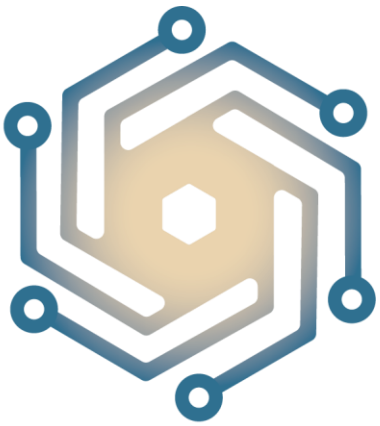

CLINT

CLIMATE INTELLIGENCE

D2.2
ML ALGORITHMS FOR EE FORECAST AND RECONSTRUCTION

April, 2024

This project is part of the H2020 Programme supported by the European Union, having received funding from it under Grant Agreement No 101003876

Programme Call:	Building a low-carbon, climate resilient future: climate action in support of the Paris Agreement (H2020-LC-CLA-2018-2019-2020)
Grant agreement ID:	101003876
Project Title:	CLINT
Partners:	POLIMI (Project Coordinator), CMCC, HEREON, CSIC, SMHI, HKV, E3M, TCDF, DKRZ, IHE, ECMWF, UAH, JLU, OGC, UCM
Work-Package:	WP2
Deliverable #:	D2.2
Deliverable Type:	Document
Contractual Date of Delivery:	30 April 2024
Actual Date of Delivery:	30 April 2024
Title of Document:	ML algorithms for EE forecast and reconstruction
Responsible partner:	IHE
Author(s):	Claudia Bertini (IHE), Schalk Jan van Andel (IHE), Paolo Bonetti (POLIMI), Yiheng Du (SMHI), Michael Maier-Gerber (ECMWF), Matteo Giuliani (POLIMI), Andrea Ficchí (POLIMI), Dennis Zanutto (POLIMI), Jorge Pérez-Aracil (UAH), Ronan McAdam (CMCC), Étienne Plésiat (DKRZ), Niklas Luther (JLU), Elena Xoplaki (JLU), Andrea Castelletti (POLIMI)
Content of this report:	This report presents the methods to forecast and reconstruct EE developed within WP2, also showing the results obtained.
Availability:	This report is public.

Document revisions		
Author	*Revision content*	*Date*
Claudia Bertini (IHE)	Version 1.0	29 March 2024
Eduardo Zorita (HEREON)	Internal revision: typos, grammar, comments on content	05 April 2024
Enrico Scoccimarro (CMCC)	Internal revision	08 April 2024
Claudia Bertini (IHE)	Version 1.1	25 April 2024
Guido Ascenso (POLIMI), Adrea Castelletti (POLIMI)	Final revision	25 April 2024

Table of Content

LIST OF ACRONYMS

AB:	Advisory Board
AI:	Artificial Intelligence
ALE:	Accumulated Local Effect
ANN:	Artificial Neural Network
AVE:	Average Variance Explained
BNN:	Bayesian Neural Network
CA:	Consortium Agreement
CCA:	Canonical Correlation Analysis
CDF:	Conditional Distribution Function
CEEI:	Concurrent Extreme Event Index
CNN:	Convolutional Neural Network
CS:	Climate Service
DL:	Deep Learning
DoA:	Description of Action (Annex I of the Grant Agreement)
EC:	European Commission
ECMWF:	European Centre for Medium range Weather Forecasts
ELM:	Extreme Learning Machine
FDR:	False Discovery Rate
GA:	Grant Agreement
GPU:	Graphical Processing Unit
GSM:	Gaussian-Scale Mixture
HPC:	High-performance Computer
KRGCCA:	Kernel Regularized Generalized Canonical Correlation Analysis
LSTM:	Long Short-Term Memory
MCMC:	Markov Chain Monte Carlo
ML:	Machine Learning
MLP:	Multi-Layer Perceptron
Mx:	Month number (where x is the month number)
NN:	Neural Network
NPSPEI:	Nonparametric Standardized Precipitation-Evapotranspiration Index
NWP:	Numerical Weather Prediction
PCA:	Principal Component Analysis
RA-U:	Residual Attention UNet
RF:	Random Forest
RL:	Reinforcement Learning
SIS:	Sure Independence Screening
SPEI:	Standardized Precipitation-Evapotranspiration Index
SPI:	Standardized Precipitation Index
STMAX:	Standardized Maximum Temperature Index
WAIC:	Watanabe-Akaike Information Criterion
WEF:	Water-Energy-Food
WP:	Work Package
PCRO-SL:	Probabilistic Coral Reefs with Substrate Layers
DPS:	Direct Policy Search

Note: the acronyms list above shows only the main acronyms commonly used across the entire document. Several other acronyms, especially referred to the name of physical variables employed, are presented and explained in each methodology section.

EXECUTIVE SUMMARY

This Deliverable report presents an overview of the Machine Learning (ML) approaches for Extreme Events (EE) forecasting and reconstruction developed in Work Package (WP) 2, by tasks T2.4 and T2.5, respectively. The document describes in detail the methods developed, results obtained in the test areas, and the next steps foreseen in their application in local and pan-European CLINT case studies, serving the purposes of WP7 and WP6, respectively.

The methods here described involve purely data-driven and hybrid approaches, using state-of-the-art or newly developed/re-adapted ML algorithms, and use a combination of observational and forecasted data.

The approaches of this deliverable target droughts, tropical cyclones, heatwaves and warm nights, and compound events, either directly (e.g. forecasting the probability of the genesis of an EE), or indirectly (e.g. forecasting and reconstructing hydrological variables and extremes, which are then used to detect EE with specific indicators).

Where available, the ML models developed have been benchmarked against existing forecasting provided by conventional Numerical Weather Prediction (NWP) models.

The report first provides an introduction on the use of ML for reconstructing and forecasting extreme events, then presents the methods and results developed to forecast EE, followed by those developed to reconstruct EE. Conclusion and next plans for further application of the developed methods are presented at the end of this report.

1 INTRODUCTION

This deliverable presents machine learning (ML) algorithms and application methods developed for enhancing extreme event forecasting at sub-seasonal to seasonal lead times, and reconstruction of extreme events in climate data sets and projections. The most suitable methods identified in the previous work package 2 (WP2) deliverable, D2.1, and milestones MS17 and MS22 have now been tested for application to extreme events, some for case study areas and regions globally in support of WP3, and some already for the specific local scale case studies of WP7. In this deliverable the ML algorithms are presented in detail, along with the methods developed to use them for extreme event forecasting and reconstruction. Sample results of each of the experiments are presented along with performance metrics to discuss the effectiveness, strong and weak points of each of the methods.

1.1 Extreme Events forecasting with Machine Learning

Extreme weather events (from now on simply Extreme Events - EEs) severely impact society and human lives. In the period between 1980 and 2022, EEs created economical losses of around 650 billion EUR only for EU member states (EEA, 2023). Moreover, the frequency of extreme events has been increasing over time and it is expected to further increase due to climate change (Seneviratne et al., 2021). Accurately forecasting extreme events is becoming increasingly critical for the well-being of humans and societies. Over time, many methods for forecasting extreme events have been developed. These methods can be classified into dynamic or statistical methods. The former methods involve the use of dynamical models, either Numerical Weather Prediction (NWP) or climate models, that reproduce the evolution of the atmosphere and climate. These models predict future states of climate variables that are in turns used to detect EE through specific definitions and/or indicators. Statistical methods, instead, derive empirical relationships between EE and predictors, using historical data. To this last category belong also Machine Learning methods.

Even though NWP forecasting capabilities have substantially improved in the last decades, issues in forecasting anomalous conditions that generate EE still remains a challenge (Chattopadhyay et al., 2020), especially at sub-seasonal lead-times.

Machine learning (ML) techniques have recently gained popularity in the field of extreme event forecasting, thanks to their ability to unravel complex non-linear relationships among large datasets. ML and Deep Learning (DL) algorithms have been widely employed to forecast extreme events. For comprehensive and relevant literature, the reader can find details in the several reviews recently published (e.g. Salcedo-Sanz et al., 2023, Prodhan et al., 2022, Wang et al., 2022, Olivetti & Messori, 2024). Moreover, ML models have been recently applied to create ML-based weather forecasting systems at the global scale, emulating and sometimes outperforming canonical NWP models from meteorological agencies. Examples are PanguWeather (Bi et al., 2023) developed by Huaweii, FourCastNet (Pathak et al., 2022) developed by NVDIA, and GraphCast (Lam et al., 2023) developed by Google DeepMind. Even though these models have high potentials of developing good quality of forecasts, they are strongly relying on existing forecasting systems and reanalysis data produced by conventional NWP and climate models for their training. Moreover, as these global forecasting systems are generally trained to minimize the average error, they still have issues in accurately predicting intensity of extreme events. (Chantry et al., 2023).

The main opportunities for ML in weather and extreme event forecasting still rely in post-processing NWP forecasts, to correct them leveraging on ML power in solving non-linear complex problems (Haupt et al., 2019), in developing data-driven forecasts tailormade for specific EE, variables and/or locations, thanks to its power in identifying links among enormous amount of data, and to facilitate data assimilation and fusion in physically-based models.

1.2 Extreme Events reconstruction with Machine Learning

Observational climate datasets are usually incomplete and contain missing values which can vary in both time and space. Information may be missing from these datasets for a variety of reasons:

- data collection and processing due to equipment malfunction, human errors, processing time, data loss, etc.
- external factors such as natural disasters, which can damage climate monitoring equipment and disrupt data collection
- geographical limitations due to the accessibility of certain regions or difficulty to perform measurements in specific conditions
- historical limitations due to the limited technology and interest for climate data

Missing values contribute to uncertainties and structural biases in the analysis of climate records such as the detection, causation, and attribution of EEs. The evaluation of EE trends is particularly impacted by the historical limitations, resulting in an increasing scarcity of data as we go back in time.

Over the years, many numerical methods have been developed to tackle this problem, essentially by performing data imputation, i.e. by replacing the missing data with estimated values based on the observed data itself. In climate science, it is common to use methods such as the Inverse Weighted Distance (IWD) (Teegavarapua et al. 2005), linear regression, thin-plate spline interpolation, or Kriging's method (Cowtan et al. 2013). More sophisticated techniques like the empirical orthogonal function (EOF) analysis or principal component analysis (PCA) (Beckers et al. 2003) are also commonly used. However, these methods suffer from well-known limitations that constrains their use for the reconstruction of climate data. For instance, the IWD method assumes a certain spatial continuity and tends to produce overly smooth interpolations. Kriging offers improved spatial variability but requires a careful definition of various parameters to define an appropriate variogram model. Furthermore, both methods can be computationally demanding and sensitive to outliers. More recently, machine learning (ML) techniques have stirred an increasing interest in the community because of their efficiency, in particular deep-learning approaches (Shibata et al. 2018, Dong J. et al. 2019, Geiss et al. 2021, Kadow et al. 2020, Hu et al. 2023, Yao et al. 2023). As epitomised by the work of Kadow et al. 2020 on the reconstruction of the HadCRUT4 dataset, deep learning-based inpainting techniques can achieve state-of-the art performance in comparison with traditional inpainting methods, both for the evaluation metrics as for the ability to reconstruct complex climatic patterns.

1.3 Objectives of this deliverable

The objectives of this deliverable are to:

- Present ML algorithms and methods developed for extreme event forecasting and reconstruction.
- Discuss their application results to extreme events addressed by CLINT: droughts, tropical cyclones, heatwaves and warm nights, and compound events.

1.4 Connection with other Deliverables and Milestones

This report is connected with the following Deliverables: *D 2.1 Review of ML algorithms for Climate Science*, which describes the state-of-the-art ML algorithms to support climate science in the detection, causation, and attribution of extreme events; *D 3.1 Extreme Events detection*, which

reviews existing indicators, datasets and potential ML drivers for the detection of all extreme events addressed in CLINT; *D 7.1 Local Climate Services*, which reports the end-users' needs for local Climate Services (CSs) in the local case studies of CLINT; *D 3.2 Preliminary AI-enhanced Extreme Events detection*, which reviews the existing knowledge on extreme events detection and candidate drivers for ML-based detection; *D 6.2 Preliminary report on AI-enhanced Climate Services for extreme impacts*, which illustrates the preliminary analysis of AI-enhanced CSs for European impacts for the water, energy, food sectors; *D 7.2 Preliminary AI-enhanced Climate Services for local decision-making*, which describes the preliminary analysis of the value of AI-enhanced CSs for the different extreme events and climate change hotspots.
Moreover, this report is also linked with the following Milestones: *MS2 Data provision for local Climate Services*, which reviews the datasets (regional and global) to be acquired and used in the local case studies; *MS 12, 13, 14 Benchmark Climate Services and local use case established for delta hotspots, snow hotspots, and semiarid hotspots* (respectively), which report on the local CS to be used for benchmarking AI-enhanced CS that will be produced in CLINT; MS19 *First prototype of algorithms for EE detection*, which describes the first ML algorithms for extreme event detection. Finally, MS17 *First prototype of algorithms for reconstructions of missing EE values*, constitutes the basis for this deliverable's section on EE reconstructions.

1.5 Structure of the document

This report is structured following the two main components: forecasting and reconstruction, which each make-up a Chapter (Chapters 2 and 3). Chapter 2 on forecasting, is divided in a sub-section for each combination of type of event and ML method developed. Chapter 3, on reconstruction, has a sub-section describing the methodology, and one on benchmarking for climate dataset infilling. Chapter 4 concludes the deliverable with main findings and recommendations.

2 EXTREME EVENTS FORECASTING WITH MACHINE LEARNING

Generally speaking, many are the ways in which ML can be used to enhance EE forecasting: 1) build data-driven prediction models that predict either the extreme event or the variables used to detect the event; 2) post-process existing forecasts, to improve the EE detection; 3) extract valuable information from existing forecasts, to make better decisions related to EE; 4) discover new mathematical relationships between predictors and EE indicators; 5) re-train existing formulation of EE indicators; 6) incorporate ML into dynamical models, to improve specific model components (e.g. data assimilation).

At the current stage of the project, efforts to enhance EE forecasting with ML have been focused on the first three methods, as machine learning showed great potential in all three applications. Discussions are also ongoing on how to develop new mathematical relationships between predictors and EE indicators (approach 4) and such methods might be developed on a later stage. Finally, the last two approaches are at the moment not taken under consideration, as there is more interest in revealing mechanisms between predictors and predictands not yet known, and because incorporating ML into existing dynamical forecasting models is out of CLINT purpose.

Hence, this deliverable focuses on the first three approaches, and more specifically presents methods that are classified as:

1. Purely data-driven models, which use only past or current information (observations or reanalysis data) to forecast extreme events. These models can forecast the variables used to define extreme events (e.g. precipitation to forecast meteorological drought), the values of EE indicators (e.g. Standardized Precipitation Index for meteorological drought), or the probability of occurrence of the event;
2. Hybrid models, which in turn can be used for: i) AI post-processing of existing forecast, if ML algorithms are used to correct or downscale existing forecasts (e.g. bias correction of meteorological forecasts); ii) AI-extension of the lead time of existing forecasts, if ML algorithms forecast future conditions by using a combination of past and forecasted data as inputs (e.g. using the forecasted average precipitation of one week ahead to predict the average precipitation of two weeks ahead); AI-extraction of valuable information from existing forecasts, which is then used for improving decision making.

The ML-based methods here presented address several forecasting horizons, also referred as to "lead-times", ranging from nowcasting (same day of the event occurrence) to seasonal (1 to 3 months). More details on the physical differences across the lead-times are presented in section 2.1.

This section collects the methods developed for forecasting extreme events at the local scale. These methods address droughts, tropical cyclones, heatwaves and warm nights, and compound events. Moreover, some methods to derive hydrological extremes, regarding both floods and droughts, are also developed. Some of these methods have already been tested in some of CLINT case studies, while some others are still under testing and will be further tuned for local applications. Moreover, for some of the forecasting approaches presented, plans have already been made to extend the application to the pan-European level (e.g. hydrological forecasting and post-processing with Long Short-Term Memory models), and their outcomes will be most likely presented at a later stage in D6.3. Finally, some of the methods presented will be replicated in other CLINT local case studies, presenting the corresponding outcomes in D7.3.

A summary of the forecasting methods presented in this chapter is provided in Table 1, including information related to the EE addressed, the ML algorithm(s) used, the scale of the application, the classification between fully data driven and hybrid methods, and the lead-times addressed.

Table 1 Summary of methods developed to forecast extreme events described in this report.

Extreme Event	Methodology	ML algorithm	Scale	Fully Data Driven/Hybrid	Lead-time
Drought	Meteorological drought forecasting with ML and climate data	ELM, MLP	Local	Fully Data Driven	Sub-seasonal
Drought	Hydrological drought forecasting with LSTM	LSTM	Local	Fully Data Driven/Hybrid*	Sub-seasonal
Tropical cyclones	Enhanced tropical cyclone rainfall forecasting	RA-U	Global	Fully Data Driven	Medium-range
Tropical cyclones	Deep learning-based approaches for tropical cyclone activity detection and forecasting	UNet	Local	Hybrid	Short to Medium-range
Heatwaves and warm nights	Two-step dimensionality-reduction method for driver identification	PCRO-SL	Local	Fully Data Driven	Nowcasting to seasonal
Concurrent extremes	Compound and concurrent events forecasting with Machine Learning	SIS, KRGCCA, BNN	Local	Fully Data Driven	Seasonal
Hydrological extremes	Machine Learning-based post-processing for hydrological prediction	LSTM, RF	Local**	Hybrid	Seasonal
Hydrological extremes	Extraction of valuable information from multi-timescale forecasts via Reinforcement Learning	DPS	Local	Hybrid	Short-term to seasonal

* Two different approaches have been developed using LSTM, one fully data-driven and one hybrid, using meteorological forecasts as inputs.
** This approach is developed for the entire country of Sweden, but is presented here as local to distinguish it from the analogous approach under development to post-process streamflow at the pan-European scale, currently under development within WP6.

2.1 Lead times

The methods proposed in this report address EE forecasting across different lead times: short and medium-range, from 1 to 14 days ahead, for the cases of tropical cyclones; sub-seasonal, from 15 to 30 days, for the case of droughts; seasonal, from 1 to 3 months ahead, for heatwaves and warm nights.
These lead times differ among each other in terms of the mathematical problem to be solved and sources of predictability. Short and medium-range weather forecasting, mathematically speaking,

can be seen as an initial value problem, since changes in weather are due to changes in the initial conditions of the atmosphere. Seasonal forecasting, instead, is increasingly a problem at the boundary conditions, as climate variability is more strongly driven by slow changes in the atmospheric lower boundary conditions (Vitart & Robertson, 2019). These differences are reflected in the sources of predictability and in the structure of the NWP models used to forecast weather and climate: short to medium-range forecasts mainly look at changes in atmospheric variables, and the NWP models are focused on modelling only that component of the Earth; seasonal forecasts, instead, have their most important source of predictability in variables related to oceans and lower boundary of atmosphere (e.g. sea surface temperature, soil moisture, snow cover), and therefore, atmospheric NWP models are coupled with ocean models as well (Vitart & Robertson, 2019).
In between the short to medium and the seasonal lead times, there is the sub-seasonal range, which is a critical horizon for weather and EE forecasting, since this lead time is short enough for the atmosphere to still have memory of its initial state, but at the same time is long enough for the ocean variability to start influencing the atmospheric circulation (Sahin, 2022). NWP atmospheric models are therefore coupled with ocean models and need to reproduce the interaction between land, ocean and atmosphere.

2.2 Meteorological drought forecasting with ML and climate data

This method aims at forecasting the total cumulative precipitation of the upcoming 30 days, by using information provided by a set of local atmospheric variables, global climate variables, teleconnection patterns and Machine Learning algorithms. Once the forecasts of total precipitation are produced, they are verified against observations and benchmarked against the existing Extended Range (ER) meteorological forecasts produced by ECMWF and simple linear regression models.
Meteorological drought can then be predicted by applying the definition of Standardised Precipitation Index (SPI) (Mckee et al., 1993) on the cumulative precipitation forecasted by the ML models.

2.2.1 Methodology

To forecast total cumulative precipitation of the upcoming month, two different ML-frameworks are developed. Both of them employ local atmospheric and global climate variables as input features, but they differ in the detail of climate information used and, as a consequence, in the ML algorithm adopted, i.e. Extreme Learning Machine (ELM) and Multi-Layer Perceptron (MLP). The two approaches are designed to investigate the influence of climate data in forecasting precipitation at monthly lead-times, specifically the influence of teleconnection patterns. Indeed, the one-month lead time belongs to the sub-seasonal forecasting horizon, which is a critical horizon in weather forecasting, since the atmosphere is still influenced by its initial conditions, but at the same time, it is affected by the slow ocean variability and large-scale phenomena (Sahin, 2022). Moreover, it has been proved that teleconnection patterns could also play a role in the genesis of extreme events and in atmospheric circulation at the sub-seasonal lead times (Specq & Batté, 2022). Atmosphere initial conditions are usually described with local atmospheric variables observed in the location of the target forecast (e.g. precipitation, temperature, wind speed, etc). Large-scale phenomena, instead, are studied through global climate variables (e.g. Sea Surface Temperature (SST), Mean Sea Level Pressure (MSLP), etc), while teleconnection patterns are defined through climate indices (e.g. North Atlantic Oscillation (NAO) Index and others).
An overview of the two frameworks is presented in Figure 1. The text in black refers to both approaches, while the text in green refers only to the one leveraging on explicit climate information, which employs climate indices of teleconnection patterns in addition to the local and global climate

variables. From now on, the approach using only local and global climate variables will be referred to as “Implicit climate information approach”, while the one adopting teleconnection patterns as well will be referred to as “Explicit climate information approach”.
For both the explicit and implicit frameworks, a set of candidate input features is pre-selected based on the climate of the target location and on the variable to be forecasted itself, choosing those atmospheric, climate variables and climate indices that might be linked to the target. The global climate variables (and the teleconnection patterns), then undergo a phase of dimensionality reduction (and teleconnection detection), at the end of which only the first Principal Component (PC) of each variable (and the relevant teleconnections) is retained. These leading PCs are then used together with the candidate local atmospheric variables to forecast total precipitation of the upcoming 30 days using an MLP (ELM) model. In this phase, different combinations of input features (local and PCs of the global variables) are employed to train several ML models, and the set of variables providing the highest model performances is retained. The same set of input features is also used to build a linear forecasting model, which is adopted as a benchmark for the ML models.
The ML models’ performance is then tested by comparing the resulting forecasts against observations. Additionally, the ML models developed are benchmarked against a simple linear forecasting model and the existing Extended Range (ER) meteorological forecasts provided by ECMWF. Both assessments are done by computing the Mean Squared Error (MSE):

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(F_i - R_i)^2 \quad (1)$$

where F_i is the forecasted value for a specific target time i, R_i is the corresponding reference observation (reanalysis in this case) for the same target time, N is the number of samples in the dataset. The score is defined in the range $[0, +\infty)$, with 0 value for perfectly accurate forecasts.

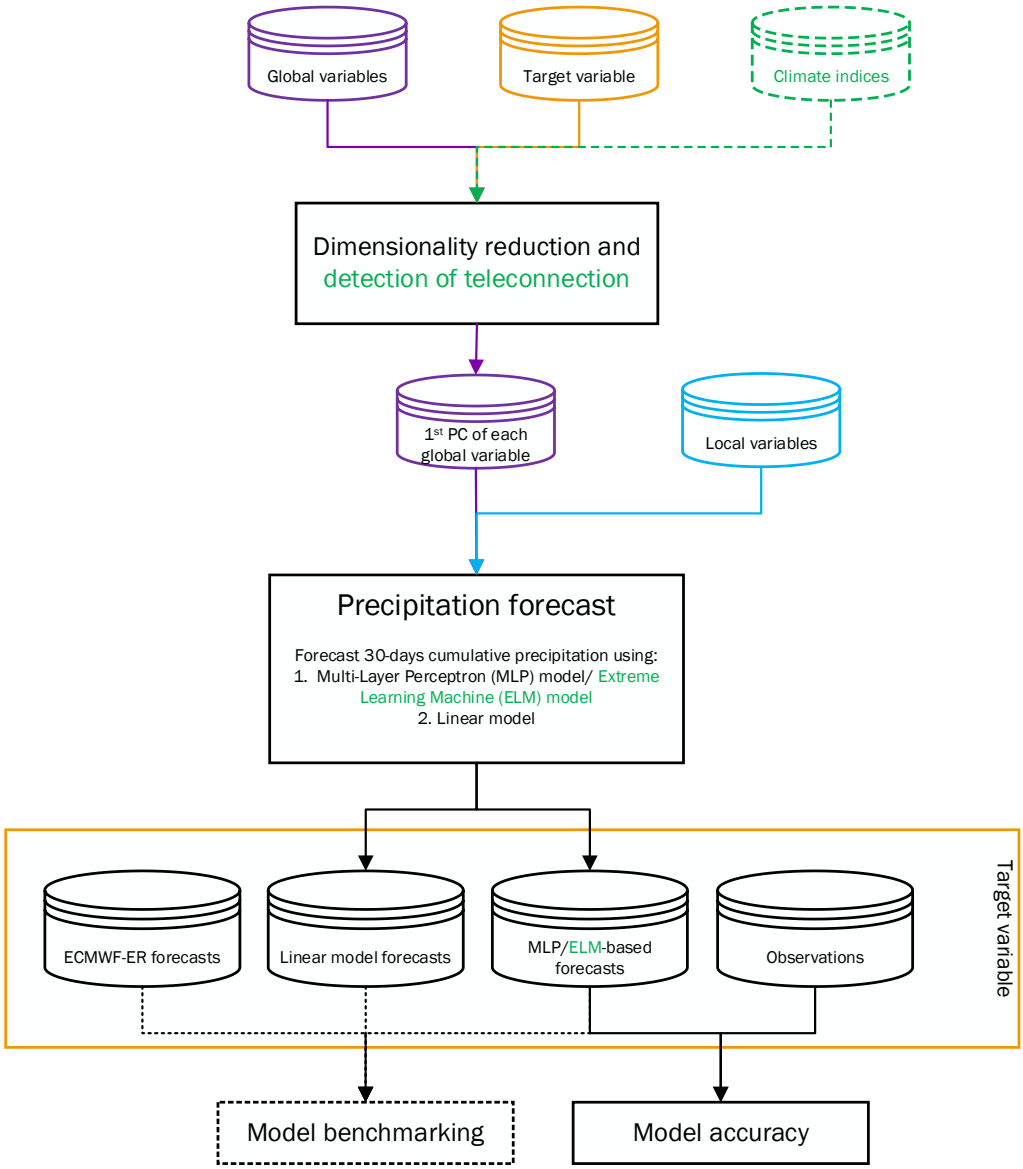

Figure 1 Overview of the two frameworks developed to forecast total precipitation. The text in black refers to both approaches, while the text in green refers only to the framework with explicit climate information.

Explicit climate information approach: Extreme Learning Machine (ELM)

Teleconnections influence the overall atmospheric circulation patterns, the genesis of extreme events (Specq & Batté, 2022), and precipitation variability at sub-seasonal lead-times (Jones & Dudhia, 2017). Based on this evidence, the explicit climate information approach developed here leverages on the detailed information of teleconnection patterns and their phases to build a forecasting model for cumulative monthly precipitation. This approach builds up from the Niño Index Phase Analysis (NIPA) developed by Zimmerman et al., (2016) and further expanded with the Climate State Intelligence (CSI) by Giuliani et al., (2019), which aimed at exploiting the behaviour of global climate variables during specific phases of teleconnection patterns to forecast seasonal precipitation. The original frameworks have been here readapted to predict precipitation at monthly lead-times and extend the pool of teleconnection patterns taken under consideration, according to the location of the target forecast. The information on the teleconnections is provided by the relative climate indices, which are provided per each calendar month of the year. For this reason, the explicit climate information approach developed 12 different forecasting models based

on ELM, one for each month of the year. This framework also employs global climate and local atmospheric variables as input feature candidates.

The NIPA framework acts as an input feature selection and dimensionality reduction algorithm, as it both detects the meaningful pairs of teleconnection patterns and global variable, and it reduces the dimensions of the global variables selected. The algorithm is run for each pair of teleconnection pattern and global variable selected. The steps followed by NIPA are described below, using as an example the investigation of the influence of one month back North Atlantic Oscillation (NAO) and Mean Sea Level Pressure (MSLP) on the total precipitation (TP) in the month of February. The same processes are repeated for any calendar month, and for any combination of teleconnection and climate variable up to three months in the past. The steps are:

1. Based on the climate indicator, divide the years into positive and negative phases, and divide the samples of the target variable (TP of February) and the global climate variable (MSLP) accordingly;
2. For each phase separately, evaluate the linear correlation between the target variable in the target location and the global variable across the world. To avoid spurious correlations, only areas with the 95% significance correlation and that are contiguous are selected;
3. Run a Principal Component Analysis (PCA) using the time series resulting from step 2, with the purpose of reducing the dimensions of the global variable;
4. The PC obtained is used to build a linear regression model to forecast the target variable. For each tuple of climate signal and global variable, one linear model per signal phase is built;
5. Pearson correlation between the forecasted and observed variable is computed, and if the value obtained is above a predefined threshold, then the PC of the considered global variable (MSLP) coupled with a specific phase of the teleconnection pattern (NAO) is selected to be one of the input features for the forecasting model.

