, represented in color scale over the geographical boundaries of the area covered by the radar. The area outside the observable radar radius is shaded.

TAASRAD19, both the original MAX(Z) images and a pre-processed version of the dataset are available. For reproducibility, the methods used for pre-processing, modeling and validation are also provided. They are composed of three main sections:

Sequence Extraction: obtaining contiguous and labelled precipitation sequences from the full image repository;

Noise Mitigation: reducing noise and systematic artefacts in the MAX(Z) product;

Data exploration: Interactive data visualization based on embedding.

In particular, the methods can be combined into a pipeline for developing nowcasting applications, with emphasis on those applying deep learning models. An overview of software methods for TAASRAD19 is displayed in Fig. 2.4; details for each main software module are provided in the following subsections.

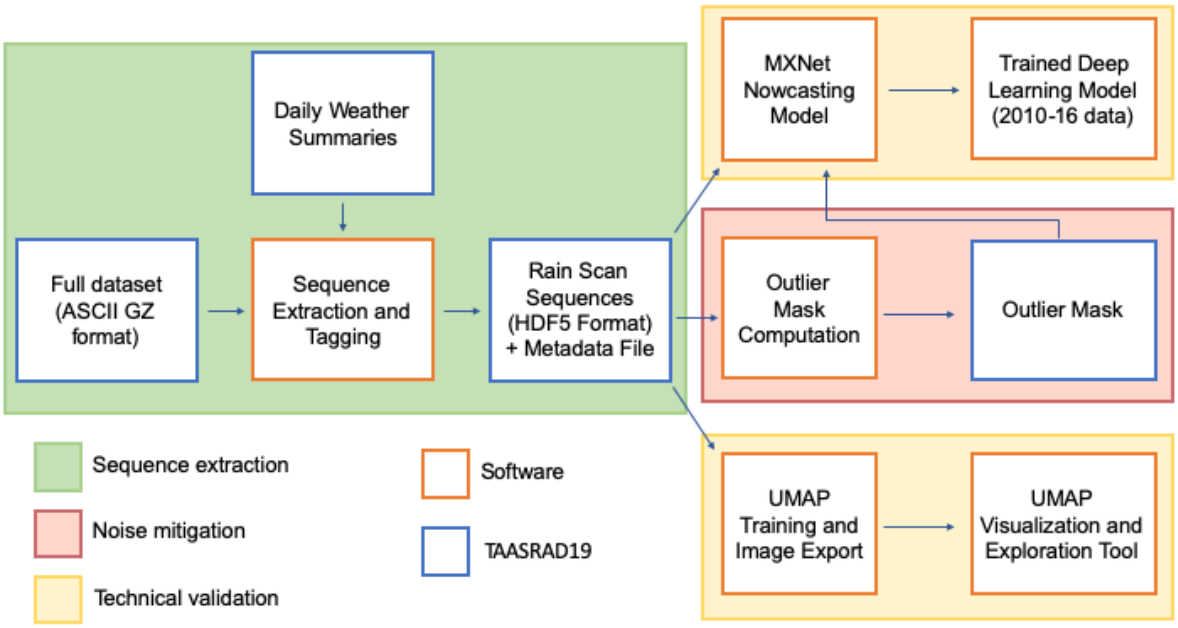


Figure 2.4: Overview of software methods for pre-processing and modeling of TAASRAD19 data.

Sequence Extraction

The elementary patterns for training and operating nowcasting systems are sequences of radar time steps (frames). The sequence extraction process applied to the raw TAASRAD19 data is based on four basic requirements:

1. each sequence must be contiguous in time (no missing frames), of sufficient length (at least two hours per sequence, to account for

operational requirements of nowcasting methods and to guarantee sufficient decorrelation time [111]);

2. each sequence should include at least one frame precipitation; sequences without precipitation signal are removed;

3. the full set of sequences should match the original data distribution in terms of seasonal occurrence (day/night, months, seasons), as well as precipitation types;

4. the sequences should be as clean as possible from noise or artefacts.

Descriptive statistics on the original data are listed in Table 2.2. The mean pixel value per frame varies from a minimum of $4.5 \cdot 10^{-4}$ to a maximum of 32.3. Clearly, a positive minimum indicates the presence of noise in images, even in the absence of precipitation. A noise-mitigation strategy is thus needed. Figure 2.5 reports the annual radar operativity, i.e., the amount of time the radar has been in operation over the ten years (thus, not in maintenance or shut down), expressed as the percentage of valid 5min frames over the total feasible in the year.

Table 2.2: Descriptive Statistics of Radar frames included in TAASRAD19

Total number of time steps	894,916
Total number of recorded days	3,292
Minimum & Maximum pixel values	0 & 52.5
Minimum & Maximum frame mean pixel values	$4.5 \cdot 10^{-4}$ & 32.3
Radar operativity between Jun 2010 and Dec 2019	89.55%

In addition to radar products, we collected the daily weather summary written by an operational meteorologist for each day. The summaries are provided in the form of a short overview, in Italian, describing the main meteorological conditions in the region during the day. A set of keywords

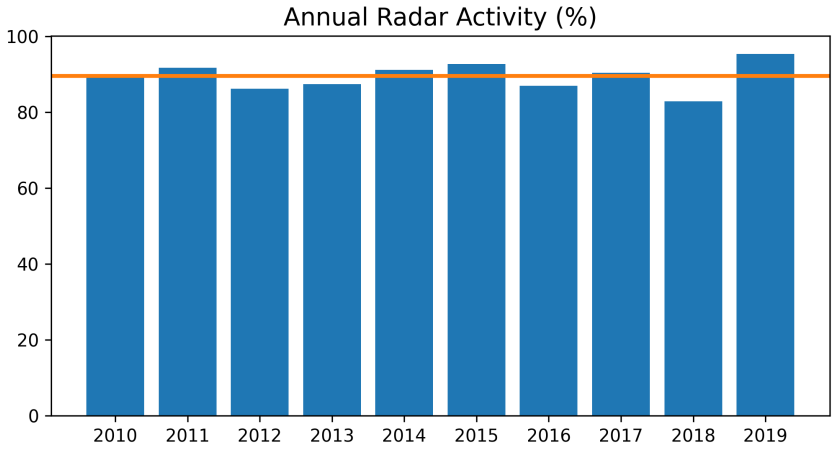


Figure 2.5: Radar operativity. Bar length represents the amount of time the radar has been in operation (not in maintenance or shut down), as percentage of valid 5min frames over the total feasible in the year; yellow line: mean radar operativity (89.55).

corresponding to specific meteorological events (e.g. *storm, rain, snow, hail*) were extracted automatically from the summaries to tag the precipitation patterns from the radar sequences by *weak-labels*, i.e. labels that should be considered incomplete, inexact and inaccurate but are nonetheless useful for machine learning purposes [112]. The annotations in `TAASRAD19` can be used in supervised or semi-supervised machine learning algorithms. The absence of those keywords has been combined with other descriptors of the radar images to identify and exclude sequences without precipitation events. The complete text of daily weather summaries are also released together with the radar data in the `TAASRAD19` repositories [35, 36]. In summary, the sequence extraction process is composed of four steps:

Data Selection To avoid seasonal imbalance, we select the interval `2010-11-01` and `2019-10-31`, corresponding to exactly 9 years of data.

Data Chunking The set of time steps is partitioned into multiple chunks of contiguous frames within a single day. Since some frames might be missing due to radar's fault or errors in the processing pipeline,

multiple chunks can account for the same day. Moreover, only chunks longer than 2 hours (i.e. 24 frames) are retained. Thus the length of each sequence varies from 25 to 288 contiguous frames, i.e. a single whole day with no missing data.

Sequence filtering Sequences with no or few precipitation events are removed. To retain useful chunks, we adopt a selection strategy based on the *Average Pixel Value* (APV) of the chunk (defined as the mean value over all pixels of the sequence), and the *weak-labels* assigned to the corresponding day. First, all the sequences s where $\text{APV}(s) < 0.5$ dBZ are immediately discarded, whereas those with $\text{APV}(s) > 1.0$ dBZ are retained. We thus filter out sequences with only background noise and retain those with at least one precipitation pattern. For all the remaining sequences (i.e. $0.5 \text{ dBZ} \leq \text{APV}(s) < 1.0$ dBZ), we leverage on the *weak-labels* annotated from the daily summaries to identify sequences with precipitation events. Sequences with no label - i.e. with no precipitation event registered for the corresponding day - are discarded. A graphical representation of the decision strategy workflow is depicted in Fig. 2.6.

Sequence labelling All the retained sequences are labelled according to the corresponding *weak-label* from the daily summary, wherever possible. The complete list of all keywords used (in the form of word stems, in Italian), and corresponding *weak-labels* is reported in Tab. 2.3.

The resulting number of sequences in TAASRAD19 is $1,732$, describing a total of $362,233$ scans, mapped to $1,258$ days of precipitation data. Sequences are available at the Zenodo TAASRAD19 repository [38], along with related metadata files, including labels and statistics.

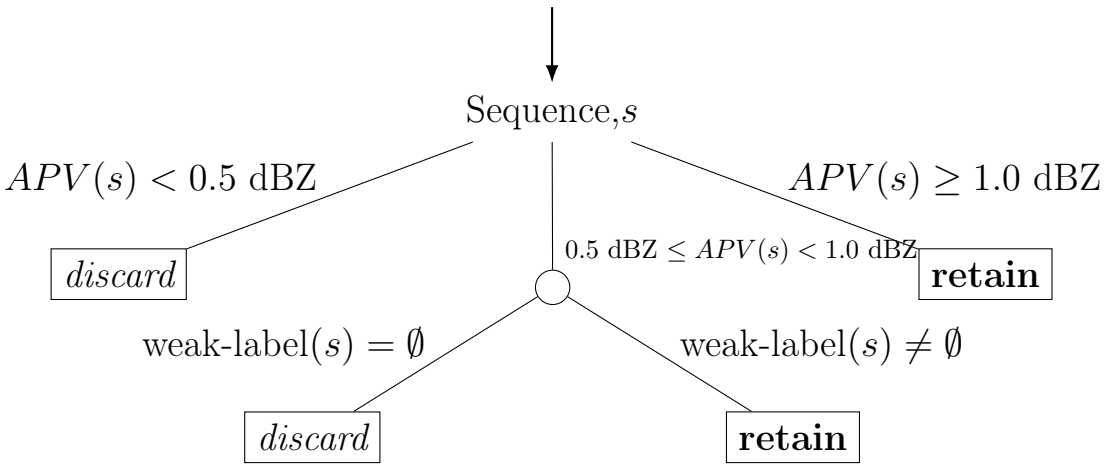


Figure 2.6: Decision tree of the strategy used to filter scan sequences based on corresponding APV value (the mean value over all pixels of the sequence), and *weak-labels*

Table 2.3: Keywords used to extract the weak labels from the daily weather summaries. Keywords correspond to word stems (in Italian) to account for plurals and other morphological inflections.

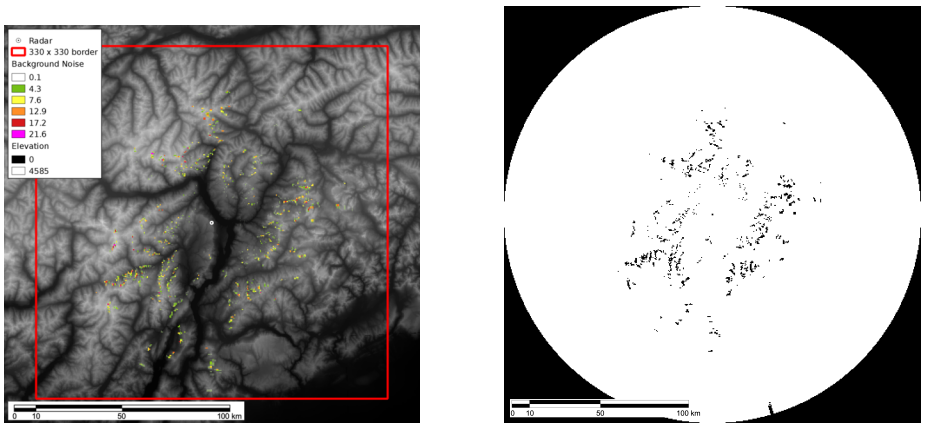
Keyword	**weak-label**
precip, piov, piog	rain
grand	hail
temporal	storm
rovesc	downpour
nev	snow

Noise Mitigation

The main goal of the noise removal step is to identify recurring noise patterns (i.e. outliers in specific pixel location) that can consistently occur in most radar images. Removing such outliers is particularly important, especially for methods (e.g. machine learning) whose performance may be affected by the presence of values (largely) out of the data distribution. We investigate the issue by generating a map to observe the presence of outlier pixels, from which we then derive a data-driven strategy for a *global outlier mask*.

Noise analysis To check for outlier pixels, we ranked each time step for increasing APV (i.e. from the the least to the most rainy) and we considered the top 0.1% of the ranking (895 frames) to compute a map of background noise. The noise map was generated as the average (per-pixel) of the 895 less rainy frames, which correspond to clear sky condition sampled at different times through the dataset.

As shown in Fig. 2.7a, where the computed map is overlaid to the digital terrain model, there is thus evidence of systematic artifacts in the signal.



(a) (b)

Figure 2.7: (a) Map of noisy pixels and orography in the operation region; noise value is computed as the average of radar signal over the most clear sky images, showing systematic artifacts. (b) The TAASRAD19 outlier mask; black pixels correspond to outliers and the area outside the radar operational range.

Outlier mask Systematic noise signals can be associated to non-fixed structures, e.g. clutter, multipath returns, or several other effects. In the case of the mt. Macaion radar, most of the noise still present in the data product is due to moving objects sensed on the terrain surface (e.g. trees moving during high wind days). The objective is thus to build a mitigation technique aimed at reducing the impact of high value noise in localized pixels present in most dataset operating days, thus managing possible non-meteorological moving

artefacts on the ground. In order to filter the noise in the frames, a *global outlier mask* can be generated based on a distance measurement between distributions of pixel values over time (Fig. 2.7b). We construct this mask using the *Mahalanobis distance*, as in [84]. In details, first the distribution histogram of the pixel values over a random sample (20% of the sequences) is computed by binning the ratios of pixel value in a location i in $N = 526$ bins x_i (each bin corresponding to a step of 0.1 dBZ). Then, we extract the corresponding sample mean

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

and covariance matrix

$$\hat{S} = \frac{1}{N-1}\sum_{i=1}^{N} (x_i - \mu)(x_i - \mu^T)$$

to evaluate the Mahalanobis distance of x as

$$D_M = \sqrt{(x - \hat{\mu})^T \hat{S}^{-1} (x - \hat{\mu})}$$

where $\hat{S}^{-1}$ is derived by using the Moore-Penrose pseudoinverse. Pixels that have a Mahalanobis distance higher than the mean distance plus three times the standard deviation are marked as outliers. We finally obtain a binary mask with $179,333$ inliers. Excluded pixels are $1,627$ outliers and $49,440$ points outside the radar operation range of 120km (equivalent to a 240 pixel radius from the Mt. Macaion site). The `TAASRAD19` outlier mask is mapped in Fig. 2.7b. Notably, the binary mask can also be used to skip calculation on the masked pixels when computing the loss function in deep learning models. The `TAASRAD19` outlier mask is also available as binary `PNG` file in the Zenodo repository [38].

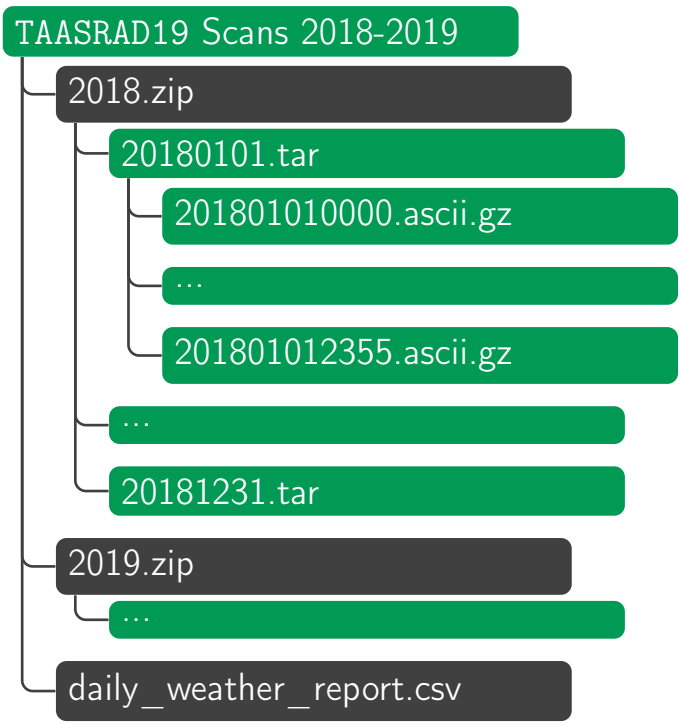


Figure 2.8: File structure of the TAASRAD19 ASCII image archive in the 2018-19 repository.

Data Records

TAASRAD19 is available on Zenodo, split in four repositories to comply with data size limits. The full MAX(Z) raw data archive is organized in two different repositories, one for years 2010 – 2016 [35] and another one for years 2017 – 2019 [36], while the sequences are available at [38] and [37], respectively. The product archive is organized by acquisition time for an easier automatic processing, using a three-level structure represented in Fig. 2.8. The organization of the data retains the hierarchy originally provided by Meteotrentino: this decision is motivated by the aim to provide a fully reproducible end-to-end data generation pipeline that can be run starting from the original raw dataset. Frames recorded the same year are archived together in a single ZIP file; each day of the year is archived in a single TAR files containing radar scan compressed using the GZIP algorithm

to reduce disk space. A CSV file with the daily weather summaries (i.e. daily_weather_report.csv) for all 9 years is also available, replicated in the two scan repositories.

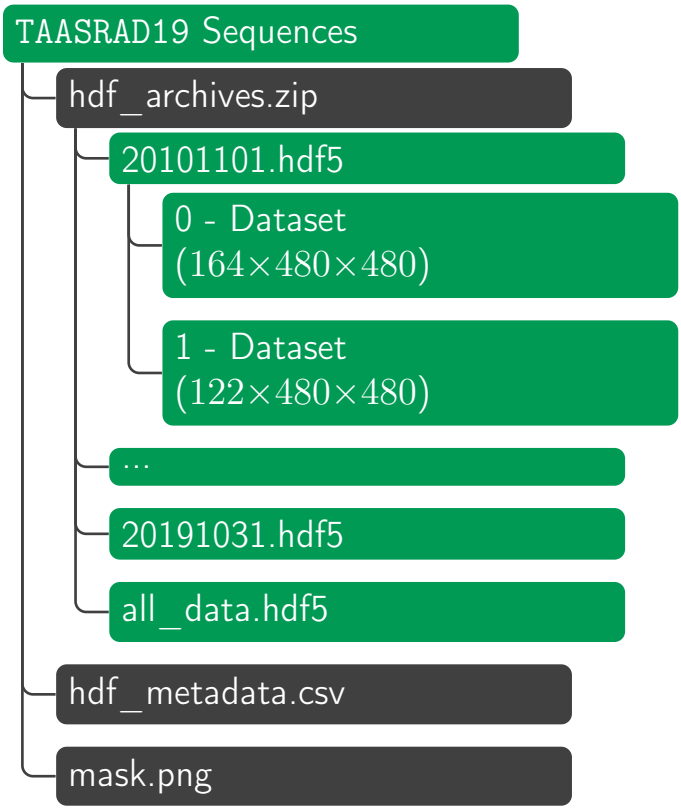


Figure 2.9: File structure of the TAASRAD19 HDF5 sequence archive

The data hierarchy has been designed to facilitate training machine learning models. The experimental setup of a classification/regression task usually requires an extensive number of repeated runs where data is supplied to the learning algorithms in small chunks, whose corresponding archived batches can be decompressed at runtime, thus optimizing the operational flow. In this scenario, the per-year organisation of the archives for the raw data is preferred over the unique monolithic alternative to allow users to set a preferred training strategy for training and testing over specific time intervals within data analysis plans, also considering appropriate cross-validation procedures. The configuration is indeed used in the analysis. For

example, practitioners usually find it convenient to keep the elementary batch file compressed, thus we keep the basic blocks as zip files. One level up, an organization for yearly time steps allows accounting for seasonalities, also maintaining large flexibility. The data structure in our case, including four different repositories, 2 reps for the raw data and 2 reps for the sequences, each split in one rep for the years 2010-2016 and one for the years 2017-2019 enables an easy generation of training/validation/test partitions in Machine/Deep Learning settings.

The text files follow the GRASS ASCII Grid [102, 43] format, which is a plain-text, human readable grid representation: each row in the file represents a row in the grid, where each cell value is a floating point number separated by a tabulation character. An optional `nodata` value (representing NULL value) can be specified: by convention, in `TAASRAD19`, the NULL data value is -99, and in our workflow it is converted to 0 upon parsing the files. The format can be parsed in any programming language without the need of specific software packages; geo-reference information is included in a header before the data rows. This structure allows a seamless data loading into Geographical Information Systems (GIS) software suites, e.g. QGIS [19], GRASS [72]. It also facilitates the conversion of scans in different formats of choice by libraries such as GDAL [101].

However, even if the compressed ASCII format is extremely easy to create and manipulate programmatically, it is very inefficient in terms of throughput, and processing power required for data ingestion. Moreover, metadata and other attributes cannot be incorporated using this data format.

For these reasons, the extracted sequences are made available in both HDF5 [90] and NetCDF [91] format, that are both widely used in meteorological applications [70]. Conversion between the two formats can be easily obtained by well established libraries such as *xarray*.

The HDF5 release [38] is mainly aimed at supporting a straight

integration of the dataset into modeling pipelines, thanks to the large support of the format in many machine learning platforms, and in the majority of scientific environments. Notably, HDF5 is the format of choice for many Deep Learning (DL) frameworks, that offer native support (e.g. Keras, TensorFlow), or straightforward integration hooks (e.g. PyTorch, mxnet/Gluon) for HDF5 datasets. Sequence data in the HDF5 release is organized similarly to the image archive (see Figure 2.9). Sequences from the same day are saved together in a single HDF5 file named after the date of the day. A file named `all_data.hdf5` stores links to all the daily files, and can be used to iterate over all the sequences. The whole HDF5 archive is stored on Zenodo on a single ZIP file (i.e. `hdf_archives.zip`). The minimum hdf5 library version to read the files is 1.10.4. Two other files are available: a PNG image (`mask.png`) representing the pre-computed outlier mask, and a CSV file (`hdf_metadata.csv`) with relevant metadata about each sequence. Metadata include: id of the sequence, start and end time, sequence length, average pixel value, corresponding weak-labels extracted from the daily weather summaries (if any).

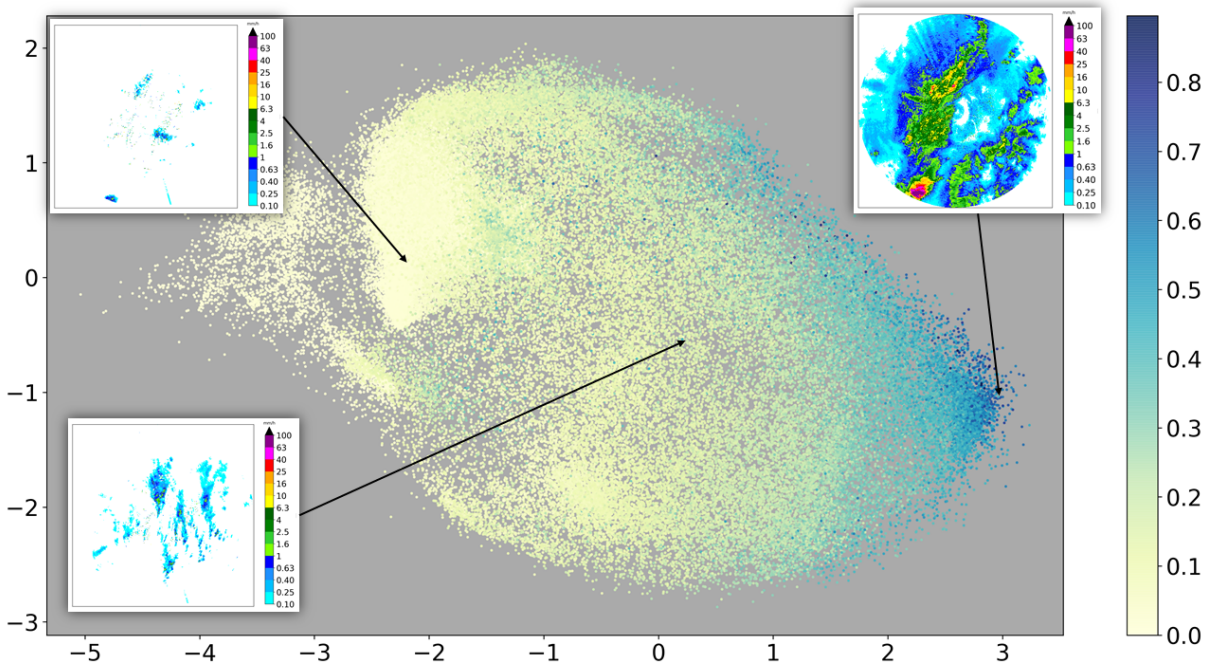
The NetCDF release [37] is aimed at maximum compatibility with existing meteorological and climatological tools for data analysis and exploration. The dataset mimics the same file structure of the HDF5 release (one file per day) and is further supplemented by extensive metadata (e.g. reference coordinates, data types, date/time reference, daily tags, sequences length). To ensure maximum compatibility, a flattened structure is used and NetCDF4 groups are avoided. Sequence lengths and date-time attributes are both reported in metadata, and can be used to determine the start and end frame of each sequence. The produced format follows the Climate and Forecast (CF) Metadata conventions and has been validated for the use with compatible tools using the CF-Checker suite (`https://github.com/cedadev/cf-checker`) against the CF-Convention 1.7 standard.

2.3 Exploring massive radar data via Embeddings

Here we leverage UMAP [68] dimensionality reduction features for the interactive visualization of similar radar images from the `TAASRAD19` dataset. To realize a real-time interaction on a massive sample of images, we first pre-processed all HDF5 sequences by resizing the images from 480×480 to 64×64 pixel using bi-linear interpolation. Normalization between 0 and 1 is obtained by dividing each pixel value by 52.5, i.e. the maximum reflectivity value supported by the radar (see Tab. 2.2). The first $200,000$ images (out of $362,233$) are used as training data for a UMAP model with the following hyper-parameters: `neighbors`=200; `components` (dimensions)=5; `min-distance`=0.1; `metric`=*euclidean*. The UMAP algorithm outputs a dimensionality reduction map (from 64×64=$4,096$ to 5), which distributes images in the reduced space by preserving the reference distance metric as in the original space: in this case, euclidean distance. Given that Euclidean distance is rank preserving with regard to mean squared error, similar precipitation patterns result closer in the reduced space. Further details on UMAP are provided in Cahpter 4. In Fig. 2.10 we show an example of UMAP planar embedding of the remaining $162,233$ scans (`TAASRAD19_u162k`), where each point is coloured by Wet Area Ratio (WAR), defined as the percentage of pixels in the scan with a rain rate higher than 0.1mm/h. Examples of different precipitation patterns in `TAASRAD19_u162k` are shown as insets within the figure. From left to right (UMAP component 1), locations in the projected space correspond to patterns of increasing WAR.

The approach has been engineered as UMAP Radar Sequence Visualizer, a tool for interactive exploration of sequence analogues in radar archives. Sets of radar sequences can be imported for visualization in an interactive web canvas built as React/NodeJS application, derived from the UMAP Explorer

Figure 2.10: UMAP Embedding for `TAASRAD19_u162k`: plot of the first (`x` axis) and third (`y` axis) components. Each point is a radar scan in the projected UMAP space, colorized by Wet Area Ratio (WAR). Scans with similar rain patterns are placed closer together. Insets show examples of three different precipitation patterns and their position in the UMAP projected space.