The entire procedure just described makes sure that the forecasting models built later on leverage on the predictability given by a certain global variable when a specific climate signal occurs, and that not relevant climate signals and/or global variables are discarded. Moreover, teleconnections impact weather and climate differently according to seasonality. The NIPA approach ensures that these impacts are captured by the forecasting models built, by selecting the relevant teleconnections in each month, and hence each season.

The teleconnections and global variables detected by NIPA are then used to group the years into specific climate states, based on the phases of each of the selected teleconnections. For instance, in case two teleconnection patterns are identified (S_1, S_2), both with positive (S_i^+, S_i^-) and negative phases only, then four different climate states (CS_j) are possible: $CS_1 = [S_1^+, S_2^-]$, $CS_2 = [S_1^+, S_2^+]$, $CS_3 = [S_1^-, S_2^-]$, $CS_4 = [S_1^-, S_2^+]$. An integer label is assigned to each of these climate states and is further used as input in the ML forecasting models, together with the PCs of the global variables of each original climate signal and the local atmospheric variables. The latter variables are additionally included with respect to the original CSI framework (Giuliani et al., 2019) because of their potential influence on precipitation predictability at sub-seasonal lead-times.

While the NIPA and CSI frameworks ensure the extraction of relevant climate information, they have a considerable drawback in the amount of data that is available for training, validating and testing the ML-based forecasting models. Indeed, by detecting the relevant climate signals per each month of the year, we segment the initial sample dataset in twelve subsets (one per month), reducing the sample size of the dataset available for each model/month. Moreover, the addition of a label for each CS implies another implicit dataset segmentation, resulting in challenges for the ML models to generalise. To address this drawback, four solutions are implemented, two related to the input features considered and two to the ML models developed. First, for each teleconnection pattern analysed, only the positive and negative phases are considered, while potential neutral phases are neglected, hence reducing the number of possible CS labels. Second, the total number of input

features for each model is fixed to five, with one variable being the CS label, with a maximum of two global variables, and the remaining are local atmospheric variables. The constraint on the minimum (0) and maximum number of global variables is applied as the goal of this approach is to test the importance of climate information in forecasting precipitation. The best combination of input features is selected based on model performance, as described later on in this section. Third, an Extreme Learning Machine (ELM) model is preferred over conventional Artificial Neural Networks (ANNs), as they randomly initialise the parameters of the hidden nodes and later optimise their values with a one-step matrix product (Huang et al., 2006), resulting more efficient for small training datasets. The same ELM was employed and proved to perform better than conventional ANNs in the original CSI framework (Giuliani et al., 2019). Fourth and final, the performance of the models has been assessed with a Leave One Out Cross-Validation (LOOCV) approach, as in the work of Giuliani et al., (2019).
The pre-selected variables are then employed to derive sets of candidate input features, ensuring to consider all possible combinations of variables that respect the constraint previously described. Each of these sets is then used to train the 12 ELM models and the set performing best in validation/testing is selected as the final model. During the training, each model (per month and per input feature set) is trained separately, allowing to choose the number of neurons and the activation function that ensure better performances.

Implicit climate information approach: Multi-Layer Perceptron (MLP)

Sources of sub-seasonal predictability are given by slow varying phenomena, such as the ocean variability, large-scale phenomena, and stratospheric initial conditions. Sea Surface Temperature (SST), Mean Sea Level Pressure (MSLP), and Geopotential Height at 500 hPa (Z500) proved to be among the global climate variables responsible for S2S increased predictability (He et al., 2019, and citations hereafter, Lee 2019, Ardilouze et al., 2021). The Implicit climate information approach has been designed to investigate whether the influence of slowly varying and large-scale phenomena on precipitation predictability can be leveraged by directly incorporating global climate variables into an ML forecasting model, without providing explicit information on the occurrence of teleconnection patterns. This second framework employs local atmospheric and global climate variables only. Also in this case, global climate variables undergo a dimensionality reduction process, which is similar to the one followed in the Explicit approach, with the main difference that there is no segmentation of the dataset based on climate signal phases. Hence, the dimensionality reduction procedure reduces to only two of the steps described before:

1. Evaluate the linear correlation between the target variable in the target location and the global variable across the world. To avoid spurious correlations, only areas with the 95% significance correlation and that are contiguous are selected;
2. The time series resulting from step 2 are used to run a Principal Component Analysis (PCA), with the purpose of reducing the dimensions of the global variable. The first PC is retained and used as candidate input feature for the forecasting model.

Discarding teleconnection patterns and their related indices induces two modifications in the forecasting problem settings. Firstly, this framework predicts the total precipitation of the upcoming 30 days and not of the upcoming calendar month, as there is no need for restrictions to monthly data due to the fact that teleconnection patterns and their related indices are discarded. More specifically, the target precipitation is computed by summing the total precipitation fallen in moving windows of 30 consecutive days, for a total of 365 (or 366) samples per year, determining a considerable increase in the number of sample data available for training, validating and testing the ML model. Secondly, only one ML model is built to forecast precipitation, as we are not interested in capturing the different impacts of climate signals in each month and season anymore. The goal

of this framework, therefore, is also to check whether the increased data availability can make up for the lack of detailed climate information provided by the teleconnection patterns.
Because of the increased sample size, moreover, the Implicit climate approach adopts a Multi-Layer Perceptron (MLP) model, which is a version of ANN made of fully connected neurons with a nonlinear activation function, as ELM is not optimal for training on large datasets. Different MLP models are trained independently, one for each set of candidate features, following the conventional training-validation-testing split. The set of variables providing the best model performance during validation is then used to further test the model and for forecasting purposes. The sets of candidate features are created by making all possible combinations of local and global (PCs) variables, constraining them to a minimum number of 5 and a maximum number of 10 variables per set. The increased number of input features per set is chosen because of the increased sample size.

Linear model

In order to check whether the performance of the forecasting models of both approaches are due simply to the input features considered or also to the capabilities of the ML model selected, the results of the two frameworks based on ML have been compared to those of simple multi-linear regression models. More specifically, a total of thirteen linear models have been built to compare against the twelve ELM and the MLP models, by using the best sets identified during the validation of the ML model as input features.

2.2.2 Implementation

The methodological frameworks described in the previous section are developed and applied in the case study of Rijnland, in the Netherlands, which faces issues of both meteorological and hydrological drought, especially during summer months.
Both local and global climate variables are extracted from the ERA5 reanalysis dataset (Hersbach et al., 2023), which has a native spatial resolution of 0.25° x 0.25°. Because of its considerable spatial extension and to improve computational speed, the global variables dataset has been downloaded at the coarser resolution of 1.5° x 1.5°. As a consequence, to match the spatial resolution of local and global climate variables, the local atmospheric features have been upscaled to the bigger grid of 1.5°. This process consisted in simply averaging, per each time step, the values of the local atmospheric features in the cells of 0.25° that belong to the same cell of 1.5°. At the moment of performing the experiments, the validated ERA5 dataset was available with hourly frequency from 1979 to present, and the back-extension to 1940 was still available in a preliminary form. For this reason, the variables considered in this implementation are from 1979 to 2022. Both global and local variables have been aggregated to a monthly time step, either through summation or averaging procedure, based on the nature of the specific variable, i.e. instantaneous or aggregated. Climate indicators for teleconnection patterns, finally, are available for every calendar month and are downloaded from the website of National Oceanic and Atmospheric Administration (NOAA).
The set of candidate input features is preselected mainly following previous studies in literature: local variables are chosen following the work of (Felsche & Ludwig, 2021), in which the authors were forecasting the SPI-1 for a case study in Portugal and in Paris, and readapting them for the case study of Rijnland and considering the available variables within ERA5 dataset; teleconnection patterns (and the indicators used to describe them) are selected based on their influence on the Netherlands and on North Europe, in general; global climate variables are chosen based on their influence on predictability according to literature (e.g. He et al., 2019) and/or accordingly to the teleconnection patterns, e.g. since North Atlantic Oscillation (NAO) is related to a difference of sea level pressure, MSLP is selected. The set of candidate input variables is summarised in Table 2:

Table 2. Summary of candidate input features selected for both the Explicit and Implicit climate information approaches.

Acronym	Full name	Type	Source
NAO	North Atlantic Oscillation	Teleconnection pattern	NOAA
SCA	SCAndinavian oscillation		
EA	East Atlantic oscillation		
ENSO	El Niño Southern Oscillation		
SST	Sea Surface Temperature	Global climate variable	ERA5
MSLP	Mean Sea Level Pressure		
Z500	Geopotential height at 500 hPa		
TCC	Total Cloud Cover	Local atmospheric variables	ERA5
MER	Mean Evaporation Rate		
MSSHF	Mean Surface Sensible Heat Flux		
TCWV	Total Column Water Vapour		
UW	U-component of Wind		
VW	V-component of Wind		
RH	Relative Humidity		
SH	Specific Humidity		
TP	Total Precipitation		
T2M	Temperature 2 meters above surface		

The best sets of input features and hyperparameters obtained for the MLP and the ELM approaches are shown in Table 4 and Table 3, respectively:

Table 3. Sets of input features and model hyperparameters employed to forecast precipitation using the MLP model.

Dataset	Hyperparameter	Best values
TP T2M SH UW VW TCC MSSHF MSLP Z500	Activation	Sigm
	Neurons layer 1	30
	Neurons layer 2	28
	Initial learning rate	0.01

Table 4. Sets of input features and model hyperparameters employed to forecast precipitation using ELM models. The sets shown in the table are those which provided the lowest MSE score when compared against reanalysis data. The local variables are indicated in black, while the global are in orange. The global variables are coupled with the teleconnection pattern that resulted relevant to forecast precipitation, with the convention of teleconnection acronym first, followed by the global variable. Moreover, these coupled acronyms are followed by a number, which represents the number of months in the past during which the global variable is averaged. For instance, in the case of January, one of the variables is EA_Z500-2, meaning that the averaged geopotential height of December and November, when EA teleconnection is in place, is one of the predictors for January precipitation. The columns

"Neurons" and "Activation" provide, respectively, the number of neurons and the type of activation function used for each of the 12 models developed.

Month	Dataset	Neurons	Activation
January	T2M TP EA_Z500-2 SCA_Z500-1	8	Relu
February	T2M TCWV ENSO_MSLP-3 SCA_SST-3	9	Sigm
March	MSSHF SCA_SST-1 NAO_MSLP-1	10	Sigm
April	TP NAO_Z500-3 NAO_SST-1	4	Sigm
May	TCC TCWV EA_Z500-2 ENSO_Z500-1	8	Sigm
June	EA_Z500-2 NAO_Z500-1	10	Sigm
July	UW VW SCA_MSLP-1 NAO_Z500-1	8	Sigm
August	SCA_MSLP-1 NAO_MSLP-2	9	Sigm
September	MER EA_Z500-2 NAO_Z500-1	8	Sigm
October	RH SH EA_MSLP-1 ENSO-mei_SST-1	8	Sigm
November	SH SCA_MSLP-2 EA_SST-1	12	Sigm
December	SD TCWV NAO_MSLP-3 EA_MSLP-3	8	Sigm

From the input features selected for the ELM approach it emerges that, in general, the relevant predictors differ for each month, and the same combination is never repeated. It is also interesting to note that for both ELM and MLP approaches, global climate variables were always selected as input features, confirming the influence of large-scale phenomena for forecasting precipitation at sub-seasonal lead times. Moreover, in two cases for the ELM approach, in the months of June and August, climate information seems to be the only relevant to forecast precipitation, with no local atmospheric information needed.

2.2.3 Results

Total precipitation of the upcoming month is forecasted using two ML approaches, which differ both for the ML algorithm used and for the level of detailed climate information used. The first approach is based on ELM coupled with the NIPA framework, which produced 12 different forecasting models (one for each calendar month), using local climate variables, global climate variables and teleconnection patterns. Each model is trained with the set of input features that maximises the model performances. The second approach is based on an MLP model that is trained for the entire year using the combination of local and global variables that gives the lowest MSE score when compared against reanalysis data.
The performances of the ML models in forecasting total precipitation are assessed in terms of Mean Squared Error (MSE), by comparing the forecasts against reanalysis data. Additionally, MSE is computed also for the linear models and for the precipitation forecasts given by the ECMWF Extended Range, to benchmark the ML models performances against more simple and numerical weather prediction models, respectively. It should be noted that as the ELM approach provides forecasts for each calendar month, all the models are tested per each calendar month. The results are presented in the radar plot in Figure 2:

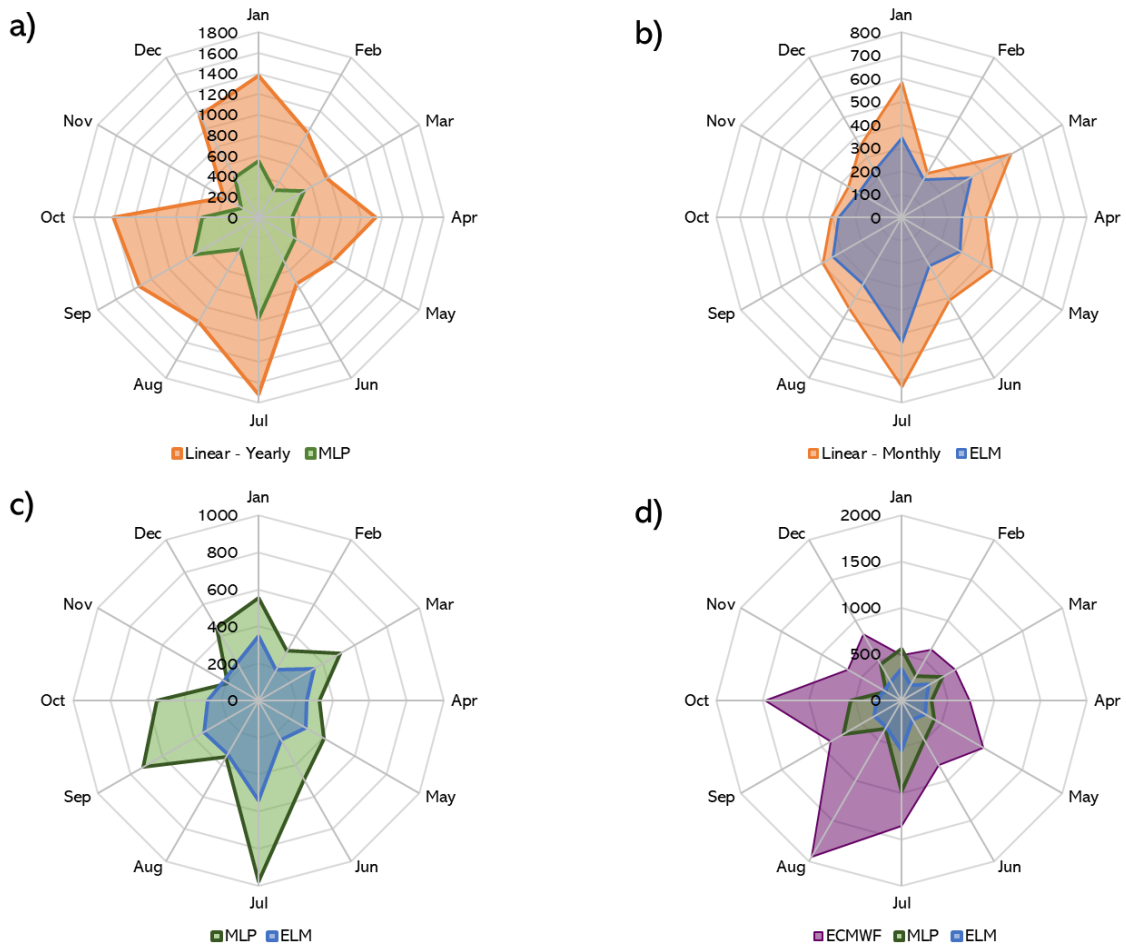

Figure 2 Radar plot with performances of: a) MLP and linear model; b) ELM and linear model; c) ELM and MLP models; d) ELM, MLP and ECMWF precipitation forecasts. Linear models are distinguished between monthly and yearly, to differentiate from the linear models built to benchmark the ELM models (monthly) and the one built to compare against the MLP model (yearly). The performances are expressed in terms of MSE, which grows from the centre towards the outer of the circles, indicating better performances when closer to the centre.

As it can be observed, both ML approaches outperform their linear counterparts (Figure 2a) and b)), showing that the relatively low MSE values are not due only to the input features selected, but to the ML-enhancement. The difference in the performances between ML and linear models, however, is not constant, and becomes lower for the autumn months, especially in the case of the Explicit climate information approach (ELM and monthly linear models).

In most of the months, the ELM model outperforms the MLP (Figure 2c)), showing that the detailed climate information provided by the Explicit approach can counterbalance the reduced sample size used by the ELM. Indeed, the Explicit climate information approach allows for the selection of the best input features for each month, while the Implicit approach forces the model to select the best input features that overall describe the different processes happening throughout the year, trying to generalize them, neglecting seasonal variability.

Finally, both the machine learning approaches developed outperform the forecasts given by the ECMWF ER, which are obtained with a numerical weather prediction (NWP) model. Higher differences in MSE values between the two approaches, i.e. based on AI and on NWP models, are found in the period July-December, while they are lower for the other months, with a minimum for the month of March. An explanation of the superior performances of the AI-based approaches could be due to the fact that both the ML models developed are tailor-made both for the target variable and the target location, while NWP models are general-purpose models calibrated and validated to forecast multiple weather variables across time and space.

Conclusion and next steps

The Machine Learning algorithms developed proved to forecast total precipitation better than existing forecasts issued by ECMWF, with the approach leveraging on explicit climate information being the best overall among those tested. Potential reasons for the improved forecasts are that ML models are tailor-made to predict one specific variable and in one specific location, whereas conventional NWP models are calibrated to predict multiple variables globally. The forecasts obtained with ML models can then be used to forecast drought events by applying commonly used indicators, such as the Standardised Precipitation Index (SPI) or the Standardized Precipitation Evapotranspiration Index (SPEI). The Explicit climate information framework is currently being adapted to forecast streamflow and seasonal precipitation in some local case-studies (Lake Como, Zambezi river basin). The results of these approaches will be provided later in the project in the deliverable D7.3 *AI-enhanced Climate Services for local decision-making*.

2.3 Streamflow forecasting with Long Short-Term Memory (LSTM) models

Since their first appearance in rainfall-runoff modelling (Kratzert et al., 2018), Long Short-Term Memory (LSTM) models have been widely used for streamflow prediction for single catchments (Li et al., 2023), ungauged basins (Arsenault et al., 2023), and global modelling (Kratzert et al., 2019; Nearing et al., 2024), both in their classical shape or in combinations with Convolutional Neural Networks (CNN-LSTM, ConvLSTM) (Ghimire et al., 2021; Oddo et al., 2024). LSTM models are now considered state-of-the-art for streamflow prediction. However, their potential in forecasting streamflow at sub-seasonal lead-times has not been fully explored yet, unless for hybrid LSTM models (Fan et al., 2023), i.e. LSTM coupled with other ML algorithms.

2.3.1 Methodology

In this work, we develop a simple LSTM model for forecasting streamflow, at daily or weekly scales, at the outlet of a catchment with lead-times up to four weeks. Instead of investigating the performance of different model coupling (e.g. Conv-LSTM, Encoder-Decoder LSTM), we explore the potential of relevant input features and target aggregation to maximise the performance of a simple stacked LSTM. More specifically, we developed two different approaches, the first leveraging only past observations or reanalysis data, and the second adopting a combination of observations and meteorological forecasts as input features. The two approaches are developed and tested in two different case studies, Rhine river and Douro catchment, respectively. In both cases, experiments with daily streamflow forecast at short to medium lead-times (up to seven days) are carried out to check the potentialities of LSTM models. In the case of forecasts with lead-times from 1 to 4 weeks ahead, weekly streamflow values (minimum or sum) are used as target, as it might not be possible to accurately predict the weather, and hence weather dependant variables such as streamflow, at daily resolution for sub-seasonal lead times (Buizza & Leutbecher, 2015).

In both approaches, a set of candidate variables is initially selected according to the hydrological processes that the LSTM is aimed at reproducing, based on literature and expert knowledge. More specifically, precipitation, temperature and past streamflow observations have been pre-selected, with temperature being considered as a proxy for seasonality and snow-melt processes. Moreover, both precipitation and temperature data are averaged over the entire catchment area. Cross-correlation analysis has been carried out for two purposes: 1) to check whether the candidate predictor has influence on the target variable and quantify this influence; 2) to determine the maximum lag time between candidate predictors and target, after which the predictor has no longer valuable influence on the target. Once the forecasting models are trained, they are compared against observations by computing the Nash Sutcliffe Efficiency (NSE) coefficient, according to the equation:

$$NSE = 1 - \frac{\sum_{i=1}^{N}(R_i - F_i)^2}{\sum_{i=1}^{N}\left(R_i - \underline{R}\right)^2} \quad (2)$$

Details on the two experiments are given in the two following sub-sections.

Streamflow forecasting with LSTM and historical data

This approach leverages on historical data, either from local observations or reanalysis data, to predict streamflow at the outlet of the catchment for lead-times up to 4 weeks ahead. Average precipitation and temperature over the catchment, past streamflow observations in the outlet and in the upstream stations are pre-selected as input candidates.
After a cross-correlation analysis between the candidate inputs and the target, a first set of experiments was carried out in order to further investigate the relevance of the retained input variables and to define the structure of the LSTM model to be further used. Experiments were carried out both at the daily and weekly aggregation, in the former case forecasting the average daily discharge (Q), while in the latter the target was the weekly minimum discharge (weekly min Q), as the method is developed to forecast hydrological drought, and hence focuses on low flows. A summary of the first set of experiments is presented in Table 5:

Table 5 Summary of experiments performed to determine the LSTM optimal structure and the input features, both at the daily and weekly aggregation.

LSTM	Experiment ID	Input Variables	Target
Univariate	**U1**	Q_{outlet}	Daily Q_{outlet}
	U2	Q_{outlet}	Weekly min (Q_{outlet})
Multivariate	**M1**	Q_{outlet}, $Q_{ups\ 1}$	Daily Q_{outlet}
	M2	Q_{outlet}, $Q_{ups\ 2}$	
	M3	$Q_{all\ stations}$	
	M4	$Q_{all\ stations}$, P, T	
	M5	Weekly min ($Q_{all\ stations}$, P, T)	Weekly min (Q_{outlet})

The experiments started considering only past streamflow observations at the target location (Q_{outlet}), to check how far in time the predictability from past observations was lost. Then, observations from upstream stations were added (Q_{ups1}, Q_{ups2}), followed by the combination of several upstream discharge stations ($Q_{allstations}$), precipitation and temperature averaged over the catchment. The LSTM model structure was then fixed to the parameters shown in Table 6. All the experiments are carried out using a 0.9-0.1 split for training and testing, a subsequent 0.7-0.3 split for training and validation purposes, Adam optimiser, and Mean Squared Error (MSE) as loss function. To avoid overfitting during training of the LSTM models, an early stopping procedure is introduced to stop the training when the validation loss function is not improving for ten consecutive steps. All the variables, moreover, are normalised in the range [0,1] to facilitate model learning.

Table 6 LSTM model architecture.

Layer	Units	Output shape	Parameters	Activation function	Batch size
LSTM	60	(None, 30, 64)	18176	ReLu	
LSTM	54	(None, 50)	23000	ReLu	64
Dense	1	(None, 1)	51	Linear	

Once the LSTM model structure was fixed, a second set of experiments was performed, to analyse the influence of different aggregations for the input features and target. These experiments were carried out only for forecasting streamflow at the weekly temporal aggregation, as the focus of this method is to forecast streamflow at sub-seasonal lead-times, and using only precipitation and temperature as input features. The details of the experiments are shown in Table 7:

Table 7 Summary of the experiments on the aggregation of the input features and the target for the LSTM model.

Experiment ID	Input Variables	Target
T1	$min(P), min(T)$	$min\ (Q)$
T2	$anomaly\ (\bar{P}), anomaly\ (\bar{T})$	$anomaly(min\ (Q))$
T3	$min(P), min(T)$	$\Delta(min(Q))$
T4	$min(P), min(T)$	$Log(min(Q))$

Experiment T1 consists in predicting the weekly minimum streamflow using as input features the weekly minimum values of precipitation and temperature. The same input features are considered for experiments T3 and T4, where the target variables are, respectively, the difference between the weekly minimum streamflow of the target week and the weekly minimum streamflow of the week in which the forecasts are made (T3), and the logarithm of the weekly minimum streamflow (T4). Both the targets of T3 and T4 have been chosen because they have been reported to have good performance for the prediction of low flows (Deng et al., 2024). Experiment T3, instead, was performed using the anomaly of average precipitation and temperature ($\bar{P}, \bar{T}$) to forecast the anomaly of the weekly minimum streamflow. In this latter experiment, anomalies were considered to evaluate the influence of deviations from the long-term signals. The anomalies were computed monthly by averaging the variables over a period of 30 years.
Finally, once the second set of experiments was carried out, the settings of the best-performing models in both sets were merged. Two different model configurations were tested, both targeting the weekly minimum anomaly of streamflow, but one using all the input variables available (F1), while the other only adopted temperature, precipitation and streamflow at the target location (F2). The performances of all models were then critically compared.

Streamflow forecasting with LSTM, historical data and meteorological forecasts

This approach leverages on the information provided by historical observations and meteorological forecasts to predict streamflow with lead-times up to 4 weeks. Different LSTM models have been trained to predict daily streamflow in short-term horizons and to predict weekly accumulated streamflow, as the framework was originally developed to forecast inflow into a reservoir.
A set of initial experiments was performed to identify the optimal model structure, by forecasting daily streamflow. It was then chosen to use a stacked LSTM model, with two LSTM layers of 32 and 40 units, respectively, and a batch size of 32. An early stopping criterion was introduced, to avoid overfitting of the model, which stopped the training in case the validation loss function was not

improving for ten consecutive steps. For the daily forecasts, a window size (also called look-back) of 30 days was found to be optimal.
Once the LSTM architecture was fixed, it was used for a second set of experiments to predict weekly accumulated streamflow, hence in terms of volumes, for lead-times ranging from 1 to 4 weeks ahead. The window size was varied again, to identify the optimal one for weekly predictions and was finally set to 16. Different input variables were tested to define the best set of features, starting initially with only observed data (as in the previous approach) and then adding precipitation and temperature forecasts from the ECMWF-ER, as the models were not performing well with only observations (see section 2.3.2 for more details). For the daily experiments (Di), precipitation (P), temperature (T) and daily streamflow (Q) were employed to predict daily streamflow. In the weekly experiments, accumulated weekly inflow (I) was used instead of streamflow, in addition to precipitation and temperature. Accumulated inflows are derived by transforming streamflow into volumes and then summing them up for a period of 7 days. Precipitation and temperatures are averaged over the entire catchment. The set of experiments and LSTM models trained for both daily and weekly predictions are shown in Table 8:

Table 8 Summary of LSTM models trained for daily and weekly streamflow predictions.

Experiment ID	Input Variables	Observations/Forecasts
D1	P, T, Q	Observations
D2	P, T, Q	Observations
	P, T	Forecasts
D3	P, T, Q	Observations
	P, T	Forecasts
D4	P, T	Observations
	P, T	Forecasts
W1	P, T, I	Observations
W2	P, T, I	Observations
	P, T	Forecasts
W3	P, T, I	Observations
	P, T	Forecasts
W4	P, T	Observations
	P, T	Forecasts

Looking at Table 8, experiments D2, D3 (and analogously W2, W3) look to be the same, however the LSTM model configuration is different. Indeed, in experiment D2 (W2), each time step of both observed and forecasted variables is treated as a different input feature and not as unique time series, as streamflow (inflow) forecasts are not available. The LSTM model is trained from the beginning with both observations and forecasts together. In experiment D3 (W3), instead, the same input configuration is retained, however, the model is pre-trained with observations, also for the time steps that are in the future. This setting is to resemble the training of the model with the perfect forecasts. Once the LSTM is trained, the model is finally fine-tuned by introducing the meteorological forecasts in place of the perfect forecasts, i.e. observations of future time steps. Model D4 (W4), finally, uses both observed and forecasted precipitation and temperature, but

merged as a single time series. This means that in this setting, the LSTM model has two input features, having as many time steps as the length of the window size.
In all the experiments, the window size was kept constant. This means that in the weekly experiments using only observations, each variable had 16-time steps in the past, while in those experiments which employed also forecasts, the sum of the time steps for the observations and forecasts was 16. In the case of one-week lead-time, for instance, 15-time steps were considered from observations and 1 from forecasts, while in the case of 2-week lead-time, 14-time steps were retained from observations and 2 from forecasts, and so on.

2.3.2 Implementation

Streamflow forecasting with LSTM and historical data

The approach to forecast streamflow with LSTM and historical data was developed and tested for the CLINT case study of Rijnland, predicting streamflow at the station of Lobith, where the Rhine river enters the Netherlands. More details on this case study can be found in D7.1 and D7.2.
Streamflow observations from Lobith station and five upstream stations located in the main channel (Koeln, Kaub, Worms, Maxau, Basel) are downloaded from the the Global Runoff Data Centre (GRDC). Rainfall and temperature data, instead, are downloaded from the ERA 5 reanalysis dataset (Hersbach et al., 2023), which originally have a native spatial resolution of 0.25° x 0.25°, and are spatially averaged over the catchment. All the data were considered in a time window from 1979 to 2018, to match the periods in which both streamflow observations and meteorological reanalysis were available.

Streamflow forecasting with LSTM, historical data and meteorological forecasts

This approach was developed and tested in the Douro river basin, which is one of the CLINT case studies. The LSTM models developed aimed at forecasting accumulated weekly inflows into the Barrios de Luna reservoir, which belongs to the Órbigo system. More details on this case study can be found in D7.1 and D7.2.
The observed precipitation and temperature were provided by the Douro River Basin Authority (RBA) and are derived from the SPAIN02 dataset, which is a 20 km daily rainfall analysis from the Spanish Meteorological Agency. The historical observations of reservoir inflows are also provided by the Douro RBA, and are computed from a water balance analysis in Barrios de Luna Reservoir. The variables are referred to the period 1958 to 2015.
The meteorological forecasts of precipitation and temperature were downloaded from the ECMWF ER open-source dataset, which has a resolution of 1.5° x 1.5 °. The reforecasts (or hindcasts) of the model issued between June 2019 and June 2020 were downloaded, and the ensemble mean of the reforecasts- of the years 1999-2015 (or 2000-2015, according to the initialisation date) was employed.

2.3.3 Results

Streamflow forecasting with LSTM and historical data

This method leveraged historical data to forecast streamflow at the outlet of a catchment with a stacked LSTM model. Several experiments have been carried out to investigate the influence of the input features considered (precipitation, temperature, streamflow in the target location and in five stations upstream) and of the aggregation of both the input and the target.

The first set of experiments was performed to identify the optimal LSTM model structure and the set of best input features, which were all retained from the initial set of candidate features after the cross-correlation analysis. The results in terms of NSE are presented in Figure 3 and Figure 4, respectively, for the daily and weekly targets.

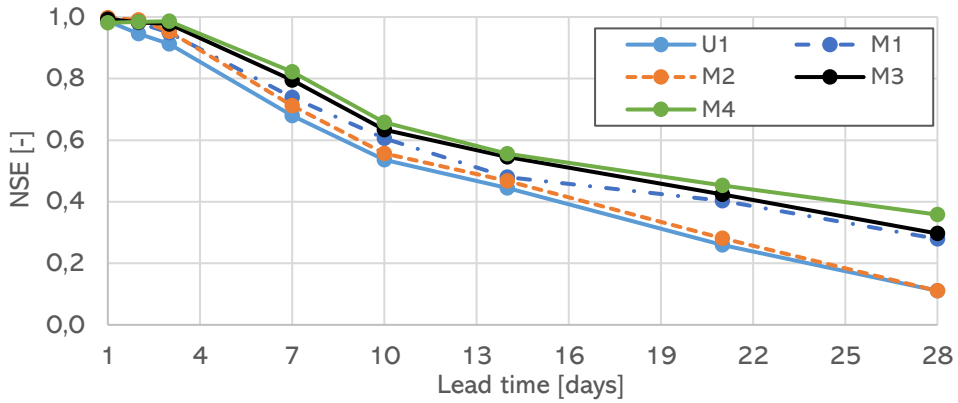

Figure 3 Performance of the LSTM models trained with different input features in predicting daily averaged streamflow (Q) in the outlet of the Rhine. Experiment U1 is carried out using only Q as input features, while those marked with "M" are the multi-variate experiments, i.e. carried out using more than one input feature at the time (M1 and M2 employ Q from the outlet and one upstream station, M3 adopts Q observations from all the selected stations, M4 uses Q observations from all the stations and precipitation and temperature averaged over the catchment).

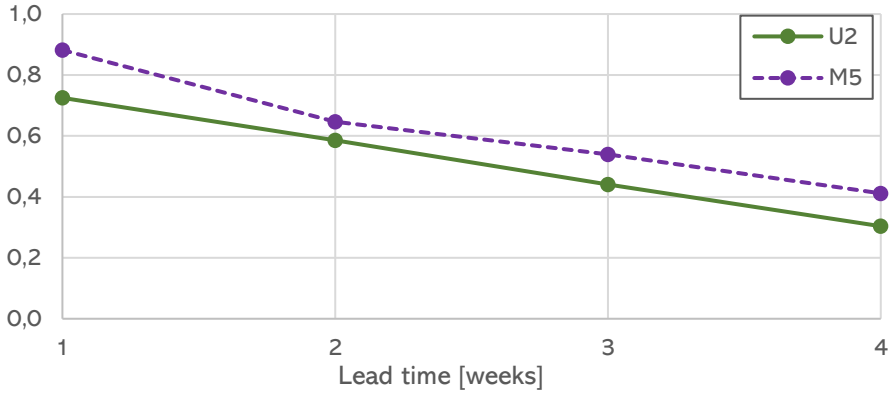

Figure 4 Performance of the LSTM models trained with different input features in predicting weekly minimum streamflow Q in the outlet of the Rhine. Experiment U2 is carried out using the weekly minimum of streamflow Q in the outlet station as only input, while experiment M5 employs the weekly minimum of Q observed in all the selected stations and the weekly minimum of the precipitation and temperature averaged over the catchment.

As it can be seen from Figure 3, augmenting the number of input features from only past streamflow at the target location, to streamflow observations at multiple locations, and finally also adding temperature and precipitation averaged over the catchment, considerably improve the model performances across all the lead-times. The best performances are obtained in the last experiment M4, when all the streamflow and meteorological inputs are considered. It is also interesting to observe that when all the past streamflow observations are included (M3), the NSE values are almost equal to the ones obtained with also precipitation and temperature. A similar behaviour is found for the weekly experiments in Figure 4. In the case of the daily forecasts (Figure 3), all the models trained with different combinations of input features have high performances for the first 3 days of lead-time, with NSE values above 0.95. With longer lead-times, however, the forecasting accuracy quickly drops, with higher decreases for the models with fewer input features, and reaching a value of 0.65 for 10 days lead-time for experiment M4. In the case of weekly predictions

(Figure 4), a similar behaviour is also found, with differences in the NSE values, which are overall lower than for the daily forecasts, reaching a maximum of 0.9 for the first week.
Once the best model settings were fixed, a set of experiments to investigate the influence of the aggregation of both target and input variables was performed. As the goal was to investigate the maximum performance achievable by changing the aggregation of these variables only, we chose not to use the best set of input features to run this new set of experiments, to voluntarily challenge the LSTM models. Spatially averaged precipitation and temperature were used as only inputs. The experiments were carried out only for the weekly forecasts, which are the focus of the work. The results are presented in Figure 5.

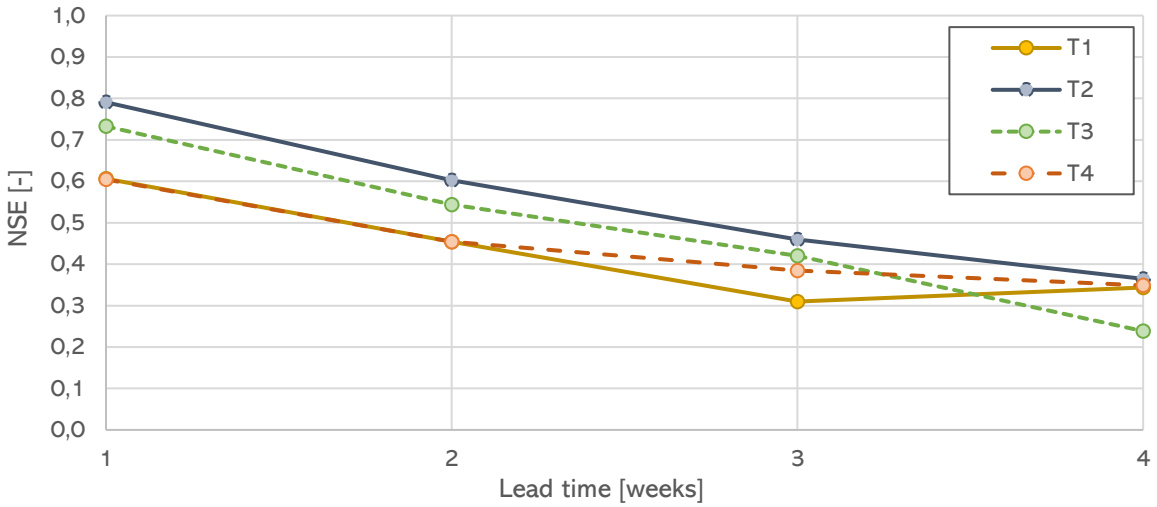

Figure 5 Performances of the LSTM model in forecasting weekly streamflow using different aggregations for the input and target variables. Experiments T1, T3 and T4 employ the weekly minimum precipitation and temperature (averaged over the catchment) to predict the weekly minimum streamflow Q (T1), difference between the minimum Q of the target week and of the initialisation week (T3), and the logarithm of the weekly minimum Q (T4). Experiment T3, instead, adopts the anomalies of the weekly averaged temperature and precipitation to predict the anomaly of the weekly minimum streamflow.

The best model performances are obtained when the LSTM is trained using the anomaly of the weekly average of precipitation and temperature to predict the anomaly of the weekly minimum streamflow in the outlet (T2). A potential reason for the improved performances might be that when transforming the variables (input and output) into anomalies, their skewness diminishes sensibly, making it easier for the LSTM model to learn. Skewed datasets, indeed, challenge LSTM models to learn. The second to best-performing experiment is the one in which the target is the difference between the minimum weekly streamflow of the target week and the same variable in the week of forecast "initialisation". Also in this case, potential reasons for the high performances are to be found in the change of distribution of the sample data. As for the previous experiments, the NSE values are high for lead-time of one week (0.79) and quickly decreases, arriving at values of nearly 0.4 for week 4.
Finally, four LSTM models were trained combining the best input variables and the best aggregations previously found, i.e. using weekly anomalies of all the streamflow time series, precipitation and temperature to predict anomalies of weekly minimum streamflow at the outlet (F1). The same experiment was repeated without using the streamflow observations from upstream stations (F2). The performance of the models are shown in Figure 6.

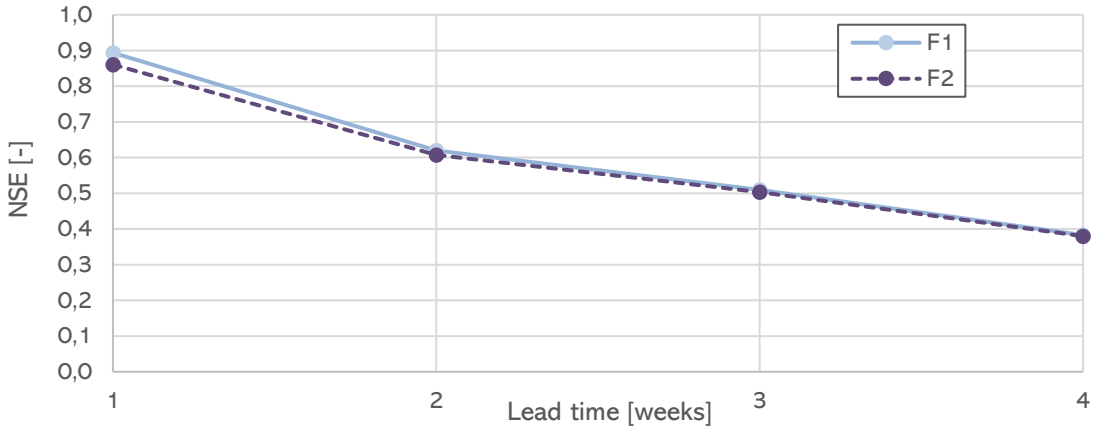

Figure 6 Performances of the best LSTM models trained to forecast weekly anomalies of minimum streamflow, using weekly anomalies of average precipitation, temperature and streamflow as inputs. F1 employs the anomalies of observed streamflow Q from all the selected stations, while F2 uses the anomaly of streamflow in the outlet.

Both experiments F1 and F2 performed almost at the same level, with F1 having slightly higher NSE values for the two initial weeks. When comparing these results with those obtained in experiment M5, i.e. the one using all input variables (weekly minimum) to forecast the weekly minimum streamflow at the outlet, the NSE values of M5 are slightly superior. This evidence shows that when enough streamflow observations from multiple locations upstream in the catchment are available, the LSTM model can reach its maximum performance. When such information is not available, however, the LSTM model can reach similar NSE values if the target variable is properly transformed (experiment F1). The combination of optimal target aggregation and input features, however, does not provide better performances, as the forecasting model has reached its maximum capabilities. Plots of experiment M5 for one-week lead-time are presented in the figures below:

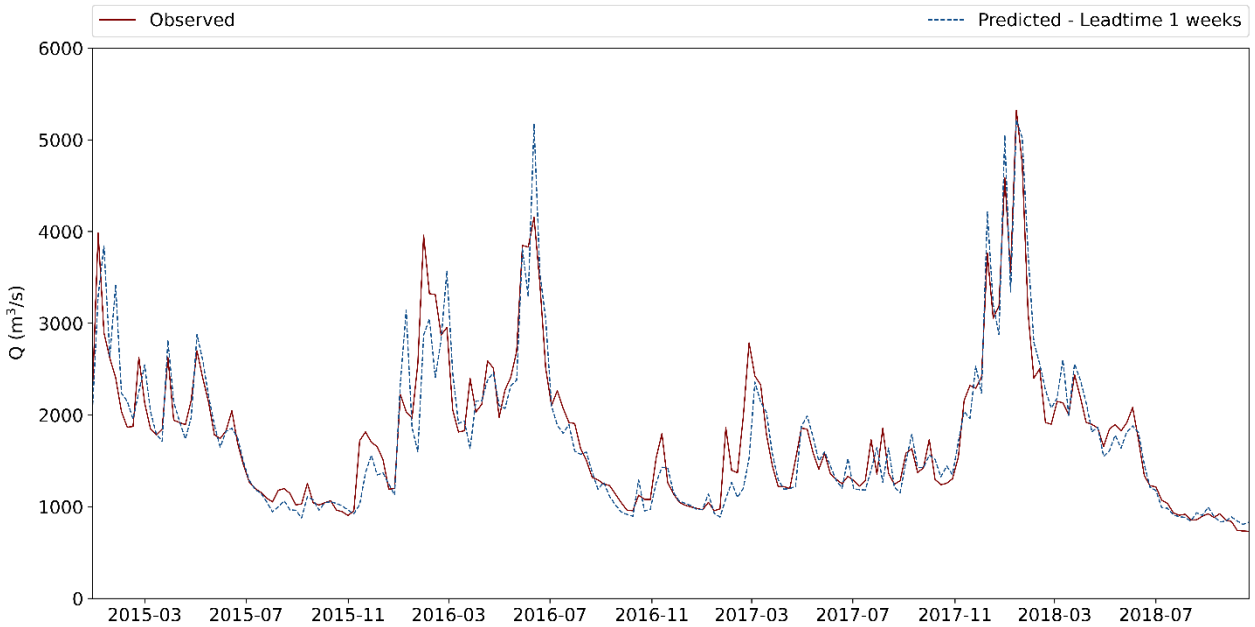

Figure 7 LSTM performances in predicting weekly minimum streamflow for one-week lead-time when using weekly minimum precipitation, temperature, and several streamflow data to predict weekly minimum streamflow at the outlet (model M5).

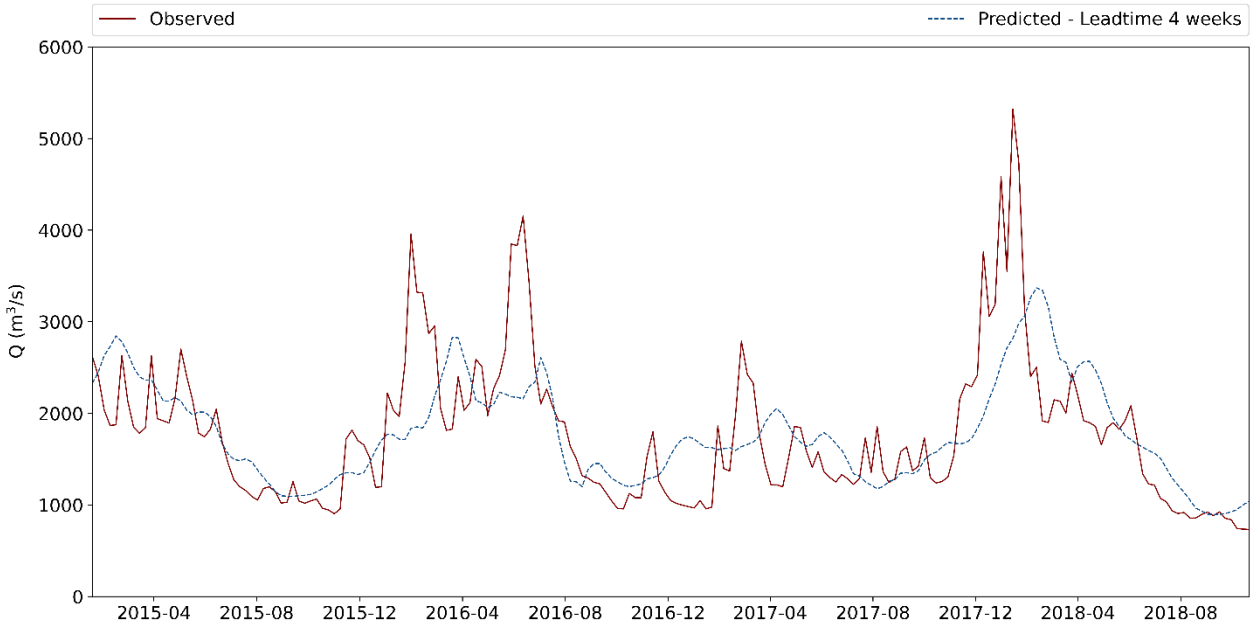

Figure 8 LSTM performances in predicting weekly minimum streamflow for four weeks lead-time when using weekly minimum precipitation, temperature, and several streamflow data to predict weekly minimum streamflow at the outlet (model M5).

As it can be seen from Figure 7, the LSTM model predicts reasonably well the overall trend of streamflow, although sometimes with a small delay. The model also seems to predict also sudden increases or decreases of the streamflow, although the predictions are slightly noisier than the observations. For four weeks lead-time (Figure 8), instead, the LSTM model can predict the overall trend, but the streamflow forecasts are smoothed, with challenges in forecasting peaks.

Streamflow forecasting with LSTM, historical data and meteorological forecasts

Multiple LSTM models were trained to forecast accumulated inflow up to four weeks ahead. Initial experiments were performed to define the best model architecture by forecasting streamflow at daily lead-times, initially using observations only as inputs. Although NSE values were reasonably high for the first day (0.8-0.9), performance deteriorated quickly afterwards and a constant delay in LSTM predictions was found, as it can be observed from Figure 9.
The LSTM model is, therefore, not able to learn from the past observations. A potential reason for that might be in the catchment size, which is very small and with concentration time around one day, which in turn means that the information from past observations, and hence the streamflow predictability, is lost after one day. For this reason, it was decided to also include forecasts in the input features. For the sake of brevity, only the results of the weekly experiments will be shown here (Figure 10).

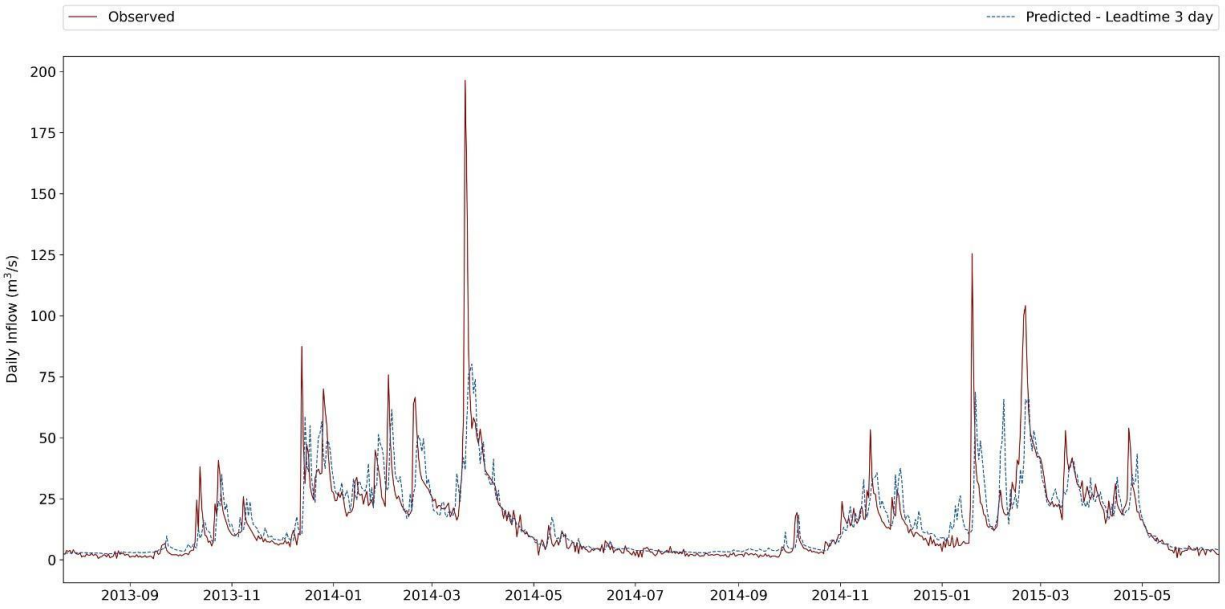

Figure 9 Daily streamflow forecasts for three days ahead using LSTM and historical information.

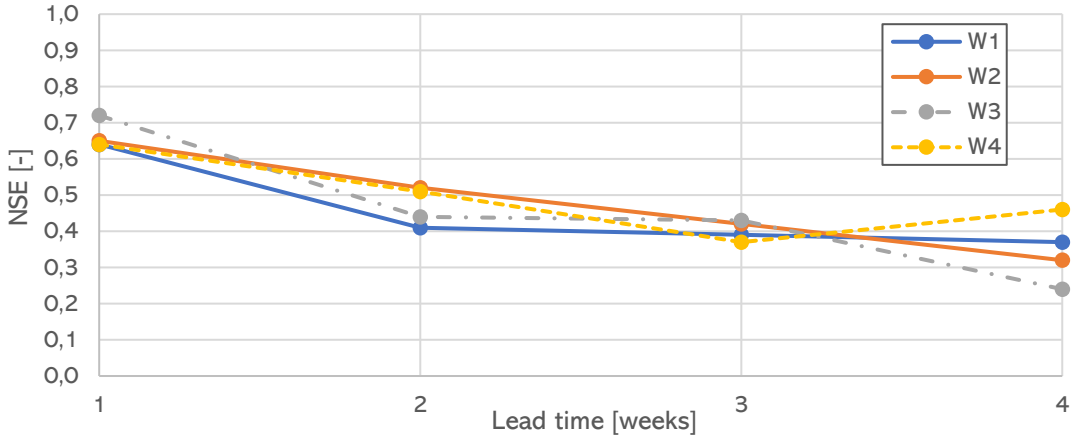

Figure 10 LSTM models performances for weekly inflow predictions from 1 to 4 weeks ahead.

As it can be observed from Figure 10 and Figure 11, pre-training the model with observations and fine tune it with forecasts (W3) provides the best performances only in the first week, while NSE values quickly deteriorates for longer lead-times and performing even worse than using only observations for week 4. This might be due to the not-so-good performances of meteorological forecasts at longer lead-times. In the first week, predictions perform satisfactorily, especially for lower inflow, and can capture the overall trend and inflow variations, although peaks are not often well reproduced.

When training the model from the beginning with both observations and forecasts, performances are very similar for the first two weeks for models W2 and W4. However, at longer lead-times, the inclusion of past inflow seems to worsen model performances (W4), while employing precipitation and temperature only, observed and forecasted, yields higher NSE values. It is also interesting to observe that the model configuration in which forecasts and observations were treated as a merged time series (W4) performed better than in the configuration in which each time step is considered as a different input (W2), as it can be seen from Figure 10 and Figure 12. A plausible reason might be that LSTM models are notoriously good to learn the temporal structure of time sequences, i.e. time series, while in W2 this structure is lost.

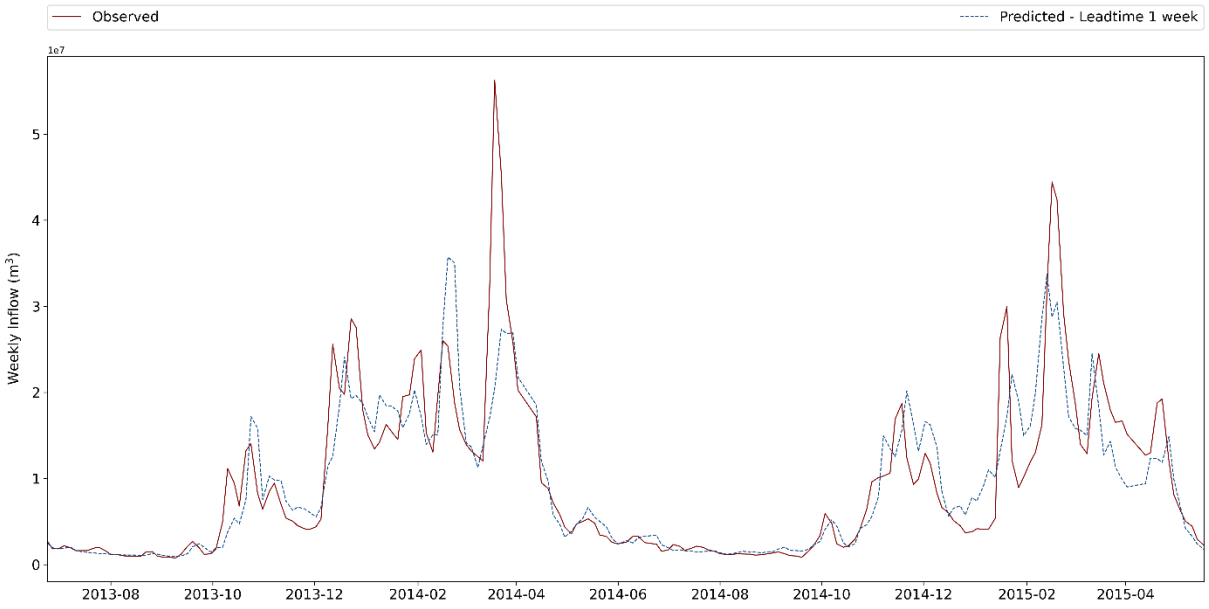

Figure 11 Weekly accumulated inflow for one week lead-time obtained pre-training the model with observations and fine-tuning it with forecasts (W3).

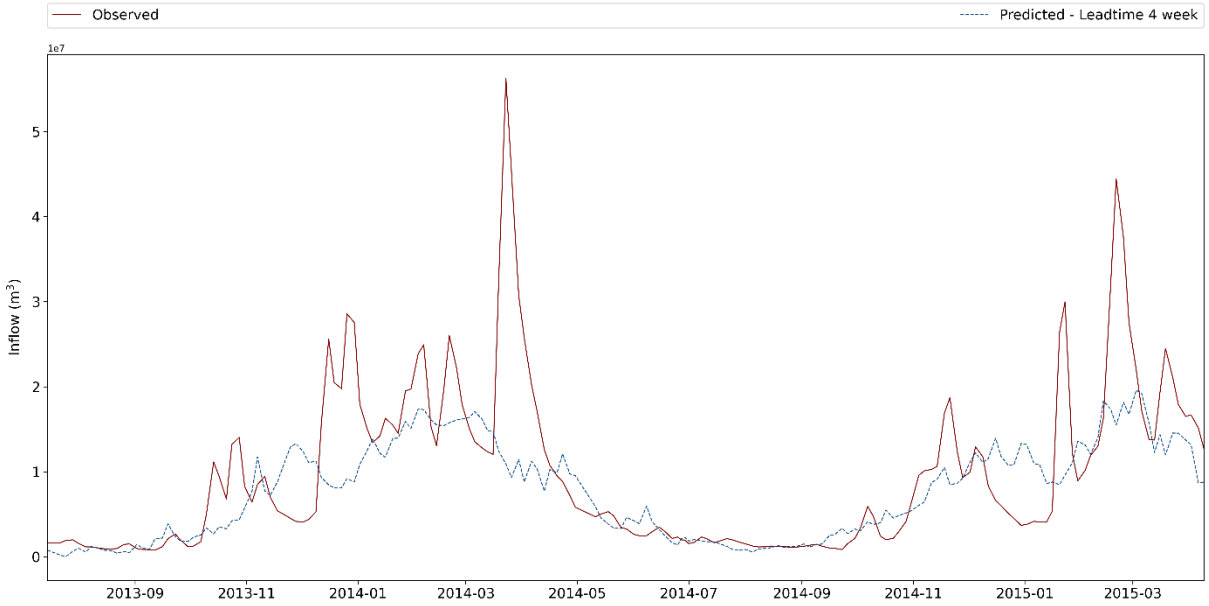

Figure 12 Weekly accumulated inflow for four weeks lead-time obtained using precipitation and temperature observations and forecasts, merged as one time series (W4).