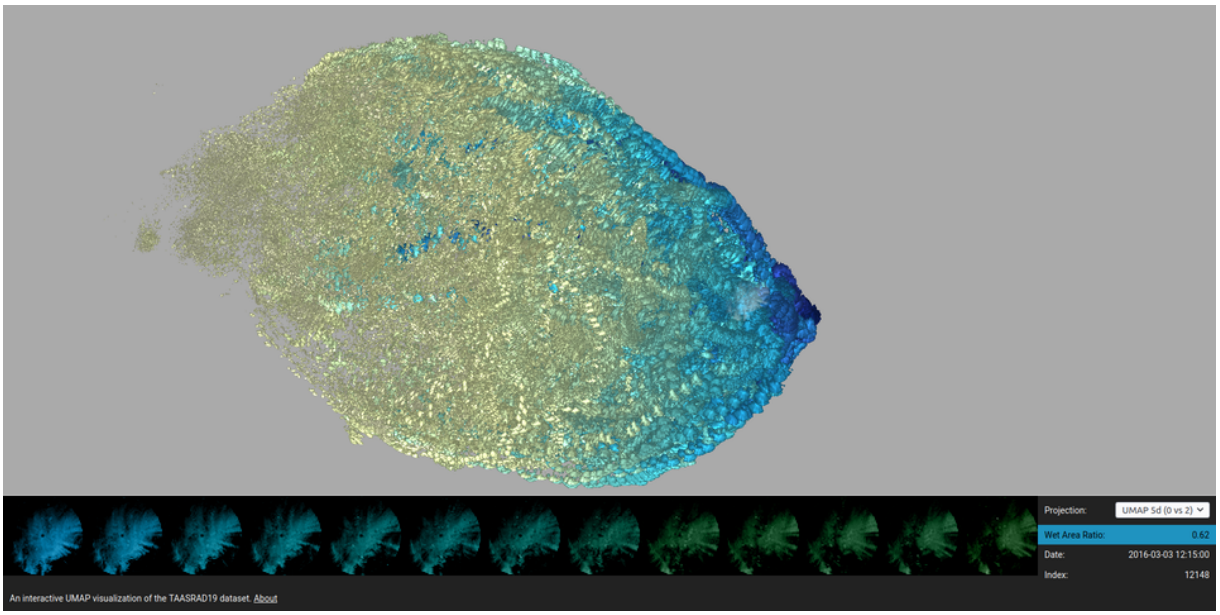
tool [24].

Each radar scan is placed as a mini image on the explorable canvas based on its coordinates in the UMAP projection. The canvas can be panned and zoomed, and each image is colored by WAR using a yellow-to-blue gradient. When an image is selected, the lower panel shows the next images in the sequence, highlighting the evolution of the precipitation pattern. Projections over different UMAP axis pairs can be selected. The source code of the tool, along with scripts and examples on how to export data for visualization, are available in the GitHub repository.

An online demo of the UMAP Radar Sequence Visualizer is also available (reachable from the `TAASRAD19` GitHub repository). The online dashboard is

currently equipped with `TAASRAD19_u50k`, a sample of moderate size (50,000 images for 18 months of observations) to allow browsing with limited RAM resources (Fig. 2.11). In the online tool, scans with precipitation concentrated in the Northern part of the region are also located on the upper area of the plot, while those with rain in the South lie on the lower part of the plot. Blue images on the rightmost sector of the plot represent extreme events, while scattered points on the left of the main cloud correspond to less severe precipitation patterns.

Figure 2.11: Interface of the interactive radar analog exploration tool.



2.4 Computational details

When using the scans or the sequences in machine learning workflows, we suggest to set no-data cells (value -99.0) to 0 and normalize the data in $[0, 1]$ interval by dividing by 52.5 before feeding the batch of data to the model; whilst for computer vision applications it can be useful to transform the scans to grayscale images by applying a lossy conversion to 8-bit integers (values from 0 to 255).

All the software described in this chapter have been published in a public GitHub repository[1], along with the Python scripts for sequence pre-processing and installation scripts. All the code was written in Python 3.6 and tested on Ubuntu releases 16.04/18.04. Some pre-processing steps (e.g. sequence and outlier mask generation) require a non trivial amount of computing resources and memory.

2.5 Rainrate conversion

It is possible to convert the `TAASRAD19` reflectivity in rain rate (mm/h) by applying the Z–R relationship developed by Marshall and Palmer[66] with the standard parameters $Z = 200R^{1.6}$.

[1] `https://github.com/MPBA/TAASRAD19`

Chapter 3

Deep Learning for Precipitation Nowcasting

Where there is data, there is machine learning. Meteorology is one of the scientific field with the longest and most curated data collections in the world. It is thus of no surprise that the deep learning community has started to spend efforts to help solving meteorological problems. In this chapter we introduce the recent advancements contributed by deep learning research to precipitation nowcasting, along with the basic meteorological concepts required to engage in this field.

3.1 Convolutional recurrent neural networks for Nowcasting

Chapter 1 briefly introduced the relevant literature outlining the two different approach to deep learning for nowcasting, with models based on pure CNN architectures, and models based on Convolution and Recurrent Layers. In this thesis we adopt the latter approach, because of the superior predictive performance and flexibility offered by this type of architectures. The primary characteristics of this class of models is to be able to extrapolate both the

spatial and the temporal information present in the data, where the recurrent unit encodes the temporal evolution, and the convolution captures the spatial correlation. In these architectures the nowcasting task is considered as a sequence-to-sequence problem, where sequences of reflectivity maps are used as input. More formally, given a reflectivity field at time T_0, radar-based nowcasting methods aim to extrapolate m future time steps $T_1, T_2, ..., T_m$ in the sequence, using as input the current and n previous observations $T_{-n}, ..., T_{-1}, T_0$.

3.2 TrajGRU Nowcasting Model

As baseline model in this work we adopt the trajectory gated recurrent unit (TrajGRU) network structure proposed by Shi et al. in [84]. The underlying idea of the model is to use convolutional operations in the transitions between RNN cells instead of fully connected operations to capture both temporal and spatial correlations in the data. Moreover, The network architecture dynamically determines the recurrent connections between current input and previous state by computing the optical flow between feature maps, both improving the ability to describe spatial relations and reducing the overall number of operations to compute. The network is designed using an encoder-forecaster structure in three layers: in the encoders the feature maps are extracted and down-sampled to be fed to the next layer, while the decoder connects the layers in the opposite direction, using deconvolution to up-sample the features and build the prediction. Fig. 3 shows the model architecture diagram. With this arrangement, the network structure can be modified to support an arbitrary number of input and output frames. In our configuration 5 frames (25 minutes) are used as input to predict the next 20 steps (100 minutes), at the full resolution of the radar (480 x 480 pixels). Given the complex orographic environment where the radar

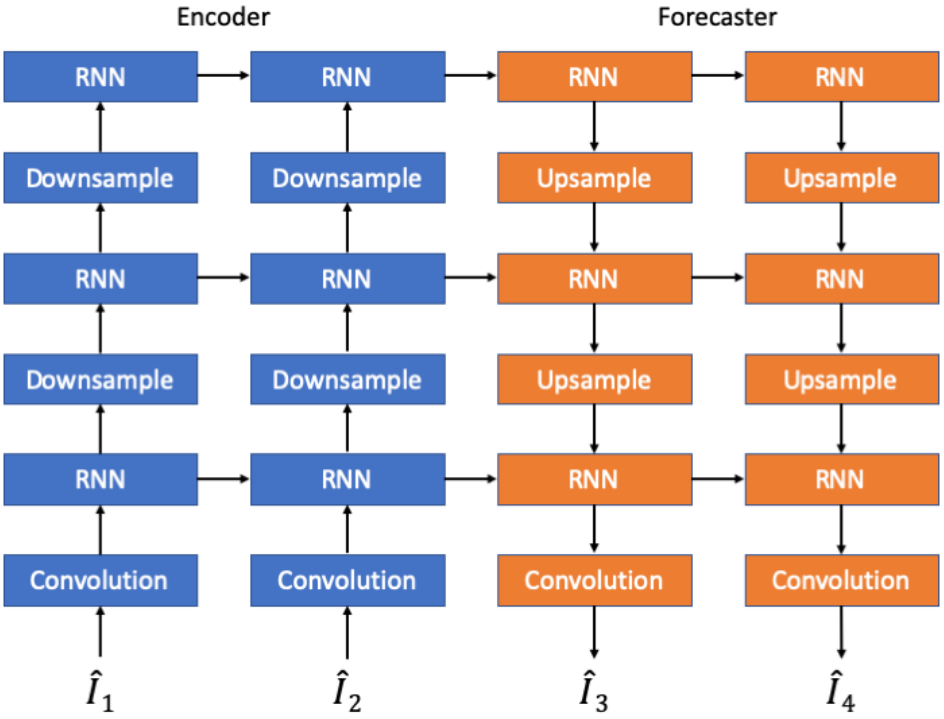


Figure 3.1: Schema of the deep learning architecture adopted by TrajGRU, in a configuration with two input and two output frames

operates, the data products suffer from artifacts and spurious signals even after the application of the polar filter correction. For this reason we use the static noise mask presented in Chapter 2 to systematically exclude out of distribution pixels when computing the loss function during training. As loss function we adopt the same weighted combination of L1 and L2 proposed by [84], where target pixels with higher rain rate are multiplied by an higher weight, while for masked pixels the weight is set to zero. Specifically, given a pixel x the weight $w(x)$ is computed as the stepwise function $w(x)$:

$$w(x) = \begin{cases} 0 & \text{if } x \in \text{MASK} \\ 1 & \text{if } R(x) < 2 \\ 2 & \text{if } 2 \leq R(x) < 5 \\ 5 & \text{if } 5 \leq R(x) < 10 \\ 10 & \text{if } 10 \leq R(x) < 30 \\ 30 & \text{if } R(x) \geq 30, \end{cases} \tag{3.1}$$

where $R(x)$ is the Z-R Marshall Palmer conversion with the parameters described in Section 2.5. The final loss equation is given by the sum of the weighted errors

$$\text{B-MAE+B-MSE} = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{480}\sum_{j=1}^{480} w_{nij}(x_{nij}-\widetilde{x}_{nij})^2 + \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{480}\sum_{j=1}^{480} w_{nij}|x_{nij}-\widetilde{x}_{nij}|, \tag{3.2}$$

where w are the weights, x is the observation, $\widetilde{x}$ is the prediction and N is the number of frames. This loss function gives the flexibility to fine-tune the training process by forcing the network to focus on specific rain regimes at the pixel level with a concept that reminds spatial attention layers [20].

3.3 Verification scores for precipitation nowcasting

The standard verification scores used by meteorological community to test predictive skills of precipitation forecasting are the Critical Success Index (CSI), the False Alarm Ratio (FAR), and the Probability of Detection (POD). These measures are somewhat similar to the concept of accuracy, precision and recall commonly used in machine learning settings. To compute the scores, it is necessary to convert the prediction and ground truth matrix of the precipitation to 1/0 values by thresholding the precipitation and compute the number of *hits* (truth=1, prediction=1), *misses* (truth=1, prediction=0)

and *false alarms* (truth=0, prediction=1) between the two matrixes. Then the skill scores are defined as:

- CSI = $\frac{hits}{hits+misses+falsealarms}$
- FAR = $\frac{falsealarms}{hits+falsealarms}$
- POD = $\frac{hits}{hits+misses}$

3.4 TrajGRU on TAASRAD19

We used TAASRAD19 to train a deep learning model that forecasts reflectivity up to 100min ahead (i.e. 20 frames) at full spatial spatial resolution of the radar (0.5×0.5 km), based on 25min (i.e. 5 frames) of input data. A Python implementation using the Apache MXNet [21] deep learning framework is available at https://github.com/MPBA/TAASRAD19[1].

In our experimental setup, TAASRAD19 sequences extracted from June 2010 to December 2016 are used for training, whilst the model is tested in inference on sequences from 2017 to 2019. Training and validation sequences are extracted with a moving-window strategy applied along the entire set of contiguous sequences included in TAASRAD19. The generated sub-sequences are 25 frames long, where the first 5 frames are used as input, and the remaining 20 ones are used as ground truth for validation. In summary, 220,054 and 122,548 sub-sequences have been generated for *training* and *validation*, respectively.

To allow a fair comparison with results reported in [84] on the Hong Kong (HKO-7) dataset, we implement the same model hyper-parameters: the model is trained for 100,000 iterations considering a batch size of 4, using two NVIDIA GTX1080 GPUs in parallel, with 8GB of memory each.

[1]For the original version see https://github.com/sxjscience/HKO-7

Network weights for our trained model are available on GitHub. We evaluate results using the Critical Success Index (CSI) score, as defined in [84]: output predictions and ground truth scans are first converted to rain rate using the Marshall-Palmer Z-R relationship [65], then binarized at different thresholds to test model performance over different rain regimes. Results on the validation data set are reported in Tab. 3.1. Scores for both models are satisfactory for potential application as a score of CSI > 0.45 (for $r \geq 0.5$) means that the model is reliable for predicting precipitation occurrence. Results reported for the HKO-7 dataset are consistently better; disregarding the use of the MAX(Z) product instead of CAPPI as inputs, differences are expected due to the higher variability of Alpine landscape and the different spatial resolutions (0.5km for TAASRAD19 vs. 1.07km for HKO-7).

Table 3.1: Critical Success Index (CSI) scores for TrajGRU on TAASRAD19 and in Shi et al, 2017 on HKO-7 dataset; r is the instantaneous rain rate (mm/h)

Model and Dataset	$r \geq 0.5$	$r \geq 2$	$r \geq 5$	$r \geq 10$	$r \geq 30$
TrajGRU on HKO-7 [84]	0.553	0.476	0.375	0.284	0.186
TrajGRU on TAASRAD19	0.487	0.283	0.190	0.144	0.078

Chapter 4

Towards Ensembles for Nowcasting

The use of analogs – similar weather patterns – for weather forecasting and analysis is an established method in meteorology. The most challenging aspect of using this approach in the context of operational radar applications is being able to perform a fast and accurate search for similar spatiotemporal precipitation patterns in a large archive of historical records. In this context, sequential pairwise search is too slow and computationally expensive. Here we propose an architecture to significantly speed-up spatiotemporal analog retrieval by combining nonlinear geometric dimensionality reduction (UMAP) with the fastest known Euclidean search algorithm for time series (MASS) to find radar analogs in constant time, independently of the desired temporal length to match and the number of extracted analogs. We show that UMAP, combined with a grid search protocol over relevant hyper-parameters, can find analog sequences with lower Mean Square Error (MSE) than Principal component analysis (PCA). Moreover, we show that MASS is 20 times faster than brute force search on the UMAP embedding space. We test

the architecture on real dataset and show that it enables precise and fast operational analog ensemble search through more than 2 years of radar archive in less than 3 seconds on a single workstation.

4.1 The Analog Ensemble Approach

The observation of repeating weather patterns has a long history [64], and the use of analogs has found its way in almost all aspect of meteorology, for the most diverse purposes. Approaches based on analogs have been proposed for the postprocessing of numerical weather predictions [27], as a statistical downscaling technique [114] and for data assimilation in numerical models [63, 89]. However, the most prolific use of analogs is by far forecasting: either as a proxy for predictability [82], or as prediction technique itself [14]. In this regard, one of the most used operational methods for analog-based forecasting is Analog Ensemble (AnEn) [26], which involves searching and using an ensemble of past analogs to generate new deterministic or probabilistic [26] predictions. Ensemble methods have been used for very complex prediction targets, like short-term wind [3] or renewable energy forecasting [2]. Analogs can be sought to match single locations (pointwise time series) or spatial distributions over an area (spatiotemporal sequences). The quality of the analogs is totally dependent on the dimensionality of the data archive [95]: depending on the context, the historical records can spawn from few years to multiple decades, making the analog search procedure a critical component of any operational analog application method. The ideal analog search method should be dependable, predictable, accurate, fast and able to return multiple ranked analogs at the same time. In this paper we present a novel search method for spatiotemporal sequences and show how it meets many of these desirable qualities.

In nowcasting applications (very short-term prediction, between 0 to 6 hours), where the available time for computation is extremely limited, it is often necessary to trade-off analog search accuracy in favor of speed to meet a given computational time threshold. In this regard, one of the most important and complex problems is nowcasting precipitation fields. Analog ensemble approaches based on radar precipitation fields have been proposed for this application [74, 85, 6]: in particular, the AnEn method is especially relevant, since the nowcasting of convective precipitations is extremely challenging to tackle by Numerical Weather Prediction (NWP) methods [87]. AnEn approaches to radar nowcasting use feature extraction [74], linear dimensionality reduction [31] or cross correlation [6] to summarize and perform an Euclidean search through the radar image archive, in combination with other indicators like mesoscale variables, seasonality and time of the day to filter the pool of valid sequences.

Here we propose a flexible framework that employs a two-step process that can be used to improve the accuracy and dramatically speedup the retrieval of spatiotemporal analog sequences. Our work combines non-linear geometric dimensionality reduction method based on Uniform Manifold Approximation and Projection [68] (UMAP) with the fastest Euclidean based profile search algorithm (Mueen's Algorithm for Similarity Search [71], MASS), that can search for arbitrarily long time sequnces in constant time. We compare UMAP dimensionality reduction with Principal Component Analysis (PCA) [53] on a original test dataset of almost 10 years of radar data, and show that UMAP finds analogs with smaller Mean Squared Error (MSE) than the one extracted by the PCA based method proposed in [31], with proper train configurations parameters. Moreover, we discuss how the MASS search algorithm is 20 times faster than linear Euclidean search and how it can be used to search the reduced UMAP space to find arbitrarily long time sequences of analogs in constant time. Finally we discuss the flexibility

of the UMAP-MASS method by showing how other indicators can be easily integrated in the search space to filter analogs and how it is feasible to fine-tune the dimensionality reduction using a custom distance function to project the embeddings.

4.2 Projection and search for analog ensembles (MASS-UMAP)

4.2.1 Introduction to Uniform Manifold Approximation and Projection (UMAP)

Uniform Manifold Approximation and Projection [68] (UMAP) is a recent manifold learning technique for dimensionality reduction aiming at reconstructing in a lower dimensional space the geometric structure of the variety where data lie. UMAP is based on algebraic techniques mapping the original manifold into a reduced projection using topological analysis of the geometric space. UMAP is thus situated in the family of k-neighbour based graph learning algorithms (e.g. Laplacian Eigenmaps, Isomap and t-SNE) and combines approximate k-nearest neighbor calculation (Nearest-Neighbor-Descent) and a stochastic gradient descent for efficient optimization (SGD). Due to its faithfulness in the representation and the low computational burden, UMAP is becoming the reference algorithm for dimensionality reduction in multiple research fields [13]. The dimensionality reduction of UMAP is driven by 4 parameters: *metric*, number of components (d), number of neighbors (n) and minimum distance (mindist) [67].

The *metric* parameter is the distance function used to compare elements in the space of the input data. By default UMAP uses the Euclidean distance function, but any non-negative symmetric function $f\colon \mathbb{R}^2 \to \mathbb{R}$ can be a valid UMAP metric.

The number of components d represent the size of the projected space where the transformed data will lie: for example with $d = 4$ every input element will be reduced to a vector of 4 values.

The n parameter controls the number of nearest neighbors used for each point to build the local distance function, taking into account the trade-off between highlighting the details rather than the global picture when rearranging the input data. Using higher values will force UMAP to look at larger neighborhoods to estimate the structure of the data, while using small values will give more weight to local structures present in the data.

The mindist parameter forces a minimum distance between points in the output space. A smaller value will allow UMAP to pack similar elements closer to each other and describe finer topological structure, that can be useful for example for clustering applications, while higher values will preserve the broader structure more.

In table 4.1 we report a summary of the main parameters used in this work.

4.2.2 The Mueen's Algorithm for Similarity Search (MASS)

One of the most important subroutines in time series analysis is searching for similar patterns to a query string. Any new algorithm or method that is able to speedup this task can potentially open new disruptive applications in any system that deals with time series and historical records. Mueen's Algorithm for Similarity Search (MASS) [71] is probably the most interesting solution in the domain of Euclidean based searches in the last few years, where it has spawn new research directions [110, 109, 25, 40, 113]. Recently its application as an analog search function for forecasting time series of renewable energy has been proposed [108]. The key idea in MASS is to perform the search in the

mindist	UMAP training parameter used to define a minimum distance between elements in the low dimensional representation. In our study this value is fixed to 0.1.
metric	UMAP training parameter used to compare images in original space. In this study we use the euclidean distance (the Euclidean distance is rank invariant with respect to the MSE).
n	UMAP training parameter used to define the number of nearest neighbors to build the local distance function. N is the set of all tested values of n.
d	Number of components (dimensions) used by the dimensionality reduction (UMAP/PCA). D is the set of all tested values of d.
t	Length of the query sequence (number of consecutive radar images) to match. T is the set of all tested values of t.
k	Number of closest analogues to consider for further processing. K is the set of all tested values of k.
l_s	Number of radar images in the search set (archive). The search set contains all the valid data from 2010 to 2016.
l_v	Number of radar images in the verification set (query data). The verification set contains all the valid data from 2017 to 2019.

Table 4.1: Table of parameters

frequency domain by computing the fast Fourier transform (FFT) on the time series, and replace the loop typically used in similarity matching algorithms with a convolution operation between the query vector and the search archive vector, thus making the search routine independent of the query length. This makes the algorithm free from the curse of dimensionality: matching a long query takes the same time than matching a short one. For this reason the algorithm complexity depends only on the size of the search archive l_s and its complexity is $O(l_s \log l_s)$ in the worst case. MASS also produces all the distances from the query to all sub-sequences of the archive, allowing to find all the most similar profile in one single pass. Notably, MASS is parameter-free and can be easily parallelized by splitting the data archive vector in chunks.

4.2.3 Application to TAASRAD19

For this task we leverage the TAASRAD19 starting from the raw scan. Since the analog sequence retrieval process expects temporal continuity in the data, we developed a strategy to remove most of the empty data while keeping the temporal discontinuity of the resulting dataset to a usable level. Instead of working on single radar images, we divide the data in chunks of contiguous images by splitting the data by day. Due to missing scans we can obtain more than one chunk of contiguous data for the same day. Chunks longer than 2 hours are kept, the rest is discarded, so each chunk contains at least 25 and at most 288 contiguous scans. Finally we thresholded all chunks with no or few precipitation events: the sum of all pixel values of all images of each chunk is computed, and all chunks with an average pixel value < 0.5 dBZ are discarded. After these cleaning steps, the total number of samples in the dataset amounts to $342,598$ radar images, corresponding to 3 years and 95 days of precipitation data. For the reason stated above, and to avoid temporal overlapping, we split the dataset temporally between search space and verification: the data from the years 2010-2016 was used as archive (search space), and the years 2017-2019 as verification (query data). The final result is $l_s = 220,050$ (2 years and 34 days) images for search space and $l_v = 122,548$ (1 year and 61 days) for verification.

A simple bilinear resize was applied to all the selected data to obtain 64 x 64 pixel images, corresponding to a resolution of 3.75 x 3.75 km. This was chosen for similar reasons to the ones described in [31]: to reduce the computational time of the experiments and extend the range of tested parameters combinations, to remove small scale variability, and in our case to also remove any possible residual noise and scatter in the MAX(Z) product. Figure 4.1 summarizes the data pre-processing pipeline.

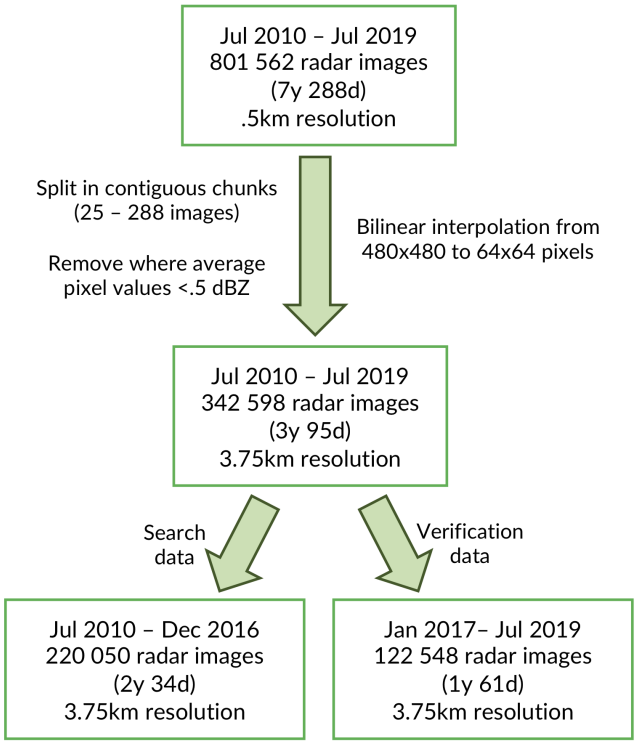


Figure 4.1: Data pre-processing pipeline. The whole dataset is first filtered to remove data chunks that do not contain a interesting amount of signal. A bilinear interpolation filter is applied to the images to reduce the resolution from 480x480 to 64x64 pixels. The transformed dataset is then split into search and verification sets.

4.2.4 The MASS-UMAP algorithm

The workflow of our method follows two phases: dimensionality reduction phase and search phase.

In the first phase we used UMAP to reduce the dimensionality of each image in the radar archive to a vector of length d (embedding), such that $d \ll p$ where p is the number of pixels in one image. UMAP will learn a transformation that maps closer together in a geometric space the input images that are closer according to a specified *metric*. In our experiment we chose the Euclidean distance as indicator *metric* to compare the images (more details for this choice are discussed in Section 4.2.5.1), but more specialized metrics are possible. The embeddings are generated for both the search space

data and the query data.

The second phase uses the MASS algorithm to search through the embedding search space and find the closest matches for the query data. To this aim, the embedded vectors of all the images in the search space are concatenated together following their natural time order. The result is a vector of length $l_r = d * l_s$. The embeddings of the queries are concatenated in the same fashion, generating vectors of size $q = d * t$ for each query, where t is the time length of the search query (the number of images in the sequence). The MASS algorithm is used to compare the query vector with the search space vector and extract the indexes of a desired number k of closest Euclidean profiles. The use of the Euclidean search is possible because the embeddings are projected by UMAP into a geometric space. As last step, the image sequences corresponding to the indexes of the top k profiles are compared and reordered by computing the MSE with respect to the query image sequence (in the low-res 64x64 image space), generating the final analog ranking. The desired number of final analogs can be selected by slicing the MSE reordered k analog vector to the final desired size of top-a analogs, with $a \leq k$. There are two reasons why performing a partial reorder of the top k results in image space before selecting the final analogs is useful: the first is that, even if the dimensionality reduction method works ideally on all cases and always returns the same top k items that an MSE search[1] would, there is hardly any guarantee that those will be returned in the same order (this is discussed in Section 4.2.5.1 and 4.3.2). The second reason is that MASS is able to perform such a fast search because it compares the vector profiles in the frequency domain: while the computed rankings for the profiles are exact, it may match a sequence with a very similar profile of the query vector but with a constant shift on all coordinates. This is indeed a rare occasion, but the partial MSE reordering is overall useful to move those

[1]the expressions "MSE search" and "MSE reorder" in the text, are always to be considered as operations computed by comparing radar images in the low-resolution 64x64 image space

spurious matches at the bottom of the ranking. So, while we want to avoid computing MSE between the query and all the image archive, we can afford a small configurable number of MSE comparisons that can greatly improve the quality of the final ranking. The scheme of all steps of the workflow is summarized in Figure 4.2.

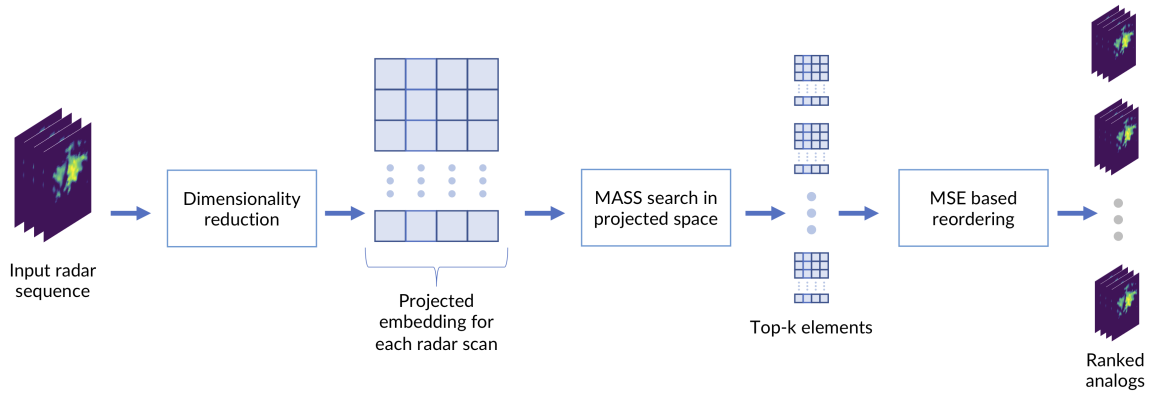


Figure 4.2: The MASS-UMAP workflow.

In an ideal setting, we want the embedding length d to be as small as possible (to reduce the memory requirements and the search time), and to be able to keep k as close as possible to a, to minimize the computation needed by the partial reorder step. For these reasons, we tested the performance of our method with different values of k and d with regard to the ability to return analogs in the same order that an MSE search would.

4.2.5 Evaluation Framework for MASS-UMAP

To better highlight the contribution of the two phases to the overall result, we divided our evaluation in two parts. In the first part (Section 4.2.5.1) we evaluate the impact of dimensionality reduction methods (UMAP at different parameter configurations and PCA) on finding analogues in terms of MSE on single images. We assessed this ability using two different metrics (4.2.5.1, 4.2.5.1). In the second part (Section 4.2.5.2) we used MASS in conjunction

with the best performing dimensionality reduction models, to test the ability of the overall solution in finding analogs of different length (t), using the error computed with respect to sequences found by MSE search in the original (64x64 pixels image) space as ground truth. Computational and memory requirements were also considered.