Conclusion and next steps

The LSTM model trained only with historical data will first be benchmarked against the existing hydrological forecasts of E-HYPE provided by SMHI. A check on the model performances for low flows also will also be done. The influence of different input features, such as soil moisture, snow water equivalent and catchment attributes, on streamflow predictability will also be investigated. Finally, this framework will be extended to homogeneous catchment clusters in Europe, to improve the overall model performances, as suggested by Kratzert et al., (2018).

The approach for forecasting inflow with observations and forecasted data will also be benchmarked against the existing forecasts of E-HYPE, provided by SMHI. Additional experiments will be

performed to directly predict the one-month accumulated inflow, but also to further fine-tune the model configurations in order to fully exploit the potential of both LSTM models and meteorological forecasts, following, for instance, the work of Deng et al., (2024).

2.4 Enhanced tropical cyclone rainfall forecasting

In this section we describe the methodology and results of the experiments to enhance state-of-the-art medium-range forecasts of Tropical Cyclone (TC) rainfall using deep learning.
Accurate predictions of rainfall are essential to predict and estimate TC-related flood risk. Rainfall forecasts can be directly used as triggers of early warning/early actions (e.g., Clark et al., 2014; Yang et al., 2015) or used as input of hydrological and hydrodynamic models for predicting fluvial, pluvial, and compound flood risk (e.g., Nederhoff et al., 2024). However, TC rainfall is one of the most challenging features of a TC to simulate and predict (Zhao et al., 2022; Cheung et al., 2018). This is due to its high spatio-temporal variability, the complex underlying physical processes, and the lack of high-quality and high-resolution observations to verify and improve forecasts via data assimilation and model development (Lamers et al., 2023). A recent study by García-Franco et al. (2023) showed that state-of-the-art forecasts of TC rainfall (from the Sub-seasonal-to-Seasonal, S2S, project) are limited by large biases even in the short range and their magnitude and spatial patterns exhibit little variation with lead time. Thus, there is an urgent need to improve medium-range and sub-seasonal forecasts, to reduce biases and enhance spatial accuracy, starting from the medium range (< 5 days) which is critical for decision making and early warning. Our approach responds to this need to improve current TC rainfall predictions to support anticipatory actions, by leveraging on deep learning techniques. In particular, we aim to not only reduce forecast biases but also improve the spatial accuracy of state-of-the-art TC rainfall data. For this goal, we post-process medium-range forecasts of total precipitation produced by ECMWF up to 5 days lead times, using a deep learning architecture and different configurations (e.g., multiple alternative forecast inputs). Over a large sample of global historical TC events, we aim to learn how medium-range forecasts can be improved by leveraging on the capacity of a convolutional neural network to learn spatial patterns and semantic information from data. To validate and demonstrate the value of our approach, we assess the forecast improvements using a multi-metric evaluation method, also considering user-oriented criteria, tailored for early warning and humanitarian anticipatory action applications (see Deliverable D7.2 for more details and an application of this evaluation). Our framework relies on observations only for model training purposes and can be considered a hybrid approach, as its focus is on post-processing existing forecasts. For this reason, our forecast enhancement tool can be applied in real-time, once it has been trained over historical (re-)forecast/observations paired data, to post-process and improve existing operational products, without the need of retrieving any new observational data, that would limit its applicability in real-time.

2.4.1 Methodology

Problem setting

Our goal here is to improve operational forecasts of Tropical Cyclone (TC) rainfall in terms of general accuracy (biases) and spatial localisation of rainfall peaks, leveraging on deep learning. Our framework is based on the use of a recent variant of a state-of-the-art deep learning architecture (RA-U) and a novel loss function that we developed and tested for the first time in CLINT on ERA5 rainfall data (Ascenso et al., under review). Here, we focus on adjusting state-of-the-art medium-range forecasts of TC rainfall events using a historical record of TCs to track them and define the regions and periods of interest. Our input and target data are gridded datasets, and the problem is

formulated as a regression problem, where the objective is to minimise errors of forecasts with respect to observations. As a reference dataset, a multi-source observational dataset (MSWEP, Beck et al., 2019a) is used, given its good performance at the global scale compared to other observational and reanalysis datasets, as reported in previous studies (Beck et al., 2019b; Sharifi et al., 2019). We train the deep learning model on specific lead times up to 5 days, with the aim to develop a tool that could be used in real-time to make lead-time specific adjustments and improve future forecasts. The same framework could also be used and tested with other gridded datasets (e.g., hydrological) if both reforecasts- and observations are available over a sufficiently long period for training purposes. The final goal of this work is to build an effective deep-learning model to enhance TC rainfall forecasts, selecting the most effective loss function and verifying the forecasts in a user-oriented framework.

Deep Learning framework

As deep learning model, we used a recent variant of the popular U-Net (Ronneberger et al., 2015) architecture (see Figure 13), originally developed in the field of medical computer vision. UNet has already been shown to be effective in performing similar tasks of rainfall post-processing (e.g., Hess and Boers, 2022). Moreover, a recent study showed that it outperforms other deep learning networks for the prediction of precipitation extremes (Otero and Horton, 2023).

In the standard U-Net architecture, information is first encoded through a series of convolutional and max-pooling layers that reduce the resolution and perform an extraction of semantic information and then decoded through a series of convolutional and up-sampling layers that restore the spatial resolution to the original one as the input maps. This is done to maintain the high-level extracted semantic information and to correct the input data by learning from the differences in patterns between the input and target samples. Encoder and decoder blocks that are at the same depth in the network are linked via the so-called 'skip connections' that aid in the transfer of spatial information across the network.

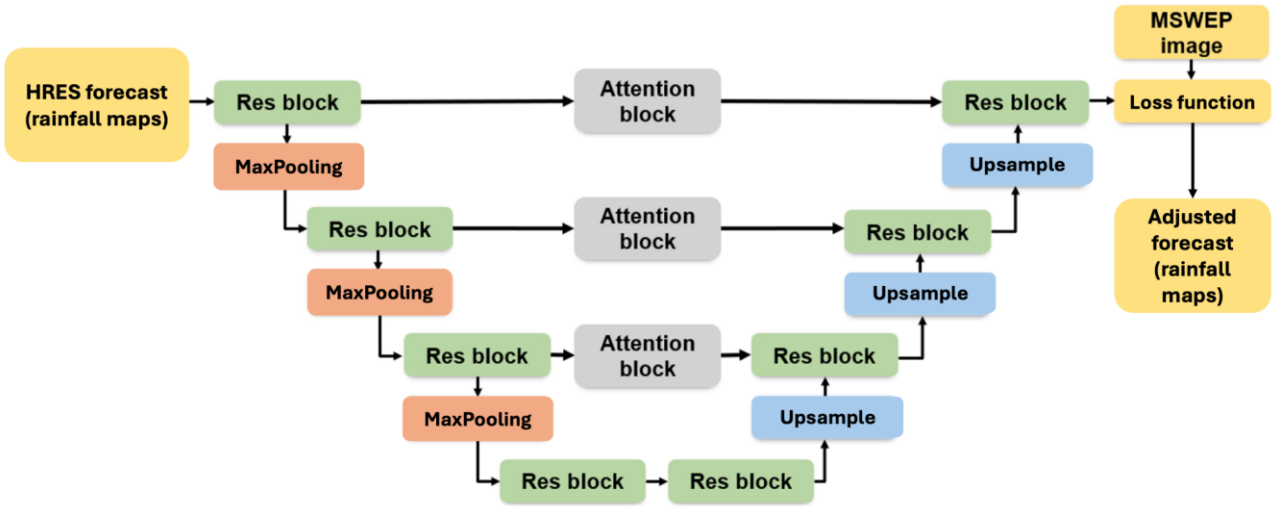

Figure 13 Flowchart of the AI-enhancement model for Tropical Cyclones rainfall based on RA-U.

We use the Residual Attention UNet (RA-U) variant, proposed by Jin et al. (2020), which shares the encoder/decoder structure and main principles of U-Net, while replacing convolutional blocks with residual blocks and integrating attention modules along skip connections (see Figure 13). Residual blocks facilitate gradient flow, mitigating the vanishing gradient issue, while attention modules augment the RA-U network's feature extraction capability, to emphasise salient features. We

decided to omit batch normalisation and dropout from the network, as we did not need these features to solve overfitting issues (see Results Section). For model training, we used the Adam optimizer with early stopping (with Adam parameters $\beta1$=0.9; $\beta2$=0.999). After some sensitivity tests, we selected a batch size of 32 (i.e., sub-sample of the training dataset used at each iteration). As training and validation strategy, we opted for a k-fold cross-validation with k=5 (fixed after preliminary tests that showed a low sensitivity of the results with varying k). Each model training and validation experiment took approximately 3h (on average, depending on the number of inputs), on an NVIDIA A100 GPU.

Loss function

A novelty of our work is, in our view, the new loss function that we developed with the objective of not only adjusting the per-pixel precipitation at each timestep of a given TC, but also to increase the accuracy of the spatial distribution of the precipitation field. Our main aim is to improve the spatial discrimination of extreme events in the forecasts, refining the localization of TC rainfall peaks and reducing errors within acceptable margins for decision making.

To achieve this, we introduce a novel loss function, called the *compound loss*, including two components: (i) the Mean Squared Error (MSE), to correct pixel-wise biases and (ii) the Fractions Skill Score (FSS, Roberts and Lean (2008)), to improve the accuracy of spatial patterns. Previous studies using deep learning for bias correction of rainfall predominantly relied on pixel-wise metrics, like the MSE, which overlook overall spatial accuracy and potentially lead to overly smoothed predictions. Recent studies have shown that such local (pixel-wise) error metrics discourage models from making predictions with sharp gradients, often leading to "blurred-out" predictions (e.g., Hess and Boers, 2022; Lagerquist and Ebert-Uphoff, 2022). By integrating the FSS into our compound function, we aim to mitigate such issues and enhance the accuracy of rainfall peak localisation, which is expected to also help in the reduction of false alarms and increase of hit rates, which are two key metrics for decision making.

The FSS is a popular spatial verification metric often used in meteorology to evaluate the resemblance between spatial patterns in two gridded datasets, typically comparing model predictions to observational data, yielding values between 0 (indicating no match) and 1 (perfect match). The FSS is computed through the following three steps: (i) the rainfall maps (prediction and observation) are converted into binary maps by applying a rainfall intensity threshold (Q) that can be either a fixed value or a percentile of rainfall intensity (calculated independently for each image); (ii) fractional coverages of threshold exceedances are computed for various neighbourhood areas; (iii) the fraction of positive pixels within patches of a specified size (N, number of grid cells along the side of a square neighbourhood) are then used to compute a skill score based on the mean squared difference of these fractions across all possible patches in the maps. Thus, the FSS measures the average overlap between N-sized patches of the two binary grids.

In our implementation, the neighbourhood size was set to 19 x 19 grid cells (N=19), following a grid search empirical testing (in the range [9, 21]). To allow the use of the FSS as a loss function for deep learning, we made some adjustments, mainly to ensure differentiability, by replacing the binary classification step based on hard threshold (Q) with an arctan function transformation. Our modified version of the FSS is referred to as FSS'. Also, we inverted the score to be used in the loss function (1-FSS'), so that the value of 0 indicates a perfect match.

Finally, our compound loss function (L_{cmpd}) consists of a weighted combination of FSS' values corresponding to different percentile thresholds (80th, 95th, and 99th percentiles) alongside the Mean Squared Error (MSE), as follows:

$$L_{cmpd} = 0.5\left(FSS'_{Q=80} + FSS'_{Q=95} + FSS'_{Q=99}\right) + 0.5\,MSE \qquad (3)$$

The weights of the loss function components (0.5, 0.5) were determined empirically through a systematic exploration of the weights space. By incorporating multiple percentile thresholds for FSS', our aim is to train the model to improve rainfall peak localisation across varying intensities, thus enhancing spatial accuracy while mitigating pixel-level biases (represented by the MSE).

Performance assessment metrics

To assess the performance of our model and the HRES benchmark, we considered multiple metrics, including standard performance measures such as MSE, MAE, absolute total rainfall bias (i.e., mean absolute difference between the total rainfall in forecast and observed maps) and spatial correlation. Additionally, we also selected and further tailored user-oriented scores to inform on the quality of forecasts (enhanced and original HRES) to inform anticipatory actions. In particular, we used custom-made action-relevant scores tailored for humanitarians and disaster management applications, i.e. a modified version of False Alarm Ratios (FAR) & Hit Rates (HR). To this end, we modified the definition of FAR and HR by including an additional parameter, i.e. the action scale, which we defined as the effective spatial scale of a warning to trigger useful actions based on the level of 'acceptable' forecast spatial error. If an observed event (threshold exceedance) occurs within a distance equal to the action scale from the forecast event, the forecast is considered a hit, otherwise a false alarm. Based on the Early Action Protocol (EAP) for TCs of the Mozambique Red Cross, we used action scale values of 100- and 50-km. These are more restrictive than the FSS patch size used in the loss function (around 210 km). Our aim is to bring the spatial accuracy to satisfy the needs stated by the Mozambique Red Cross (errors < 240 km), as described in more detail in Deliverable D7.2. As a basic target level for these scores, we considered FAR < 0.5 & HR>0.5, as for example, often adopted by Red Cross National Societies in Early Action Protocols (see Deliverable D7.1).

2.4.2 Implementation

Case study and data

In this deliverable, we present all the results at the global scale focusing on a validation sample of 2875 time steps of TC rainfall accumulated at 6-h resolution. Building on the need for improving forecast accuracy for early action for the Zambezi climate hotspot of CLINT (see Deliverable D7.1), an analysis of the performance of our RA-U model was extracted for four TCs affecting the Zambezi River basin and is reported in Deliverable D7.2. The case study results are similar to those presented at the global scale in this deliverable (we refer the reader to D7.2 for more details).

As baseline forecast to enhance, we use the High-Resolution (HRES) medium-range forecasts produced by the ECMWF Integrated Forecasting System (IFS, see https://www.ecmwf.int/en/publications/ifs-documentation for a detailed description). The HRES forecasts are widely considered the top-performing global deterministic operational forecasting system and are often chosen as a benchmark in machine learning–based weather prediction studies (e.g., Lam et al., 2023; Rasp et al., 2020). HRES forecasts are produced operationally twice daily (at 00 and 06 UTC) with a maximum lead time of 10 days. Since March 2016 (IFS Cycle 41r2), HRES forecasts are run at the horizontal grid resolution of approximately 9 km x 9km (about 0.08 degrees) and stored at a temporal resolution between 1h to 6h, varying with lead time (1h up to 90h lead time, 3h up to 144h lead time, and 6h onwards). Being provided at higher resolution than other global forecasting models, HRES is the most accurate single-run realisation of broadscale weather patterns. For several years, its use enabled more detailed analysis of precipitation patterns than the ECMWF's ensemble forecasts (ENS). In June 2023, the ENS resolution was also upgraded to the same 9-km resolution as HRES, but a 9-km ENS reforecast- is still under production and no multi-year ensemble reforecast- run at 9 km is available yet. Using a short reforecast- was not a viable solution,

given our need for a multi-year-long data record of forecasts (overlapping with observations) for training and validating our machine learning model. Thus, the use of operational HRES forecasts (since 2016) at the consistent current high resolution was considered the best choice. After matching HRES with observations, we selected a period of analysis of 4 years (2016-2019), in common with the observational data available (see below). In terms of forecast horizons, we decided to focus on lead times up to 5 days, given the current levels of predictability and TC track errors that limit the use of forecasts for decision-making over longer time scales than a few days. For example, a 3-day lead time is used so far in the Red Cross' Early Action Protocol for TCs in Mozambique. We selected a 6-h target resolution, using a common resolution across the lead times considered to make the comparison of scores consistent and to avoid increasing the correlation of the dataset used for ML training using higher temporal resolutions (for time steps of a same TC event). Similarly, as HRES forecasts are issued at 00 and 12 UTC, but forecast maps issued 12 hours apart are expected to be highly correlated, only one daily forecast issue step was considered (00 UTC).

To help the deep learning model correct the biases of TC rainfall forecasts and improve spatial accuracy, we considered five different candidate inputs from HRES in addition to total precipitation (see Table 9): (i) total column of water, (ii) temperature at 850 hPa, (iii) total cloud cover, (iv) relative humidity at 850 hPa, and (v) mean sea level pressure. Their choice was based on first model development experiments on ERA5 (Ascenso et al., *under review*) and on previous studies (e.g., Sha et al., 2020; Ling et al., 2022).

Table 9 Parameters extracted from HRES forecasts and considered as inputs of the deep learning model, with their original units and validity (for more details, see parameter database at: https://codes.ecmwf.int/grib/param-db/). All data was downloaded via ECMWF's MARS API.

Acronym	Full name	Validity [units]
TP	Total Precipitation	Accumulated [m]
TCW	Total Column of Water	Instantaneous [kg m^{-2}]
TCC	Total Cloud Cover	Instantaneous [-]
T (850hPa)	Temperature at 850hPa pressure level	Instantaneous [K]
RH (850hPa)	Relative Humidity at 850hPa pressure level	Instantaneous [%]
MSLP	Mean Sea Level Pressure	Instantaneous [Pa]

As reference for the forecast skill assessment and enhancement target, we used the Multi-Source Weighted Ensemble Precipitation (MSWEP) dataset (Beck et al., 2019a), which was downloaded from the Glo-H2O repository (https://www.gloh2o.org/mswep/). MSWEP provides global observational precipitation data derived from multiple sources, including ground-based observations, satellites, and reanalysis products, at a spatial resolution of 0.1 degrees (approximately 10 km) and a temporal resolution of 3 hours. The multi-source data integration of

MSWEP enhanced its performance and robustness with respect to other single-source datasets (Beck et al., 2019b) and showed the highest accuracy in multi-datasets comparative studies (e.g., Sharifi et al., 2019), making it a suitable reference for validating the accuracy of AI-enhanced rainfall forecasts. To define the domains for rainfall forecast evaluation and post-processing, we located Tropical Cyclone (TC) centres using the International Best Track Archive for Climate Stewardship (IBTrACS) best-track data (version v04r003), which offers a global TC dataset at 3-hourly temporal resolution (Knapp et al., 2010).

Data pre-processing

As IBTrACS reports instantaneous TC data every 3 hours starting at 00:00 UTC, considering the target 6-h resolution of the two other products (HRES and MSWEP) and the forecast issue time (00 UTC), we sub-sampled the IBTrACS data to get instantaneous data aligned with the centre of the MSWEP and HRES accumulation window considered (e.g., TC data at 03:00 UTC were used to match rainfall maps accumulated between 00:00 and 06:00 UTC). Subsequently, for each time step in IBTrACS, we cropped the HRES and MSWEP fields surrounding a 14-degree-side (i.e., about 1550 km) box centred on the TC location both temporally and spatially; this approach (and box size) allows us to encompass an area large enough to include all the grid cells with TC rainfall at each time step, as this distance (>1500 km) is larger than extreme TC sizes. This approach resulted in squared domains of dimension 141x141-grid-cellss, with one channel for MSWEP and more channels for HRES, i.e. one for each selected input variable (e.g., total precipitation, total column of water, and temperature at 850 hPa). As MSWEP is a 3-hourly aggregated product and HRES rainfall forecasts at close hourly time steps are highly correlated, we performed a temporal aggregation at a common temporal window of 6 hours for both products, i.e., to get 6-hourly accumulated values for HRES total precipitation and 6-hourly average values for HRES temperature and other instantaneous variables tested (like relative humidity; see Table 9).

In summary, to prepare the inputs for the deep learning model from the HRES forecast data (precipitation and other input variables), we followed these steps:

i. we downloaded HRES data at the global scale using ECMWF's Meteorological Archival and Retrieval System (MARS) API;
ii. 6-h step values were obtained from the original cumulated ones (e.g., de-accumulating values of two adjacent steps for rainfall or averaging close-step values for other instantaneous variables; see Table 9);
iii. a spatial regridding was performed from the reduced gaussian grid system of IFS (octahedral from 2016) to the regular 0.1° x 0.1° lon-lat grid resolution of MSWEP, applying the conservative interpolation method (using ECMWF's MetView Python library);
iv. the regridded data is then cropped over the regions of interest (14-degree-side boxes), defined based on the TC locations retrieved from IBTrACS;
v. also, the observed gridded data (at 0.1° x 0.1° resolution), MSWEP, is cropped over the same regions of interest.

Preliminary work and developments on ERA5

Preliminary developments of the model were performed on ERA5 precipitation reanalysis, to test and compare the performance of the network with different loss functions using a reference target data that is expected to be closer to observations and available over a longer period than HRES. This work was carried out at the global scale (Ascenso et al., under review) and the key findings are briefly summarised here. This foundational work on ERA5 showed that the RA-U model with our new compound loss function improves ERA5 over all metrics considered (see previous Section) by 3-28%. This improvement is statistically significant and larger than the one achievable with an RA-U model using a standard pixel-wise bias-based loss function (MSE) or another recent loss function

proposed in the literature (Hess and Boers, 2022), both of which worsen the total rainfall bias of ERA5. While improving on the total rainfall bias, our network significantly improves spatial accuracy.

Further developments on HRES

Building on the preliminary work on ERA5, after testing the model on HRES using the same configuration (Ascenso et al., under review), we re-trained the model with some alternative settings (e.g., with data augmentation) and performed more sensitivity analysis tests on the parameters of the loss function and of the FSS component (N and Q values). These tests led to the following main changes in the implementation for HRES:

- We selected new weights of the components of the loss function, selecting slightly different values for HRES with respect to ERA5.
- We included a k-fold cross-validation procedure, which was deemed necessary in our case due to the smaller sample size with respect to the ERA5 work (where a static training/validation/test split was adopted); including a k-fold validation proved successful to reduce overfitting and improve performance in validation (see Results).
- We added a data augmentation step during training (with randomly selected transformations of the paired input/target images at each iteration), to increase the robustness of the training, particularly to changes in the position of the TC in the image; this proved successful, as no significant change in the performance was observed adding the data augmentation step.
- After testing 20 possible multi-input combinations using HRES data fields (as additional model inputs to total precipitation), we reduced the number of additional inputs to two (from three used for ERA5); adding more inputs was not beneficial here, at the expense of increased computation, while two inputs were necessary to reduce overfitting (see Results).

2.4.3 Results

Results in Figure 14Figure 16 report the performance at 5-day lead time for a large sample of the multi-input configurations tested, for which no overfitting was detected (except for MSLP) after inspecting the learning curves (e.g., see Figure 17). RA-U successfully improves the original forecasts with respect to the different accuracy metrics, with all the combinations of inputs considered. Already by using only total precipitation (TP) as input, the performance is substantially improved with respect to HRES. Further minor improvements can be obtained in terms of the scores by adding one or two inputs. In fact, model performances are very similar across different input combinations, with most of the two-input combinations using TCW, TCC, T, and RH (see Table 9) being the top-performing ones. MSL is the only input bringing smaller performance improvements (and more overfitting issues) than others, with a clear reduction in correlation (Figure 16). This indicates that this larger-scale variable is less useful than the other local-scale atmospheric variables considered, as it can be expected.

A significant improvement in all scores moving from HRES to RA-U enhanced forecasts is found by the Wilcoxon signed rank test that detects statistically significant differences in mean ranks (Figure 14-Figure 16). On the other hand, only a few remarkable (and significant) changes across distributions for different three-input configurations are found. Including more inputs than three does not help or would keep the performance at the same level at the expense of increased computation.

For further analysis, here we select the configuration using TCW and T as additional inputs to total precipitation, as this solution appears systematically in the top categories of significantly superior options tested and shows some of the best learning curves, achieving lower validation loss values (see Figure 17).

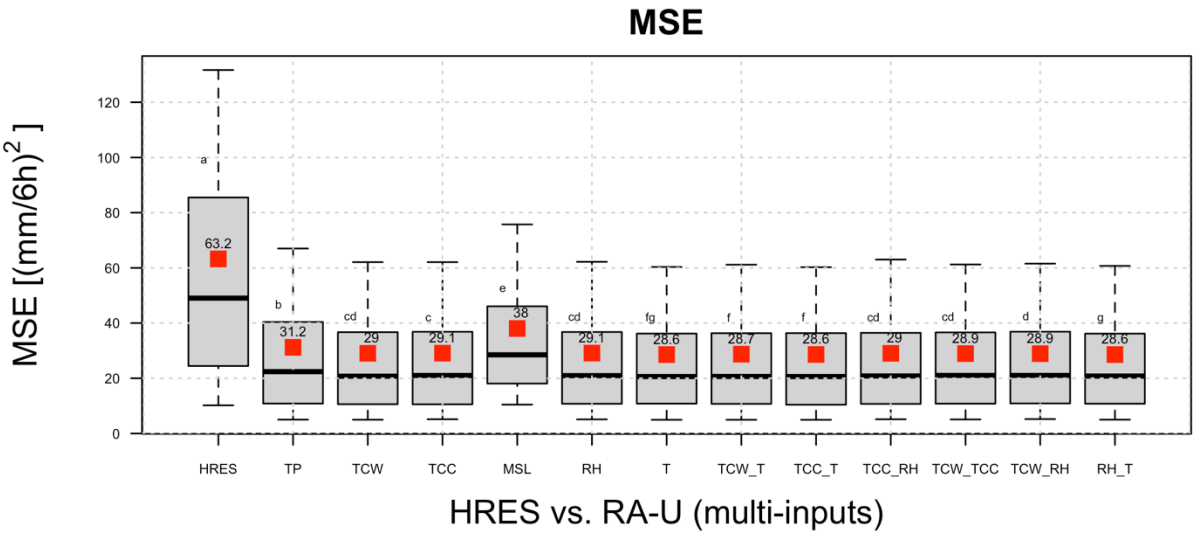

Figure 14 Distribution of the MSE score for original and enhanced forecasts (using different inputs) over the test set (2875 samples) in cross-validation (k-fold=5) at 5-day lead time. While the first boxplot (HRES) represents the original forecast, all the others correspond to RA-U with different inputs, starting from only total precipitation (TP), and then adding one or more inputs (see Table 9 for all acronyms,). The letters above each boxplot indicate the significant differences detected by the Wilcoxon test at significance level 0.05 (distributions with the same letters are not significantly different).

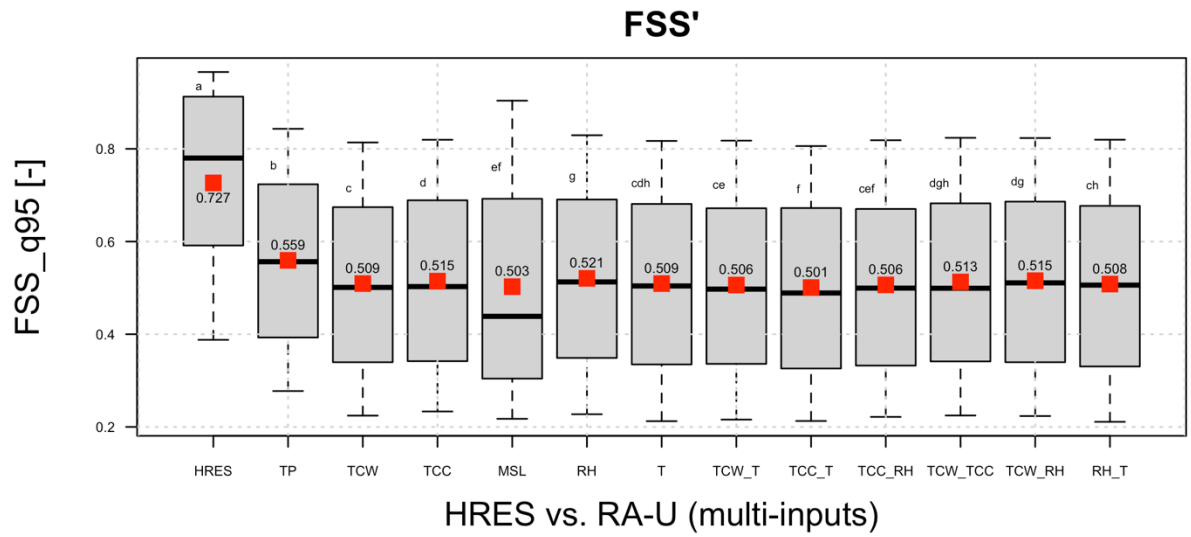

Figure 15 Distribution of the FSS' score (with Q=95th percentile threshold and N=19; ideal score: 0) for original and enhanced forecasts over the test set (2875 samples) in cross-validation (k-fold=5) at 5-day lead time. While the first boxplot (HRES) represents the original forecast, all the others correspond to RA-U with different inputs, starting from only total precipitation (TP), and then adding one or more inputs (see Table 9 for all acronyms). The letters above each boxplot indicate the significant differences detected by the Wilcoxon test at significance level 0.05 (distributions with the same letters are not significantly different).

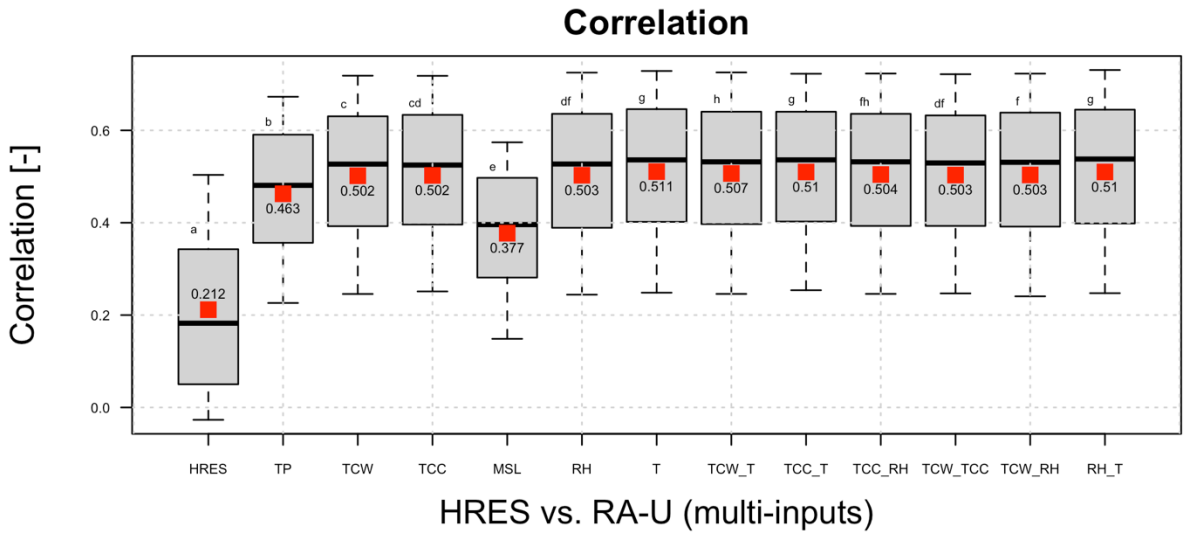

Figure 16 Distribution of the spatial correlation between forecasts (original and enhanced) and observations over the test set (2875 samples) in cross-validation (k-fold=5) at 5-day lead time. While the first boxplot (HRES) represents the original forecast, all the others correspond to RA-U with different inputs, starting from only total precipitation (TP), and then adding one or more inputs (see Table 9 for all acronyms). The letters above each boxplot indicate the significant differences detected by the Wilcoxon test at significance level 0.05 (distributions with the same letters are not significantly different).

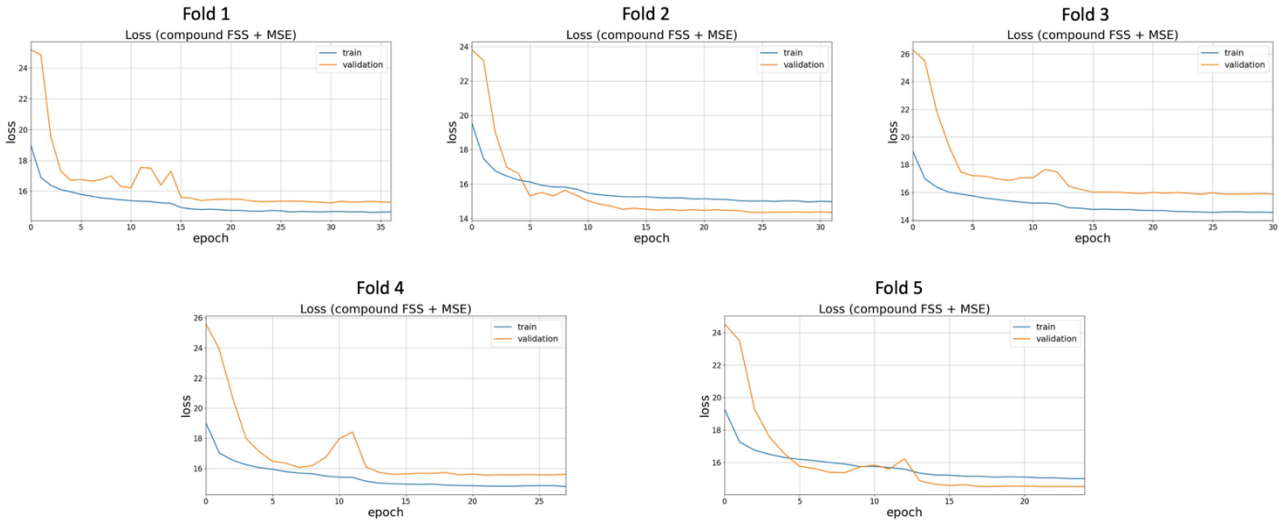

Figure 17 Learning curves in training (blue) and validation (orange) reporting the evolution of the loss with the number of iterations (epochs) for our RA-U model with the selected 3 inputs (TP + TCW + T) over the 5 cross-validation folds.

For the RA-U model selected, Table 10 summarises the performance across lead times in terms of general attributes of forecast quality (mainly focusing on biases and spatial accuracy and correlation).

These results indicate that our enhanced forecasts improve with respect to the HRES forecast benchmark across all lead times and for all metrics, especially in terms of local biases (MSE and MAE) and spatial correlation and accuracy (FSS). In terms of user-oriented performance, Figure 18 shows how False Alarm Ratios (in our definition tailored for decision making) are largely reduced, especially

at 3- and 5-day lead time and for the highest rainfall thresholds, where there is more room for improvement in HRES. Given the low acceptability of false alarms for early warning systems, the adoption of our post-processing model is critical in making TC rainfall forecasts acceptable to support early warning and early actions (see more results and discussion on this application in Deliverable D7.2).

Table 10 Performance of the original HRES vs enhanced forecasts (RAU with the selected 3 inputs, i.e., TP, TCW and T) for different metrics, i.e. MSE, MAE, absolute total delta bias (mean absolute difference between the total rainfall in forecast and observed maps), spatial correlation, FSS' for 95-th percentile threshold. The ideal scores are: 0 for MSE, MAE, absolute total bias, FSS' (being inverted with respect to the original formulation) and 1 for correlation.

	MSE [(mm/6h)^2]		MAE [mm/6h]		Absolute Total Delta Bias [mm/6h]		Spatial correlation [-]		FSS' (q95) [-]	
Lead time [d]	HRES	RAU	HRES	RAU	HRES	RAU	HRES	RAU	HRES	RAU
1	35.3	17.4	2.1	1.7	8583	8286	0.58	0.71	0.33	0.33
3	53.6	24.9	2.8	2.2	12207	12370	0.36	0.58	0.56	0.46
5	63.2	28.7	3.2	2.4	16410	15981	0.21	0.51	0.73	0.51

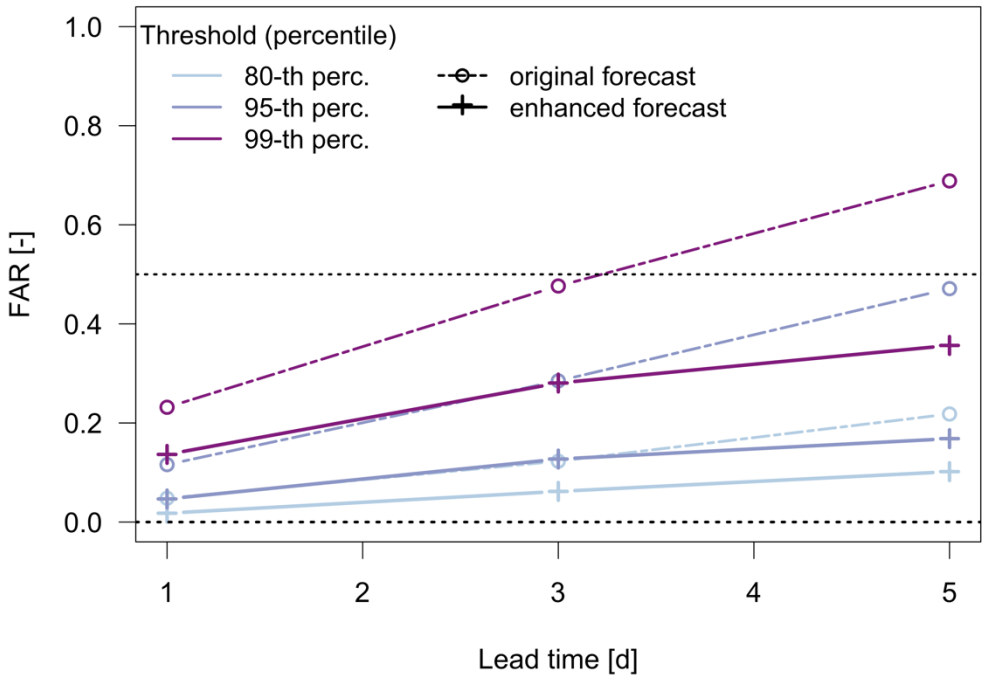

Figure 18 False Alarm Ratios (FAR) of original HRES vs AI-enhanced TC rainfall forecasts (RAU with the selected 3 inputs, i.e., TP, TCW and T) at lead times from 1 to 5 days over the model test set (average values over 2875 time-steps from 5-fold cross-validation sets) at the global scale.

Conclusions and next steps

This work focused on enhancing the operational ECMWF's High-Resolution forecasts (HRES) of TC rainfall using deep learning. We show a substantial improvement of the forecasts across different attributes of forecast quality, reducing local biases of rainfall and increasing spatial accuracy with respect to observations. The results presented in this deliverable at the global scale suggest a clear potential of our enhancement tool for reducing false alarms and increasing the value of forecasts of early action, while offering a longer window of opportunity, extending the actionable lead times, as also confirmed by our more user-focused analysis on the Zambezi River Basin (Deliverable D7.2).

Recently (in Summer 2023), the ensemble (ENS) operational forecasts at ECMWF moved to the same resolution as HRES, but ENS reforecasts- are not available over a sufficiently long period yet, so we could not consider training our model over the ensemble forecasts (or its control member). However, future work could look at validating the benefits of our model trained on HRES and applied in test mode over the ensemble at high-resolution for a few events or for a longer period, as soon as more ENS reforecast- data will become available. Given that the ECMWF ensemble members are inter-exchangeable and produced by the same model (IFS) as HRES, re-training a deep learning model for each ensemble member is not considered a good use of resources. On the other hand, the RA-U model we presented in this deliverable could be directly deployed in test mode over all members of the ensemble (after training on HRES, as done here, or on the control member of ENS). Thus, this can be considered a final step of the work that could give the opportunity to assess and exploit enhanced ensemble forecasts of TC precipitation to better support decision making.

2.5 Deep learning-based approaches for tropical cyclone activity detection and forecasting

This section describes different deep learning-based approaches for the application of tropical cyclone (TC) activity. In particular, different time lags between input variables (features) and target variables are analysed, ranging from detection (i.e. 0 lag days) to forecasting (i.e. 1-13 days). An individual model for each lag day is trained and tested, in order to understand the prediction skills of fully data-driven approaches in exploiting past features to predict future activity of TCs. The starting point is to consider no time lag between features and target, to check if various ML model architectures are able to extract meaningful information on the actual occurrence of TCs from a set of features provided. Then, increasing the time lag between 1 and 13 days tests the ability of these ML models, trained at lag 0, to exploit the features for predicting the target further ahead in time. As will be described in the next subsections, when used for forecasting (i.e. having time lags greater 0), the best-performing neural-network-based approach considered is not able to outperform ECMWF's ensemble forecast system, while it achieves compatible skill for detection (lag 0). Therefore, to enhance the forecast skill of the model, a hybrid approach is the natural conclusion of this analysis, combining dynamical ensemble forecasts of the input features with the deep learning-based model for lag 0.

Considering a set of potentially informative features, this work compares the performances of different ML model types/architectures, spanning from basic and advanced tabular approaches to modern convolutional neural networks-based architectures. Then, the addition and removal of various features with different characteristics is tested, to understand the eventual improvement of the forecast provided by autoregression, climate indices, time and location indicators, or additional features related to the climatological probability of TC occurrence. Additionally, multiple areas are considered, to test the ability of the models to generalise over different ocean basins, which would allow to eventually deploy a trained model able to produce robust forecasts without having to focus on a specific region only.

From an ML perspective, the challenges and contributions of this work are:

- Test different architectures and models on a challenging real-world binary classification problem with spatio-temporal input data and a target variable, where the interest is not only on the predicted label but also on the specific value assigned to its probability.
- Find a suitable low-dimensional representation of a high-dimensional problem where each variable at each location is considered, with a number of features comparable with the number of samples.
- Address a highly imbalanced classification task, with the occurrence of TCs being very rare (more than 99% of samples have class 0, i.e. no TC).
- Manage the trade-off between the dimension of the hypothesis space and the number of parameters to tune, since the number of observational data available does not allow to train complex models with several hyperparameters.
- Balance between the exploitation of the invariant information shared between the underlying phenomenon (i.e., TC occurrence) and the features and the bias introduced by the local characteristics of different regions.
- Provide an exhaustive evaluation pipeline that not only includes classical evaluation metrics (e.g., accuracy) but that assesses the model's skills and reliability in terms of probabilities.

2.5.1 Methodology

Problem setting

This work focuses on predicting the probability of TC activity in a region of interest, therefore considering spatio-temporally distributed input features and target variable, in the form of gridded data with daily values. The main purpose of this work resides in daily predictions of the probability of TC activity. For this reason, the problem is formulated as a classification problem with binary targets, where the evaluation metrics will also take into account the calibration of the predicted probabilities. In particular, for each day and gridpoint, the target value in a specific location is equal to 1 if there is a TC occurring within the following two days and a radius of 300 km, while the maximum wind speed needs to reach tropical storm strength (>=17m/s).

Baselines

Firstly, classical ML algorithms for tabular data have been considered. They serve as a first inspection of the relationships between features and target variable, and they constitute valid ML-based baselines w.r.t. the more advanced approaches that will be described in the next paragraph, which consider the spatial distribution and the temporal sequentiality of the data.

In our setting, one model has been trained for each time lag of one day between 0 and 13, considering as features the values of the identified candidate drivers lagged by the selected number of days. The main assumption that resides behind these tabular approaches is that a set of *N* samples, drawn from the same joint distribution of features and target, are available to train, validate and test the models. Therefore, in our specific procedure, this assumption resides in considering that at each grid point of the region and on each day of the year, the underlying multivariate distribution of the features and the target does not change. Although this may be restrictive, it allows training a single model with all the data available for the region under analysis. Indeed, although no spatial and temporal information is exploited by these baselines, the main advantage of these models is to consider a large number of samples (number of days x number of grid points), since data at each location are indeed individual input-output pairs, and a reduced number of features (only the ones at a specific location), reducing the risk of overfitting. On the contrary, the advanced approaches that will be described in the next paragraph consider all data at all locations as a collective single input-output pair, which results in a significant reduction in the number of samples (one for each day), but with a possibly better identification of the spatial relationships between data. Specifically, *logistic regression* (Kleinbaum, Dietz, Gail, Klein, & Klein,

2002), *AdaBoost* (Freund & Schapire, 1997), *and extremely randomized trees* (Geurts, Ernst, & Wehenkel, 2006) have been applied. Also *Feedforward Neural Networks* (FFNN) (Schmidhuber, 2015) architectures have been considered. These approaches focus on different aspects, and they represent a set of baselines with different characteristics. Logistic regression has the advantage of being easy to interpret but the disadvantage of linearity assumptions. Adaboost and extremely randomized trees are ensemble models that produce a strong predictor by combining weak predictors, designed for boosting and bagging, respectively (i.e., respectively aimed to reduce the bias and the variance). Finally, FFNN are modern approaches for tabular data that combine consecutive layers of atomic non-linear transformations to reproduce complex non-linear functions. In terms of hyperparameters, a validation set has been exploited for the selection of the type and magnitude of regularisations for the logistic regression, number of trees and propensity to perform splits for the extremely randomised trees, number of estimators and learning rate for the AdaBoost. Regarding the FFNN, the validation led to the selection of activation function, dropout, number of nodes and layers.

Additionally, ML techniques designed to deal with imbalanced classification tasks have been explored. Indeed, given the rare occurrence of TCs, the task is highly imbalanced (as an example, the accuracy of a trivial model that always predicts probabilities equal to 0 is equal to 0.998). Therefore, two variants have been considered: oversampling and undersampling techniques (Ganganwar, 2012) and sample weighting (Chawla, 2010). However, these balancing techniques lead to a highly miscalibrated model, since the training set distribution becomes balanced between samples of class 0 and 1, while the real underlying distribution is highly unbalanced. For this reason, since the main interest of this task is to forecast probabilities and not only to enhance the accuracy of the identification of the binary classes, models trained on the original data are much better calibrated and hence have been preferred.

Summarising, logistic regression, AdaBoost, extremely randomised trees, and FFNN, have been trained, validated, and tested considering meteorological input features. Then, FFNN, which are the best performing and more advanced ML-based baselines, have also been tested with different configurations of inputs, as will be described in the next subsections, where the applicative analyses and results will be overviewed.

Advanced methods

Starting from the results associated with the methods designed for classical tabular data discussed in the previous paragraph, this paragraph describes the ML algorithms considered to try to exploit spatial correlation patterns and temporal dependencies of the feature data. In the first case, convolutional approaches have been considered, while for the second one recurrent neural network have been tried.

Firstly, Convolutional Neural Networks (CNN, (LeCun, Bottou, Bengio, & Haffner, 1998)) have been explored. They are deep learning approaches designed to deal with images, exploiting the spatial dependencies of pixels. Given the spatial distributions of the meteorological fields considered, these techniques are particularly suited to exploit spatial patterns and extract meaningful information. To adapt the data available to encode them as images, each feature has been considered as a channel of an image with a number of pixels equal to the product of latitudes and longitudes considered, and the target has been encoded as a black and white image, where each pixel assumes a value between 0 and 1 representing the probability of TC activity in the considered region.

Depending on the different configurations of inputs, we firstly considered images of the same shape and related to the same location of the target. Then, we considered input images of refined resolution (when using 1x1 degrees instead of 2.5x2.5 degrees), global images, or different numbers of channels (when adding or removing features from the set of inputs provided). On the other hand, the target variable can only be equal to 0 or 1 in observational data, where we know if a TC occurred

or not, while the predicted target at each grid point is a real number between 0 and 1, representing the probability of TC activity.
The CNN architecture designed for this forecasting problem follows the structure of autoencoders (Baldi, 2012), with the difference that the output is not designed to reconstruct the input image but the target one. Therefore, an encoder structure compresses meaningful features into a latent space and a subsequent decoder part reconstructs an image from the latent space, minimising the reconstruction error w.r.t. the target image. Highly complex models with many parameters may be susceptible to overfitting. For this reason, the considered architectures have been designed to keep the number of parameters of the same order of magnitude as the number of samples, such that they remain comparable w.r.t. the number of training samples. As an example, Figure 19 shows an architecture with 1313 parameters that follows the principle described.

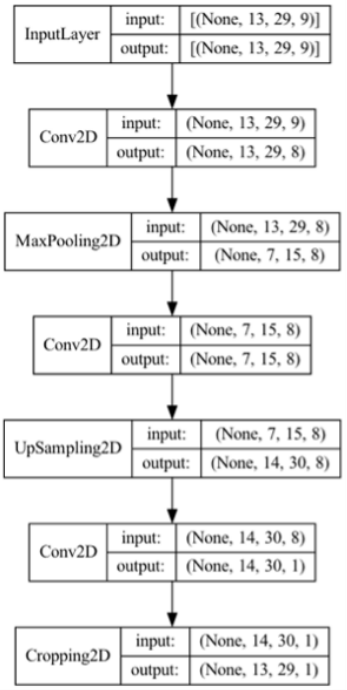

Figure 19 CNN architecture with one downscaling and one upscaling layer. This model has been used in the basic configuration with nine input features, and adapted depending on the different configurations of inputs, to evaluate the CNN-based approach.

Then, we consider a more recent state-of-the-art approach, U-Net (Ronneberger, Fischer, & Brox, 2015), that is based on convolutional layers and has specifically been designed for image segmentation. This way, we compare the relatively simple structure of an autoencoder-based CNN with another, more complex, state-of-the-art CNN-based architecture that is suitable for the current problem setting. Indeed, we need to identify important groups of grid points that identify a specific element, i.e., the TC, similarly to what is usually done for pixels in image segmentation. Figure 20 shows the architecture of the U-Net implemented in this work.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 16, 32, 9)]	0	[]
conv2d (Conv2D)	(None, 16, 32, 16)	1312	['input_1[0][0]']
dropout (Dropout)	(None, 16, 32, 16)	0	['conv2d[0][0]']
conv2d_1 (Conv2D)	(None, 16, 32, 16)	2320	['dropout[0][0]']
max_pooling2d (MaxPooling2D)	(None, 8, 16, 16)	0	['conv2d_1[0][0]']
conv2d_2 (Conv2D)	(None, 8, 16, 32)	4640	['max_pooling2d[0][0]']
dropout_1 (Dropout)	(None, 8, 16, 32)	0	['conv2d_2[0][0]']
conv2d_3 (Conv2D)	(None, 8, 16, 32)	9248	['dropout_1[0][0]']
max_pooling2d_1 (MaxPooling2D)	(None, 4, 8, 32)	0	['conv2d_3[0][0]']
conv2d_4 (Conv2D)	(None, 4, 8, 64)	18496	['max_pooling2d_1[0][0]']
dropout_2 (Dropout)	(None, 4, 8, 64)	0	['conv2d_4[0][0]']
conv2d_5 (Conv2D)	(None, 4, 8, 64)	36928	['dropout_2[0][0]']
max_pooling2d_2 (MaxPooling2D)	(None, 2, 4, 64)	0	['conv2d_5[0][0]']
conv2d_6 (Conv2D)	(None, 2, 4, 128)	73856	['max_pooling2d_2[0][0]']
dropout_3 (Dropout)	(None, 2, 4, 128)	0	['conv2d_6[0][0]']
conv2d_7 (Conv2D)	(None, 2, 4, 128)	147584	['dropout_3[0][0]']
max_pooling2d_3 (MaxPooling2D)	(None, 1, 2, 128)	0	['conv2d_7[0][0]']
conv2d_8 (Conv2D)	(None, 1, 2, 256)	295168	['max_pooling2d_3[0][0]']
dropout_4 (Dropout)	(None, 1, 2, 256)	0	['conv2d_8[0][0]']
conv2d_9 (Conv2D)	(None, 1, 2, 256)	590080	['dropout_4[0][0]']
conv2d_transpose (Conv2DTranspose)	(None, 2, 4, 128)	131200	['conv2d_9[0][0]']
concatenate (Concatenate)	(None, 2, 4, 256)	0	['conv2d_transpose[0][0]', 'conv2d_7[0][0]']
conv2d_10 (Conv2D)	(None, 2, 4, 128)	295040	['concatenate[0][0]']
dropout_5 (Dropout)	(None, 2, 4, 128)	0	['conv2d_10[0][0]']
conv2d_11 (Conv2D)	(None, 2, 4, 128)	147584	['dropout_5[0][0]']
conv2d_transpose_1 (Conv2DTranspose)	(None, 4, 8, 64)	32832	['conv2d_11[0][0]']
concatenate_1 (Concatenate)	(None, 4, 8, 128)	0	['conv2d_transpose_1[0][0]', 'conv2d_5[0][0]']
conv2d_12 (Conv2D)	(None, 4, 8, 64)	73792	['concatenate_1[0][0]']
dropout_6 (Dropout)	(None, 4, 8, 64)	0	['conv2d_12[0][0]']
conv2d_13 (Conv2D)	(None, 4, 8, 64)	36928	['dropout_6[0][0]']
conv2d_transpose_2 (Conv2DTranspose)	(None, 8, 16, 32)	8224	['conv2d_13[0][0]']
concatenate_2 (Concatenate)	(None, 8, 16, 64)	0	['conv2d_transpose_2[0][0]', 'conv2d_3[0][0]']
conv2d_14 (Conv2D)	(None, 8, 16, 32)	18464	['concatenate_2[0][0]']
dropout_7 (Dropout)	(None, 8, 16, 32)	0	['conv2d_14[0][0]']
conv2d_15 (Conv2D)	(None, 8, 16, 32)	9248	['dropout_7[0][0]']
conv2d_transpose_3 (Conv2DTranspose)	(None, 16, 32, 16)	2064	['conv2d_15[0][0]']
concatenate_3 (Concatenate)	(None, 16, 32, 32)	0	['conv2d_transpose_3[0][0]', 'conv2d_1[0][0]']
conv2d_16 (Conv2D)	(None, 16, 32, 16)	4624	['concatenate_3[0][0]']
dropout_8 (Dropout)	(None, 16, 32, 16)	0	['conv2d_16[0][0]']
conv2d_17 (Conv2D)	(None, 16, 32, 16)	2320	['dropout_8[0][0]']
cropping2d (Cropping2D)	(None, 13, 29, 16)	0	['conv2d_17[0][0]']
conv2d_18 (Conv2D)	(None, 13, 29, 1)	17	['cropping2d[0][0]']

Total params: 1,941,969
Trainable params: 1,941,969
Non-trainable params: 0

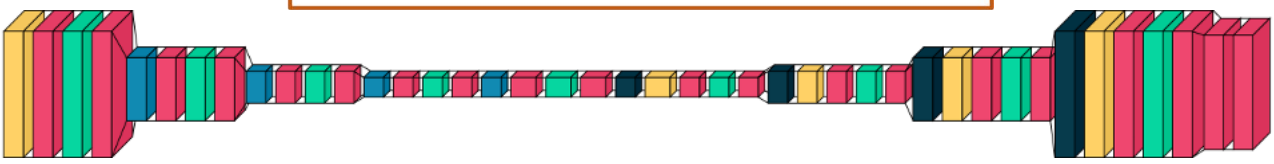

Figure 20 Main Unet architecture considered in this work. This model has been used in the basic configuration with nine input features, and adapted depending on the different configurations of inputs, to evaluate the Unet-based approach.

Another characteristic of the data is their sequential dependence, given the temporal structure of the daily data considered. For this reason, some approaches have also been tested to take into account this characteristic, in combination with the spatial one. Firstly, the same CNNs/Unets have been trained considering, as inputs, the features related to the current day and one day before, or the current day and the three previous days, to try to exploit the features from the recent past. Then, the previous values of the target or the predicted values of the target have been considered

(based on the concept of persistence forecasting), trying to exploit its autoregressive component. Finally, a convolutional-LSTM (Shi, et al. (2015)) layer has been considered to train a model whose architecture combines convolutional and recurrent neural network's properties. This way, the method is specifically designed to take into account temporal dependencies of spatially distributed data. However, as will be further discussed, the inclusion of these recurrent layers did not further enhance the performances.

2.5.2 Implementation

Data

Most of the experimental analysis has been conducted focusing on the Southern Indian Ocean region, which is known for its susceptibility to TC activity (Singh, Ali Khan, & Rahman, 2000). Although still a very rare event, TC occurrence is considered frequent enough to allow for training of skilful ML models predicting their activity. Additionally, it is particularly important to develop reliable models for early detection and forecasting of TC activity in this area, with the aim to mitigate its hazards. Preliminary results also show the ability of the ML algorithms under analysis to generalise over different oceans. In this report, we will focus on the description of the methodological analysis performed, considering the Southern Indian Ocean region as the main applicative example, while a more extensive analysis of the result and their generalisation will be reported in the deliverables related to WP3. Specifically, this work focuses on predicting the probability of TC activity in a region of interest, using data from a 42-year period (1980-2022). The target variable, derived from the International Best Track Archive for Climate Stewardship (IBTrACS), is defined as the occurrence of at least one TC exceeding tropical storm strength (>= 17m/s) evaluated within a 48-hour time window and a radius of 300 km at each grid point in the region of interest. Meteorologically relevant features are taken from the ERA5 atmospheric reanalysis dataset, produced by the Copernicus Climate Change Service (C3S, Buontempo, C. et al., 2022) at ECMWF. All predictor data have also been standardised to make features independent from their unit of measure and scale.

As a standard approach in ML to deal with temporal data, the dataset has been divided into three sequential subsets for training, validation, and testing purposes. Indeed, to design a sound ML model, it is crucial to select a validation set that contains data from a time period independent of the training set, and the same applies to the relationship between test and validation sets. Additionally, to allow for real-time applications, the model cannot be trained on data that refer to a period that comes after the unseen data used to evaluate its performance, that otherwise would be predicted considering data from the future. The training dataset covers the years 1980 to 2010, providing a substantial amount of historical data to build a robust predictive model (11323 daily values). The validation dataset includes the years from 2011 to 2015 (1826 daily values), allowing for the evaluation and fine-tuning of the model performance. Finally, the test dataset spans from 15 April 2016 to 31 December 2022 (2452 daily values), serving as an independent set to evaluate generalizability to new unseen data in terms of skill and reliability. The choice to define a test set without considering the first three and a half months of 2016 is to temporally divide the validation and the test sets, such that the assumption that there is no dependence between the test set and the data that have been used to train and optimise the model is more robustly respected.