4.2.5.1 Evaluation part I: dimensionality reduction training and verification on single images

In Section 4.2.1 we discussed the four parameters driving the UMAP dimensionality reduction algorithm. For the purpose of this study we were especially interested in testing two of them: number of components d (that corresponds to the embedding length) and number of neighbors n. Default settings were used for the two other parameters (metric=Euclidean and mindist = 0.1). The rationale for this choice is that for the aim of analog retrieval we were not interested in the absolute distance values, but our objective is to keep the same ranking distance between the elements in the original and the embedded space. This means that any distance function that preserves ranking with regard to MSE can be used, such as the default Euclidean distance used by UMAP. The same holds true for the minimum distance between the points where the projected data will lie. This allowed us to concentrate our effort on the remaining parameters where we chose to setup a grid of 6 values for $D = [2, 5, 10, 15, 20, 100]$ and 6 values of $N = [5, 10, 50, 100, 200, 1000]$ for the model optimization.

Using as input the whole set of search data, we fit 36 UMAP transformations with different parameters, given by the cartesian product of 6 choices of d and n. We then proceed to produce the embeddings of all the UMAPs for both the search data and the query data.

To evaluate the impact of UMAP to preserve rankings, we took the daunting task of computing the MSE distance matrix between all the images

in the archive vs all the images in the query set, thus generating a matrix of $l_s * l_v = 220,050 * 122,548 = 26,966,687,400 = \approx 2.7 \times 10^9$ distances. This matrix allowed us to create an extensive and accurate verification setup of the ability of UMAP to rank and find the same analogs compared to MSE, considering different thresholds of top k elements.

Figure 4.3 illustrates the workflow of the UMAP model training and verification.

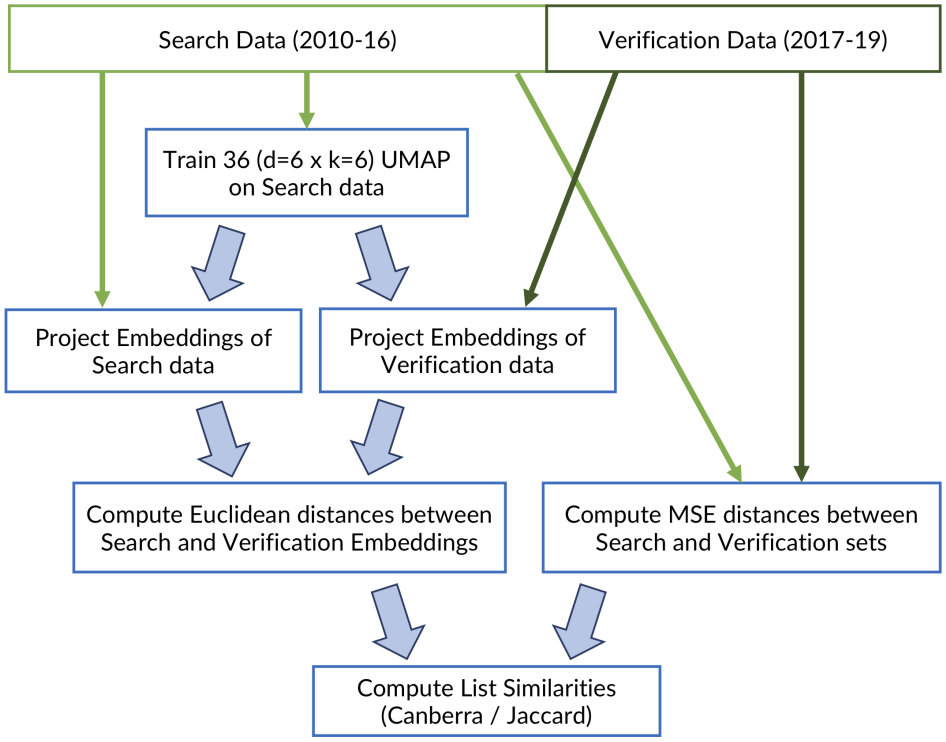


Figure 4.3: Workflow of the model development for the UMAP training and verification. The same workflow is used for training and verification of the PCA, which is used as a comparison method.

As a baseline comparison method we considered the embedding space generated with Principal Component Analysis (PCA) [53]: its use as dimensionality reduction technique for radar analog search has been proposed in [31]. Like UMAP, the PCA embedding maps the original space to a geometric space, so we can use Euclidean calculation to compute the distances. All the steps described for UMAP were applied also for PCA with

only two notable differences: the first one is that we had to train only one configuration since PCA is non-parametric, the second is that we had to apply more normalization steps to our data archive before computing the PCA, because of the variance maximization used to find the principal components is extremely sensitive to unbalanced values. For this step, we followed the same workflow described in [31]: we applied Box-Cox transformation [29] adding a 0.01 offset to the rainfall rate of each radar scan, and centered each element by removing the corresponding mean before computing the PCA.

We tested all the trained dimensionality reduction configurations methods by introducing two evaluation metrics that helped us to measure the characteristics of the ranking results returned by UMAP/PCA against the ideal rankings found via MSE. The first metric is a weighted ranking correctness measure (4.2.5.1), the second measures the proportion of correct elements found in the nearest top k (4.2.5.1).

Stability of ranked lists The Canberra stability indicator [54] $I_{\mathrm{Ca}}(\mathcal{L})$ is a group-theoretical measure for assessing the similarity of a set $\mathcal{L}$ of ranked lists of n shared items. The indicator is based on the Canberra distance [61], a weighted version of the L1 norm whose main features is to penalize more the differences occurring in the top part of the ranked list rather than those occurring at lists' bottom. The indicator is normalized by the expectation E of the Canberra distance on the whole permutation group S_n of cardinality $n!$, so that $0 \leq I_{\mathrm{Ca}}(\mathcal{L}) \leq \max_{\rho,\sigma \in S_n}\{\mathrm{Ca}(\rho,\sigma)\} \approx 1.42$, with $I_{\mathrm{Ca}}(\mathcal{L}) \approx 0$ denotes a set $\mathcal{L}$ of very similar lists, while $I_{\mathrm{Ca}}(\mathcal{L}) \approx 1$ indicate that $\mathcal{L}$ is a randomly ranked set of lists [55]. By using the locator parameter k [55], the same measure can be restricted to evaluate the similarity of the top-k sublists of $\mathcal{L}$ including only the highest ranked items. We used this measure to evaluate how well the top k embedding elements are ranked compared to the MSE ranking.

Jaccard distance The Jaccard distance is a dissimilarity measure between two sets. The Jaccard distance (J_d) is the complementary of the Jaccard index (J) [52], and it is defined as:

$$J_d = 1 - J(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$

where A and B are two set of elements and $\cap/\cup$ are the intersection/union operators.

Intuitively this distance helps to understand what is the proportion of items that are present in both sets, normalized by the total number of distinct elements. A value of $J_d = 0$ is obtained between two identical sets, while a value of $J_d = 1$ corresponds to two disjoint sets. We used this measure to evaluate how many of the top k elements found by searching the embedding space corresponded to elements returned by the top k MSE search. The comparison is also useful to assess and compare the performances between PCA and UMAP for analog retrieval. As with the Canberra stability indicator, the Jaccard distance is used to evaluate the similarity of the top-k sublists of $\mathcal{L}$.

4.2.5.2 Evaluation part II: Sequence search evaluation

In the second part of the evaluation we assessed the complete workflow: we combined UMAP and MASS to test the retrieval of analog sequences of different lengths (t) and different number of top-k sequences. For this part we evaluated the solution by comparing the straight cumulative MSE error between the sequences found by MASS-UMAP and MASS-PCA and the sequences retrieved by MSE. We found this comparison to be a more faithful representation of the performances of the overall solution, than using the two metrics introduced in part I. Indeed, since the testing occurs with sequences of different lengths, the total number of possible matches available

between the query data and the archive is different for every value of t: this makes the interpretation of the metrics much less straightforward. We also benchmark the use of computing resources required both theoretically and experimentally. Wall execution times are reported when available and discussed.

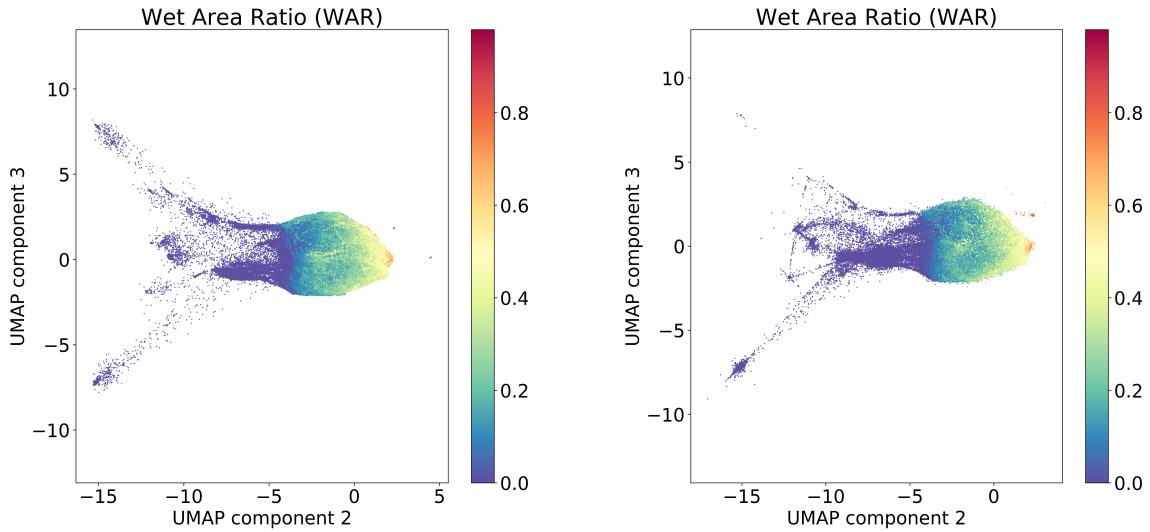
4.3 Results

4.3.1 Exploration of UMAP embeddings

We investigated some of the manifolds generated by UMAP projections and plotted the resulting embeddings for the search and the verification sets (Figure 4.4). The hyper-parameters for the represented model are the one found for the best model in 4.3.2, with number of components $d = 5$ and the number of neighbor $n = 200$. If we consider for example the second and third components, the visualization of the two embeddings belonging to the search set (on which the UMAP is constructed) and the verification set are quite similar, where the embedding points are colored by the Wet Area Ratio (WAR), defined as the percentage of pixels with a rain rate higher than 0.1 mm/h. The stability analysis shows that UMAP is able to project the space maintaining the general distances between radar scans with different rain rates. Notably, the two sets are composed of radar scans collected from 2010 to 2016 for the search set, and from 2017 to 2019 for the verification set. UMAP generalizes well across the two years, and it is applicable to scans coming from future time windows without the need for retrain.

4.3.2 Evaluation part I: UMAP vs PCA

The evaluation of the dimensionality reduction step was implemented as explained in Section 4.2.5.1. 36 UMAP configurations were fitted on the



(a) search set (b) verification set

Figure 4.4: UMAP embedding visualization of the second and third components for search space (a) and for verification space (b). The embeddings are colored by Wet Area Ratio (WAR).

search data, and the embedding of both search and verification data were generated. The UMAP embeddings were used to compute on the fly the ranking distances for all the verification images and compared with the MSE ranking to compute the cardinality of the set intersections and the Canberra indicator for a number of top-k results. The average and standard deviation for Jaccard and Canberra distances on all validation images were then computed for all possible permutations of k (limits), d (components) and n (neighbors). The final number of computed results is given by the cross product of the configuration space built with the following parameters:

- limits: $|K| = 8$ with configurations $K = [5, 10, 15, 20, 50, 100, 200, 500]$
- components: $|D| = 6$ with configurations $D = [2, 5, 10, 15, 20, 100]$
- neighbors: $|N| = 6$ with configurations $N = [5, 10, 50, 100, 200, 1000]$

The total number of parameters tested for the UMAP projection is $|K| * |D| * |N| = 8 * 6 * 6 = 288$. Conversely, for PCA the size of the tested

configuration space was $|K| * |D| = 8 * 6 = 48$, where D is mapped to limit the number of principal components used by the PCA decomposition to the same number of components of UMAP.

The results of the 48 configurations tested on PCA are reported in Fig. 4.5 for Canberra stability index and in Fig. 4.6 for Jaccard. For each configuration we group together the means, the standard deviations, and the suboptimal scenario, namely the sum of mean and standard deviation, describing the retrieval performance of the dimensionality reduction with suboptimal results.

Figure 4.5: Canberra stability indicator results for PCA with different values of limit k and components d (darker/lower is better). Lower values indicates that the configuration better preserves the rankings found computing MSE on the original images. Mean, standard deviation and the suboptimal scenario given by sum of mean and standard deviation are reported.

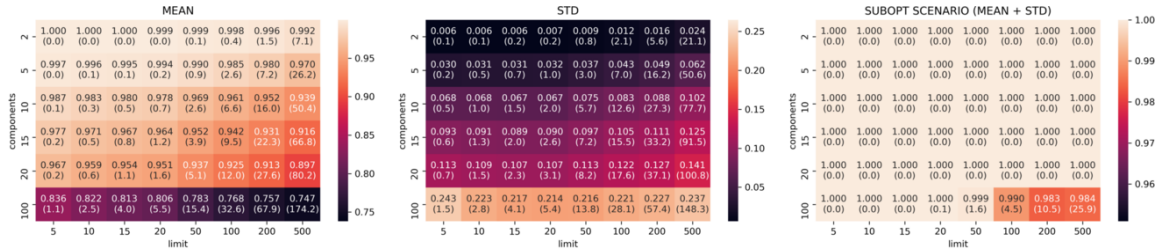


Figure 4.6: Jaccard values for PCA with different values of limit k and components d (darker/lower is better). The number in parentheses is the cardinality of the intersections between the top-k PCA list and the top k MSE list. Mean, standard deviation and the "suboptimal scenario" given by the sum of mean and standard deviation are reported.

Reduction by PCA is consistent and predictable: it shows systematic linear improvements in both Canberra and Jaccard metrics by adding more

components and extending the size of top-k considered elements. On the other hand, PCA needs to use at least 20 components and 500 top-k elements to start showing consistently good ranking results (an average of 80 elements in common with the top 500 MSE elements).

In Figs. 4.7-4.11 the analysis of search reduction by Jaccard distance for different values of n of UMAP are reported. The first observation is that UMAP does not follow a linear trend: the algorithm improves dramatically (40%) between 2 and 5 components, to then subsequently plateau. Going from 5 to 100 dimensions makes hardly any difference in the ability to find better analogs, and this behavior is consistent even with different values of n. Thus, we conjecture that this saturation in the dimensionality is dataset dependent, and that UMAP has already maximized its ability to describe the data manifold using 5 components. On the other hand, choosing a different value for n drastically changes the performance of UMAP with regard to the choice of k. The two values appear to be positively correlated: to train a transformation that finds a consistent number of good analogs in the top-k, we need to set n around the value of k (usually a step lower).

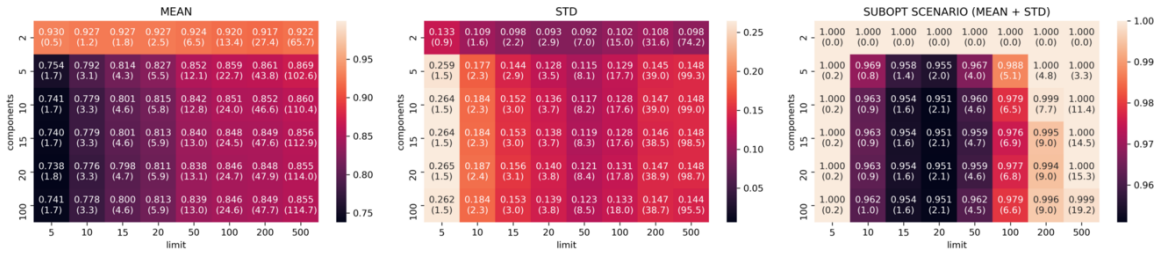


Figure 4.7: Jaccard results for UMAP models trained with neighbors $n = 5$.

Given the consistently good results that UMAP showed using just 5 dimensions and the positive correlation between n and k, we choose the configuration with $d = 5$, $n = 200$ and $k = 500$ as a benchmark for the second part of our evaluation. Figure 4.12 summarizes the findings for the chosen values. In Appendix A.1, we report the plots for all the remaining configurations not included in this section.

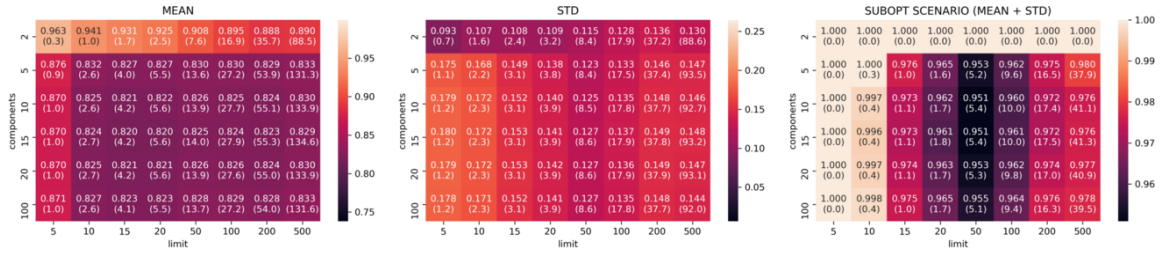


Figure 4.8: Jaccard results for UMAP models trained with neighbors $n = 10$.

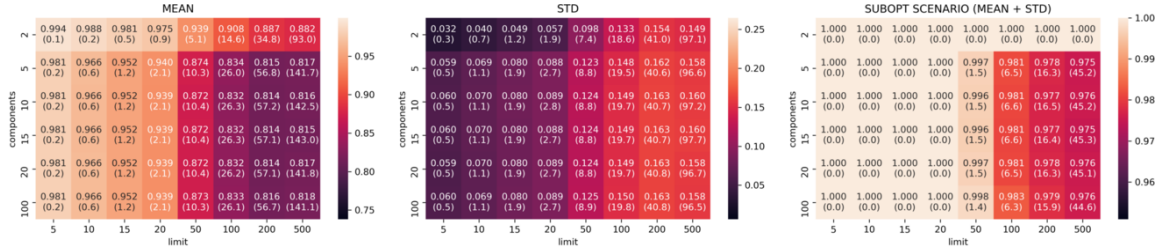


Figure 4.9: Jaccard results for UMAP models trained with neighbors $n = 50$.

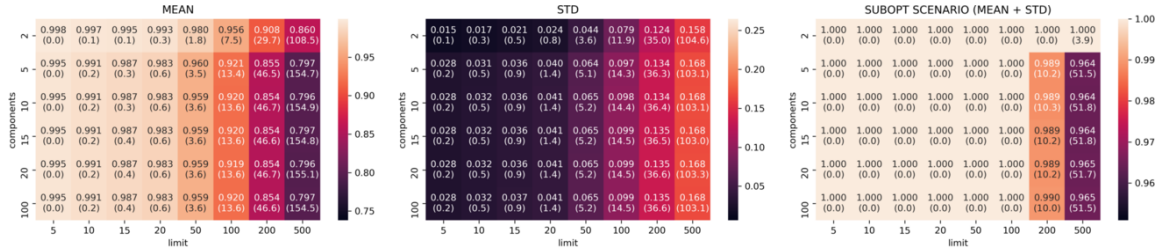


Figure 4.10: Jaccard results for UMAP models trained with neighbors $n = 200$.

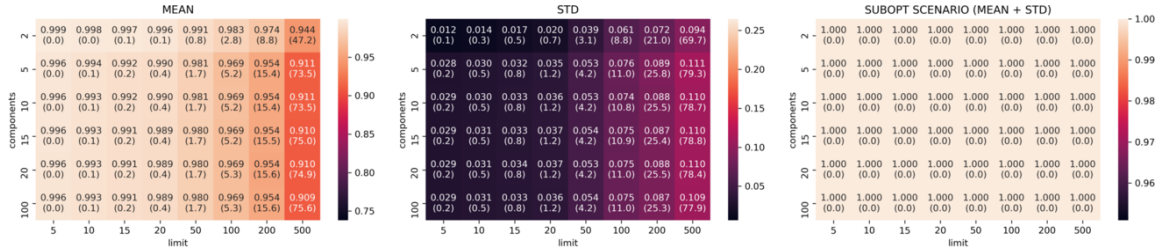
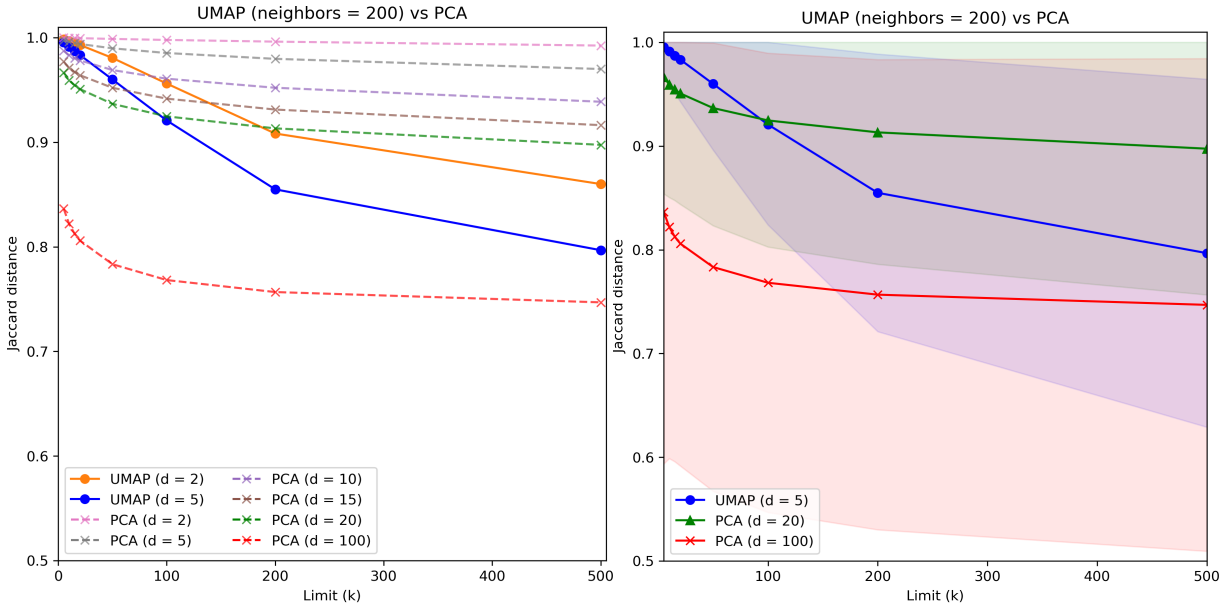


Figure 4.11: Jaccard results for UMAP models trained with neighbors $n = 1000$.

4.3.3 Evaluation part II: sequence search performance

4.3.3.1 Fidelity of the Analog search

The evaluation of the spatiotemporal analog search performance was implemented as explained in Subsection 4.2.5.2. We used the combination of



(a) UMAP ($d = 2$, $d = 5$) vs all PCA dims. (b) UMAP $d = 5$ vs PCA ($d = 20$, $d = 100$).

Figure 4.12: UMAP Jaccard score for the chosen value of neighbor $n = 200$ vs PCA. Only $d = 2$ and $d = 5$ are drawn for UMAP, as the values are overlapping for d from 5 to 100. In (b) the shade represent the standard deviation.

UMAP and MASS (MASS-UMAP) to find analogs for sequences of $|T| = 4$ different lengths. We tested values $T = [3, 6, 12, 24]$ corresponding to sequences of 15, 30, 60 and 120 minutes respectively. For comparability, we used the same number of sequences with the same start times for all values of t. The sequences were chosen from the query set, starting from the first index and leaving whenever possible 100 images of gap between the beginning of the next sequence: this avoided sequence overlapping and also such a gap was sufficiently long to guarantee complete spatiotemporal de-correlation between the chosen sequences. The total number of extracted sequences after such processing was 1226. We compared the best UMAP configuration ($d = 5$, $n = 200$) with respect to the sequences found by MSE and the sequences found by PCA with 5 and 20 components. The figures 4.13, 4.14, 4.15, 4.16 show the mean MSE, of the models with different values of t and a MSE reorder on the top $k = 500$ elements.

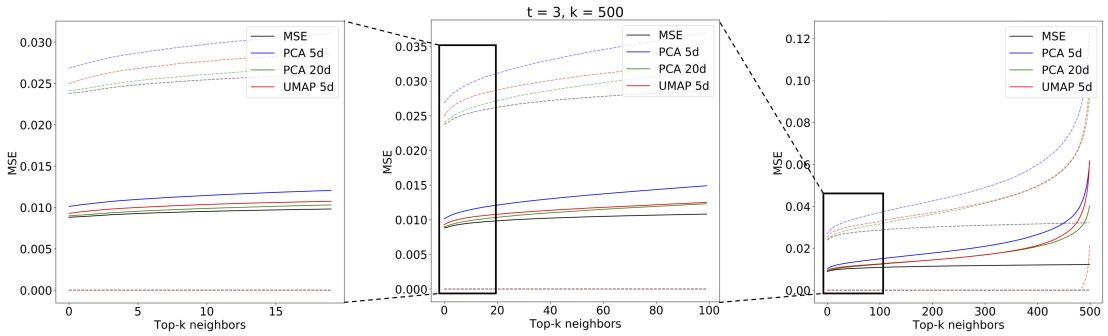


Figure 4.13: Mean MSE values for analog sequences of $t = 3$ obtained with PCA ($d = 5$ and $d = 20$ components), UMAP ($d = 5$ components) and MSE search in original space. Dotted lines represent the standard deviation of the MSE.

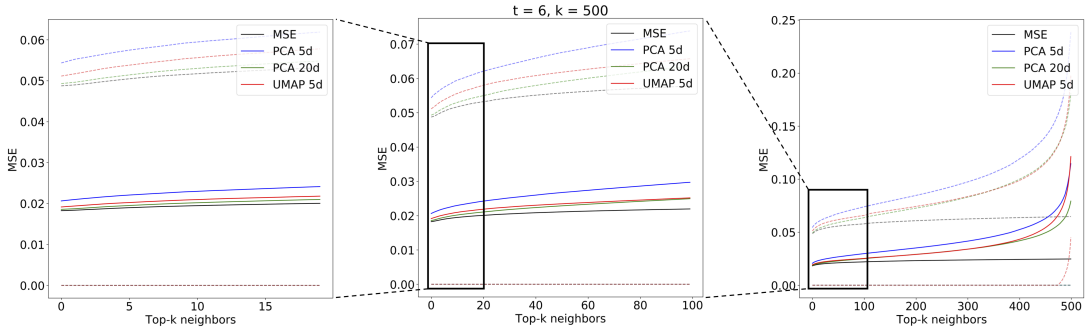


Figure 4.14: Mean MSE values for analog sequences of $t = 6$ obtained with PCA ($d = 5$ and $d = 20$ components), UMAP ($d = 5$ components) and MSE search in original space. Dotted lines represent the standard deviation of the MSE.

The figures show the plot of the average MSE error between the query sequences and the top $k = 500$ sequences for each i-th analog in the ranked list. The results are consistent with the performance on single images: the 5 component UMAP consistently outperforms 5 component PCA on all T by a wide margin. The UMAP results are actually on par with 20 component PCA, where UMAP accounts for slightly higher MSE error for $t = 3$ and $t = 6$ an lower for $t = 12$ and $t = 24$. Results of the top-2 most similar sequences found given a query sample (Fig. 4.17) of $t = 6$ radar scans are shown for the 3 compared methods: using MSE on the original scans (Fig. 4.18) and

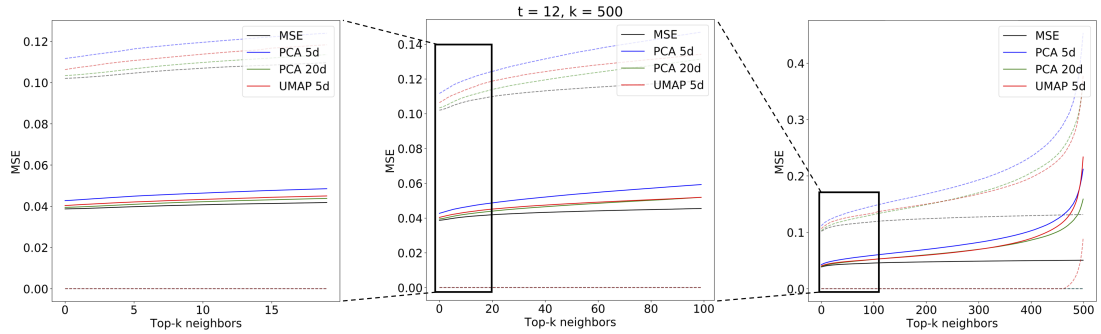


Figure 4.15: Mean MSE values for analog sequences of $t = 12$ obtained with PCA ($d = 5$ and $d = 20$ components), UMAP ($d = 5$ components) and MSE search in original space. Dotted lines represent the standard deviation of the MSE.

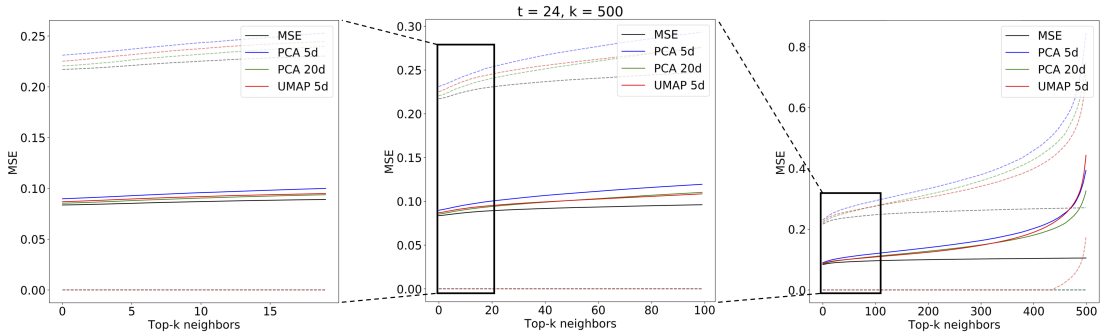


Figure 4.16: Mean MSE values for analog sequences of $t = 24$ obtained with PCA ($d = 5$ and $d = 20$ components), UMAP ($d = 5$ components) and MSE search in original space. Dotted lines represent the standard deviation of the MSE.

using MASS and MSE-based reordering on the top-500 closer embeddings on PCA (Fig. 4.19) and on UMAP (Fig. 4.20) embeddings.

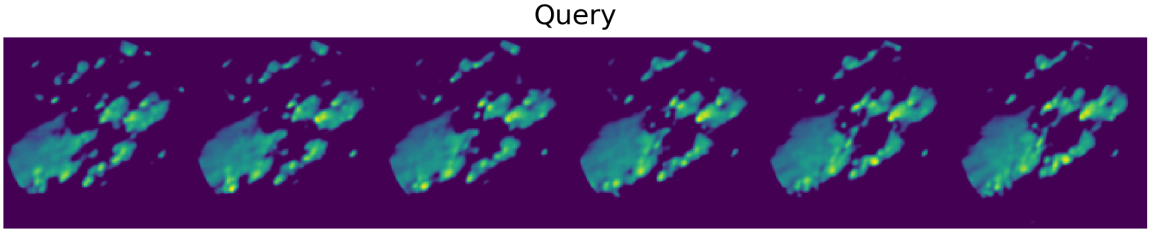


Figure 4.17: Sample query sequence of $t = 6$ radar scans sampled from the verification set.

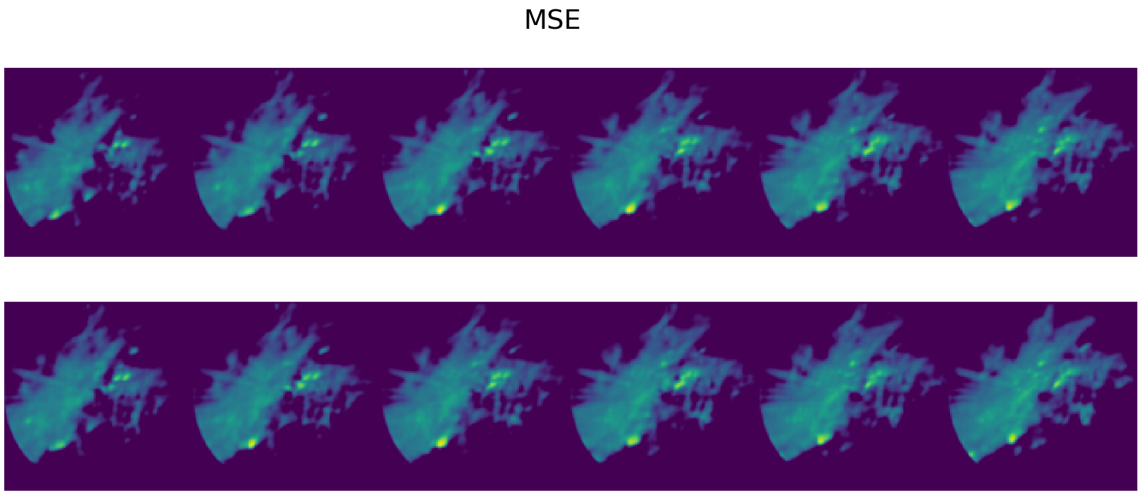


Figure 4.18: Top-2 most similar sequences found in training set for the query sequence shown in Fig. 4.17 using MSE comparison on the original radar scans.

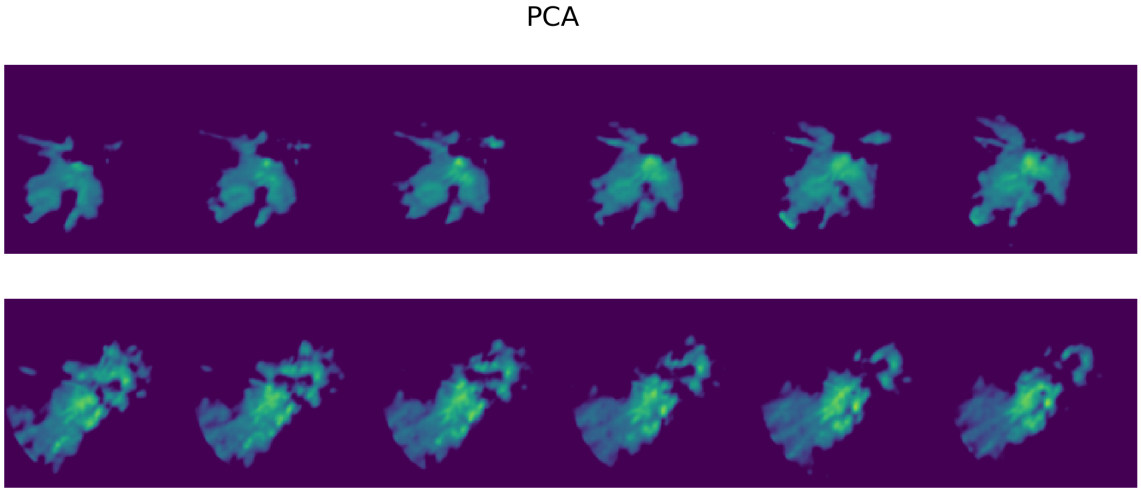


Figure 4.19: As in Fig. 4.18 but searching PCA embeddings ($d = 5$) with MASS. PCA embeddings fail to provide any correspondence with the reference sequences found by MSE.

4.3.3.2 Execution times and memory requirements

We tested the execution times of MASS-UMAP on our dataset first by benchmarking each component of the method separately and then by executing the whole workflow end to end. Tab. 4.2 shows the result of our

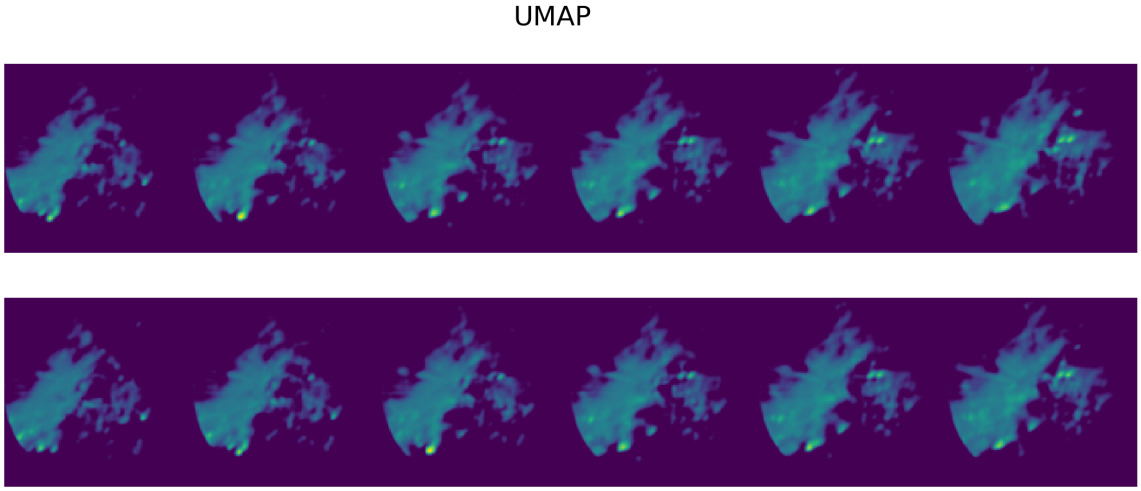


Figure 4.20: As in Fig. 4.18 but searching UMAP embeddings ($d = 5$) with MASS. Although the match is not perfect, UMAP sequences provide at least a partial match with the reference ones in Fig. 4.18.

benchmarking. The UMAP configuration chosen for the test is the same used in 4.3.3.1 with parameters $d = 5$, $n = 200$ and $k = 500$.

All reported tests were executed ten times; confidence intervals are reported. The test platform used was a non-burstable cloud instance with processor Intel(R) Xeon(R) E5-2673 v4 running at 2.30GHz and 425GB of RAM. For a fair comparison all algorithms were executed in single thread mode. The speedups given by MASS-UMAP against linear MSE search were computed by pre-loading all the image archive and query images in memory, so the reported results for the linear MSE search are the fastest possible, with no disk access. While this approach is useful for testing purposes, in real applications this is usually not possible because of memory limits or operational choices, and thus the gap between MASS-UMAP and MSE search will skew even more in favor of the former because of less or no needs of disk access. We remark that the entire archive and query embeddings can fit in 342598 * 5 * 4 bytes = 6.5 MB of memory, while the original radar scans account for more than 5 GB of data even after the 64 x 64 pixel resize.

Sequence Length	3	6
1) UMAP Transform	194 ms ± 6.72 ms	303 ms ± 8.87 ms
2) MASS search	1.01 s ± 9.11 ms	1.05 s ± 13.4 ms
3) top-k MSE reorder	11.1 ms ± .12 ms	43.6 ms ± .72 ms
MASS-UMAP (1+2+3)	1.22 s ± 15.6 ms	1.37 s ± 23.0 ms
MASS-UMAP end-to-end	1.18 s ± 22.5 ms	1.37 s ± 48.4 ms
linear MSE search	9.59 s ± 1.08 s	20.4 s ± 1.6 s
MASS-UMAP speedup	**8.1x**	**14.9x**
Sequence Length	12	24
1) UMAP Transform	451 ms ± 11.3 ms	745 ms ± 15.5 ms
2) MASS search	1.12 s ± 23.1 ms	1.31 s ± 25 ms
3) top-k MSE reorder	86.4 ms ± 1.27 ms	172 ms ± 1.11 ms
MASS-UMAP (1+2+3)	1.66 s ± 35.7 ms	2.23 s ± 35.67 ms
MASS-UMAP end-to-end	1.65 s ± 82.9 ms	2.3 s ± 11.9 ms
linear MSE search	39.5 s ± 3.74 s	1min 24s ± 1.02 s
MASS-UMAP speedup	**23.9x**	**36.5x**

Table 4.2: MASS-UMAP execution times comparison with linear MSE search

4.4 Discussion

The MASS-UMAP method proved to be a flexible and effective method: not only it allows for searching analog sequences of arbitrary length in a few seconds over several years of data archive, but it also improves accuracy over published results [31]. While in this work we focused our analysis only on the minimization of MSE as objective, any positive distance measure can be used to tune the UMAP dimensionality reduction. An example of this would be using a distance measure that is robust to a certain degree of rotation or translation, like the Complex wavelet structural similarity [80], allowing to find analogs accounting for a certain degree of displacements or rotation [6]. The MASS-UMAP method allows also the integration of external variables: synoptic or seasonality descriptors can be integrated by concatenating the

desired variables to the UMAP embedding generated for each image and weighted during MASS search. We also envision the possibility to test functions different from euclidean distance to search the projected space. Finally, the UMAP neighbors parameter n allows to derive the embedding distribution that fine tunes the search results for a specific number of top k analogs. We believe that our verification setup was extensive enough that our findings about the optimal values of d, n and k can be reused as baseline parameters at least for other radar datasets, but we envision the use of the method also for any other remote sensing applications where spatiotemporal search is needed.

The drawback of our solution is that its flexibility comes with a price: some combination of parameters can give worse result than PCA. To avoid this edge cases a proper verification like the setup proposed in this paper is needed. We show UMAP gives substantially better results than PCA on all reasonable number of dimensions in this setup. The same warning holds for MASS: to avoid search results with spurious matches we provide the top-k MSE reordering mechanism to filter spurious matches as analogs. Another aspect that could be investigated is the effect of the use of different spatial resolutions in the training of UMAP: while the image resolution we used was sufficient to describe the variability [97] for the general task we presented, specific applications may require the use of radar images at higher resolution, for example to characterize specific type of precipitation in convective rain cells.

As future research directions we plan to use this methodology as an operational application in probabilistic precipitation nowcasting. Another possibility we envision the usage of ensembles of UMAP trained with different configurations and metric functions to improve the retrieval of analogs in embedded space.

Chapter 5

Stacked Generalization for Nowcasting

One of the most crucial applications of radar-based precipitation nowcasting systems is the short-term forecast of extreme rainfall events such as flash floods and severe thunderstorms. While deep learning nowcasting models have recently shown to provide better overall skill than traditional echo extrapolation models, they suffer from conditional bias, sometimes reporting lower skill on extreme rain rates compared to Lagrangian persistence, due to excessive prediction smoothing. This chapter presents a novel method to improve deep learning prediction skills in particular for extreme rainfall regimes. The solution is based on model stacking, where a convolutional neural network is trained to combine an ensemble of deep learning models with orographic features, doubling the prediction skills with respect to the ensemble members and their average on extreme rain rates, and outperforming them on all rain regimes. The proposed architecture is applied on the TAASRAD19 dataset: the initial ensemble is built by training four models with the same TrajGRU architecture over different rainfall thresholds on the first 6 years

of the dataset, while the following 3 years of data are used for the stacked model. The stacked model can reach the same skill of Lagrangian persistence on extreme rain rates while retaining superior performance on lower rain regimes.

5.1 The problem of Conditional Bias in Deep Learning for Nowcasting

The main challenge faced by nowcasting methods is the progressive accumulation of uncertainty: DL architectures deal with uncertainty by smoothing prediction over time, using the intrinsic averaging effect of loss functions such as mean squared error (MSE) and mean absolute error (MAE), commonly used as loss functions to train DL architectures in regression problems [42]. This smoothing problem can be seen as *conditional bias* (CB): the minimization of MSE leads to models where peak values are systematically underestimated and compensated by overestimation in weak rain-rates [22, 34, 39]. Moreover, the minimization of these two errors is at odds. Quoting Ciach [22]:

> *A dilemma between the minimization of these two errors is demonstrated. Removing CB from the estimates significantly increases MSE, but minimizing MSE results in a large CB that manifests itself in underestimation of strong rainfalls.*

In other words, measures taken to remove CB lead to an increase in MSE, and vice versa, the minimization of MSE results in a higher CB, manifested in an underestimation of high and extreme rain rates.

While not addressing the problem directly, some DL approaches try to cope with CB by introducing weighted loss functions [98], by integrating loss functions used in computer vision [92], or by optimizing for specific rain

regimes [18]. Others avoid the problem altogether by renouncing to a fully quantitative prediction and threshold the precipitation at specific rain-rates, approaching the nowcasting as a classification problem [1, 86]. Unfortunately, while applying modification on the loss function can result in improvement for the general case, the current knowledge on loss functions suggests that this approach alone cannot be used to improve predictions of extreme events [15].

Instead of solely relying on loss function, in this work we improve the prediction skills of DL models, especially for extreme rain rates, by combining orographic features with a model ensemble. Ensemble models are extensively used in meteorology for improving predictions skills, to estimate prediction uncertainty, or to generate probabilistic forecasts [11]. Despite their potential, the use of model ensembles is problematic for deterministic precipitation nowcasting, because model averaging exacerbates the conditional bias problem, leading to attenuation on extreme rain rates [88]. Thus, we use model stacking [106, 96], where the outputs of a DL ensemble and orographic features are combined by another DL model to enhance the skill of existing predictions.

Our contribution is threefold:

1. We introduce the concept of *thresholded rainfall ensemble (TRE)*, where the same DL model and dataset are used to train an ensemble of DL models by filtering precipitation at different rain thresholds.

2. We present *ConvSG*, a DL stacked generalization (SG) model for nowcasting based on convolutional neural networks, trained to combine the ensemble outputs and reduce conditional bias in the prediction.

3. We introduce the concept of *enhanced stacked generalization (ESG)*, where we extend the SG approach with the integration of external variables (in our case orographic features), to further improve prediction accuracy on all rain regimes.

5.2 Tackling Conditional Bias by Stacked Generalization

5.2.1 Adapting the data workflow

For the purpose of this study we leverage the `TAASRAD19` presented in Chapter 2. We split the data by day, and group the radar scans in chunks of contiguous frames, generating chunks of at least 25 frames (longer than 2 hours) and a with a maximum length of 288 frames (corresponding to the whole day). Only chunks with precipitation are kept. Then, we divide the data in two parts: the first period from June 2010 to December 2016 is used to train and validate the model ensemble *TRE*, while the precipitation events from January 2017 to July 2019 are used to generate the ensemble predictions. These are in turn used to train, validate and test the stacked model *ConvSG*. During the latter stage we also test the integration of orographic features in the model chain. Fig. 5.1 summarizes the overall flow of the data architecture used in the study.

5.2.2 Thresholded Rainfall Ensemble for Deep Learning (TRE)

An endless number of approaches can be used to build an ensemble: from using different models on the same dataset, to using the same architecture with different sampling strategies on the input data, to the generation of perturbations in the input or the output of the model, to the use of past similar situations. We base our ensemble on different realizations of the TrajGRU model, given its strength and flexibility for the task. Ideally, a reliable ensemble should be able to sample the complete underlying distribution of the phenomenon[103]. For precipitation nowcasting, the ensemble should be able to fully cover the different precipitation scenarios

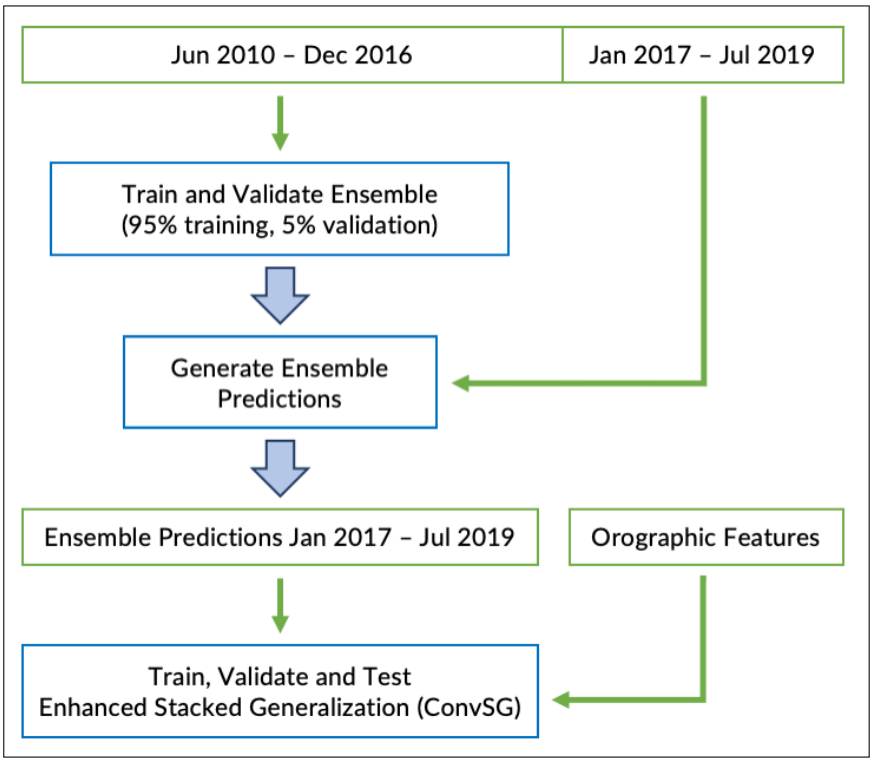


Figure 5.1: Data Architecture of the study. The predictions generated by the ensemble on the test set are used to train, validate and test the Stacked model.

into which the input conditions can develop: in the specific case of extreme precipitations we are interested in modeling the variability of the boundary conditions that can lead to an extreme event. The solution we propose is based on the idea of generating an ensemble that can mimic these different scenarios.

5.2.3 Adapting the base model (TrajGRU)

Two are the common approaches for building an ensemble from a single DL model: either adding random perturbations to the initial conditions of the model or training the model on a different subset of the input space, e.g via bagging [16]. Our solution differs from these approaches and it based on a modification of the loss mechanism described in Chapter 3 to modify the loss

weights of lower rain rate pixels. Specifically we set to zero the weight for pixels under a certain threshold by modifying the computation of the loss as follows:

$$w(x) = \begin{cases} 0 & \text{if } (x \,\epsilon \text{ MASK}) \wedge (R(x) < \text{T}) \\ 1 & \text{if } \text{T} \leq R(x) < 2 \\ 2 & \text{if } 2 \leq R(x) < 5 \\ 5 & \text{if } 5 \leq R(x) < 10 \\ 10 & \text{if } 10 \leq R(x) < 30 \\ 30 & \text{if } R(x) \geq 30, \end{cases} \tag{5.1}$$

where T is a threshold value in the set $T \,\epsilon\, \{0.03,\ 0.06,\ 0.1,\ 0.3\}$, thus building an ensemble of 4 models. With this approach, the model does not need to optimize for all precipitation regimes under the threshold during training, and considers as an optimization target only the higher rain rates. The main consequence of this manifests in a progressive overshooting of the total amount of rain estimate when rising the threshold, which in turn helps targeting higher rain regimes. Fig. 5.2 shows the progressive rise in the average pixel value of the generated predictions of the 4 models on the test set.

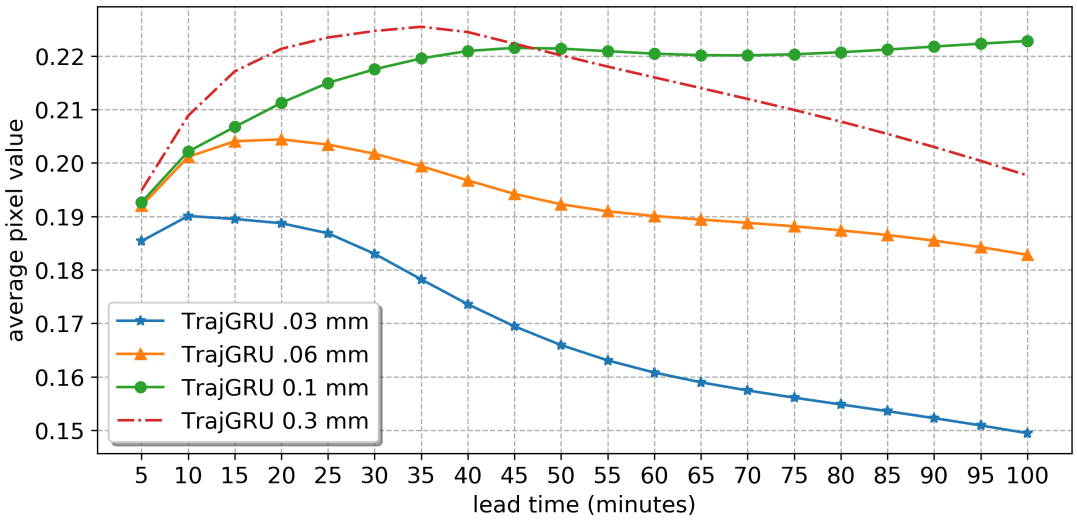


Figure 5.2: Average pixel values (normalized dBZ) of the predictions generated by the 4 models on the test set. When progressively raising the rainfall threshold in the loss, the resulting models progressively increase the total amount of predicted precipitation.

We call this approach *thresholded rainfall ensemble (TRE)*. TRE has a number of very desirable properties: it does not require any sampling in the input data, whereas bagging usually requires to implement complex and often error-prone resampling strategies on the input, and it is able to generate models with significant different behaviors using a single model architecture. Moreover, all the ensemble members in TRE keep as primary objective in the loss function the minimization of the error on the high rain rates. Finally, TRE allows tuning the ensemble spread by choosing a more similar or a more distant set of thresholds, a property that it is not achievable with random data re-sampling or via random parametrization. The only drawback of this method is that the choice of thresholds is dependent on the distribution of the dataset, and thus the generated spread can only be empirically tested. However, the presented thresholds can be reused as is at least on other Alpine radars, and with minor modifications in continental areas. Indeed the thresholds are considered on the actual rainfall rate calculated after the conversion from reflectivity, where all variability given from the physical

characteristics of the radar, background noise and environmental factors have already been taken into account and corrected.

An example of the prediction behavior of the four models is shown in Fig. 5.3, along with the input and observed precipitation.

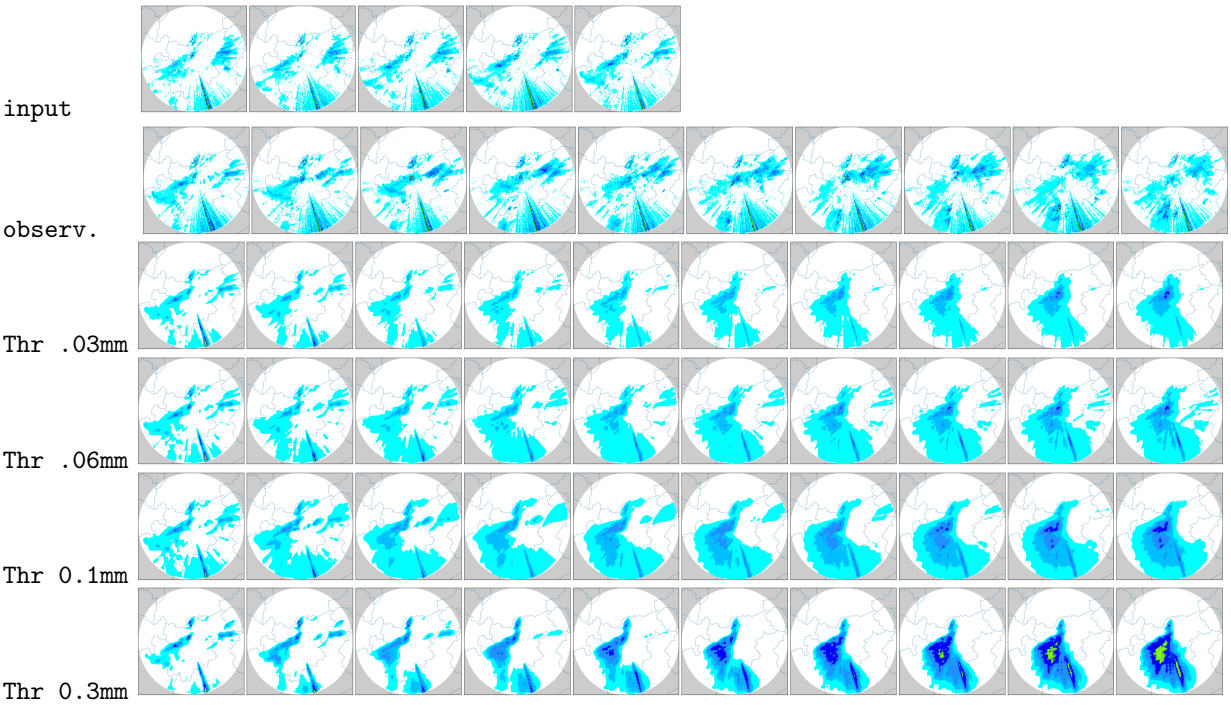


Figure 5.3: Ensemble prediction with TRE valid at 0020 UTC 26 April 2017 (best viewed in color). The first row shows the 5 input scans (25 minutes), while the subsequent rows show the observation (ground truth) and the 4 models output. Observation and prediction are sub-sampled one every two images (10 minutes) to improve representation clarity. The ensemble spread can be observed when rising the threshold value.

As introduced in Fig. 5.1, the four models composing the TRE ensemble are trained on the `TAASRAD19` data from years 2010 to 2016. Using a moving window of 25 frames on the data chunks we extract all the sequences with precipitation in the period, for a total of $202,054$ sequences: 95% $(191,952)$ are used for training while 5% $(10,102)$ are reserved for validation and model selection. All models are trained with the same parameters except for the threshold: fixed random seed, batch size 4, Adam optimizer[56] with learning

rate 10^{-4} and learning rate decay, 100,000 iterations with model checkpoint and validation every 10,000 iteration. For each threshold value the best model in validation is selected as member of the ensemble.