As a first set of features, nine candidate drivers have been selected, representing the environmental influence on a TC. Three of them are related to the components that are the inputs to compute the Genesis Potential Index (Emanuel & Nolan, 2004), (Camargo, Sobel, Barnston, & Emanuel, 2007). This choice has been made to allow the model to learn data-driven patterns independently from the GPI index, but considering the features that compose it, which have been identified as relevant by the experts who defined the index. Specifically, the relative vorticity (*vo*) at 850 hPa refers to the measure of local horizontal air rotation in the atmosphere. TC occurrence requires an initial vortex

of sufficient strength and high absolute values of *vo* are therefore required. The relative humidity (*r*) at 700 hPa represents a measure of how much water vapour is in a water-air mixture compared to the saturated state. Finally, the wind shear (*shear*) is a calculated field that expresses how wind vectors change with height. Low wind shear conditions are favourable for TC development and intensification, as they allow for the necessary organisation of convection. Conversely, high wind shear acts adversely as it weakens the convection embedded in a TC. The *shear* is computed as the magnitude of the vector difference between winds at two levels, here 200 and 850 hPa. Additionally, the components (*u,v*) of the wind vectors at these two levels are added singularly as input features. This is motivated by the idea that an ML model may extract complementary information from the underlying wind field compared to the *shear* itself. Therefore, *u_200*, *u_850, v*_200, and *v_850* are the zonal and meridional wind components at 200 hPa and 850 hPa, respectively. They are also included to assess the influence of the large-scale overturning circulations present in the atmosphere.

In addition to the seven features coming from the GPI and the underlying wind field, more features are considered. The top net thermal radiation (*ttr*) is equivalent to the outgoing longwave radiation, which is emitted to space at the top of the atmosphere, and serves as a proxy for convection. Sea surface temperature (*sst*) refers to the temperature of the ocean surface. Warm ocean waters are a key factor to establish surface heat and moisture fluxes that supply convection with energy, and thus foster the formation and intensification of TCs. Additionally, the effect of considering the total column water vapour (*tcwv*) in place of the *ttr* variable has been analysed, since *tcwv* represents instantaneous values, while ttr is accumulated over a particular time period.

Finally, some additional features have been tested, to observe eventual changes in the models' skills based on their value. Firstly, the climatological probability (*clim*) has been added, since it provides the probability for TC activity based on long-term observation statistics of the target variable. It comprises historical information on the location and frequency of TCs in the Indian Ocean region. Leveraging past climatological data provides valuable context and historical patterns that contribute to the overall understanding of TC behaviour. For this reason, this feature will also be considered as a statistical-based baseline. Also, the addition of climate indices has been analysed, since they provide some scalar values that represent large-scale flow patterns of the atmosphere.

Considering the Southern Indian Ocean region as the target area, it is composed of 377 grid points at 2.5x2.5 degrees resolution. Features are considered locally and globally, with the same resolution, for a total number of 10512 grid points. 1-degree resolution of features has also been analysed. As a conclusive illustrative example, Figure 21 reports two examples of features and target for a specific day. Prior to training, the data were standardised to ensure input values were of the same scale.

To practically implement convolutional approaches on the benchmark region considered, 377 gridded points of the Southern Indian Ocean region have been considered as 13x29 (13 latitudes, 29 longitudes) input images. As an example, when the 9 original features have been considered, the input image has been composed of 9 channels, rather than the usual three channels that represent the value of red, green, and blue for each pixel, if encoded with RGB encoding scheme. Therefore, in this case the input tensor of the CNN has shape (11323, 13, 29, 9) for the training dataset, where 11323 daily values are available, for the 13x29 region considered and the nine features. The corresponding target images have a shape of (11323, 13, 29, 1), with only one channel representing the occurrence of a TC for each pixel with class 1 and 0 otherwise. The same extraction has been repeated for the 1826 daily images of the validation set and the 2452 days of the test set.

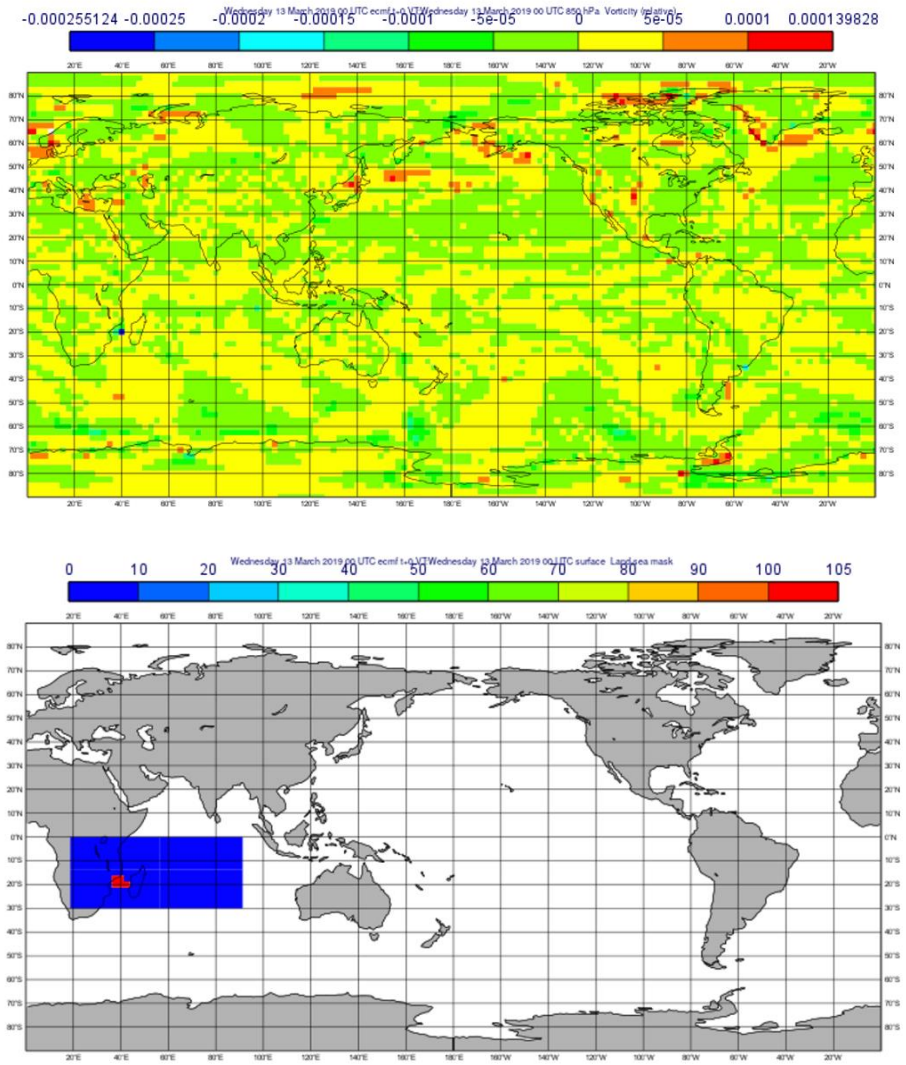

Figure 21 Example of a daily feature and target for the 19th of March 2019. The panel on the left shows relative vorticity at 850hPa. The considered target is a class (0 or 1) representing the occurrence of a TC at a certain location. The panel on the right reports the target class.

Applications

Multiple configurations of the models described above, with different input features, have been considered to test the predictive skills of the models under analysis when different information is available. This subsection describes all the attempts and the models that have been considered for each configuration.

Initial configuration: Firstly, the baseline models (Logistic Regression, AdaBoost, ExtraTrees, FFNN), the CNN, and the Unet have been trained on the nine original features (*vo, r, shear, u_200, u_850, v_200, v_850, ttr, sst*) of the same area of the target, considering for each day daily averages of each variable at each grid-point as pixel. Additionally, oversampling has been tried on FFNN, and it is showing miscalibrated outputs, as mentioned above.

clim: Since we consider the climatological probability of TC occurrence as benchmark, its addition to the inputs of the CNN as additional channel, and the CNN model that only learns on it have been tested, showing that it does not add predictive skills w.r.t. the nine features under analysis, and, on the other hand, that it does not provide enough meaningful information when considered alone.

Previous values of features or predicted targets: As already discussed, given the time dependencies of the data, we tested (through the CNN) if the addition of nine additional channels corresponding

to the features of the previous day is beneficial (for a total of 18 channels). Also, the addition of more days has been considered, up to three previous days (total of 36 channels). In parallel, we tested the benefit, still through CNNs, of the addition of the prediction for the previous day (e.g., when learning a target 3 days ahead, the prediction of the target 2 days ahead is considered as additional input), or the addition of the prediction for all the previous days (in the example above, the prediction of the target with lag 0,1 and 2 have been considered as additional inputs to predict the one with lag 3). No meaningful improvements in the prediction skills were observed considering previous features, while the addition of predicted targets improves the performance.

TCWV and daily averages: Then, we focused on FFNN, CNN and Unet, representing the best baseline and the two best performing approaches. For these three models, we checked for eventual changes in the performances when the TTR feature is replaced by TCWV. Additionally, we tested for the three ML model types if meaningful differences in skill are shown when feature data are represented by a single point in daytime (at UTC00 time) every day, rather than averaging the 6-hourly data over a day. In the first case TCWV does not lead to a worsening of skill, while daily data lead to better prediction than taken from a single point in daytime. Therefore, from this point onwards we focused on models with TCWV in place of TTR, keeping daily measurements of features.

Space and time variables, autoregression: To test the eventual benefit of the knowledge about seasonality and geographical location of each pixel, we added two more features to the FFNN, CNN and Unet. In particular, we firstly evaluated the effect of adding latitude and longitude as additional channels of the inputs of the three models. Then, the further addition in the inputs of the year (1980 to 2022) and day of the year (from 0 to 365) has been tested. Finally, the inclusion of the last observed value of the target has been inserted as an individual additional channel or in combination with latitude and longitude. The results show that the only improvement w.r.t. the nine initial features is identified when the autoregressive component (i.e., adding the previously observed target) is added, showing that the model can better exploit the feature variables combined with the last observed value of the target to predict its changes.

Global fields, recurrent methods: To further inspect the possible improvements of the models, we increased the number of input pixels both considering gridded data of the entire Earth at 2.5x2.5 degrees resolution or considering higher resolution (1-degree) input data for the Southern Indian Ocean region. In both cases, the model may benefit from the additional information provided, but it would also probably need millions of data samples to learn patterns, given the huge increase in the number of parameters. For this reason, given the limited number of observations, we continued focussing on the 2.5x2.5 degrees data for restricted regions such as the Southern Indian Ocean. On the other hand, we also tested the possibility of combining convolutional approaches with LSTM networks, specifically designed for time series. In particular, we tested these recurrent-based approaches using consecutive daily time steps of features to predict the lagged target Additionally, we tested the behaviour of LSTM predicting lagged features, and subsequently performing a detection task on the forecasted inputs to predict the lagged targets. However, also in these cases no significant improvements were observed.

Multiple areas: To test the ability of the models to generalise over different regions, meanwhile increasing the number of available samples, we extended the approach to multiple regions used for training. Firstly, considering FFNN, CNN, and Unet, we considered the Southern Indian Ocean region and the North Atlantic region, identifying two areas of the same shape, leading to different feature and target images for each day (i.e., two samples for each day). Additionally, we further include the latitude and longitude to let the model distinguish between peculiar characteristics of each region. Then, we extended this approach to seven areas, as shown in Figure 22. In this case, we train a single model, having seven times the number of original features. On the other hand, the learning may be more difficult due to the different characteristics of different areas. From the results, it is possible to see that it is slightly beneficial to include different regions, with the advantage that the same

model has learned to generalise over different areas. Therefore, more regions will be considered altogether to produce the final results.

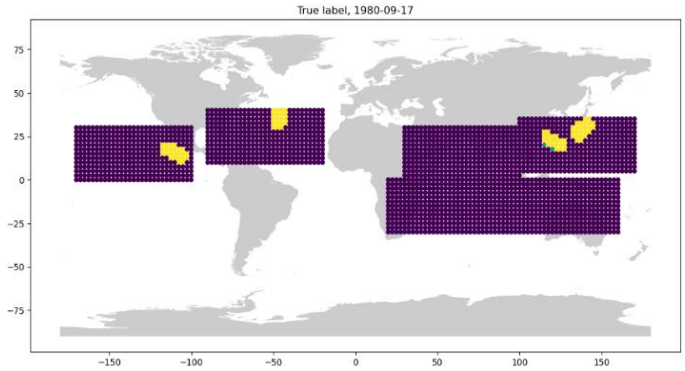

Figure 22 Example of target values for the seven regions considered on the 17th of September 1980. Class 0 is reported in purple, class 1 in yellow. Each region is composed of 377 pixels (13x29 latitude-longitude combinations).

Climate indices: As an additional analysis, we considered the addition of 40 climate indices (see deliverable D3.1 for further details) to the set of inputs. These are daily scalar values that express the current state of various large-scale atmospheric flow patterns, which were determined by empirical orthogonal functional analysis applied to geopotential height at 500 hPa or SST. Their addition as channels would be detrimental, since the convolutional models would have 40 constant input channels, highly increasing the number of parameters, without an added value in terms of their spatial distribution. Therefore, we included climate indices in two ways as inputs of the Unet model. Firstly, we applied PCA to the indices, retaining from 5 or 9 components when considering 95% or 99% explained variance, respectively. This way, only a reduced number of constant channels are added. But even in this case, their addition is detrimental, since, again, a number of constant channels is seen by the network. Therefore, we included the principal components of indices directly in the intermediate layer, between the encoder and the decoder part of the network, adding a flattening layer. This way, we retrieved similar scores w.r.t. only considering the original features, without improvements or deterioration of performances. Therefore, we will not consider the inclusion of climate indices in the final model.
In conclusion, the best-performing ML approach among the considered configurations is the most advanced convolutional-based one, the Unet, as will be discussed in the next subsection. The main benefits w.r.t. to the inclusion of the nine original features reside in considering daily averages rather than single measurements, in the inclusion of the autoregressive component of the target, and in the learning over multiple areas to both increase the number of samples and to generalise the model's skills across different areas.

2.5.3 Results

As discussed in the methodological section, *logistic regression, AdaBoost, extremely randomised trees* and *FFNNs* have been considered as baselines, training one separate model for each time lag between the features and the target from 0 to 13, considering SST, TTR, the three features derived from the GPI and the wind components at 200 and 850 hPa as inputs. Then, CNN-based approaches have been considered to further take into account the spatial correlations of the data, additionally combining them with approaches that allow for modelling temporal dependencies.
All models trained are tested on the period 15 April 2016 to 31 December 2022 to evaluate different aspects of their predictive performance. Because the forecasts made are probabilistic, meaning that they take on values between 0 and 1, while observations are binary (i.e., either 0 or 1), the joint

distribution of forecasts and observations has a larger dimensionality compared to the case of non-probabilistic forecasts. Therefore, verification measures are needed to evaluate the increased complexity. An important property of good probabilistic forecasts is that they are calibrated, i.e., they behave like random draws from the observational distribution. In other words, a calibrated forecast 'means what it says'.

A tool that is widely used in the ML community to analyse predictive skill is the so-called ROC (Receiver Operation Characteristics) curve, which displays the hit rate as a function of the false alarm rate for all possible probability thresholds. It is useful to assess a model's potential predictive ability, but it has the major problem of being insensitive to miscalibration. For this reason, instead, we use the mean Brier score (BS; Brier, 1950, Rufibach, 2010), which is defined as the squared difference between the forecast probability and the observed outcome, averaged over all forecast instances. The BS is negatively oriented and strictly proper, meaning that the estimated score can only be optimised if a model predicts the underlying observational distribution. For the actual comparison between models, the Brier skill score (BSS) is used, which measures the ratio of relative BS improvement, with respect to a reference model. The BSS is positively oriented and ranges from $-\infty$ to +1, with positive values indicating a better (and negative values a worse) skill compared to the reference model. The climatological probability is used as a reference because it is independent of lead time and thus allows comparing BSS values of other models across lead times.

All grid points in the Southern Indian Ocean region (0-30°S, 20-90°E) are pooled during verification, so that conclusions drawn are more robust. Grid points over land are not considered, as their inclusion would further worsen the already existing imbalance in the target dataset.

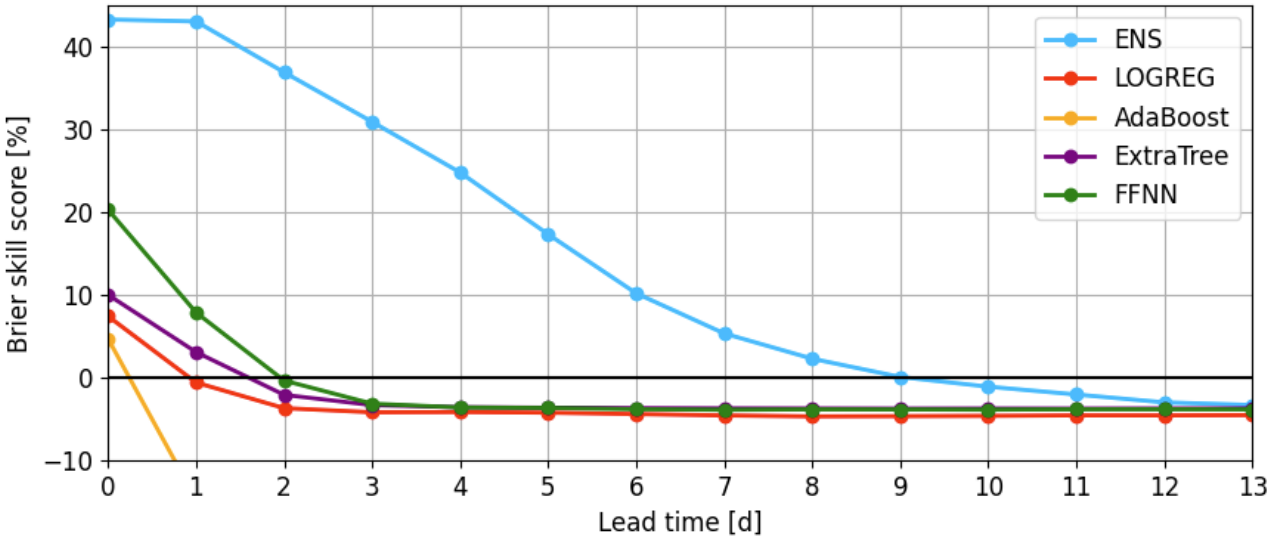

Figure 23 Brier skill score (in %) of tropical storm strike probability with respect to the climatological probability as a function of lead time for baseline and benchmark models.

While the climatological probability forecast is used as the reference in the BSS, ECMWF's ensemble forecast system provides a dynamical benchmark model (denoted 'ENS' in Figure 23). Those dynamically based forecasts turn out to outperform the climatological forecasts by more than 40% in BSS at 0-1 days lead time. With increasing lead time, the skill continuously decreases and drops below the climatological reference beyond day 9. The baseline models clearly perform worse than the dynamical model and are found to have the following descending order in performance at short lead times: The *FFNN* performs best, followed by *extremely randomised trees*, *logistic regression*, and *AdaBoost*. Note that the BSS of the latter is so low that they are not shown for the sake of readability of the other models.

Figure 24 presents BSS over lead time for the best-performing versions of the advanced ML methods considered here, in comparison with the dynamical ensemble benchmark and the FFNN baseline model. While the FFNN model only reached about half of the skill compared to the performance of the dynamical ensemble model at day 0, the U-Net slightly exceeds the dynamical model skill. The benefit of using the advanced methods shows up to 4 days lead time. The results shown and the ones from additional experiments (not shown) suggests the following descending ranking in skill for the advanced methods: U-Net, CNN, LSTM. Beyond 4 days, all models trained so far cannot outperform the climatological and dynamical reference models.

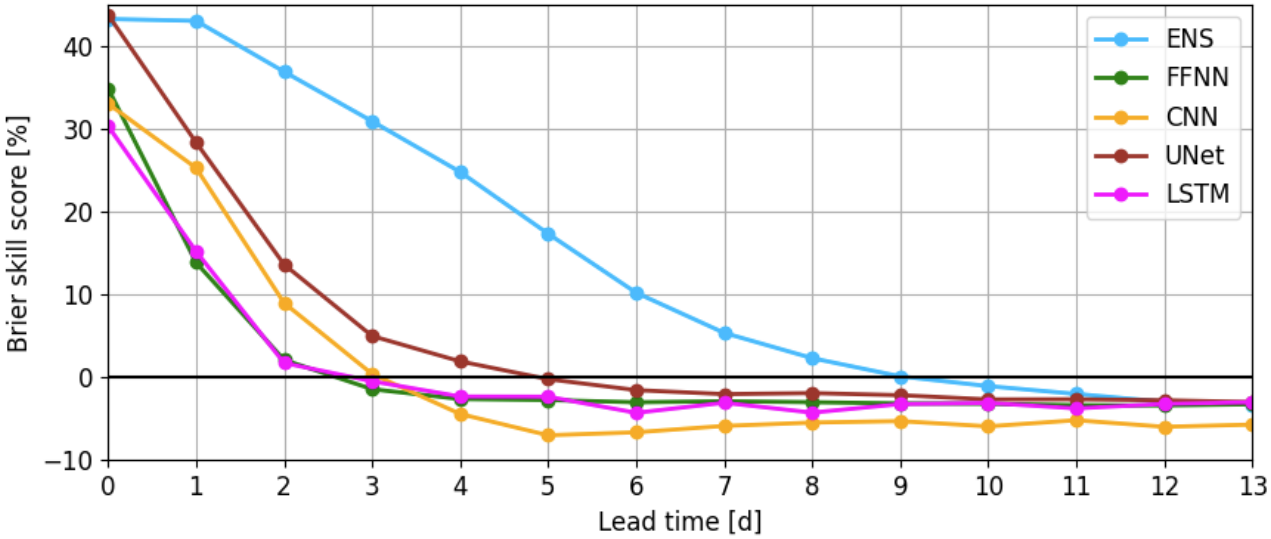

Figure 24 Brier skill score (in %) of tropical storm strike probability with respect to the climatological probability as a function of lead time for advanced models and the benchmark forecasts.

As in many other applications, the Unet model proved to be most useful for detecting and predicting TC activity, certainly due to the advantageous characteristics of its architecture, namely its ability to exploit spatial correlations in input fields through convolutions, and to compress and reconstruct data through the encoder-decoder architecture, together with the possibility to pass information through skip connections.

We conclude this section by summarising the effect of different input configurations and pre-processing, which have been discussed and commented on in the previous section. In particular, additional experiments have revealed that using oversampling techniques to combat the imbalance in the target dataset, as well as training on global input fields, leads to a strong deterioration in skill and has, therefore, not been considered for further development. Overall, a neutral effect was found for including the climatological probability, geographical information (latitude and longitude), and temporal information (year and day of year) as additional predictors, as well as for adding predictors from previous days. Another test showed that the initially used proxy variable for convection, top net thermal radiation, can be replaced by total column water vapour, which provides instantaneous values and hence simplifies pre-processing in any real-time application. Considerable improvements were gained through expanding the predictor set by real-time observations (i.e., previous targets) and the predictions for the previous day(s). Although operationally preferred, due to the reduced data volume and the lower pre-processing costs, the trial to replace the daily averaged predictor data by only the 00UTC values resulted in a non-desirable reduction in predictive skill and is therefore not pursued further.

Conclusion and next step

Given the detection skills of the Unet- model, which is able to perform comparably to the ECMWF ensemble forecasting system, and the decrease in the performance of the purely data-driven models for greater time lags, the best-performing set of ML models and features will be considered in a hybrid approach. This means that features are taken from predictions of a dynamical model and fed into the ML model trained for lag 0. Preliminary results show a significant improvement in this approach, but the details of this configuration and verification results will be presented in the deliverable D3.3 of WP3.

2.6 Two-step dimensionality-reduction method for driver identification

2.6.1 Methodology

This method has been developed to identify drivers of heatwaves (HWs) and warm nights in the detection and forecast task. It is a two-step dimensionality-reduction approach, with the two steps being clustering and an evolutionary algorithm. In the configuration here presented, this method aims at improving the forecasting of HWs from the very same day of the event occurrence (nowcasting) to seasonal horizons. The novelty of the method comes from (1) the use of clusters to reduce the dimensionality of the problem and (2) the exploitation of an evolutionary algorithm to identify and interpret the drivers which provide the forecast skill of the target time series (i.e. heatwave occurrence). This approach allows the selection of spatiotemporal drivers by using a reduced input data dimension. Relevant predictive information of extremes may be contained in large-scale/regional clusters of atmospheric/ocean variables (i.e. dimensionality reduction), e.g. tropical Atlantic SST signals impact on European climate.

The aim is to identify the critical predictor variables by considering the region, lead time, and lag time to detect and forecast HWs and warm nights. The HW occurrence over Lake Como, a case study region in CLINT, is selected. Here, we provide an overview of the methodology being developed (see Figure 25 for visualisation), before elaborating on specific details:

1. Candidate predictors are chosen. We use ERA5 reanalysis. Variables include, but are not limited to, sea surface temperature (SST), mean sea level pressure (MSLP), soil moisture (SM), sea ice content (SIC), and 2-meter air temperature (T2m). The following indexes are also included: NAO and ENSO. Seasonal climatology is removed.
2. K-means clustering is applied to each predictor to reduce the dimensionality of the input data. It also allows for spatial feature selection. The time series of each cluster is then taken as a potential predictor. Here, the pre-processing step has finished.
3. A wrapper feature selection process is then applied. This approach addresses the feature selection as an optimization problem, by minimising the fitness function. The selected predictors (i.e. cluster time series) are used, with the target variable (i.e. time series of heatwave occurrence), to train an ML model. The algorithm outputs a forecast that allows obtaining a skill score, which can be compared later with the test dataset. Although different ML models have been tested, the final proposal is done with a Logistic Regression classifier in search of simplicity.
4. Step 3 is repeated as desired to test a range of clusters, variables, and time lags.

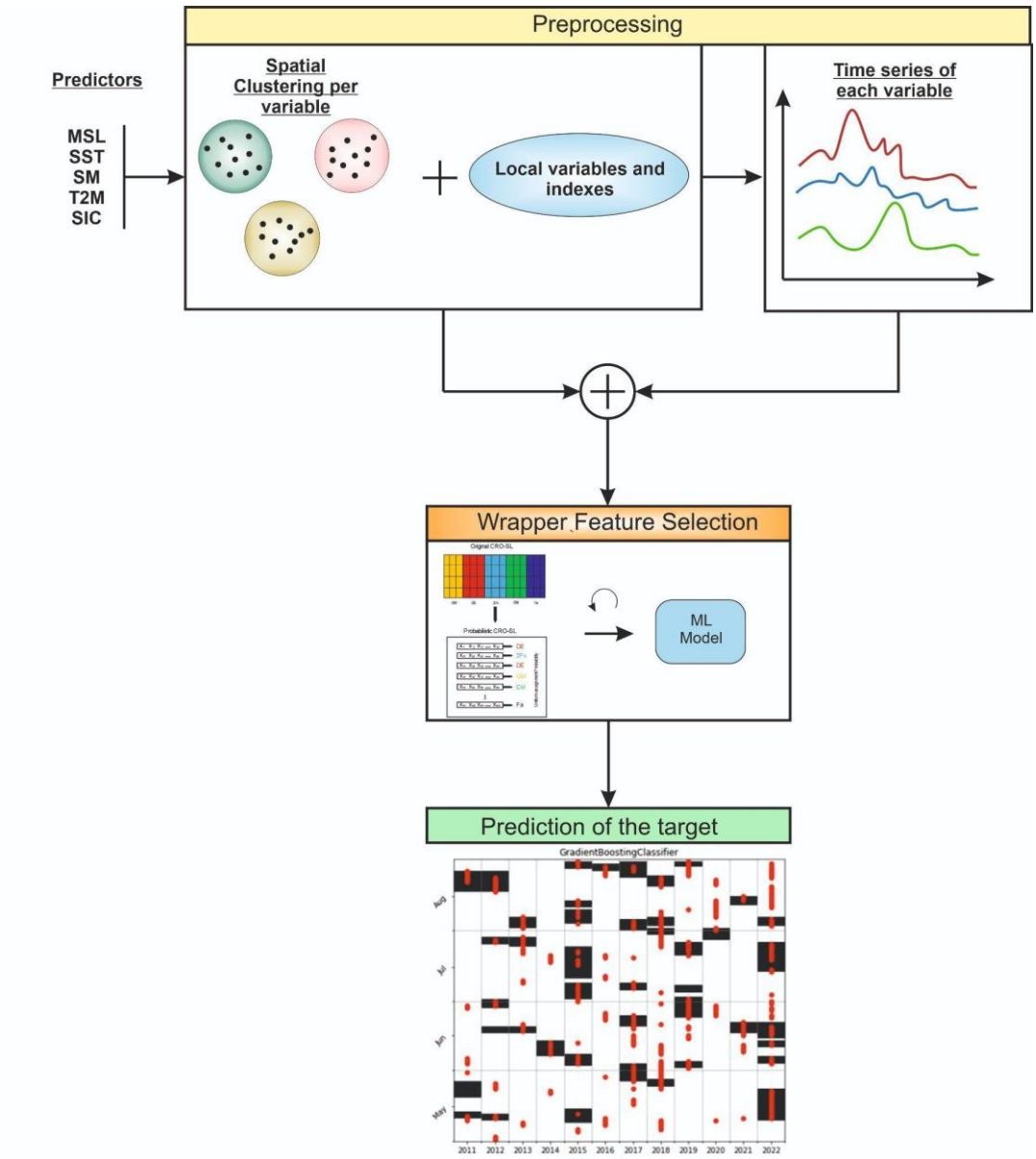

Figure 25 General outline a two-step dimensionality-reduction method for driver identification to reduce the dimensionality of feature space.

A set of data characteristics must be provided to the model, but mainly: variable intensity, either raw or anomaly, spatiality, and time evolution of the variable. In most of cases, some regions have this same information for all the nodes within their domain. This motivates the use of non-supervised techniques, that allows obtaining regions (clusters) that summarise the data for each variable and reduce the number of candidate features. For each variable, there is a fixed number of clusters. Once the clusters are generated, the downscaling procedure is applied by using either the cluster's average or the centre's data.

The feature selection method aims to find the best set of input variables for making the prediction. For that purpose, three data aspects are selected during the process: 1) spatial selection – clusters; 2) time lag selection; 3) length of the timeseries. The problem is addressed as an optimisation problem, in which the main objective is to reduce the error of the validation dataset, based on a cross-validation approach. The algorithm used is the Probabilistic Coral Reefs with Substrate Layers (PCRO-SL) (Pérez-Aracil, J, 2023). It is an evolutionary-based ensemble for optimisation problems. It combines different methods to seek the optimal or near-optimal solution to a problem. The

proposed method is very versatile and can be used for both integer and real codification of the problem. In this case, the problem is addressed with integer codification since input variables are being selected. One of the particularities of this new version of the algorithm is that it could be set up a competition between the methods. They will try to maximise the percentage of the population in which it operates.

The PCRO-SL is based on the CRO-SL, which is based on the CRO, that is a low-level ensemble based on evolutionary computation. The original CRO uses a rectangular-shaped reef of size *M*x*N*, where the possible solutions to the problem (corals) are set. The basic CRO approach is as follows:

1. Initialization: A fraction ρ of the total reef capacity is occupied with randomly generated corals. The reef position that each coral occupies is also randomly selected.
2. Evolution: Once the reef has been populated, the evolution process begins. This process is divided into five phases per generation:
    a. Sexual reproduction: In this phase, new solutions (larvae set) are created from the ones belonging to the reef to compete for a place in the reef. Sexual reproduction can be performed in two ways: external and internal. A percentage F_b of the corals settled in the reef perform external reproduction (Broadcast spawning) and the rest of them (F_b) reproduce themselves through internal sexual reproduction (Brooding). Both reproduction processes are performed as follows:
        i. Broadcast spawning: from the set of corals selected for external sexual reproduction (F_b), new solutions (larvae) are generated and released.
        ii. Brooding: each one of the remaining corals (F_b) produces a larva by means of a small perturbation and releases it.
    b. Larvae setting: In this step, all the larvae produced by Broadcast spawning or Brooding try to find a spot in the reef to grow up. A reef position is randomly chosen, and the larva will settle in that spot only in one of the following scenarios:
        i. The spot is empty.
        ii. The larva has a better health function value (fitness) than the coral currently occupying that spot. Each larva can try to settle in the reef a maximum of three times. If the larva has not been able to settle down in the reef after that number of attempts, it is discarded.
    c. Asexual reproduction: In this phase (also called budding) a fraction F_a of the corals with better fitness present in the reef duplicate themselves and, after a small mutation, are released. They will try to settle in the reef as in the previously described step.
    d. Depredation: Finally, each coral belonging to the worst fraction of the population, $F_{dep,}$ can be predated (erased from the reef) with a low probability, P_d.

In the CRO-SL, several substrates are considered instead of having a single surface *M*x*N*. Each substrate represents an operator. Thus, the CRO-SL is a multi-method ensemble algorithm with several strategies within a single population. In the PCRO-SL, the fundamental idea behind the CRO-SL is maintained, but the operator assignation to the population is made based on probability distributions. Thus, depending on the strategy, the operators can be uniformly assigned to the individuals, or the probability of being assigned to an individual is based on its performance.

The model is run several times to obtain statistical significance of the results. Once the most critical input variables have been identified, a simulation with those variables is developed in the test dataset to evaluate the model's actual skill. The codification of the problem allows an understanding of the importance of the different variables in both spatial and time dimensions. The wrapper excludes some variables. It is essential to consider that these variables have been discarded after considering the results of several simulations, hence avoiding an unlucky exclusion. The different

time steps are selected for the selected variables. Thus, the time lag, blank cells between the bottom line and the first dark-blue cell, and the sequence length, dark-blue cells, are obtained.

2.6.2 Implementation

To extract the drivers, the wrapper feature selection is run ten times to get the statistical significance of the results. The Cross-Validation technique is used for each run to obtain the fitness values. Thus, the metric score of the classifier, which is a Logistic Regressor, is the F1-score, whilst the fitness value is the average of the F1-score for 5 folds (Figure 26).

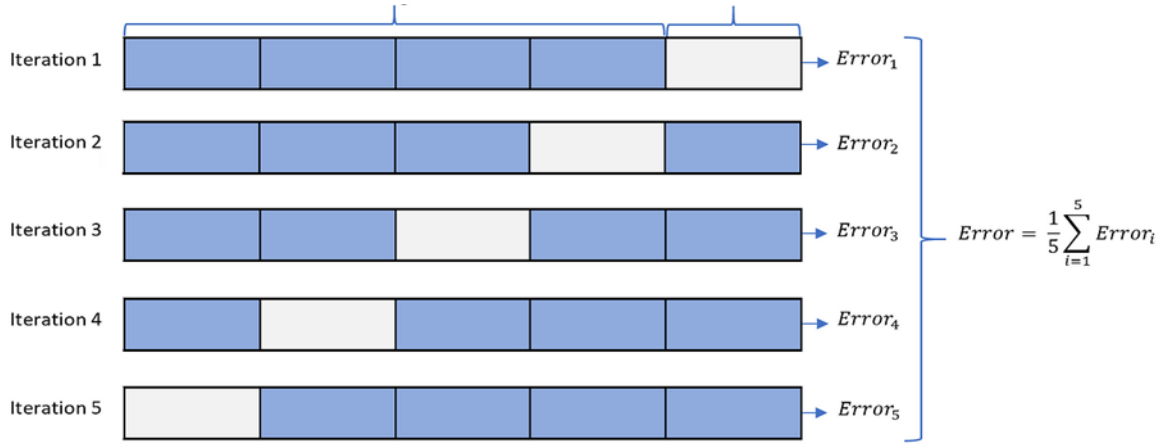

Figure 26 Cross-Validation for error calculation used in the wrapper feature selection method. The Cross-Validation is used in the training process to calculate the fitness value.

Once the model has run ten times, each simulation's best 10% (orange region in Fig. 27) is chosen. Thus, as detailed above, clusters and time series are selected. For the experiments, the data is split into train and validation datasets. The code runs on Python, and standard ML libraries such as scikit-learn are used. The code is on a GitHub repository under development, jperezaracil/ML2EE (github.com). An example of the model performance is shown in Figure 27:

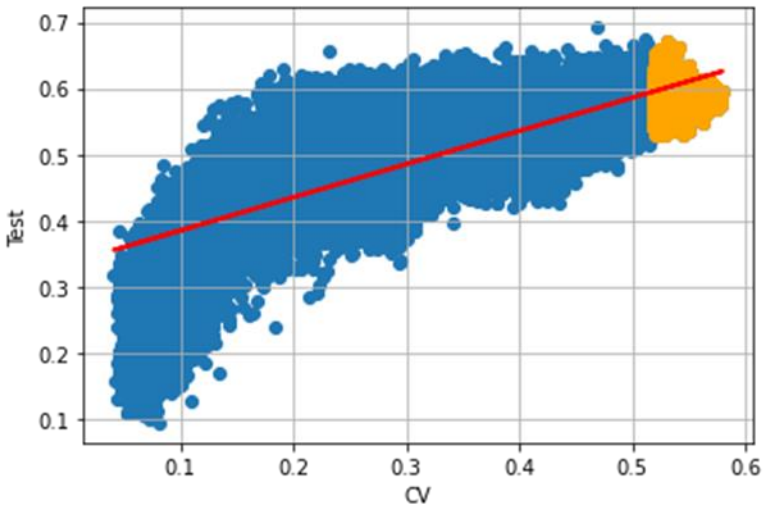

Figure 27 Example of ML performance. Vertical axis represents the F1-score of the test dataset. Horizontal axis represents the F1-score of the Cross-Validation score done during the training process.

For the optimization algorithm, some parameters need to be set up before the simulation, such as population size, number of evaluations, operators, and those regarding the evolutionary nature of the algorithm. All of them are selected based on a trial-and-error method. The configuration is as follows:

- Population size: 100
- Number of evaluations: 15000
- Operators: BLX-alpha; Multipoint crossover; Harmony Search; XOR
- Depredation Probability: 0.8
- Broadcast spawning proportion: 0.98
- Percentage of initial occupation of the reef: 0.6
- Maximum attempts for larvae setting: 3

2.6.3 Results

The model's performance is evaluated against the observed data to refine some aspects of the model. As mentioned above, the target data is the HW occurrence in Lake Como. Since the method is based on a two-step process, in the first one, the data must be accordingly revised to ensure that the clusters and regions are correctly selected. Although clustering is a non-supervised task, the variables must be previously selected. An example of the cluster for the mean sea level pressure (MSLP) variable is shown in Figure 28. The method allows selecting the number of clusters to be obtained by each variable. Thus, five clusters for each variable have been selected.

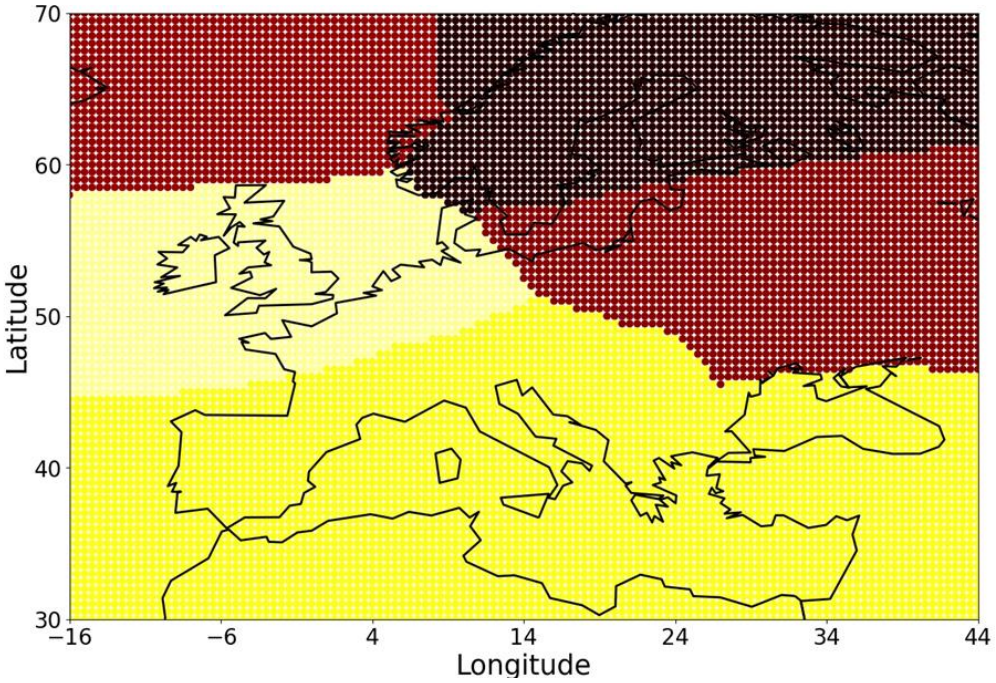

Figure 28 Example of a clustering of mean sea level pressure (MSLP).

The performance of the model should be understood dually: 1) the proper performance of the model detecting HW events; 2) the physical interpretation of the variables. For the first point, there is an objective criterion over the cross-validation method. The second point allows an understanding of the physical problem. This interpretation allows introducing/removing some variables according to the expert opinion and promotes the sensibility analysis for the candidate variables. The best 10% of 10 simulations is selected for selecting the variables. A heatmap of them is shown in Figure 29, for different thresholds, which indicate the percentage of times those variables have been selected.

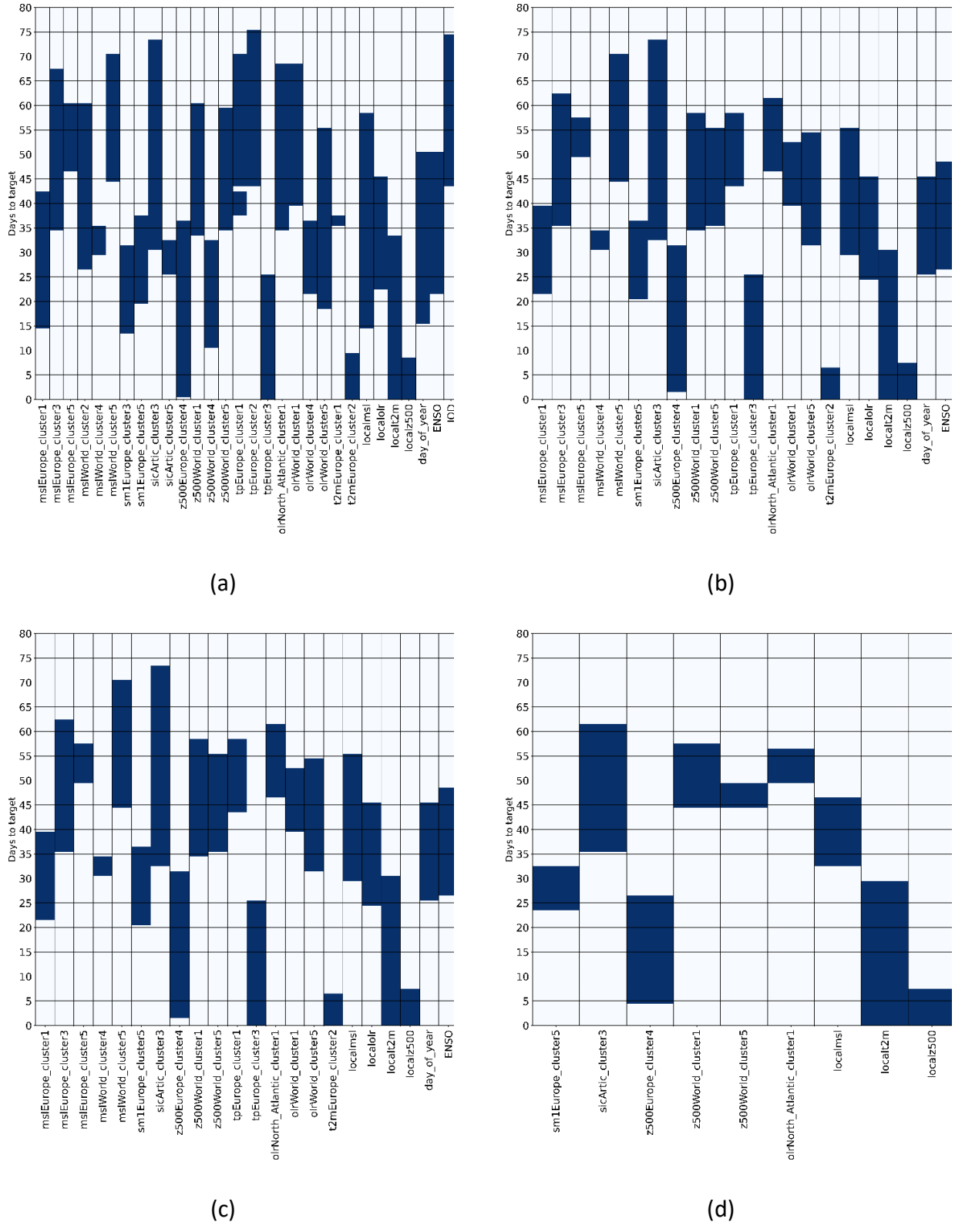

Figure 29 Most frequently selected variables for different thresholds: (a) 25%; (b) 50%; (c) 75%; (d) 95%.

These thresholds allow the calculation of the different F1-scores. This allows for understanding how noisy the data is and the importance of some variables. The results are shown in Figure 30. Low threshold values imply that more variables are used for calculating the F1-score. The F1-score increases with the threshold, up to 75%. However, slight differences can be encountered between 0.75 and 0.95.

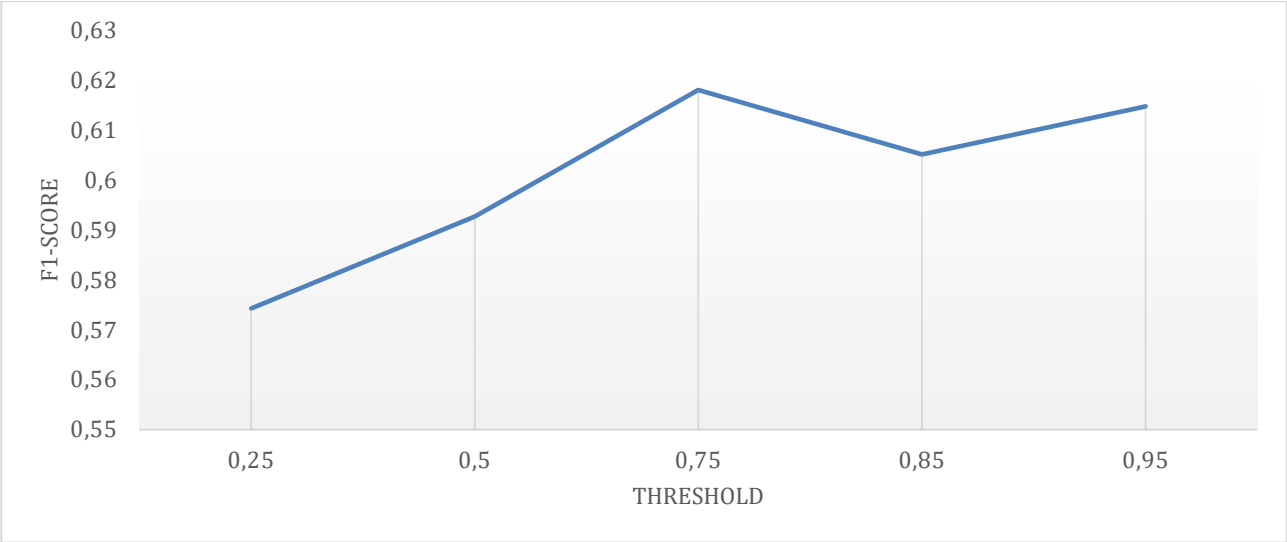

Figure 30 Evolution of the F1-score in the test dataset for the different thresholds of the best solution selected. The curve shows the importance of choosing a suitable threshold and also the quality of the selected variables by the wrapper, since many of them are required to achieve a good performance.

The performance of the Logistic Regressor appears adequate for the problem. Furthermore, the problem's complexity decrease is significantly noticeable as the number of input variables is greatly reduced. Different analyses, such as those based on feature extraction, allow an understanding of the data's noise. It also applies a second step in the reduction of the dimensionality. An example is shown in Figure 31. For achieving similar F1-scores, the model needs, at least, 90% of the variance, which can only be achieved with a high number of principal components. For the case of 99% of variance, 216 components are used.

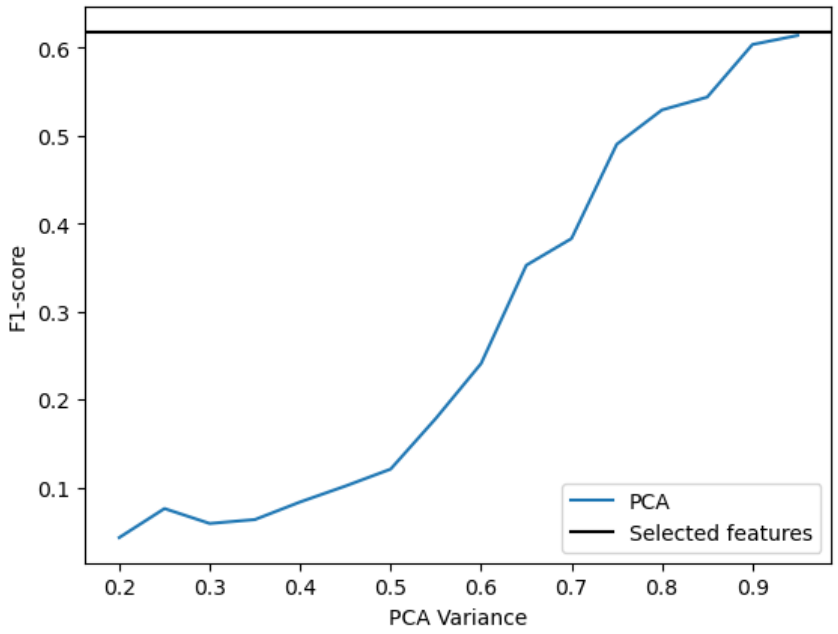

Figure 31 F1-score for different PCA explained variances in the case of 75% of threshold. As the degree of variance explained in the data increases, a higher F1-score is attained. The horizontal black line represents the maximum F1-score achieved without PCA.

It is important to note that this method is only applicable to select variables of the problem, the HW detection in this case. The final forecasting method is not proposed here. Once the spatiotemporal domain for each variable has been defined, more sophisticated models could be developed.

2.7 Concurrent extremes

While the influence of individual extreme events on socio-economic sectors is notable and potentially damaging, the confluence of multiple extreme events across space and time can exacerbate their impact, often with a non-linear relationship (Zscheischler et al., 2018). Consequently, concentrating solely on a single extreme event may substantially underestimate its overall impact and risk (Wahl et al., 2015; Zscheischler and Seneviratne, 2017). Hence, it is imperative to analyse the collective variability of multiple interconnected extreme events to facilitate accurate risk assessment and effective risk management strategies.
This subchapter delves into probabilistic forecasting of concurrent events with seasonal predictability. Seasonal forecasting capabilities typically stem from slow-varying oscillations in the climate system, such as teleconnection patterns or Sea Surface Temperatures (SST). This form of predictability enables the derivation of probabilities regarding shifts in the climate system towards specific conditions, such as drought or heat. Such predictions are crucial for decision-making and can provide well in advance valuable insights for socio-economic sectors like agriculture (Ceglar and Toreti, 2021). In the following sections, we will explore the development of an AI-based approach to assess the probabilities of concurrent extremes on a seasonal basis.

2.7.1 Methodology

Meteorological drivers, data and setup

To develop an effective forecasting model, it's crucial to select meteorological drivers or predictor variables carefully. Detailed discussions on the meteorological drivers contributing to concurrent extremes are outlined in reports for deliverables D3.1, D3.2, and D4.1. Drawing from these deliverables, the following variables are identified as drivers:

1. Relative humidity at 700 hPA serves as an indicator for droughts and/or precipitation.
2. Temperature at 850 hPA provides additional insights into heat conditions.
3. The 500 hPA geopotential height (GPH) describes large-scale atmospheric circulation, such as blocking highs.
4. Soil moisture, as a local variable, acts as a critical modulator of land-atmosphere feedbacks and plays a pivotal role in the development and characteristics of heatwaves and droughts (Miralles et al., 2019).
5. Sea surface temperature (SST), a typical slowly varying climate variable, is essential for seasonal predictability and can significantly influence heatwaves and droughts (Domeisen et al., 2022).

For all experiments, the ERA5 dataset is utilized (Hersbach et al., 2020), except for SST, where the DOISST dataset is employed (Huang et al., 2021).
We exemplify the proposed method by predicting the probability of concurrent extreme events during August in the Valle region of Lake Como (Figure 35), an area of significance for agricultural activities. Various studies have highlighted the impact of concurrent extreme events, such as combinations of dry and hot conditions, on crop yields (refer to, for instance, Hao et al., 2022, for a comprehensive review). To ascertain which of the preceding months exert a predominant influence on the development and characteristics of concurrent extremes, we analyze all six months preceding August for all input variables, representing a standard temporal window for seasonal predictability. The large-scale input features are interpolated onto a coarser 1x1° grid. Bilinear interpolation is applied for SST, temperature at 850 hPA, and 500 hPA GPH, whereas first-order conservative remapping (Jones 1999), is used for relative humidity at 700 hPA. The chosen resolution has been deemed adequate for capturing large-scale patterns and aligns with that utilised

in SEAS51, enabling the identified drivers to inform seasonal forecasts, thereby facilitating the development of hybrid forecasting models.
We tune our model using data from the reforecast- period 1993-2016, with the years 2015 and 2016 designated as test sets. Depending on the specific application, longer timeframes may be of relevance, and thus, the period could be extended up to the present to encompass significant record-breaking years such as 2022 and 2023. Finally, we generate anomalies for all variables based on the climatology from 1993-2016.

Nonparametric climate Indices for extreme event detection

In order to accurately analyse large-scale heatwaves and droughts, the selection of appropriate climate indices is paramount. In this regard, we utilise the nonparametric indicators outlined in deliverable D3.1, which rely on local-likelihood-based approaches (Loader, 1996) of the cumulative distribution function (CDF). These methods offer advantages over classical plugin estimators as they take into account higher moments (Loader, 1999; Geenens et al., 2017). Additionally, it has been demonstrated in D3.1 that these estimators can be leveraged to construct climate indices superior to classical ones, such as the standardised precipitation-evapotranspiration index (SPEI; Vicente-Serrano et al., 2010). The indices are computed for each day to account for the annual cycle of climate variables.
To identify heatwaves, we implement the above-described method by applying it to maximum temperatures, yielding the Standardized Maximum Temperature Index (STMAX). We opt for an aggregation period of three days, resulting in the final index called STMAX-3. Therefore, a value of 1 or higher for this index signifies the persistence of temperature conditions above one standard deviation for at least three consecutive days, aligning with the typical definition of heatwaves (e.g., Perkins, 2015).
For the development of nonparametric estimators for droughts, we employ similar techniques, this time focusing on water balance calculated as the total precipitation minus the atmospheric evaporative demand. The latter is defined as the maximum amount of actual evapotranspiration from land under conditions where surfaces are not constrained by water availability (IPCC, 2021). We estimate this maximum evapotranspiration using the modified Hargreaves-Samani method for daily variables, as implemented in the R-package Clisagri (Ceglar et al., 2020). This approach enables us to generate a daily nonparametric SPEI (NPSPEI) for the Valle region of Lake Como. Utilising an aggregation interval of three days, as for STMAX-3 and merely for demonstration purposes of the method, a value of NPSPEI-3 less than or equal to -1 would indicate the persistence of dry conditions for three consecutive days. Other, appropriate aggregations can be selected for drought analysis according to the local precipitation regime.

Feature selection

Given that the large-scale variables under consideration comprise an extensive array of input features, it is crucial to meticulously filter out irrelevant variables from the feature space intended for prediction. Merely inputting all available information into the machine learning algorithm can result in decreased performance due to the inclusion of excessive extraneous features. This consideration holds particular significance for our case study of Valle, which encompasses a relatively small region.
An effective approach for filtering out inactive variables is Sure Independence Screening (SIS; Fan et al., 2020), which ranks predictors based on a marginal dependence statistic. The SIS property ensures that the method will identify relevant features with a probability tending to one (Fan and Lv, 2008). However, to implement this procedure, one must choose a bound for the marginal dependence measure to determine if a variable possesses sufficient predictive power. Opting for a high upper bound may yield a conservative number of features, while selecting a low bound can lead to the inclusion of too many and irrelevant input variables, resulting in a high False Discovery

Rate (FDR). To control the FDR and maintain the SIS property, Tong et al. (2023), utilise a recently proposed Reflection via Data splitting procedure by Guo et al. (2023), which relies on the conditional independence statistic introduced by Cai et al. (2021). This method supports multivariate output, is robust to heavy-tailed variables, and allows for constraints based on prior knowledge, thereby enhancing the reliability of identifying relevant input features. Additionally, it is nonparametric, permitting consideration of nonlinear relationships between input and output variables without specifying the functional relationship. Another advantageous aspect of this method is that the dependence measure remains invariant to monotone transformations. This property implies that detrending the input variables to account for trends arising from climate change is unnecessary, making it particularly suitable for our applications, given the anticipated changes in the frequency of droughts and heatwaves under climate change (IPCC, 2021).

Kernel Regularized Canonical Correlation Analysis

Given our focus on large-scale heatwaves and droughts, our aim is to pinpoint dominant components and their associated drivers capable of encapsulating the essential features of the system. Principal component analysis (PCA) and canonical correlation analysis (CCA), or their combinations, are commonly employed dimension reduction methods in climate science (e.g., Wilks, 2011). However, these methods have limitations, as they can only discern linear relationships, are restricted to two spatial fields, and are sensitive to pre-processing choices (e.g., retained number of PCAs, orthogonality of input features, etc.).

Tenenhaus et al. (2015) introduced the Kernel Regularized Canonical Correlation Analysis (KRGCCA), which is able to circumvent these challenges by implementing appropriate regularisation schemes capable of handling high-dimensional predictors and mitigating high collinearity or spatial dependencies in features, as observed in our case. Additionally, this approach accommodates multiple variables and can be augmented by a priori graph structures for initial hypotheses and is able to process multiple variables, thereby establishing a highly adaptable framework encompassing various well-known multi-block methods as special cases (Garali et al., 2018). Finally, the kernel trick can be used to extend the methods to the analysis of non-linear relationships by making implicit calculations through the use of kernels thus circumventing the need to specify a non-linear aggregation function (Schölkopf, 2002).