5.2.4 The ConvSG Stacking model

The stacked generalization (or model stacking) approach is a strategy that employs the predictions of an ensemble of learners to train a model on top of the ensemble predictions, with the goal of improving the overall accuracy. The objective of our stacking model is to combine the ensemble outputs to reduce CB in the prediction.

We first generate the stacked model training set, i.e., the predictions for each ensemble member for the data in years 2017 to 2019, for a total of $76,151 \times 4$ set of prediction sequences, where each sequence is a tensor of size $20 \times 480 \times 480$. Given that extreme precipitations are very localized in space and time, we need to preserve both the spatial an temporal resolution of the prediction. Since the theoretical input size for the stacked model results in a tensor of size $4 \times 20 \times 480 \times 480$, memory and computing resources are to be carefully planned: to avoid hitting the computational wall, we developed a stacking strategy based on the processing of a stack of the first predicted image of each model. The approach is driven by the assumption that ensemble members introduce a systematic error that can be recovered by the stacked model and that this correction can be propagated to the whole sequence. For this reason we use only the first image of each prediction for the training of the stacked model, while all the 20 images of the sequences are used for validation and testing.

Given that our target is the improvement of extreme precipitation prediction, we reserve as test set for the stacked model a sample of 30 days extracted from the list of days with extreme events in the years 2017-2019 compiled by Meteotrentino. The resulting number of sequences for the test

set is 6840, corresponding to 9% of the total dataset, while for the validation we random sample 3% of the remaining ($76151 - 6840 = 69311$) dataset, for a total of 2189 sequences. The reason for such low validation split is that, while the training process is only on the first predicted frames, the test and validation are computed on the whole sequence, expanding the test and validation sets 20 times. The final number of images for each set is reported in Table 5.1

Dataset	Sampling strategy	Nr. images
Training	67122 first image of each seq	67122
Validation	2189 (3%) seq. x 20 images	43780
Testing	6840 (9%) seq. x 20 images	136800

Table 5.1: Dataset sampling strategy for the stacked generalization model.

As a sanity check towards excessive distribution imbalances between the 3 sets, we report the data distribution, both in terms of pixel value and rain rate in Tab. 5.2.

The architecture of the Stacked model, ConvSG, is built with the aim to preserve the full resolution of the input image during all the transformations from input to output. The architecture is partially inspired by the work presented in[7]: we adopt the idea to use a resolution-preserving convolutional model with a decreasing number of filters, but we add a batch normalization[51] layer after each convolutional layer to improve training stability and we adopt a parametric ReLU (PreLU) activation and initialize all the convolutional weights sampling from a normal distribution [44] to help model convergence. As a loss function we integrate the loss described in Equation 3.2, by assigning more weight to pixels in the higher rain thresholds. The final architecture is composed by 5 blocks of 5x5 Convolution with stride 1, Batch Normalization and PreLU, and a final 5x5 convolutional output

Rain Rate (mm/h)	(%)	Rainfall Level
$0 < x < 0.5$	95.94	Hardly noticeable
$0.5 \leq x < 2$	3.37	Light
$2 \leq x < 5$	0.47	Light to moderate
$5 \leq x < 10$	0.14	Moderate
$10 \leq x < 30$	0.07	Heavy
$30 \leq x < 1000$	0.01	Extreme

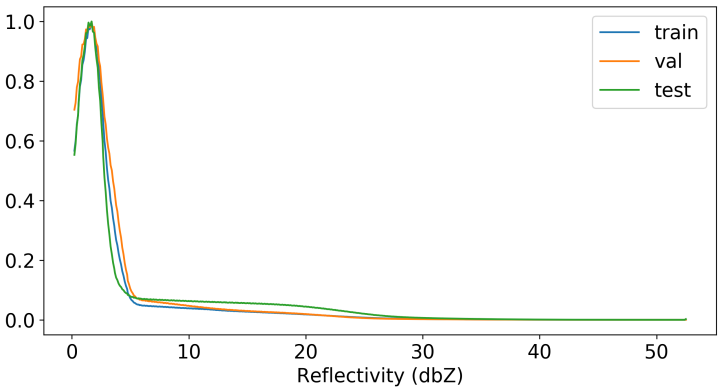
(a) Train

Rain Rate (mm/h)	(%)	Rainfall Level
$0 < x < 0.5$	96.59	Hardly noticeable
$0.5 \leq x < 2$	2.90	Light
$2 \leq x < 5$	0.34	Light to moderate
$5 \leq x < 10$	0.11	Moderate
$10 \leq x < 30$	0.05	Heavy
$30 \leq x < 1000$	0.01	Extreme

(b) Validation

Rain Rate (mm/h)	(%)	Rainfall Level
$0 < x < 0.5$	90.69	Hardly noticeable
$0.5 \leq x < 2$	7.85	Light
$2 \leq x < 5$	1.02	Light to moderate
$5 \leq x < 10$	0.28	Moderate
$10 \leq x < 30$	0.13	Heavy
$30 \leq x < 1000$	0.04	Extreme

(c) Test



(d) Distribution plot of the three sets.

Table 5.2: Distribution of the rainrate values for the three sets used for train (5.2a), validation (5.2b) and test (5.2c). Fig. 5.2d shows the plot of the distribution of the reflectivity values in the three sets. Zero values are removed since they dominate the distribution.

layer. Fig. 5.4 shows the architecture diagram of the ConvSG model along with the expected input and outputs.

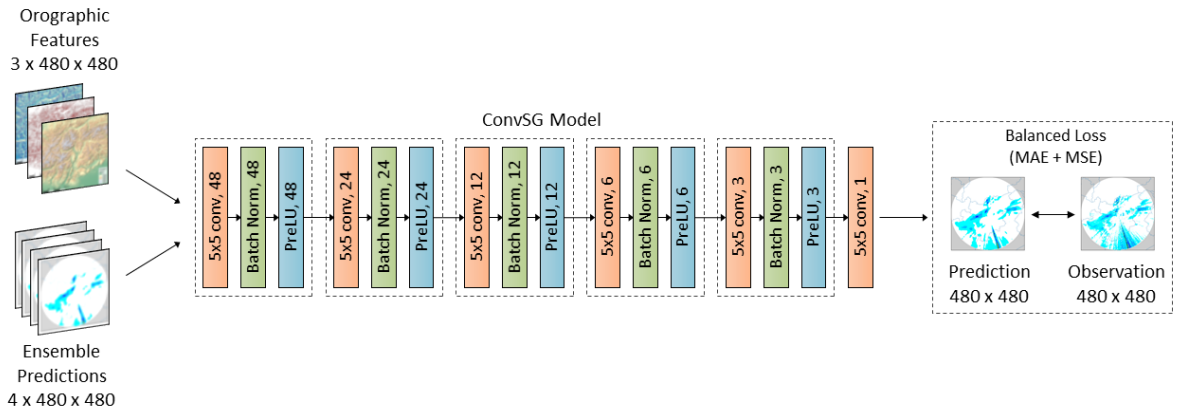


Figure 5.4: The architecture of the DL ConvSG model

For the training of the ConvSG model, we adopt the following training strategy:

- Batch size: 20
- Optimizer: Adam with learning rate $1e^{-3}$
- number of epochs: 100
- validation and checkpoint every 1000 iteration.

For each configuration the best model in validation is selected for testing.

5.2.5 Enhanced Stacked Generalization (ESG)

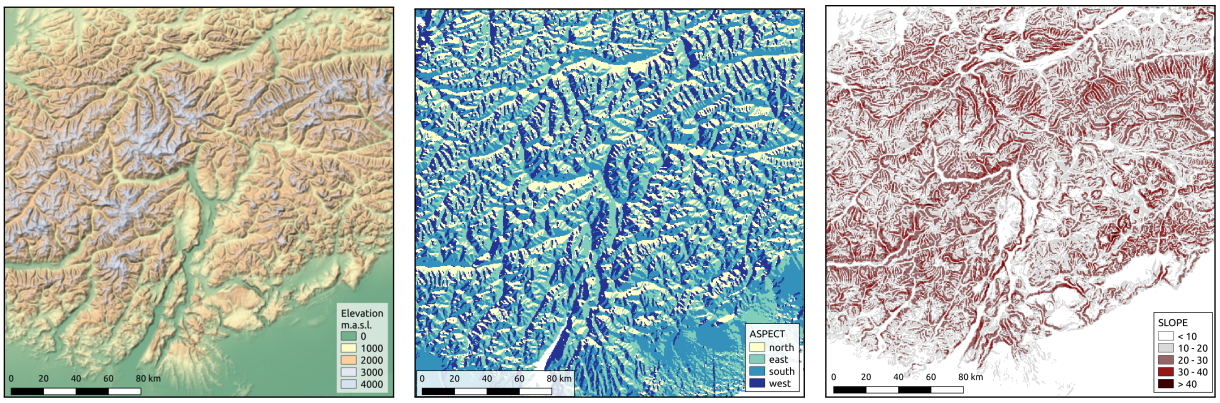
5.2.5.1 Combining Assimilation into ConvSG

We can extend the standard stacked generalization approach by feeding as input to the stacked model not only the prediction of the ensemble, but also additional data sources that can be expected to improve the target prediction: we call this method *Enhanced Stacked Generalization (ESG)*.

There are various reasons why integrating new data during the stacked phase can be helpful. The first is that the integration allows to break down the computation in smaller and faster independent steps, with an additive process. This allows the use of intermediate model outputs in the processing chain to be used for operations that accept to trade off accuracy for a more timely answer, as in operational nowcasting settings. The second reason is that composing different inputs at different stages adds explainability to the overall system. Finally, ESG can help to meet operational budgets in terms of computation or memory resources: in our case, adding the orographic features directly as input to the TrajGRU training process would almost double the memory requirements for the model, forcing us to compromise either resolution or prediction length.

5.2.5.2 Orographic features

Given the complex Alpine environment of the area covered by the `TAASRAD19` dataset and the direct known relationships between convective precipitation and the underlying orographical characteristics [50, 32, 9, 33, 34] we choose to add to the stack of the input images three layers of information, derived from the orography of the area: the elevation, the degree of orientation (aspect), and the slope percentage. The three features are computed by resampling the digital terrain model [30] of the area at the spatial resolution of the radar grid (500m), and computing the relevant features in a GIS suite [73]. Fig. 5.5 shows an overview of the three features, while the distributions of the values are reported in Fig. 5.6.



(a) Elevation map resampled over the radar grid at 500 x 500 m resolution

(b) Orientation derived from the elevation map. The colors show the nearest cardinal direction N (0), E (90), S (180), W (270).

(c) Percentage slope derived from the elevation.

Figure 5.5: Overview of the 3 orographic features used for the ESG model.

The three orographic layers are normalized and stacked along the channel dimension to the four ensemble images, generating an input tensor of size (4 + 3) x 480 x 480 as input to the ConvSG model.

5.2.6 S-PROG Lagrangian extrapolation model

We compare the *ConvSG* model with the S-PROG Lagrangian extrapolation model introduced by [81], here applied following the open-source implementation presented in [75]. S-PROG is a radar-based advection or extrapolation method that uses a scale filtering approach to progressively remove unpredictable spatial scales during the forecast. Notably, the forecasting considers the extrapolation of a motion field to advect the last input radar scan. As a result, S-PROG produces a forecast with increasingly smooth patterns, while only the mean field rainfall rate is conserved throughout the forecast, that is, the model assumes the Lagrangian persistence of the mean rainfall rate. The model is chosen here as a benchmark to the ability of Lagrangian persistence to predict extreme rain rates.

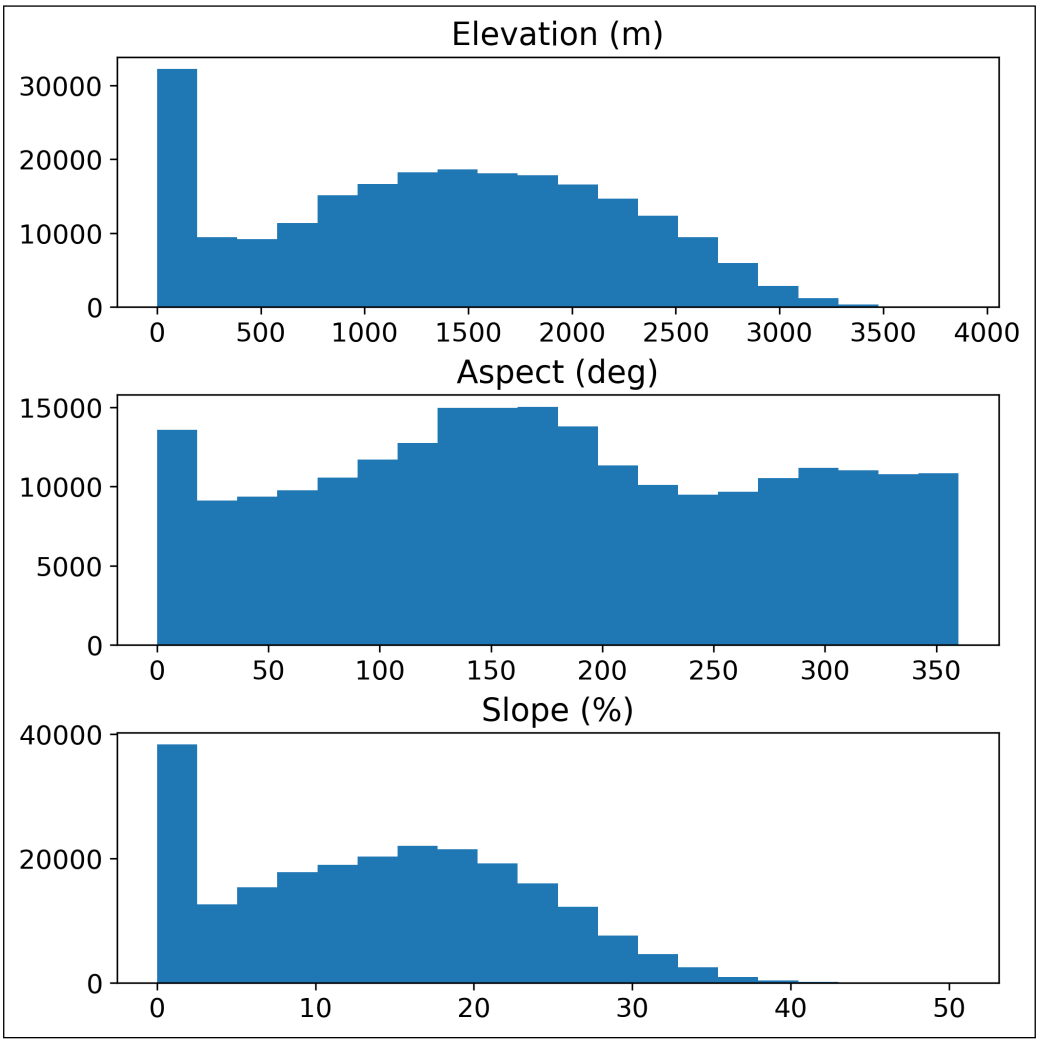


Figure 5.6: Histograms of the 3 topographic features, elevation, aspect and slope (from the top to the bottom). The Y axis of the histogram represents the pixel count for each bin, while the X axis is the value of the elevation in meters, the degree of orientation and the slope percentage respectively. No data values are zeroed.

5.3 ESG model performance

We evaluate the behavior of the various configuration of the ESG models in comparison with S-PROG, with each single member of the ensemble, and with respect to the ensemble mean, by averaging pixel-wise the 4 predictions tensors. To better assess the contribution of each component to the final solution, we perform an ablation analysis showing the contribution

of each of the introduced features (Thresholded Rainfall Ensemble, Stacked Generalization, and Orographic Enhancement) to the final result, presenting and discussing not only continuous and categorical scores but also bias measures.

5.3.1 Categorical Scores

The overall categorical evaluation results are summarized in Tab. 5.3 and Fig. 5.7, that report the comparison of the Critical Success Index (threat score) on the test set for three combinations of ConvSG, along with ensemble members, the mean and S-PROG. The three combinations of ConvSG are shown: (i) the standard Stacked Generalization approach composed by all the four members of the ensemble *ConvSG (Ensemble)*; (ii) the orographic enhanced stacked generalization *ConvSG (Ens + Oro)*; (iii) the best of the four combination of each single model plus the orogrpahy *ConvSG (Single + Oro)*. In this configuration, the best performance are achieved by the *TrajGRU 0.03 mm* model combined with orography.

Except for the threshold 0.5mm, the full ESG model always outperforms all the other DL combinations. The margin grows larger at the increase of the score threshold, and for very heavy rain rates (20 and 30 mm) all the ESG models combinations register noticeable improvements over all members of the ensemble. At 30mm, the full ESG model records a skill that is twice as good as the best performing ensemble member, with a 117.5% increase in skill, and it is on par with the score reported by S-PROG, while retaining superior skills on all the other thresholds. Considering the average of the higher rain rates, the full ESG model shows at least 30% more skill that all the other non ConvSG methods (average, TrajGRU and S-PROG).

The second best performing model is ConvSG (Single + Oro), confirming that the addition of the orographic features induces substantial improvements on all rain regimes and particularly on the extremes. This is also reflected in

CSI Threshold (mm/h)	0.1 (Δ%)	0.2	0.5	avg[0.1, 0.2, 0.5]
S-PROG	0.557 (12.6)	0.502 (14.9)	0.377 (23.4)	0.479 (16.2)
TrajGRU 0.03 mm	0.618 (1.6)	0.553 (4.4)	0.444 (4.8)	0.538 (3.5)
TrajGRU 0.06 mm	0.611 (2.7)	0.567 (1.9)	0.449 (3.7)	0.542 (2.7)
TrajGRU 0.1 mm	0.580 (8.2)	0.567 (1.8)	0.457 (1.9)	0.535 (4.1)
TrajGRU 0.3 mm	0.611 (2.8)	0.570 (1.3)	**0.468 (-0.5)**	0.549 (1.3)
Ensemble AVG	0.625 (0.5)	**0.577 (0.0)**	0.466 (0.0)	0.556 (0.2)
ConvSG (Ensemble)	0.624 (0.5)	0.546 (5.8)	0.420 (10.8)	0.53 (5.0)
ConvSG (Single + Oro)	0.627 (0.1)	0.575 (0.5)	0.463 (0.6)	0.555 (0.4)
ConvSG (Ens + Oro)	**0.628**	**0.577**	0.466	**0.557**

(a) Low Rain-rates

CSI Threshold (mm/h)	1 (Δ%)	2	5	avg[1, 2, 5]
S-PROG	0.241 (49.7)	0.140 (94.8)	0.076 (124.4)	0.152 (76.0)
TrajGRU 0.03 mm	0.353 (2.0)	0.270 (1.2)	0.155 (10.4)	0.259 (3.4)
TrajGRU 0.06 mm	0.350 (3.0)	0.268 (1.8)	0.165 (3.6)	0.261 (2.7)
TrajGRU 0.1 mm	0.353 (2.0)	0.259 (5.3)	0.166 (2.7)	0.260 (3.3)
TrajGRU 0.3 mm	0.345 (4.3)	0.256 (6.5)	0.162 (5.7)	0.254 (5.4)
Ensemble AVG	0.357 (0.8)	0.270 (1.1)	0.171 (0.2)	0.266 (0.8)
ConvSG (Ensemble)	0.344 (4.8)	0.272 (0.4)	0.164 (4.3)	0.260 (3.2)
ConvSG (Single + Oro)	0.357 (1.0)	0.269 (1.4)	0.166 (3.1)	0.264 (1.6)
ConvSG (Ens + Oro)	**0.360**	**0.273**	**0.171**	**0.268**

(b) Mid Rain-rates

CSI Threshold (mm/h)	10 (Δ%)	20	30	avg[10, 20, 30]
S-PROG	0.053 (86.0)	0.037 (30.0)	**0.027 (-0.1)**	0.039 (48.7)
TrajGRU 0.03 mm	0.067 (47.4)	0.016 (210.2)	0.004 (598.0)	0.029 (100.8)
TrajGRU 0.06 mm	0.089 (12.0)	0.031 (56.4)	0.012 (117.5)	0.044 (32.2)
TrajGRU 0.1 mm	0.090 (10.0)	0.031 (58.1)	0.011 (149.7)	0.044 (32.5)
TrajGRU 0.3 mm	0.080 (24.8)	0.028 (73.8)	0.010 (160.5)	0.039 (48.1)
Ensemble AVG	0.081 (22.9)	0.025 (95.0)	0.007 (288.9)	0.037 (54.9)
ConvSG (Ensemble)	0.086 (14.9)	0.034 (43.6)	0.014 (95.6)	0.045 (30.3)
ConvSG (Single + Oro)	0.098 (1.8)	0.046 (4.3)	0.022 (20.7)	0.055 (5.0)
ConvSG (Ens + Oro)	**0.099**	**0.048**	0.026	**0.058**

(c) High Rain-rates

Table 5.3: CSI forecast skill of the ESG models compared with the ensemble and S-PROG (higher is better). In bold the best result, the second best is underlined. The numbers in parentheses show the percentage of improvement of ConvSG (Ens + Oro) over the other methods.

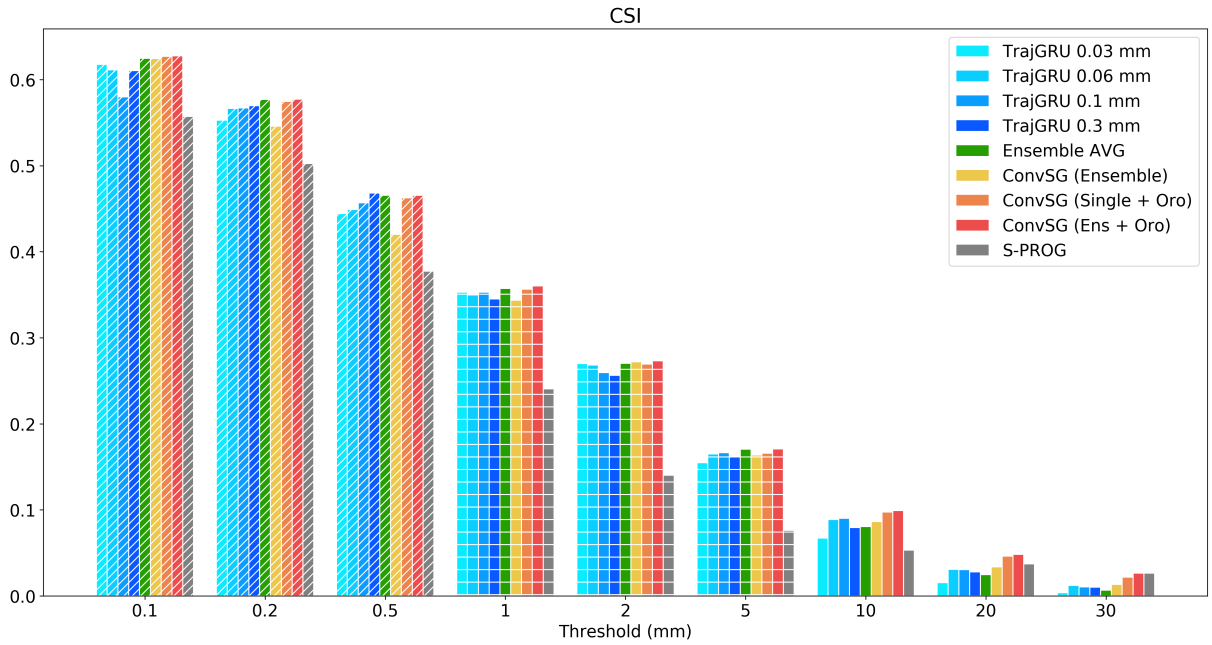


Figure 5.7: CSI score on test set. The dashed, squared and plain pattern in the bars represent the three set of light, medium and heavy precipitation thresholds respectively.

the performance of the ConvSG (Ensemble) model, where a skill increase on the high rain rates is penalized by an poorer performance at lower rain rates.

The framewise comparison shown in figure 5.8 confirms that the increase in skill learned by all the ESG combinations is systematic and does not depend on temporal dimension: as such, the performance increases are consistent across all the predicted timesteps.

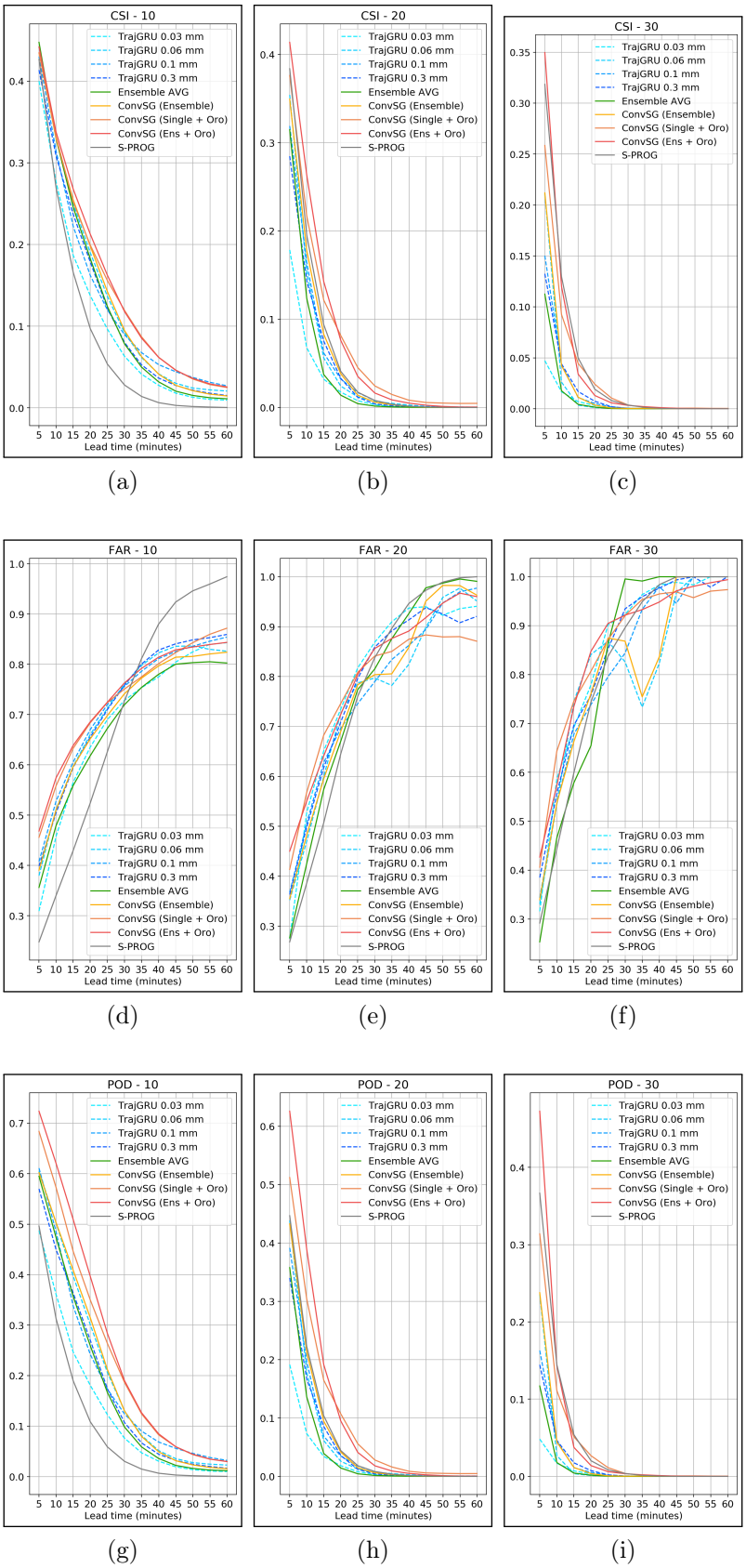
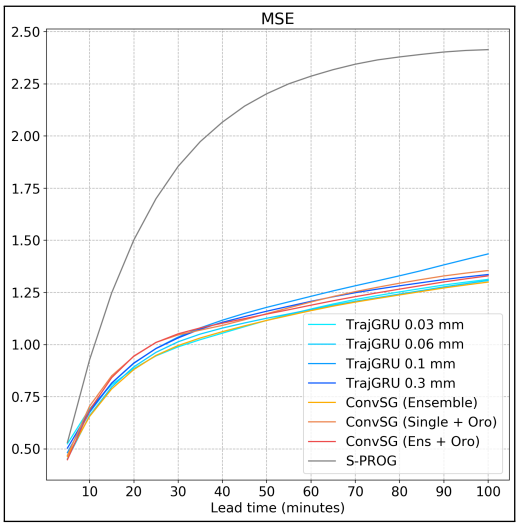


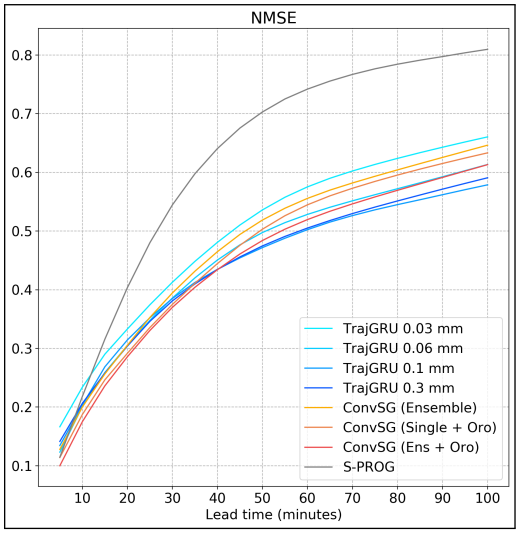
Figure 5.8: Comparison of ESG, ensemble members and average for CSI, FAR and POD scores on heavy and severe rain-rates (10, 20 and 30 mm/h).

5.3.2 Continuous Scores

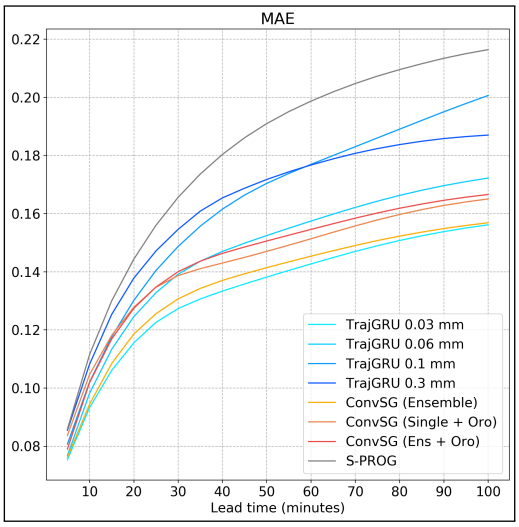
For the continuous scores, along with the standard Mean Squared Error (MSE) and Mean Absolute Error (MAE), we consider two scores that highlight the ability to forecast extreme events. One is the Conditional Bias itself (beta2), the other is the Normalized Mean Squared Error (NMSE), a measure where differences on peaks have a higher weight than differences on other values.



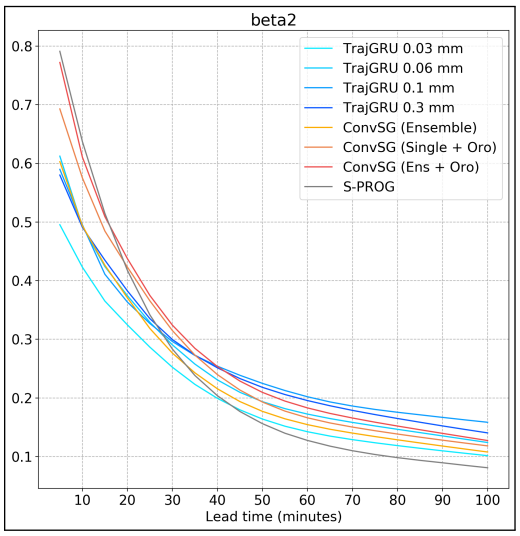
(a) Mean Squared Error



(b) Normalized Mean Square Error



(c) Mean Absolute Error



(d) Conditional Bias (closer to 1 is better)

Figure 5.9: Continuous score performance of the model

The NMSE is expressed as:

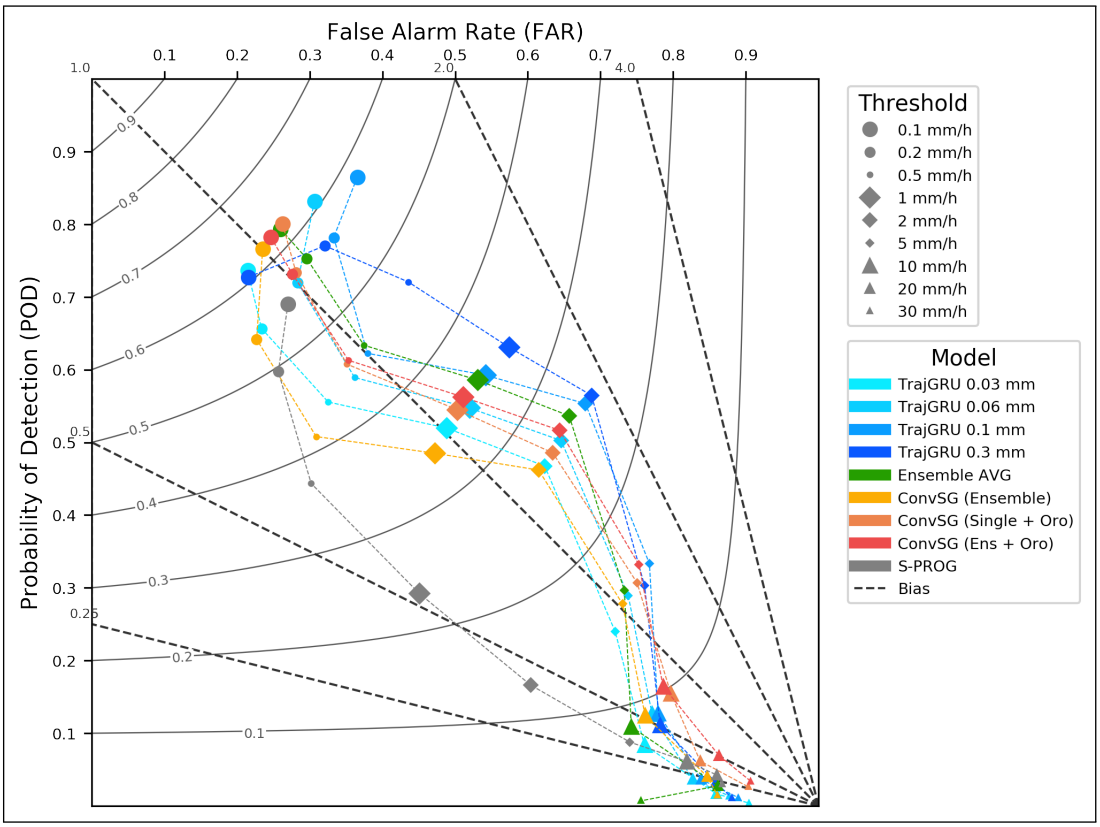
$$\text{NMSE} = \frac{(P - O)^2}{(P + O)^2} \tag{5.2}$$

where P is the prediction and O is the observation, while the CB is computed as the linear regression slope (the covariance divided by the variance of the observation). All the scores are reported in Figure 5.9. As expected the stacked models substantially improve the overall bias (beta2) (Fig. 5.9d) and NMSE (Fig. 5.9b), but have an higher Mean Squared Error (Fig. 5.9a). S-PROG has a comparable bias with the full ESG model in the first lead times, but it is substantially outperformed by all the DL models on all the other measures.

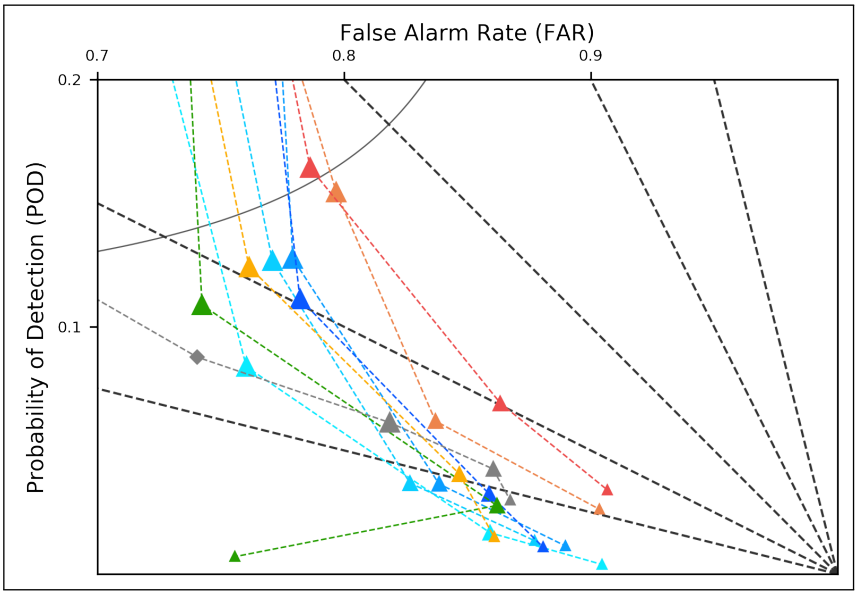
5.3.3 Conditional Bias

To analyze the reduction in conditional bias introduced by ConvSG we introduce the Performance Plot as proposed by Roebber[78] in Figure 5.10. The plot highlights the relationship between all the introduced categorical scores (FAR, CSI, POD) and the frequency bias for all the analyzed rain thresholds on all the presented methods.

All the TRE ensemble members show similar behavior, with a visible conditional bias expressed with the tendency of overestimate low and mid rain rates and a drastic underestimation of high rain rates. Figure 5.10b and Table 5.4 highlights the ability of the ConvSG methods to recover bias on the higher rain regimes. It is interesting to observe the comparison between the full ESG method and S-PROG on the highest rainfall regime (30mm/h): even if the CSI skill is equivalent between the two, the ESG model presents a substantially better bias (61.6% improvement over S-PROG), confirming the superiority of this solution in all cases.



(a) Full CSI contour plot (Performance Diagram)



(b) Zoom of the CSI Contour Plot highlighting the behavior on higher rain-rates.

Figure 5.10: CSI contour plot. Marker locations reflect the average FAR, POD, and CSI values of the 20 time steps for each rainfall rate and model. Dashed lines represent bias (underestimation or overestimation) relative to observation.

BIAS Threshold (mm/h)	10 (Δ%)	20	30	avg[Δ%]
S-PROG	0.338 (127.8)	0.305 (66.2)	0.226 (61.6)	85.2
TrajGRU 0.03 mm	0.350 (119.7)	0.119 (326.5)	0.041 (798.7)	415.0
TrajGRU 0.06 mm	0.553 (39.0)	0.212 (138.3)	0.110 (232.8)	136.7
TrajGRU 0.1 mm	0.578 (33.0)	0.226 (124.0)	0.103 (253.7)	136.9
TrajGRU 0.3 mm	0.511 (50.6)	0.230 (119.8)	0.092 (298.2)	156.2
Ensemble AVG	0.423 (81.9)	0.201 (151.8)	0.029 (1164.4)	466.0
ConvSG (Ensemble)	0.521 (47.4)	0.264 (91.6)	0.107 (239.6)	126.3
ConvSG (Single + Oro)	0.760 (1.1)	0.380 (33.2)	0.271 (34.5)	22.9
ConvSG (Ens + Oro)	**0.769**	**0.506**	**0.365**	**0.0**

Table 5.4: Average bias over the 20 time steps for high rain-rates. The ESG models are compared with the ensemble and S-PROG (closer to 1 is better). In bold the best result, the second best is underlined. The numbers in parentheses show the percentage of improvement of ConvSG (Ens + Oro) over the considered method.

5.3.4 ESG output example

Figure 5.11 shows an example of the outputs obtained by the ensemble members and their average, along with the result of the ESG. It can be observed that ESG retains more variability in prediction especially in the higher rain regimes.

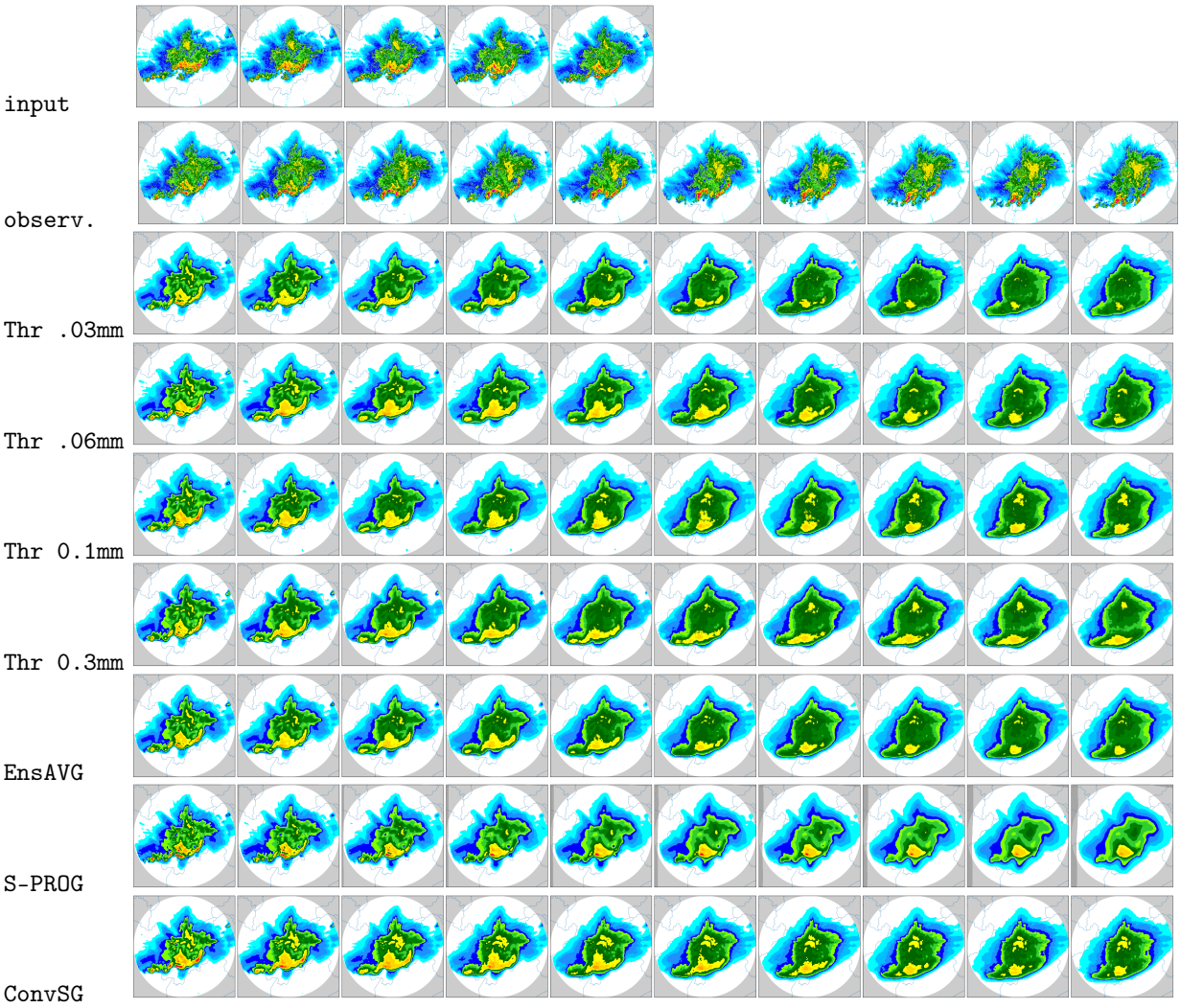


Figure 5.11: TRE Ensemble members, Ensemble average, S-PROG and ConvSG (Ens + Oro) prediction on test at 1535 UTC 03 July 2018 (best viewed in color). The first row shows the 5 input scans (25 minutes), while the subsequent rows (50 minutes) show the observation (ground truth), the 4 models output, the ensemble average, the lagrangian extrapolation model and the stacked generalization output.

5.4 Discussion and Considerations

5.4.1 ConvSG behavior

The results reported in Section 5.3 show that *ConvSG* can substantially improve both the predictive skill and reduce the bias of DL models on extreme

rain rates. When the SG is trained only on the ensemble predictions, with no additional information, the *ConvSG* model is able to leverage the ensemble spread to trade off predictive performance on the lower rain rates for an improvement in skill and bias in high and extreme thresholds. This behavior may be an instance of the *"no free lunch"* duality between the choice of reducing either CB or MSE. On the other hand, the integration of orographic features extracted from the digital terrain model results in a gain in predictive skills over all the rain rates, with the largest improvements registered on the high rain regimes along with a substantial improvement in bias (less underestimation).

As expected, the best performing model is thus given by the combination of both the ensemble and the orography, where the skill score on the extremes is on par with S-PROG, whose skill is mainly driven by persistence.

5.4.2 Comparing ConvSG and S-PROG

While the scores of S-PROG and ConvSG are similar on the extremes, there is also a fundamental qualitative difference between the predictions generated by the DL approach and the Lagrangian extrapolation. Indeed, the ability of the DL to substantially reduce the bias, and to correctly model the growth and decay of the precipitation patterns in different locations in space. An example can be observed in Fig. 5.11, where the ConvSG model is able to forecast the intensification of the rain rate in the upper section of the precipitation front, whereas S-PROG models a gradual decay. This ability opens the possibility for the DL model to eventually forecast new extremes, a behavior not possible by assuming Lagrangian persistence. This reflects in the trend reported by the CB score shown in Fig. 5.9d: S-PROG has the best score in the first few frames but quickly decays to the worst score after 40 minutes of lead time. For the NMSE score (Fig. 5.9b), S-PROG is competitive only in the first lead time, and quickly decays thereafter. Finally,

for MSE and MAE (Fig. 5.9a-5.9c), ConvSG is superior to S-PROG because the two scores are more indicative of the skills obtained in the lower rain rates. This yields that an effective model evaluation and comparison can be correctly performed only when multiple thresholds for the categorical scores and multiple continuous scores are included in the analysis.

5.5 Conclusions and Future work

We presented a novel approach, leveraging a DL ensemble and stacked generalization, aimed at improving the forecasting skills and reducing the conditional bias of DL nowcasting models on extreme rain rates. The proposed method doubles the forecasting skill of a DL model on extreme precipitations and significantly improves the bias, when combining the ensemble along with orographic features. Our contribution is threefold:

1. the *thresholded rainfall ensemble (TRE)*, where the same DL model and dataset can be used to train an ensemble of DL models by filtering precipitation at different rain thresholds;

2. the Convolutional Stacked Generalization model (*ConvSG*) for nowcasting based on convolutional neural networks, trained to combine the ensemble outputs and reduce CB in the prediction of high rain rates;

3. the *enhanced stacked generalization (ESG)*, where the SG approach is integrated with orographic features, to further improve prediction accuracy and bias reduction on all rain regimes.

The approach can close the skill gap between DL and traditional persistence based methods on extreme rain rates featuring significantly improved bias, while retaining and improving the superior skill of the DL methods on lower rainfall thresholds, thus reaching superior performance than all the analyzed

methods in all situations. As a drawback, its implementation requires a non trivial amount of data and computation to train and correctly validate all model stack, along with some knowledge of the data distribution for the selection of the thresholds. Indeed, the presented ensemble size of four models was chosen as the minimum working example for *TRE*, to satisfy the computational budget limits for the DL stack. We thus expect that, incrementing the number of members and the corresponding thresholds, the contribution of the ensemble to the overall skill of the Stacked Generalization will increase. Further experiments are needed to more formally determine the thresholds and the number of the ensemble members required to maximize the desired skill improvements on the extremes. Moreover, despite the presented improvements, the absolute skill provided by nowcasting systems on extreme rainfall is still lagging in the single digit percentage, leaving the problem of extreme event prediction wide open for improvements. As future work, we plan to test the integration of new environmental variables in the ESG model along with orography, and to leverage the ensemble spread to model prediction reliability and develop an extreme event detection index.

Chapter 6

Conclusions

We focused on developing new advancements in spatiotemporal nowcasting using deep learning models, with a specific focus on extreme precipitations. We contributed a new dataset specifically aimed at the development of machine learning models, to foster reproducibility of nowcasting experiments. We briefly reviewed some state-of-the-art solutions and established a baseline for improvements on the newly released dataset. On top of this work, we developed two new solutions with growing complexity based on ensembles. In the first part, we presented MASS-UMAP, an approach to reduce the computational burden of analog search by demonstrating the efficiency of a combined approach based on 3 steps: dimensionality reduction, fast search in constant time in the frequency domain, and MSE-based reordering on a subset of potential candidates. MASS-UMAP is able to reduce the computational complexity of analog retrieval by a factor of 20 with respect to published results; further, its performance are independent of the length of the retrieved sequences and the similarity of the retrieved analogs is improved over state-of-the-art methods. Such a framework is paving the way for future implementation of a probabilistic nowcasting system based on the retrieved analogs. We also presented ESG, a novel deep learning architecture aimed at improving the forecasting skill on extreme precipitations, through the reduction of the conditional bias. To achieve this result we introduce both

a suite of new methods to construct a deep learning ensemble and a novel stacking model that can combine the ensemble outputs with new data sources. Leveraging this architecture we are able to more than double forecasting performance on extreme precipitations by introducing orographic features in the model. As a possible limitation of the ESG architecture we notably list the substantial amount of data required to correctly train all the models in the stack. Possible future research directions include the study of the integration of multiple weather variables both in the ensemble and in the stacking phase.

6.1 A unifying vision for the operational nowcasting of extreme events: the Extreme Nowcasting Framework

Weather nowcasting of extreme events presents specific challenges that, as has been demonstrated in this thesis, can find a valuable ally for their solution through machine learning and deep learning methods. It is expected that the contribution of these approaches in all the aspect of meteorology will grow substantially in the next decade, leading to several new improvements in forecasting abilities, in terms of improved skills, lower computation times and increase in spatiotemporal resolution.

While future research directions are open for many possible paths, it is already possible to delineate a unifying vision for the operational usage of all the techniques presented in this thesis. The proposed framework, which we call the Extreme Nowcasting Framework, is inspired by the Extreme Forecast Index (EFI)[60, 115] developed by ECMWF, a product that is operationally available to be used by meteorologists, and that classifies if the currently produced forecast can be considered an extreme for a certain location in the current period of the year. The EFI for a location is computed by comparing the cumulative distribution function (CDF) built

on the current ensemble forecast, with the CDF computed from all the past forecasts of the same period of the year in the last 20 years: this CDF is called Model Climate (MClimate). Given the peculiar characteristics of precipitation nowcasting (very short temporal extension of the prediction, and high spatiotemporal resolution and variability), restricting extraction of the past forecast only to the same time of previous years can lead to extremely misleading classifications. We thus propose to compare the current nowcasting CDF with the CDF built by the ensemble of the most similar past forecasts: we call this CDF the Model Analog (MAnalog). The MAnalog for current nowcasting can be quickly computed by leveraging the MASS-UMAP framework to extract analogues from the archive of past forecasts and then compare it with current nowcasting CDF computed on the TRE ensemble, generating an Extreme Nowcasting Index (ENI). Given the capabilities offered by deep learning nowcasting models to complete model inference in a few seconds on modest hardware, and the ability to perform a fast search for analogues, we envision the possibility of executing this framework every few minutes with modest hardware requirements, thus enabling the operational usage of the ENI by national and local meteorological agencies for real-time nowcasting and alerting purposes.

Acknowledgements

First, I would like to express my gratitude to my supervisor, Cesare Furlanello, and all the colleagues at FBK. Without the guidance of Cesare and the tremendous support from all MPBA lab I would have never been able to accomplish what I did. I would like also to thank Guido Cervone at PSU for my wonderful period abroad in State College. I also want to thank the Civil Protection Departments of the two Autonomous Provinces of Trento and Bolzano for their help and their daily commitment in the maintenance of all weather data collection systems. A special thanks also to all the people working at Meteotrentino.

My last words are for everyone out there that are supporting me: thank you, thank you so much. You know who you are.

Publications

6.2 International Journals and Conferences

1. Franch, G; Jurman, G; Coviello, L; Furlanello, C. MASS-UMAP: Fast and Accurate Analog Ensemble Search in Weather Radar Archives; Remote Sensing, 11(24), 2922, 2019

2. Franch, G; Maggio, V ; Coviello, L; Pendesini, M.; Jurman, G ; Furlanello, C. TAASRAD19, a high-resolution weather radar reflectivity dataset for precipitation nowcasting. Sci Data 7, 234, 2020

3. Franch, G.; Nerini, D.; Pendesini, M.; Coviello, L.; Jurman, G.; Furlanello, C. Precipitation Nowcasting with Orographic Enhanced Stacked Generalization: Improving Deep Learning Predictions on Extreme Events. Atmosphere 11, 267, 2020

4. Xu, F.; Cervone G.; Franch, G.; Salvador M. Multiple geometry atmospheric correction for image spectroscopy using deep learning; J. Appl. Rem. Sens. 14(2) 024518, 2020

5. Franch, G; Nardelli, A; Zarbo, C; Maggio, V; Jurman, G.; Furlanello, C. Deep Learning for rain and lightning nowcasting. (Poster) NIPS 2016 workshop on ML for Spatiotemporal Forecasting, 2016.

6.3 Datasets and software

1. Franch, G; Maggio, V.; Coviello, L; Pendesini, M.; Jurman, G.; Furlanello, C. TAASRAD19 Radar Scans 2017-2019. Zenodo, 2019

2. Franch, G; Maggio, V.; Coviello, L; Pendesini, M.; Jurman, G.; Furlanello, C. TAASRAD19 Radar Scans 2010-2016. Zenodo, 2019

3. Franch, G; Maggio, V.; Coviello, L; Pendesini, M.; Jurman, G.; Furlanello, C. TAASRAD19 Radar Sequences 2010-2019. Zenodo, 2019

4. Franch, G; Maggio, V.; Coviello, L; Pendesini, M.; Jurman, G.; Furlanello, C. TAASRAD19 Code Release. GitHub, 2019

Appendix A

Appendix

A.1 Jaccard and Canberra extended results

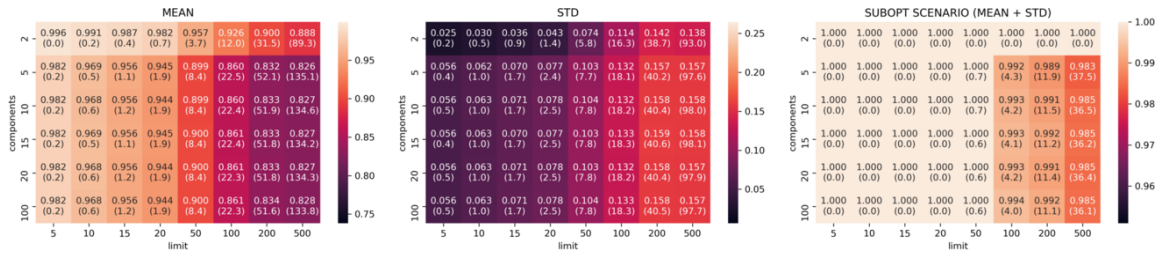


Figure A.1: Jaccard results for UMAP models trained with neighbors $n = 100$.

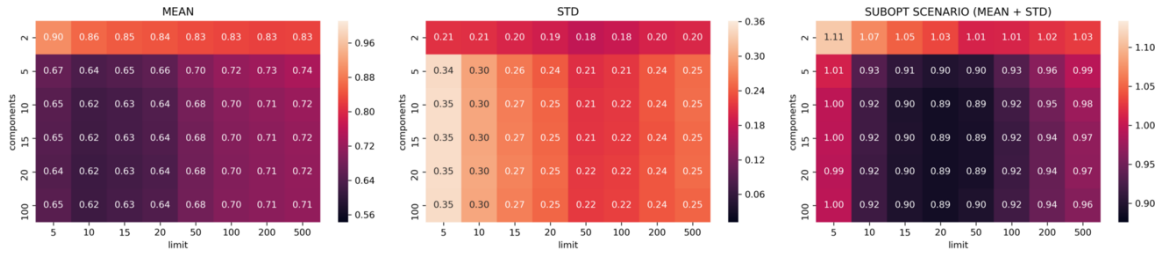


Figure A.2: Canberra results for UMAP models trained with neighbors $n = 5$.

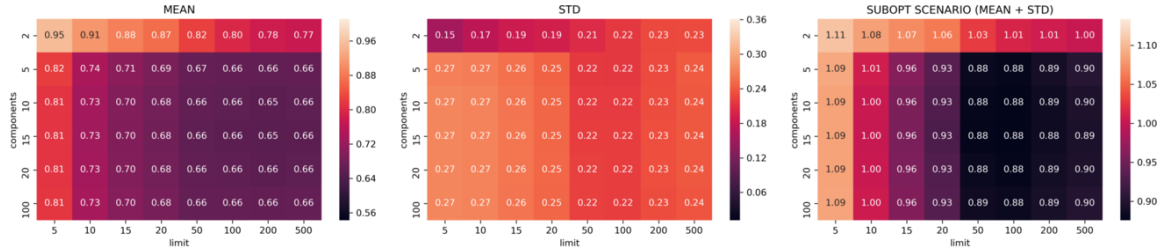


Figure A.3: Canberra results for UMAP models trained with neighbors $n = 10$.

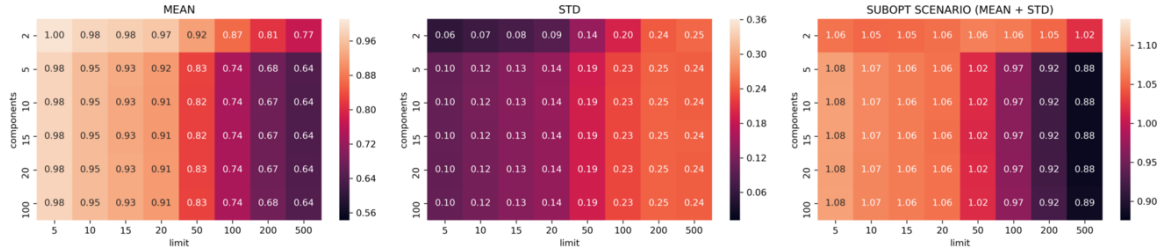


Figure A.4: Canberra results for UMAP models trained with neighbors $n = 50$.

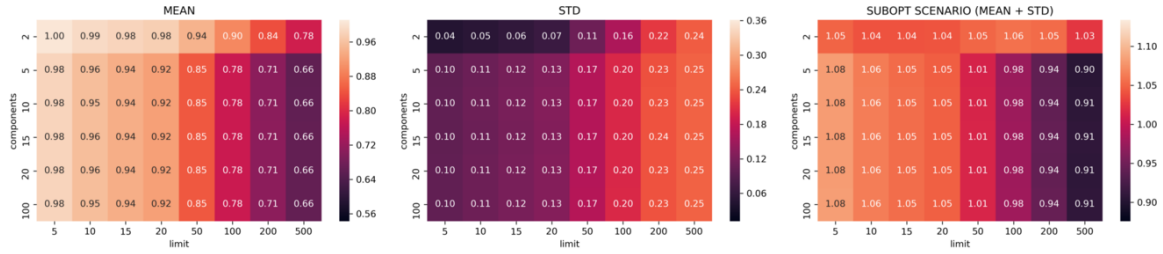


Figure A.5: Canberra results for UMAP models trained with neighbors $n = 100$.

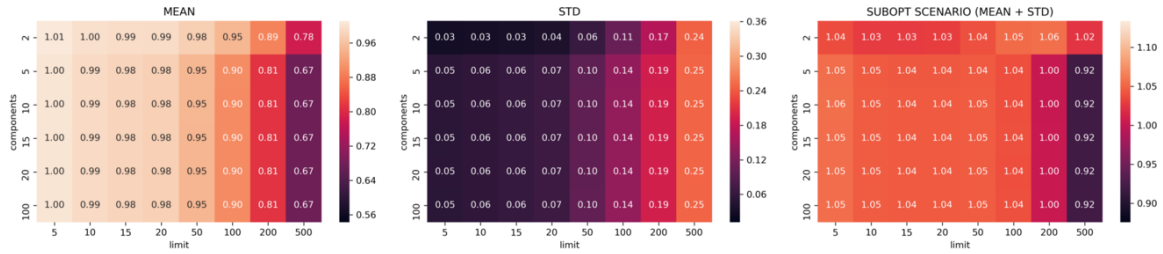


Figure A.6: Canberra results for UMAP models trained with neighbors $n = 200$.

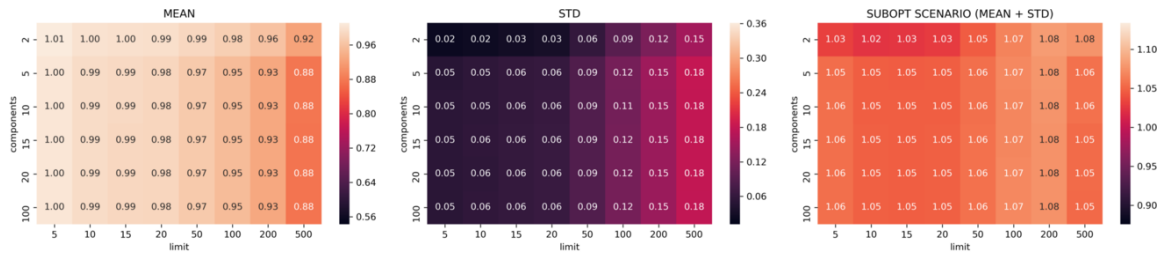


Figure A.7: Canberra results for UMAP models trained with neighbors $n = 1000$.

A.2 Umap Embedding Mosaics

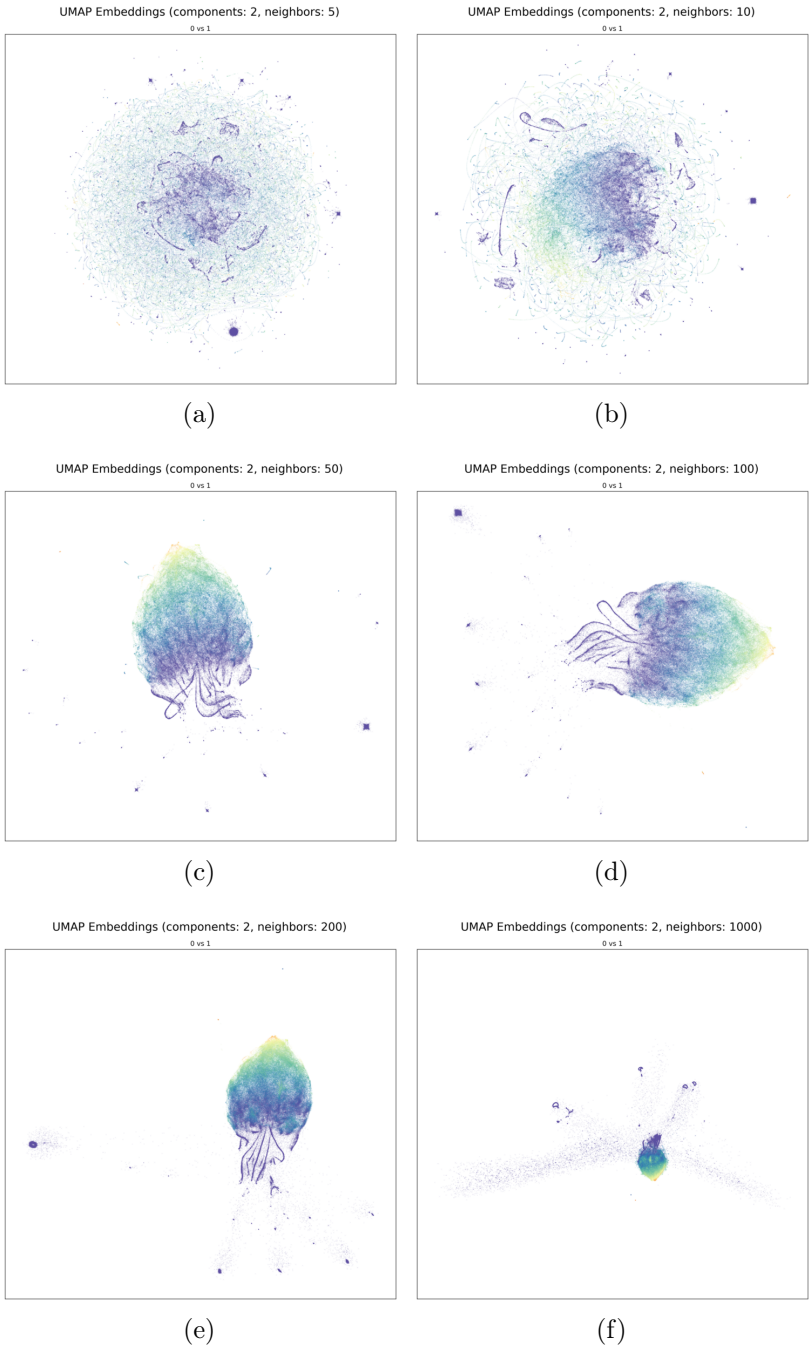


Figure A.8: Example of UMAP Embeddings that show the effect of using different neighbors parameters (n) in two dimensions ($d = 2$) on the training set, colored by Wet Area Ratio

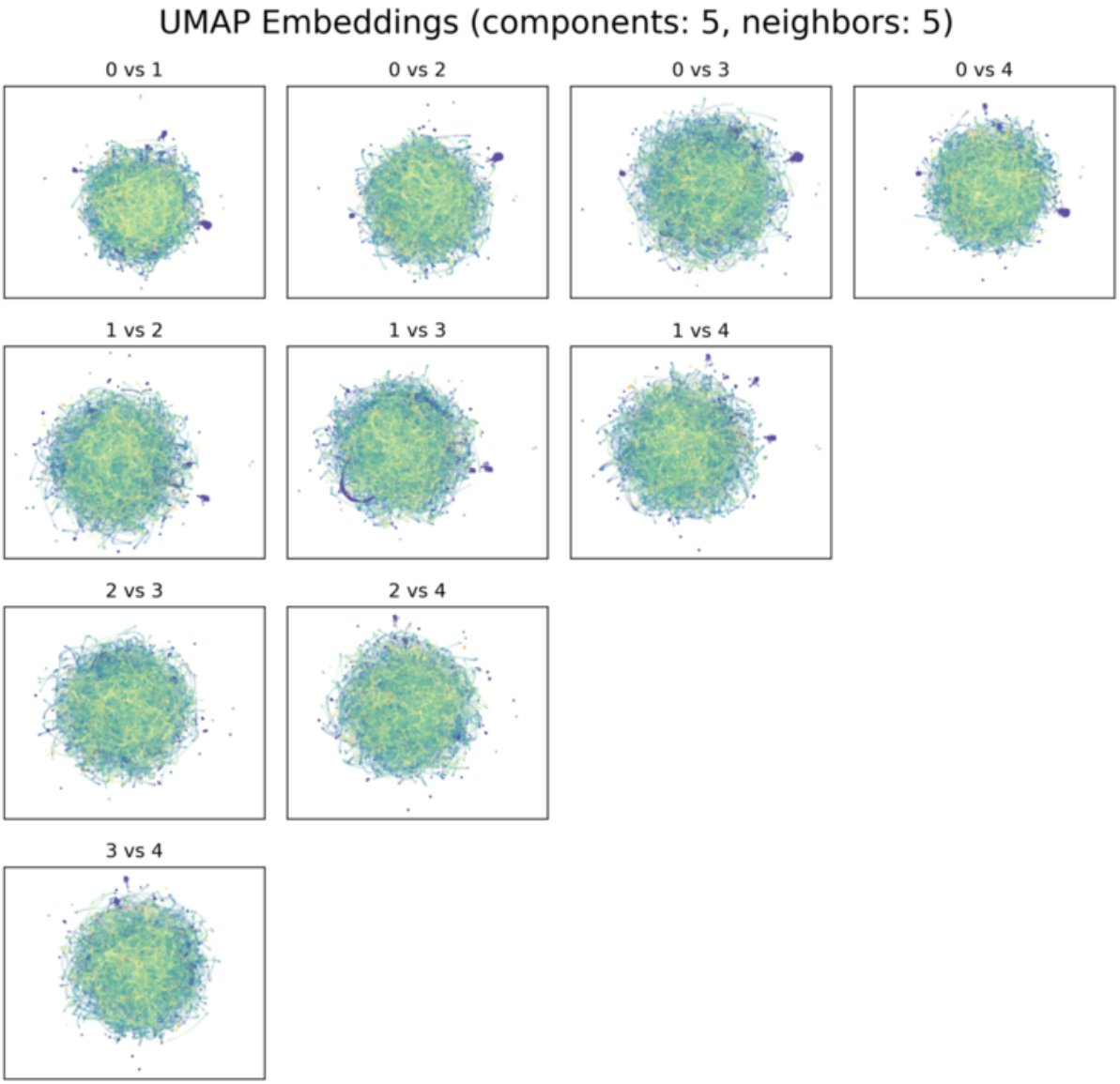


Figure A.9: UMAP Embeddings of the training set colored by Wet Area Ratio for $d = 5$ and $n = 5$.

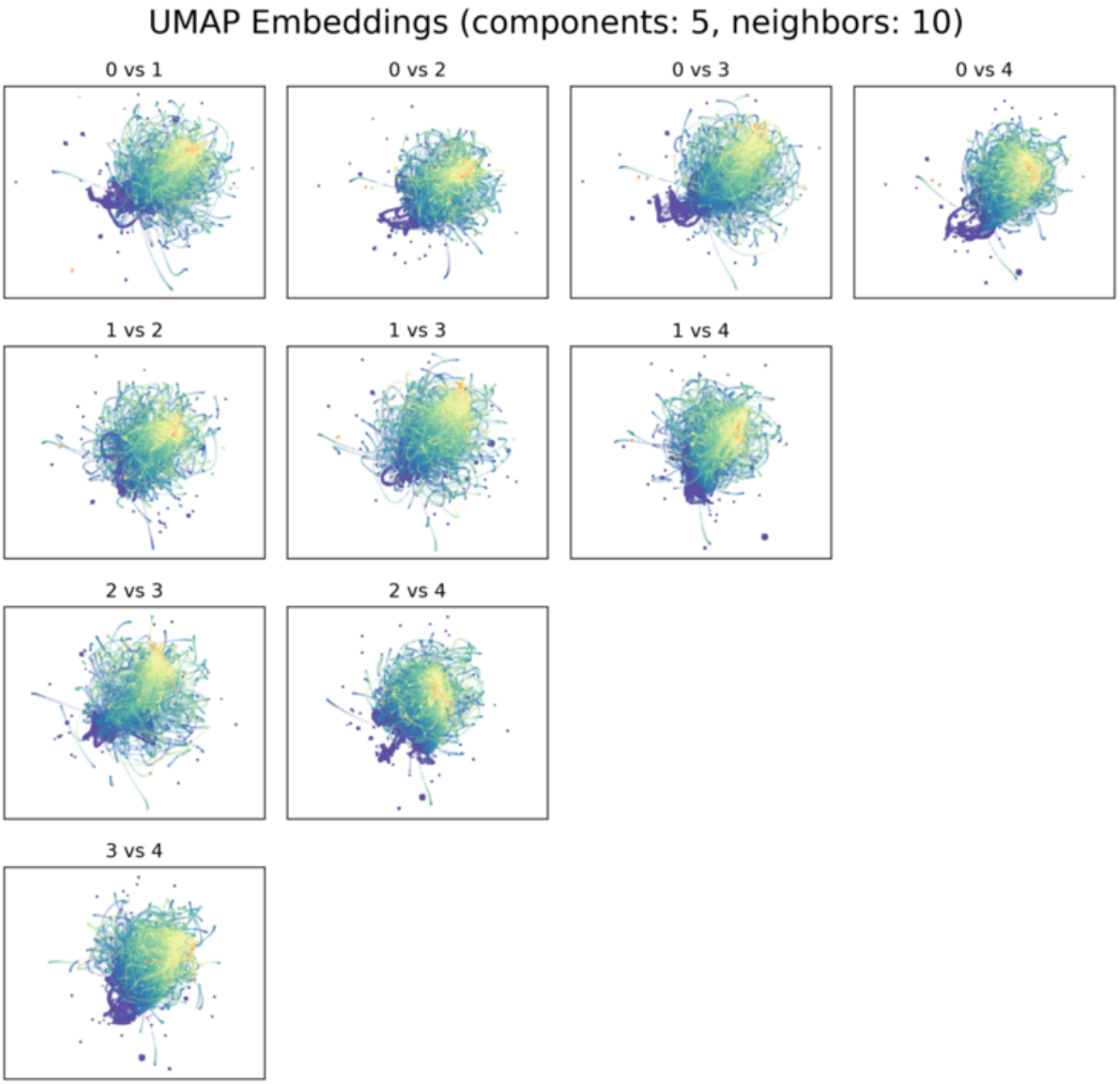


Figure A.10: UMAP Embeddings of the training set colored by Wet Area Ratio for $d = 5$ and $n = 10$.

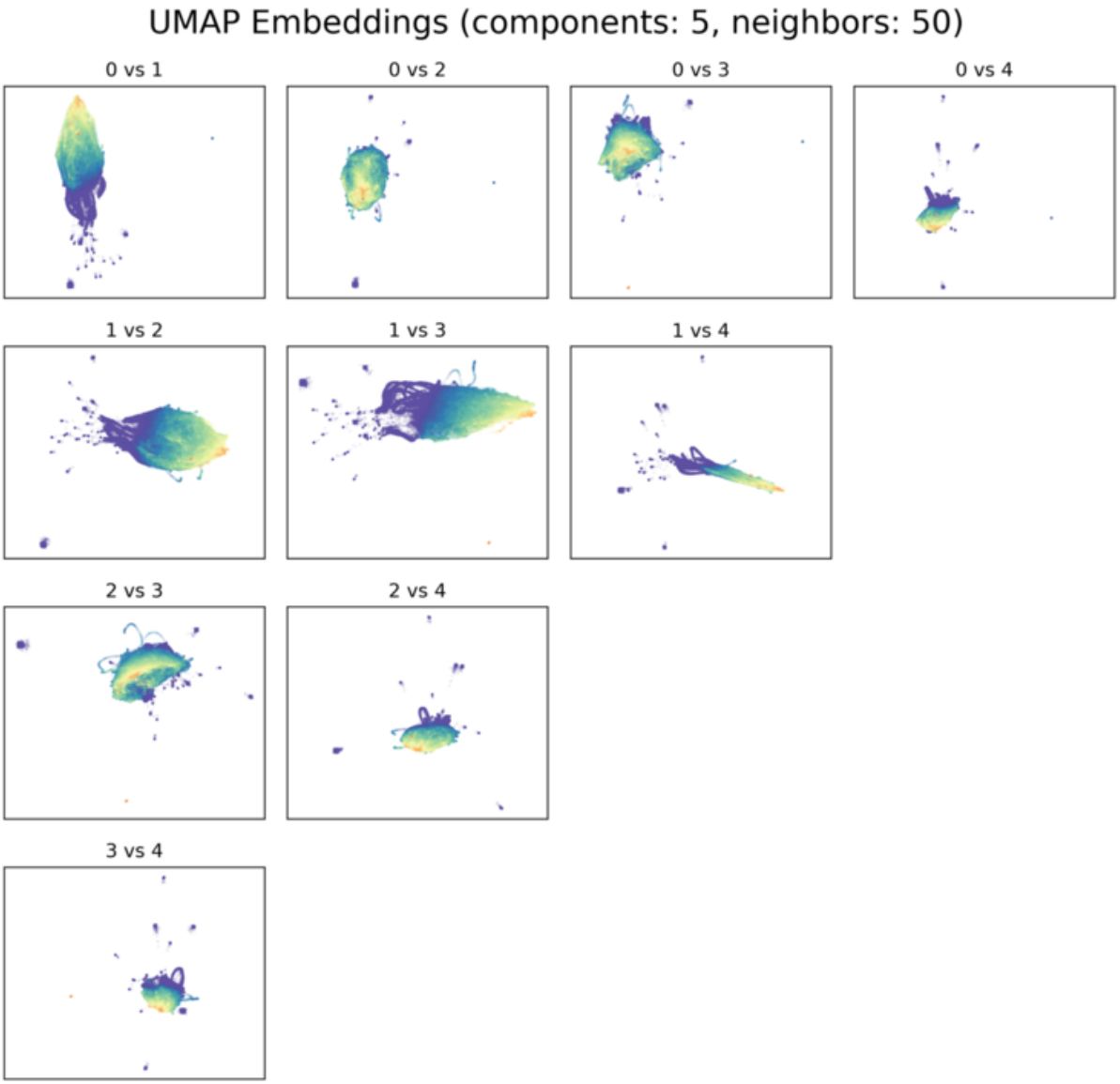


Figure A.11: UMAP Embeddings of the training set colored by Wet Area Ratio for $d = 5$ and $n = 50$.

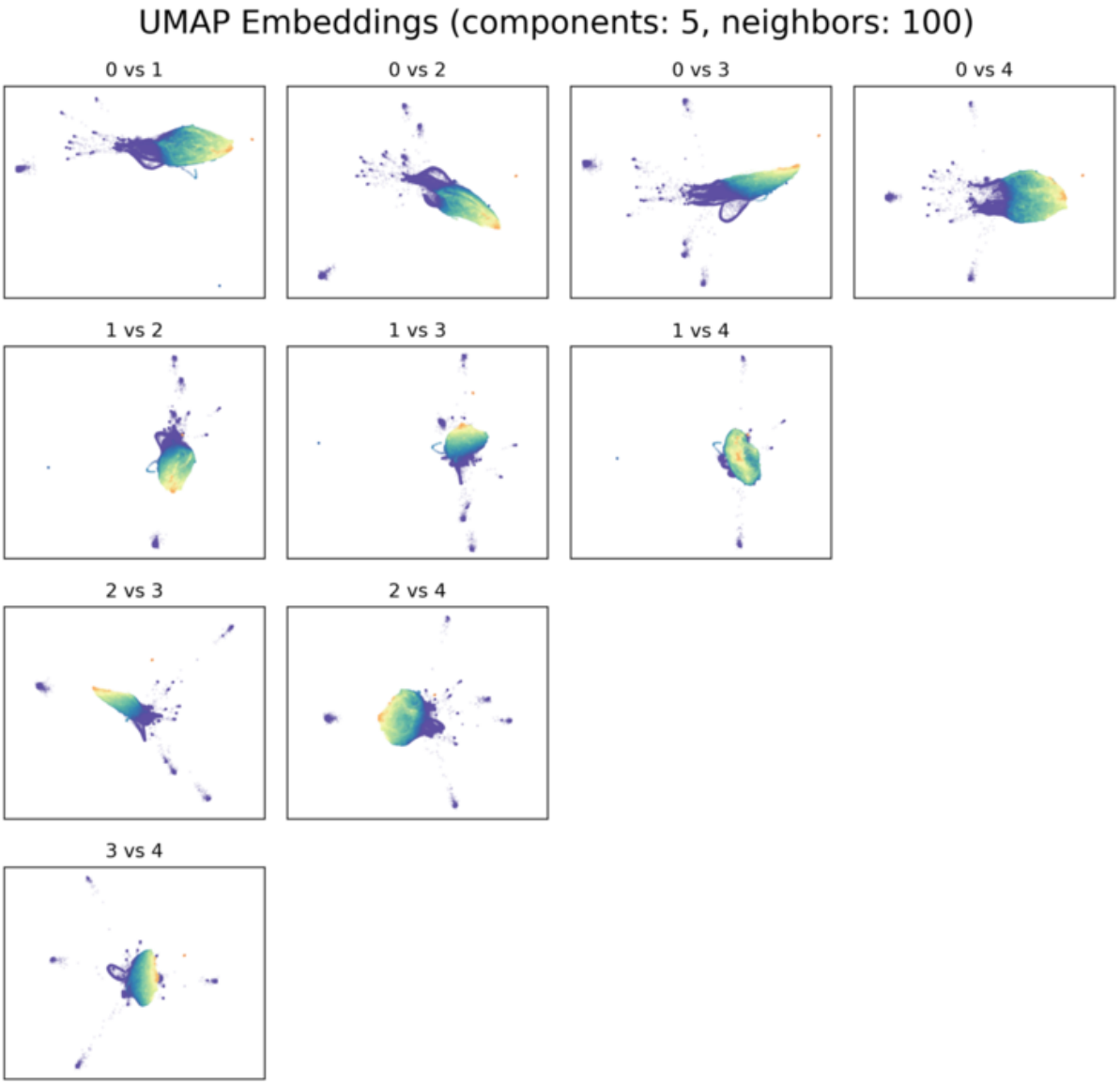


Figure A.12: UMAP Embeddings of the training set colored by Wet Area Ratio for $d = 5$ and $n = 100$.

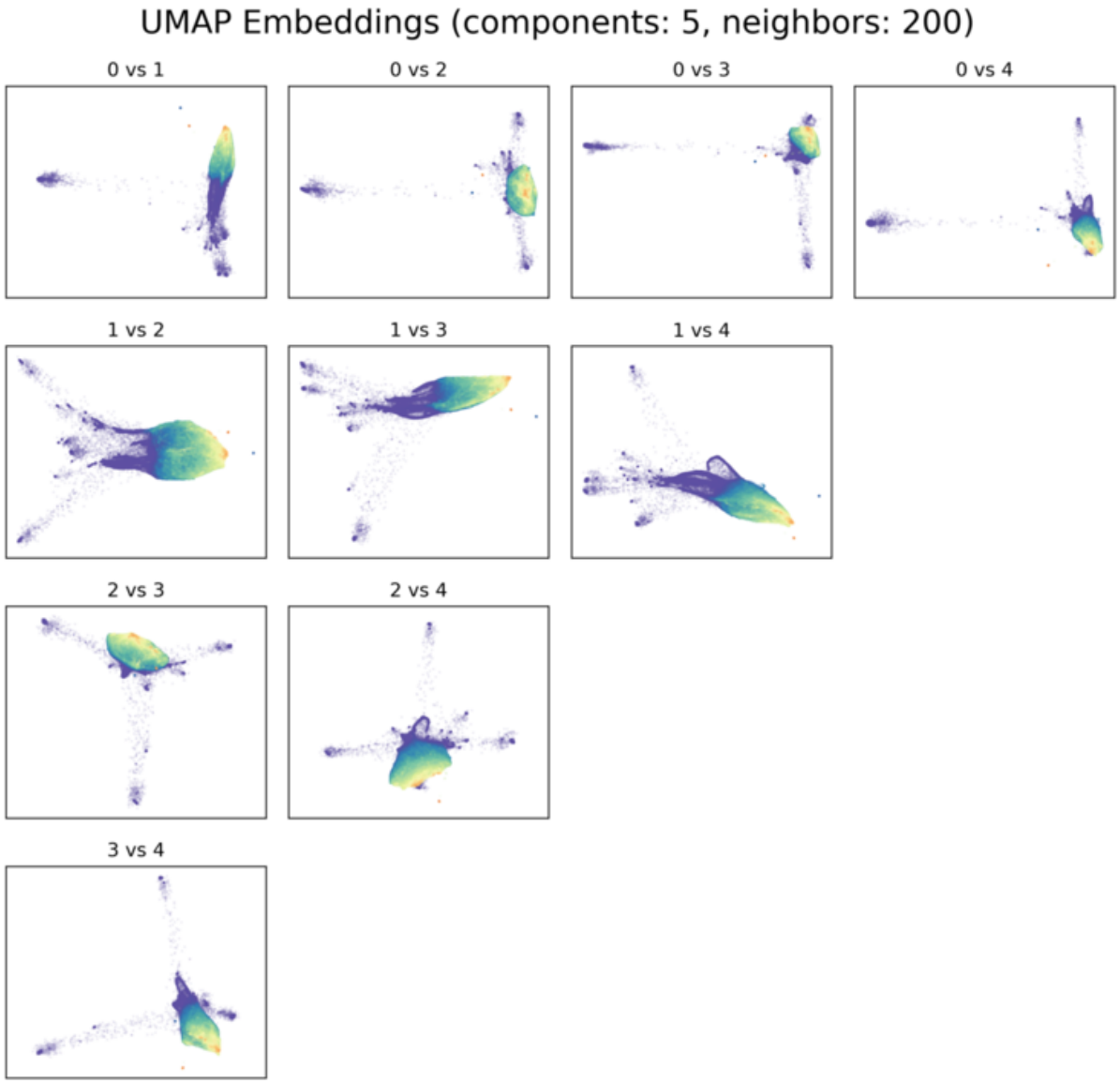


Figure A.13: UMAP Embeddings of the training set colored by Wet Area Ratio for $d = 5$ and $n = 200$.

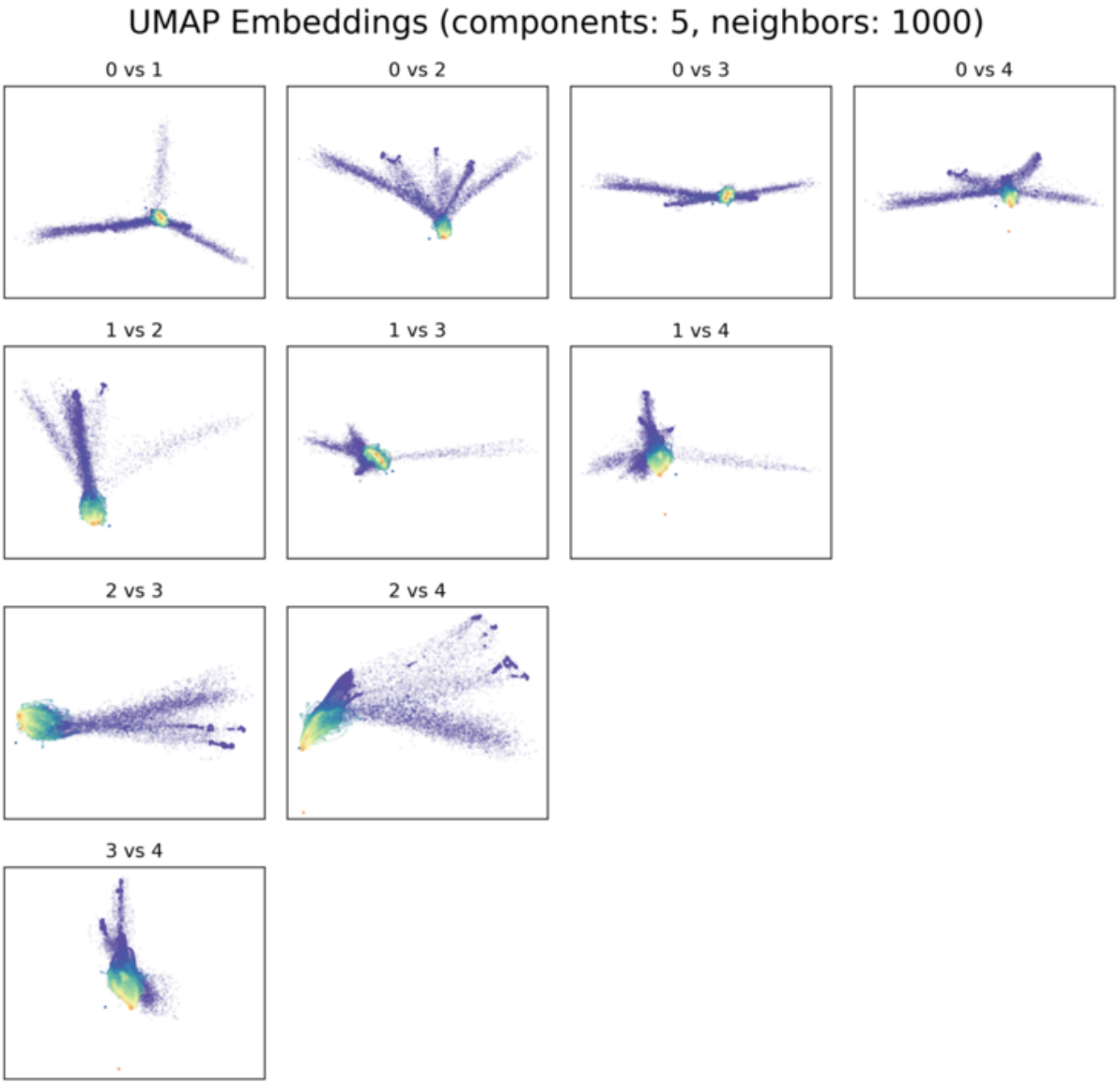
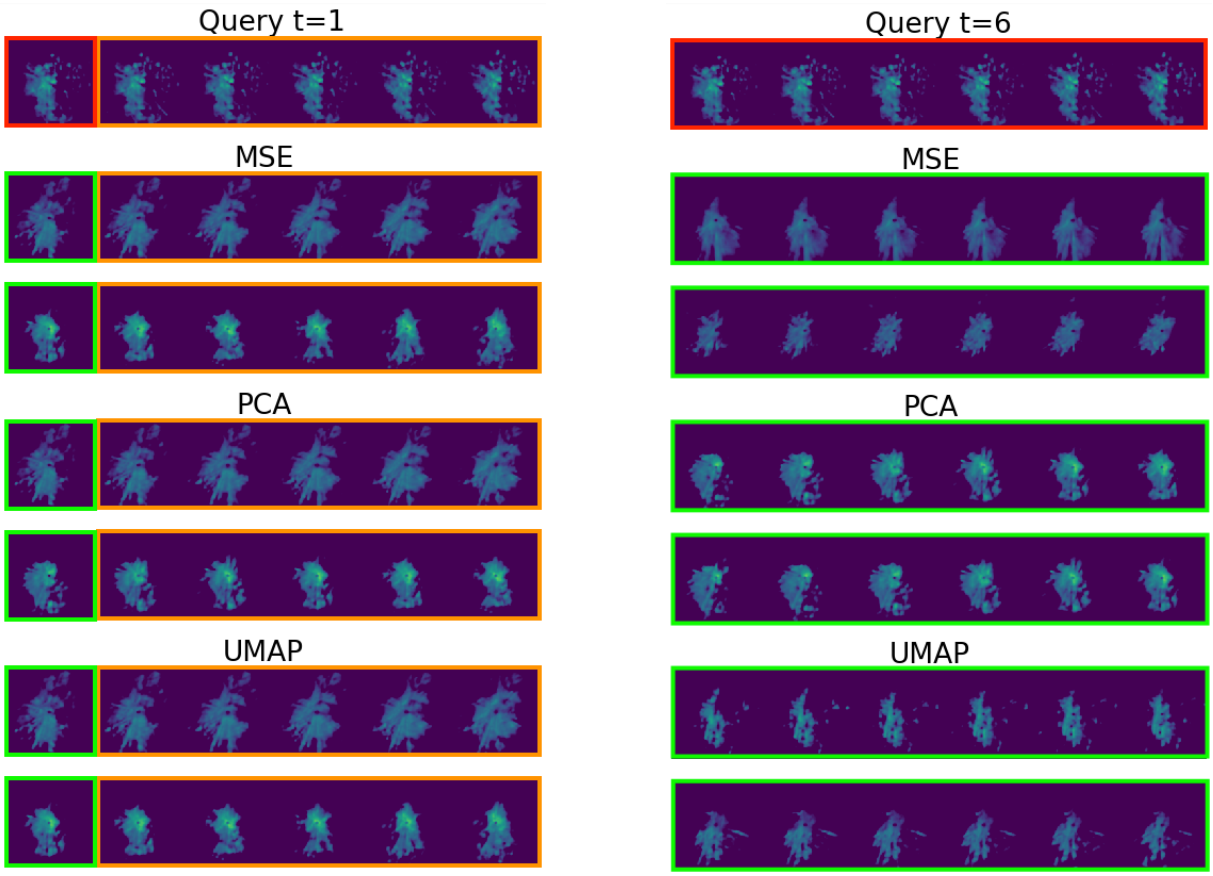


Figure A.14: UMAP Embeddings of the training set colored by Wet Area Ratio for $d = 5$ and $n = 1000$.

A.3 Effect of different t on analog retrieval

To assess the improvement in analog retrieval given by using sequence of images instead of extending in time the results from single frame search, we computed for the 1226 sequences used in Section 4.3.3 the average MSE score difference between the queries and the top-50 results at $t = 6$ and $t = 12$ considering the whole sequence or only the first image for the match: we found that using sequences to query reduces the average MSE of the analogs by 4.6% and 10.9% for $t = 6$ and $t = 12$ respectively. Figure A.15 shows an example of this behavior: a longer query helps to better match the evolution of the precipitation patterns.



(a) (b)

Figure A.15: Example of a query result for $t = 6$ frames when using as input (red box) a single radar scan (a) or the whole sequence (b). The matching sequences are marked in green, while in orange are highlighted the time extensions.

Bibliography

[1] S Agrawal, L Barrington, C Bromberg, J Burge, C Gazen, and J Hickey. Machine Learning for Precipitation Nowcasting from Radar Images. arXiv preprint arXiv:1912.12132, 2019.

[2] S Alessandrini, L Delle Monache, S Sperati, and G Cervone. An analog ensemble for short-term probabilistic solar power forecast. *Appl Energy*, 157:95–110, 2015.

[3] S Alessandrini, L Delle Monache, S Sperati, and J N Nissen. A novel application of an analog ensemble for short-term wind power forecasting. *Renew Energy*, 76:768–781, 2015.

[4] L Alfieri, P Salamon, F Pappenberger, F Wetterhall, and J Thielen. Operational early warning systems for water-related hazards in Europe. *Environ Sci Policy*, 21:35–49, 2012.

[5] S Ansari, S Del Greco, E Kearns, O Brown, S Wilkins, M Ramamurthy, J Weber, R May, J Sundwall, J Layton, A Gold, A Pasch, and V Lakshmanan. Unlocking the Potential of NEXRAD Data through NOAA's Big Data Partnership. *Bull Amer Meteor Soc*, 99(1):189–204, 2018.

[6] A Atencia and I Zawadzki. A Comparison of Two Techniques for Generating Nowcasting Ensembles. Part II: Analogs Selection and Comparison of Techniques. *Mon Weather Rev*, 143(7):2890–2908, 2015.

[7] G Ayzel, M Heistermann, A Sorokin, O Nikitin, and O Lukyanova. All convolutional neural networks for radar-based precipitation nowcasting. *Procedia Comput Sci*, 150:186–192, 2019.

[8] G Ayzel, M Heistermann, and T Winterrath. Optical flow models as an open benchmark for radar-based precipitation nowcasting (rainymotion v0. 1). *Geosci Model Dev*, 12(4):1387–1402, 2019.

[9] K Bachmann, C Keil, and M Weissmann. Impact of radar data assimilation and orography on predictability of deep convection. *Q J R Meteorol Soc*, 145(718):117–130, 2019.

[10] M Banbura, D Giannone, and L Reichlin. Nowcasting. ECB working paper, 2010.

[11] P Bauer, A Thorpe, and G Brunet. The quiet revolution of numerical weather prediction. *Nature*, 525(7567):47–55, 2015.

[12] A Bazzani, B Giorgini, L Giovannini, R Gallotti, and S Rambaldi. Now casting of traffic state by GPS data. The metropolitan area of Rome. In *2011 Proceedings of the 34th International Convention MIPRO*, pages 1615–1618. IEEE, 2011.

[13] E Becht, L McInnes, J Healy, C-A Dutertre, I W H Kwok, L G Ng, F Ginhoux, and E W Newell. Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnol*, 37:38–44, 2019.

[14] R E Bergen and R P Harnack. Long-range temperature prediction using a simple analog approach. *Mon Weather Rev*, 110(8):1083–1099, 1982.

[15] Jonas R Brehmer, Kirstin Strokorb, et al. Why scoring functions cannot assess tail properties. *Electronic Journal of Statistics*, 13(2):4015–4034, 2019.

[16] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[17] V N Bringi and V Chandrasekar. Doppler radar signal theory and spectral estimation. In *Polarimetric Doppler Weather Radar: Principles and Applications*, pages 211–293. Cambridge University Press, 2001.

[18] Y Cao, Q Li, H Shan, Z Huang, L Chen, L Ma, and J Zhang. Precipitation Nowcasting with Star-Bridge Networks. arXiv preprint arXiv:1907.08069, 2019.

[19] L Casagrande, P Cavallini, A Frigeri, A Furieri, I Marchesini, and M Neteler. GIS Open Source: GRASS GIS, Quantum GIS e SpatiaLite, 2012.

[20] L Chen, H Zhang, J Xiao, L Nie, J Shao, W Liu, and T-S Chua. Sca-CNN: Spatial and channel-wise attention in convolutional networks for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5659–5667. IEEE, 2017.

[21] T Chen, M Li, Y Li, M Lin, N Wang, M Wang, T Xiao, B Xu, C Zhang, and Z Zhang. MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems. arXiv:1512.01274, Neural Information Processing Systems, Workshop on Machine Learning Systems, 2015.

[22] G J Ciach, M L Morrissey, and W F Krajewski. Conditional bias in radar rainfall estimation. *J Appl Meteorol*, 39(11):1941–1946, 2000.

[23] L Cuo, T C Pagano, and Q J Wang. A Review of Quantitative Precipitation Forecasts and Their Use in Short- to Medium-Range Streamflow Forecasting. *J Hydrometeorol*, 12(5):713–728, 2011.

[24] G Custer. An interactive umap visualization of the mnist data set. `https://grantcuster.github.io/umap-explorer/`, 2019. [Online; accessed 28-December-2019].

[25] H A Dau and E Keogh. Matrix Profile V: A Generic Technique to Incorporate Domain Knowledge into Motif Discovery. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 125–134. ACM, 2017.

[26] L Delle Monache, F A Eckel, D L Rife, B Nagarajan, and K Searight. Probabilistic Weather Prediction with an Analog Ensemble. *Mon Weather Rev*, 141(10):3498–3516, 2013.

[27] L Delle Monache, T Nipen, Y Liu, G Roux, and R Stull. Kalman filter and analog schemes to postprocess numerical weather predictions. *Mon Weather Rev*, 139(11):3554–3570, 2011.

[28] C F Durman, J M Gregory, D C Hassell, R G Jones, and J M Murphy. A comparison of extreme European daily precipitation simulated by a global and a regional climate model for present and future climates. *Q J R Meteorol Soc*, 127(573):1005–1015, 2001.

[29] R Erdin, C Frei, and H R Künsch. Data Transformation and Uncertainty in Geostatistical Combination of Radar and Rain Gauges. *J Hydrometeorol*, 13(4):1332–1346, 2012.

[30] G Falorni, V Teles, E R Vivoni, R L Bras, and K S Amaratunga. Analysis and characterization of the vertical accuracy of digital elevation models from the Shuttle Radar Topography Mission. *J Geophys Res*, 110(F2), 2005.

[31] L Foresti, L Panziera, P V Mandapaka, U Germann, and A Seed. Retrieval of analogue radar images for ensemble nowcasting of orographic rainfall. *Meteorol Appl*, 22(2):141–155, 2015.

[32] L Foresti and A Seed. The effect of flow and orography on the spatial distribution of the very short-term predictability of rainfall from composite radar images. *Hydrol Earth Syst Sci*, 18(11):4671–4686, 2014.

[33] L Foresti and A Seed. On the spatial distribution of rainfall nowcasting errors due to orographic forcing. *Meteorol Appl*, 22(1):60–74, 2015.

[34] L Foresti, I V Sideris, D Nerini, L Beusch, and U Germann. Using a 10-Year Radar Archive for Nowcasting Precipitation Growth and Decay: A Probabilistic Machine Learning Approach. *Weather Forecast*, 34(5):1547–1569, 2019.

[35] G Franch, V Maggio, G Jurman, L Coviello, M Pendesini, and C Furlanello. TAASRAD19 Radar Scans 2010-2016. `http://dx.doi.org/10.5281/zenodo.3577451`.

[36] G Franch, V Maggio, G Jurman, L Coviello, M Pendesini, and C Furlanello. TAASRAD19 Radar Scans 2017-2019. `http://dx.doi.org/10.5281/zenodo.3591396`.

[37] G Franch, V Maggio, G Jurman, L Coviello, M Pendesini, and C Furlanello. TAASRAD19 Radar Sequences 2010-2019 NetCDF. `https://doi.org/10.5281/zenodo.3866204`.

[38] G Franch, V Maggio, G Jurman, L Coviello, M Pendesini, and C Furlanello. TAASRAD19 Radar Sequences 2017-2019. `http://dx.doi.org/10.5281/zenodo.3591404`.

[39] C Frei and F A Isotta. Ensemble Spatial Precipitation Analysis from Rain-Gauge Data – Methodology and Application in the European Alps. *J Geophys Res Atmos*, 124(11):5757–5778, 2019.

[40] S Gharghabi, Y Ding, C-C M Yeh, K Kamgar, L Ulanova, and E Keogh. Matrix profile VIII: domain agnostic online semantic segmentation at superhuman performance levels. In *Proceedings of the 2017 IEEE International Conference on Data Mining (ICDM)*, pages 117–126. IEEE, 2017.

[41] A Gobiet, S Kotlarski, M Beniston, G Heinrich, J Rajczak, and M Stoffel. 21st century climate change in the European Alps: A review. *Sci Total Environ*, 493:1138–1151, 2014.

[42] I Goodfellow, Y Bengio, and A Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[43] GRASS Development Team. GRASS GIS Documentation. `https://grass.osgeo.org/grass78/manuals/r.in.ascii.html`, 2019. [Online; accessed 27-December-2019].

[44] K He, X Zhang, S Ren, and J Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034. IEEE, 2015.

[45] M Heistermann, S M Collis, M J Dixon, S Giangrande, J J Helmus, B Kelley, J Koistinen, D B Michelson, M Peura, T Pfaff, and D B Wolff. The emergence of open-source software for the weather radar community. *Bull Amer Meteor Soc*, 96(1):117–128, 2015.

[46] M Heistermann, S Jacobi, and T Pfaff. An open source library for processing weather radar data (wradlib). *Hydrol Earth Syst Sci*, 17(2):863–871, 2013.

[47] J J Helmus and S M Collis. The Python ARM Radar Toolkit (Py-ART), a library for working with weather radar data in the Python programming language. *J Open Res Soft*, 4:e25, 2016.

[48] D Heuvelink, M Berenguer, C C Brauer, and R Uijlenhoet. Hydrological application of radar rainfall nowcasting in the Netherlands. *Environment International*, 136:105431, 2020.

[49] I Holleman. Bias adjustment and long-term verification of radar-based precipitation estimates. *Meteorol Appl*, 14:195–203, 2007.

[50] R A Jr Houze. Orographic Effects on Precipitating Clouds. *Rev Geophys*, 50(2011):1–47, 2012.

[51] S Ioffe and C Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

[52] P Jaccard. The distribution of the flora in the alpine zone. 1. *New Phytol*, 11(2):37–50, 1912.

[53] I Jolliffe. *Principal component analysis*. Springer, 2011.

[54] G Jurman, S Merler, A Barla, S Paoli, A Galea, and C Furlanello. Algebraic stability indicators for ranked lists in molecular profiling. *Bioinformatics*, 24:258–264, 2008.

[55] G Jurman, S Riccadonna, R Visintainer, and C Furlanello. Canberra distance on ranked lists. In S. Agarwal, C. Burges, and K. Crammer, editors, *Proceedings of Advances in Ranking NIPS 2009 Workshop*, pages 22–27, 2009.

[56] D P Kingma and J Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[57] J Koistinen, A von Lerber, S Pulkkinen, H Sinisalo, M Berenguer, S Park, D Sempere, C Prudhomme, W K Wong, C Baugh, A Burtsoff, and S D Molina. Seamless probabilistic MUlti-source Forecasting of heavy rainfall hazards for European Flood awareness–SMUFF project, 2019. Geophysical Research Abstracts 21.

[58] Z Kothavala. Extreme precipitation events and the applicability of global climate models to the study of floods and droughts. *Math Comput Simulat*, 43(3):261–268, 1997.

[59] J Kreklow. Facilitating radar precipitation data processing, assessment and analysis: a GIS-compatible Python approach. *J Hydroinform*, 21(4):652–670, 2019.

[60] Francois Lalaurette. Early detection of abnormal weather conditions using a probabilistic extreme forecast index. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 129(594):3037–3057, 2003.

[61] G N Lance and W T Williams. Computer programs for hierarchical polythetic classification ("similarity analysis"). *Comput J*, 9:60–64, 1966.

[62] I Lehtonen, K Ruosteenoja, and K Jylhä. Projected changes in European extreme precipitation indices on the basis of global and regional climate model ensembles. *Int J Climatol*, 34(4):1208–1222, 2014.

[63] R Lguensat, P Tandeo, P Ailliot, M Pulido, and R Fablet. The analog data assimilation. *Mon Weather Rev*, 145(10):4093–4107, 2017.

[64] E N Lorenz. Atmospheric predictability as revealed by naturally occurring analogues. *J Atmos Sci*, 26(4):636–646, 1969.

[65] J S Marshall and W Mc K Palmer. The distribution of raindrops with size. *J Meteorol*, 5(4):165–166, 1948.

[66] J S Marshall and W McK Palmer. The Distribution of Raindrops with size. *J Meteorol*, 5(4):165–166, 1948.

[67] L McInnes. How UMAP Works. `https://umap-learn.readthedocs.io/en/latest/how_umap_works.html`. [Online accessed: 2019-11-18].

[68] L McInnes, J Healy, N Saul, and L Großberger. UMAP: Uniform Manifold Approximation and Projection. *J Open Source Softw*, 3(29):861, 2018.

[69] G A Meehl, F Zwiers, J Evans, T Knutson, L Mearns, and P Whetton. Trends in Extreme Weather and Climate Events: Issues Related to Modeling Extremes in Projections of Future Climate Change. *Bull Amer Meteor Soc*, 81(3):427–436, 2000.

[70] D B Michelson, R Lewandowski, M Szewczykowski, H Beekhuis, and G Haase. *EUMETNET OPERA weather radar information model for implementation with the HDF5 file format*, 2011. OPERA deliverable OPERA_2008_03.

[71] A Mueen, Y Zhu, M Yeh, K Kamgar, K Viswanathan, C Gupta, and E Keogh. The Fastest Similarity Search Algorithm for Time Series Subsequences under Euclidean Distance. `http://www.cs.unm.edu/~mueen/FastestSimilaritySearch.html`, August 2017.

[72] M Neteler and H Mitasova. *Open source GIS: a GRASS GIS approach*, volume 689. Springer Science & Business Media, 2013.

[73] M Neteler and H Mitasova. *Open source GIS: a GRASS GIS approach*, volume 689. Springer Science & Business Media, 2013.

[74] L. Panziera, U. Germann, M. Gabella, and P. V. Mandapaka. NORA–Nowcasting of Orographic Rainfall by means of Analogues. *Q J R Meteorol Soc*, 137(661):2106–2123, 2011.

[75] S Pulkkinen, D Nerini, A Pérez Hortal, C Velasco-Forero, U Germann, A Seed, and L Foresti. Pysteps: an open-source Python library for probabilistic precipitation nowcasting (v1.0). *Geosci Model Dev Discuss*, 12:4185–4219, 2019.

[76] S Pulkkinen, D Nerini, A A Pérez Hortal, C Velasco-Forero, A Seed, U Germann, and L Foresti. Pysteps: an open-source Python library for probabilistic precipitation nowcasting (v1.0). *Geosci Model Dev*, 12(10):4185–4219, 2019.

[77] T Ramsauer, T Weiss, and P Marzahn. Comparison of the GPM IMERG Final Precipitation Product to RADOLAN Weather Radar Data over the Topographically and Climatically Diverse Germany. *Remote Sens*, 10(12):2029, 2018.

[78] Paul J Roebber. Visualizing multiple measures of forecast quality. *Weather and Forecasting*, 24(2):601–608, 2009.

[79] Elena Saltikoff, Günther Haase, Laurent Delobbe, Nicolas Gaussiat, Maud Martet, Daniel Idziorek, Hidde Leijnse, Petr Novák, Maryna Lukach, and Klaus Stephan. OPERA the radar project. *Atmosphere*, 10(6):320, June 2019.

[80] M P Sampat, Z Wang, S Gupta, A C Bovik, and M K Markey. Complex wavelet structural similarity: A new image similarity index. *IEEE T Image Proc*, 18(11):2385–2401, 2009.

[81] A. W. Seed. A Dynamic and Spatial Scaling Approach to Advection Forecasting. *J. Appl. Meteor.*, 42(3):381–388, 2003.

[82] M Shahriari, G Cervone, L Clemente-Harding, and L Delle Monache. Using the analog ensemble method as a proxy measurement for wind power predictability. *Renew Energy*, 146:789–801, 2020.

[83] X Shi, Z Chen, H Wang, D-Y Yeung, W-k Wong, and W-C Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In C Cortes, N D Lawrence, D D Lee, M Sugiyama, and R Garnett, editors, *Proc Adv Neural Inf Process Syst 28*, pages 802–810, 2015.

[84] X Shi, Z Gao, L Lausen, H Wang, D-Y Yeung, W-K Wong, and W-C Woo. Deep learning for precipitation nowcasting: A benchmark and a new model. In I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett, editors, *Proc Adv Neural Inf Process Syst 30*, pages 5617–5627, 2017.

[85] Z Sokol, J Mejsnar, L Pop, and V Bližňák. Probabilistic precipitation nowcasting based on an extrapolation of radar reflectivity and an ensemble approach. *Atmos Res*, 194:245–257, 2017.

[86] K Song, G Yang, Q Wang, C Xu, J Liu, W Liu, C Shi, Y Wang, G Zhang, X Yu, et al. Deep Learning Prediction of Incoming Rainfalls: An Operational Service for the City of Beijing China. In *Proceedings 2019 International Conference on Data Mining Workshops (ICDMW)*, pages 180–185. IEEE, 2019.

[87] J Sun, M Xue, J W Wilson, I Zawadzki, S P Ballard, J Onvlee-Hooimeyer, P Joe, D M Barker, P-W Li, B Golding, M Xu, and J Pinto. Use of NWP for nowcasting convective precipitation: Recent progress and challenges. *Bull Amer Meteor Soc*, 95(3):409–426, 2014.

[88] M Surcel, I Zawadzki, and M Yau. On the filtering properties of ensemble averaging for storm-scale precipitation forecasts. *Mon Weather Rev*, 142:1093–1105, 2014.

[89] P Tandeo, P Ailliot, J Ruiz, A Hannart, B Chapron, A Cuzol, V Monbet, R Easton, and R Fablet. Combining analog method and ensemble data assimilation: application to the Lorenz-63 chaotic system. In *Machine learning and data mining approaches to climate science*, pages 3–12. Springer, 2015.

[90] The HDF Group. Hierarchical data format version 5. `https://www.hdfgroup.org/solutions/hdf5/`, 2000–2019. [Online; accessed 26-December-2019].

[91] The National Snow and Ice Data Center. Netword Common Data Format (NetCDF). `https://nsidc.org/data/netcdf`, 2019. [Online; accessed 26-December-2019].

[92] Q-K Tran and S-k Song. Computer Vision in Precipitation Nowcasting: Applying Image Quality Assessment Metrics for Training Deep Neural Networks. *Atmosphere*, 10(5), 2019.

[93] Q-K Tran and S-k Song. Multi-Channel Weather Radar Echo Extrapolation with Convolutional Recurrent Neural Networks. *Remote Sens*, 11(19), 2019.

[94] `OpenRadarScience.org`. Open Data. `https://openradarscience.org/opendata/`, 2018. [Online; accessed 12-December-2019].

[95] H M Van den Dool. Searching for analogues, how long must we wait? *Tellus A*, 46(3):314–324, 1994.

[96] Mark J Van der Laan, Eric C Polley, and Alan E Hubbard. Super learner. *Statistical applications in genetics and molecular biology*, 6(1), 2007.

[97] J Von Hardenberg, L Ferraris, and A Provenzale. The shape of convective rain cells. *Geophys Res Lett*, 30(24):CiteID 2280, 2003.

[98] Y Wang, M Long, J Wang, Z Gao, and P S Yu. PredRNN: Recurrent Neural Networks for Predictive Learning using Spatiotemporal LSTMs. In I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett, editors, *Proc Adv Neural Inf Process Syst 30*, pages 879–888, 2017.

[99] Y Wang, M Long, J Wang, Z Gao, and P S Yu. PredRNN++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. *PLMR*, 80:5123–5132, 2018.

[100] Y Wang, J Zhang, H Zhu, M Long, J Wang, and P S Yu. Memory In Memory: A Predictive Neural Network for Learning Higher-Order Non-Stationarity from Spatiotemporal Dynamics. arXiv preprint arXiv:1811.07490, 2019.

[101] F Warmerdam. The geospatial data abstraction library. In *Open source approaches in spatial data handling*, pages 87–104. Springer, 2008.

[102] F Warmerdam and E Rouault. GRASS ASCII Grid raster driver. `https://gdal.org/drivers/raster/grassasciigrid.html`, 2019. [Online; accessed 27-December-2019].

[103] A P Weigel. *Ensemble forecasts*, chapter 8, pages 141–166. Wiley-Blackwell, 2012.

[104] M Werner and M Cranston. Understanding the Value of Radar Rainfall Nowcasts in Flood Forecasting and Warning in Flashy Catchments. *Meteorol Appl*, 16(1):41–55, 2009.

[105] J W Wilson, N A Crook, C K Mueller, J Sun, and M Dixon. Nowcasting Thunderstorms: A Status Report. *Bull Amer Meteor Soc*, 79(10):2079–2100, 1998.

[106] D H Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

[107] W-c Woo and W-k Wong. Operational Application of Optical Flow Techniques to Radar-Based Rainfall Nowcasting. *Atmosphere*, 8(12):48, 2017.

[108] D Yang and S Alessandrini. An ultra-fast way of searching weather analogs for renewable energy forecasting. *Sol Energy*, 185:255–261, 2019.

[109] C-C M Yeh. Towards a Near Universal Time Series Data Mining Tool: Introducing the Matrix Profile. arXiv preprint arXiv:1811.03064, 2018.

[110] C-C M Yeh, Y Zhu, L Ulanova, N Begum, Y Ding, A Dau, D Silva, A Mueen, and E Keogh. Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 1317–1322. IEEE, 2016.

[111] I Zawadzki, J Morneau, and R Laprise. Predictability of Precipitation Patterns: An Operational Approach. *J Appl Meteorol*, 33(12):1562–1571, 1994.

[112] Z-H Zhou. A brief introduction to weakly supervised learning. *Natl Sci Rev*, 5(1):44–53, 2017.

[113] Y Zhu, C-C M Yeh, Z Zimmerman, K Kamgar, and E Keogh. Matrix profile XI: SCRIMP++: time series motif discovery at interactive speeds. In *Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM)*, pages 837–846. IEEE, 2018.

[114] E Zorita and H Von Storch. The analog method as a simple statistical downscaling technique: comparison with more complicated methods. *J Clim*, 12(8):2474–2489, 1999.

[115] Ervin Zsoter, Florian Pappenberger, and David Richardson. Sensitivity of model climate to sampling configurations and the impact on the extreme forecast index. *Meteorological Applications*, 22(2):236–247, 2015.