Quantile Regression and Bayesian Neural Networks

Machine Learning (ML) methods are employed to forecast extreme events, with a particular focus on predicting likelihoods of concurrent extremes in our case. To develop probabilistic forecasts, we aim to estimate the CDF of the output variables based on the drivers. Xu and Reich (2023) proposed an elegant approach for estimating the CDF, which combines shape-constrained splines and neural networks. Specifically, they utilise I-splines, which are non-negative splines with unit integral, to construct a basis for the CDF. Each I-spline represents a valid CDF, and a convex combination of these splines also results in a valid CDF (Ramsay, 1988). The concept is to learn the weights for constructing the convex combination using a Bayesian multi-layer perceptron, employing softmax activation for the last layer to ensure appropriate weights are generated. Making use of the universal approximation theorem (Hornik et al., 1989), which states that a neural network can approximate any nonlinear function, this approach is capable of approximating any conditional distribution function.

As outlined above, a Bayesian framework is employed to estimate the weights of the neural network. The process of selecting priors and approximating posterior distributions is extensively detailed in Xu et al. (2022). For our case study, the best results were obtained by utilising the Gaussian-scale mixture model (GSM; Neal, 1996), which enables the network to discern the relevance of each input and each latent feature constructed in deeper layers of the model. Posterior distribution approximation is conducted using Markov Chain Monte Carlo (MCMC) algorithms,

specifically, the no-U-turn sampling (NUTS; Hoffman and Gelman, 2011), which facilitates adaptive adjustment of the leapfrog steps during warmup and iteration (McElreath, 2020). To fine-tune hyperparameters, prediction accuracy is assessed using the expected log pointwise predictive density and the Watanabe-Akaike Information criterion (WAIC; Watanabe 2010; Vehtari et al., 2017), as recommended by Xu et al. (2022). The Bayesian neural network (BNN) yields an ensemble of forecasts that are utilised to construct credible intervals, further quantifying uncertainty.
Estimating the conditional distribution function allows us to compute conditional quantiles through numerical inversion readily. These quantiles are crucial for comprehending the impact of covariates on various segments of the conditional distribution function, particularly emphasising the significance of the upper and lower tails for characterising extreme events. Strategies for leveraging covariate effects will be elucidated in the subsequent section.

Summarising drivers' importance

Although neural networks offer great flexibility in modelling non-linear relationships between drivers and response variables, they are frequently perceived as black boxes that may learn physically inconsistent relationships (e.g., Lazer et al., 2014; Lapuschkin et al., 2019). However, recently developed methods provide at least approximate insights into the effects of non-linear relationships, and we employ accumulated local effects (ALE) plots for this purpose (Apley and Zhu, 2020). ALE plots are particularly effective for strongly correlated regressors, which are common in climate data due to the physical relationships among climate variables and spatial correlations.
In essence, the ALE method summarises the impact of a given predictor variable by examining the derivative of the (adequately weighted) prediction function when the predictor is varied. This is based on the intuition that important variables should significantly affect the function's values, while unimportant features have minor effects. Using this concept, a variable importance criterion can be defined based on the empirical standard deviation of the ALE as strong oscillations of the ALE indicate that the loss function is highly influenced by the input variable, suggesting its importance for prediction (Xu and Reich 2023). Furthermore, the ALE is straightforward to interpret: Positive values of ALE indicate that the response variable increases for the given values of the predictors, while negative values suggest the opposite. Values around zero indicate that the output variable is not influenced by the corresponding values of the predictor.
By combining this approach with the aforementioned BNN, we can investigate how input variables influence different parts of the conditional distribution using conditional quantiles. Certain input features may affect higher conditional quantiles (e.g., 0.90) differently than values around the median or centre of the distribution. Identifying such pre-conditions is crucial for accurately predicting extreme events. Additionally, ALE plots can visualise higher-order effects of input variables, describing the predictive power of interactions among input variables. This allows us to explore how combinations of input variables affect specific parts of the distribution, which is valuable for characterising the compounded nature of desired climate events often considered as pre-conditioned events (Zscheischler et al., 2020).

2.7.2 Implementation and results

The forthcoming section showcases a visualisation of the previously outlined approach, focusing on predicting concurrent heatwaves and droughts in the Lake Como region. These predictions will be specifically conducted for the month of August, selected for demonstration purposes. However, it is essential to acknowledge that in Europe, droughts and heatwaves are predominantly observed during the summer months, influencing agricultural practices as well as the health sector, among others.

Climate indices and feature selection

Initially, we create the local climate indices in the Valle region and compute the NPSPEI-3 and STMAX-3. Subsequently, we conduct feature screening to pinpoint the pertinent drivers for different lead months for both STMAX-3 and NPSPEI-3. We adopt the procedure introduced by Tong et al. (2023), enabling the inclusion of preliminary knowledge into the analysis by conditioning on specific input features as additional information sources.

We choose soil moisture in August as a conditioning variable due to its significance in describing the local climate in Valle, also in the previous month(s), and its crucial role as a modulator of land-atmosphere feedbacks (Miralles et al., 2019). Conceptually, we identify relevant features of the large-scale variables by conditioning on the local climate in August. We then apply the knockoff procedure to each large-scale variable, depicting the relative reduction in features in Figure 32. The features reduction for the drivers is performed for STMAX-3 and NPSPEI-3 separately as there can also be single extreme events or lagged relationships between the output variables and the drivers are denoted as "STMAX-" when the target variable is STMAX-3 and similarly as "NPSPEI-" when the output variable is NPSPEI-3. We will also use this notation for all other experiments.

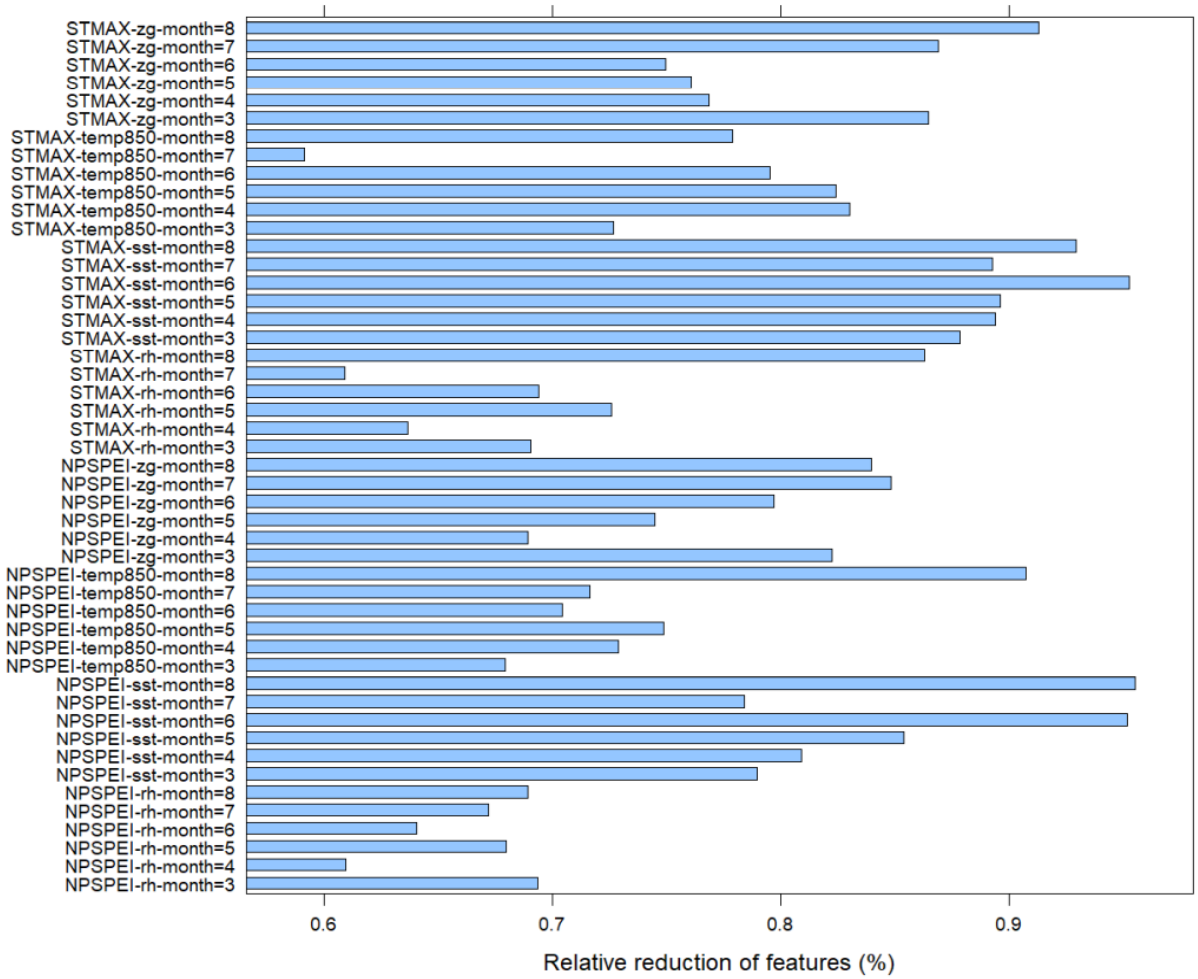

Figure 32 Relative reduction of input features using the SIS algorithm of Tong et al. (2023). The conditional variable is soil moisture in August. Variables described as "NPSPEI-" indicate that NPSPEI-3 was the output variable, while variables denoted with "STMAX" correspond to experiments with STMAX-3 as output. The abbreviation "-month-" describes to which month they correspond. "zg500" corresponds to the 500 hPA geopotential height, "temp850" to the temperature at 850 hPA, "sst" to sea surface temperatures and "sm" to soil moisture. NPSPEI represents NPSPEI-3, STMAX represents STMAX-3 and CEEI is the developed concurrent extreme event index.

The results demonstrate that, for the large-scale variables, the reduction in the total number of features ranges from at least 59% to a maximum of around 96%. While such substantial reductions

may not be surprising given our target region's small size, they underscore the potential efficiency losses that would have occurred had all input features been indiscriminately fed into the algorithm. Leveraging the SIS property of the approach, we can confidently assume that all remaining variables possess predictive power with a probability tending to one (Fan and Lv, 2008). In the subsequent phase, we identify the dominant patterns within this refined feature space.

Dimension reduction and construction of a concurrent extreme event index

To extract large-scale features, we employ KRGCCA for dimension reduction. We explore three types of kernels: linear, Gaussian, and the first-order arc-cosine kernel. For the Gaussian kernel, we utilise an automatic bandwidth selection procedure proposed by Chaudhuri et al. (2017). We select the kernel that maximises the objective function (i.e, the function that is optimised) of the KRGCCA approach. Our findings indicate that the first-order arc-cosine kernel yields the best performance for the drivers, while we retain the linear kernel for STMAX-3, NPSPEI-3, and the newly developed concurrent index.

KRGCCA allows us to use a graph of a priori hypotheses. We connect each input variable for STMAX-3 and NPSPEI-3 individually to differentiate between drivers for droughts and heatwaves, to allow single events or lagged effects between extreme events. Additionally, we connect STMAX-3 and NPSPEI-3 to capture the covariability of droughts and heatwaves. Lastly, we construct a multiblock (Tenenhaus et al., 2017) that consolidates the information from STMAX-3 and NPSPEI-3 to create a new concurrent index for forecasting, which we call Concurrent Extreme Event Index (CEEI). This index is solely linked to STMAX-3 and NPSPEI-3, which, in turn, are connected to the drivers, thereby indirectly linking the CEEI to the drivers of STMAX-3 and NPSPEI-3. Figure 33 illustrates this approach schematically.

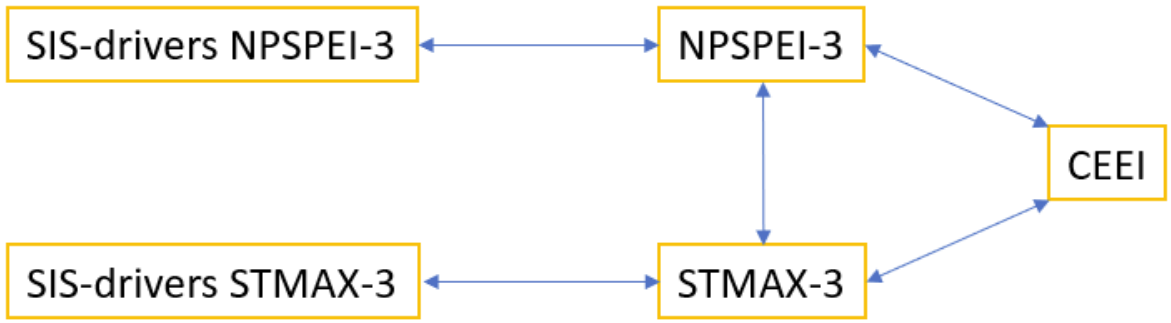

Figure 33 Concept of the graph structure imposed for the KRGCCA analysis. “SIS-drivers NPSPEI-3” refers to the drivers identified by the knockoff procedure of Tong et al. 2023 using NPSPEI-3 as an output variable. “SIS-drivers STMAX-3” refers to the active input features using STMAX-3 as the response variable. Drivers of STMAX-3 are connected to STMAX-3 and drivers of NPSPEI-3 to NPSPEI-3. STMAX-3 and NPSPEI-3 are connected to each other and this information is used to construct the CEEI using a multiblock containing the information of STMAX-3 and NPSPEI-3.

We employ blockwise scaling (Garali et al., 2018) to ensure that climate variables with stronger variability do not dominate the decomposition. Additionally, we incorporate spatial weighting using the method proposed by North et al. (1982) to account for spatial information. Through a regularisation scheme, the algorithms afford the user the flexibility to decide whether to focus on dominant patterns by applying the covariance criterion or on high correlations between the output variables while taking their co-linearity due to spatial dependencies into account (for more details, see Schäfer and Strimmer (2005) and Tenenhaus et al. (2015).

For our study in the Valle region, we seek to identify dominant patterns of STMAX-3 and NPSPEI-3 to describe these sub-systems adequately. This approach is also used for the driver variables, assuming that all retained variables exhibit predictability due to the prior SIS screening, such that a

dominant feature of these components captures essential predictive information, resulting in high predictive power.

To construct CEEI, we prioritize that it has a high correlation with NPSPEI-3 and STMAX-3 such that it captures the covariability of the indices. Since these two indices represent the primary information in their respective systems, the CEEI indirectly represent this information as well, as it is designed to maximize correlation with both.
In summary, our procedure aims to identify the dominant patterns of NPSPEI-3 and STMAX-3, as well as of the individual drivers, by extracting the dominant components of the filtered feature space. These interconnections are utilized to construct the CEEI, maximising its correlation with NPSPEI-3 and STMAX-3. As these two are connected to their drivers, by the properties of correlation the CEEI will also be (indirectly) linked to those.
Following the analysis, we focus on the results displaying the total systems variances through the outer Average Variance Explained (AVE) and individual systems variances are simply called the AVE (Tenenhaus and Tenenhaus 2011), illustrated in Figure 34 for the first two components.

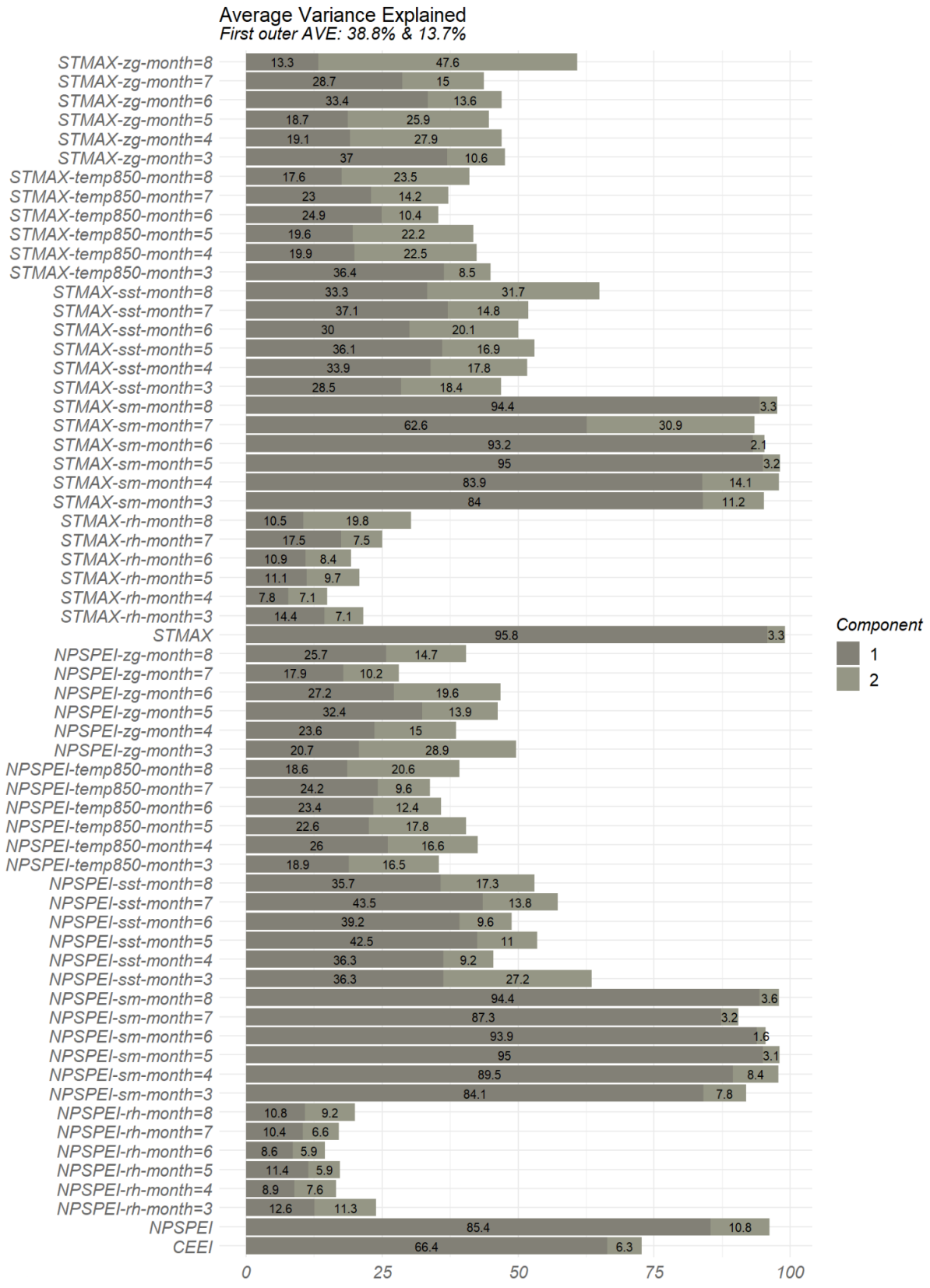

Figure 34 Summary of explained variances from the KRGCCA analysis. The bar plots describe the fraction of variance explained by the components of each individual system, called the AVE. The outer AVE displayed at the top of the Figure corresponds to the total system's explained variance for the first two components. Variables described with “STMAX-” were connected to STMAX-3, while input features denoted with “NPSPEI-” correspond to drivers linked to NPSPEI-3. Variable names are described in Figure 31.

The AVEs represent the fraction that the constructed components are able to explain in their individual systems and are displayed in the bar plot in Figure 34. The outer AVE displayed at the top of the figure corresponds to the fraction of variance of the total system, that can be explained by the constructed components. The findings reveal that the first extracted components can account for 38.8% of the total system variance as indicated by the outer AVE; the second one explains 13.7 %. Concerning the local climate variables, soil moisture, NPSPEI-3 and STMAX-3, the first extracted component explains (with one exception) at least 83.9 % of the variance in each subsystem. For STMAX-3 the first component captures 95.8 % of the variance, while it is 85.4 % for the NPSPEI-3. Thus, the first constructed feature appears to adequately capture the regional information of STMAX-3, NPSPEI-3, and the soil moisture. For the large-scale variables, the system's individual variances mostly range between 10 % and 37 %.
We now shift our focus to evaluating how effectively the newly constructed CEEI captures the covariability of droughts and heatwaves by correlating the first components from the KRGCCA of the CEEI with the ones of STMAX-3 and NPSPEI-3. The results are shown in Table 11.

Table 11 Correlation matrix of NPSPEI-3, STMAX-3 and the CEEI. The percentage in brackets indicates the AVE or the explained variance of the respective first components for their individual systems (Figure 34).

	STMAX-3 (95.8 %)	NPSPEI-3 (85.4 %)	CEEI (66.4 %)
STMAX-3	1.00	-0.47	0.88
NPSPEI-3	-0.47	1.00	-0.83
CEEI	0.88	-0.83	1.00

The CEEI exhibits high correlations with both components of STMAX-3 and NPSPEI-3, suggesting it effectively captures the co-variability of the large-scale hot and dry conditions. Moreover, owing to the construction of KRGCCA, it encapsulates essential information from these systems, with the derived variables for STMAX-3 and NPSPEI-3 explaining 95.8% and 85.4% of the variance in each sub-system, as depicted in Figure 34. Furthermore, the analysis ensures that the CEEI depends on the drivers of STMAX-3 and NSPEI-3 as it is constructed to be as strongly correlated as possible to the two. As the latter are, in turn, correlated to their drivers (Figure 33), the CEEI is also indirectly connected to those by the properties of correlations, thereby enhancing its physical consistency with the large-scale drivers. To check whether the CEEI is also well correlated with the local climate, we correlated the CEEI with each grid point of the Valle region, and the results are shown in Figure 35.

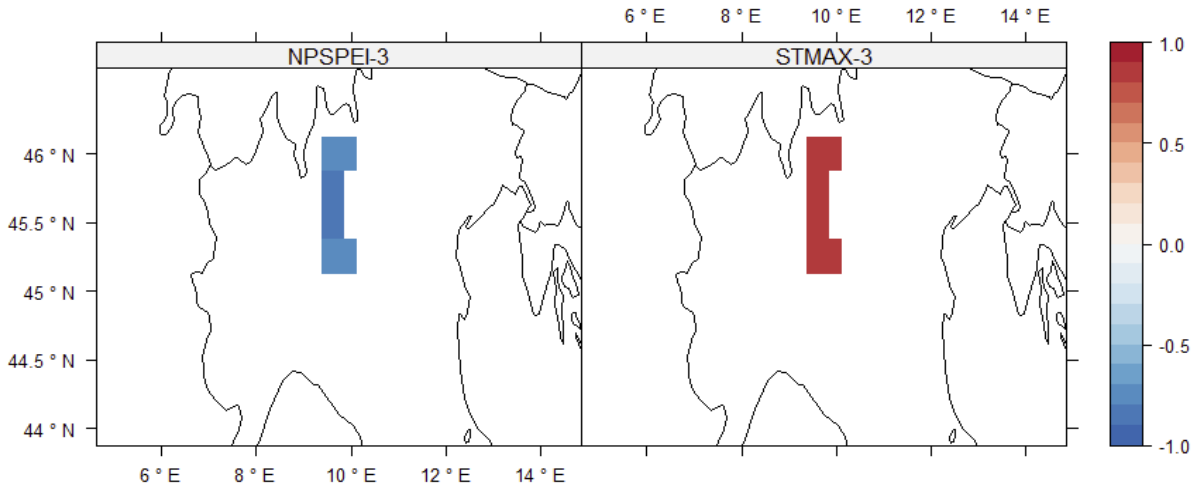

Figure 35 Correlation of each grid point with the NPSPEI-3 (left) and STMAX-3 (right) over the Valle region with the first component of the CEEI constructed by the KRGCCA.

We note that the correlation with local NPSPEI-3 varies from -0.71 to -0.82, whereas for STMAX-3, it ranges from 0.82 to 0.88. The latter suggests that positive phases of the CEEI correspond to dry and warm conditions such that a positive phase of CEEI can be interpreted as concurrent extreme and rising values of the CEEI suggest increased magnitude. As such, our index can be used to perform detection as well as assess severity. This observation leads us to conclude that the dimension reduction experiment has effectively generated a new index for concurrent extreme events rendering it well-suited for forecasting purposes.

Bayesian Neural Network and conditional quantile estimation

Having acquired the large-scale drivers and CEEI, we employ a Bayesian Neural Network (BNN) model to predict the likelihood of observing concurrent extremes. Utilising the BNN-based approach introduced by Xu and Reich (2023), we estimate the conditional distribution function, enabling us to assess likelihoods. Employing the GSM distribution as a prior, we approximate the posterior distribution using MCMC algorithms, specifically the NUTS-sampler. We conduct 2000 warm-up iterations followed by 5000 iterations, discarding every second iteration to generate an ensemble of 1500 samples. Subsequently, we fine-tune the number of knots, neurons, and hidden layers using the WAIC.

The output variable is the CEEI constructed in the previous chapter, while the input variables comprise the drivers derived from the KRGCCA. The input features are scaled to the [-1,1] range, a pre-processing method we found to be more effective than z-scaling and is also used in the simulation studies of Xu and Reich (2023).

To initiate our experiment, we predict the CEEI in August using input features from the KRGCCA ranging from March to July. Thus, this setup enables probabilistic forecasts of concurrent extremes in August with knowledge of the climate conditions up to July. To decide how many constructed components from KRGCCA needs to be retained to represent the input feature space adequately, we also treat the number of those components as a hyperparameter and tune it with the WAIC.

The final estimated model comprises 20 Neurons and I-spline basis function with one hidden layer, and is thus relatively low dimensional for a neural network. Also, the tuning by the WAIC suggested that using only the first constructed component from the KRGCCA is sufficient. With an acceptance ratio of approximately 95.1%, the model demonstrates efficient convergence. Despite conducting 5000 iterations and 2000 warm-up iterations, the total computing time amounted to only 30.51 minutes. Given those numbers of iterations and warm-ups, the ensemble size of the final model is 1500, enabling robust statistical inference and characterisation of uncertainty.

To assess the model's performance, we employ the probability integral transformation, which posits that the predicted values for the CDF should be (standard) uniformly distributed. This evaluation can be visualized and verified through Q-Q plots. Figure 36 presents the ensemble mean as well as the full ensemble, providing insights into model accuracy and uncertainty characterisations.

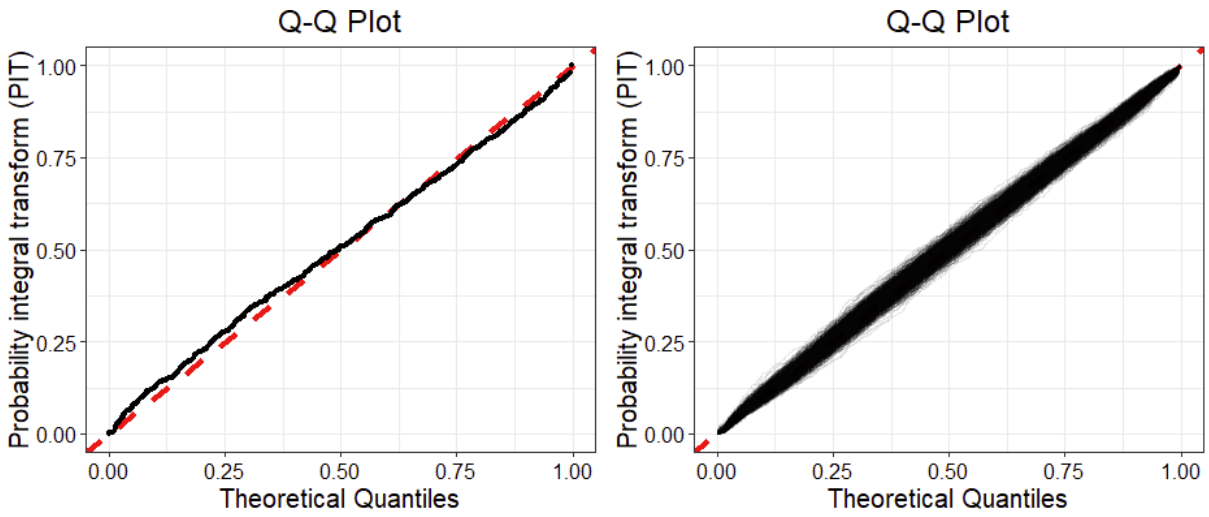

Figure 36 Q-Q-plots for evaluating the conditional distribution function. The panel on the left-hand side represents the ensemble mean, and the one on the right represents the full ensemble. In red is the identity line used for assessing the fit of the distribution.

The ensemble mean appears to deviate slightly from the identity line (equality of quantiles), especially for low quantiles, while the full ensemble appears satisfactory overall, suggesting it is well suited to evaluate the CDF of CEEI. However, both models appear to be well suited to predict high conditional quantiles corresponding to the concurrent extremes, as the high percentiles are very close to the identity line. To facilitate comprehension, we now discuss how the model can be utilized for prediction based on the ensemble mean.

Figure 37 illustrates the estimated conditional quantiles for the ensemble mean. We generated these by predicting the conditional quantiles of the ensemble mean on a fine grid, spanning quantiles from q=0.01, 0.02, …, 0.99. The model demonstrates proficiency in predicting the time series, effectively capturing most points and the overall evolution of the time series. An exception is noted around 2000 and 2003, corresponding to the minimum and maximum values of the time series, respectively. These extreme values correspond to conditional quantiles of q=0.00 and q=1.00, which are not explicitly predicted due to the chosen grid resolution. A probability of observing a concurrent extreme could now be calculated by estimating the probability that the CEEI exceeds certain thresholds like, for instance, its 90^{th} percentile. Based on this summary statistics can also be computed.

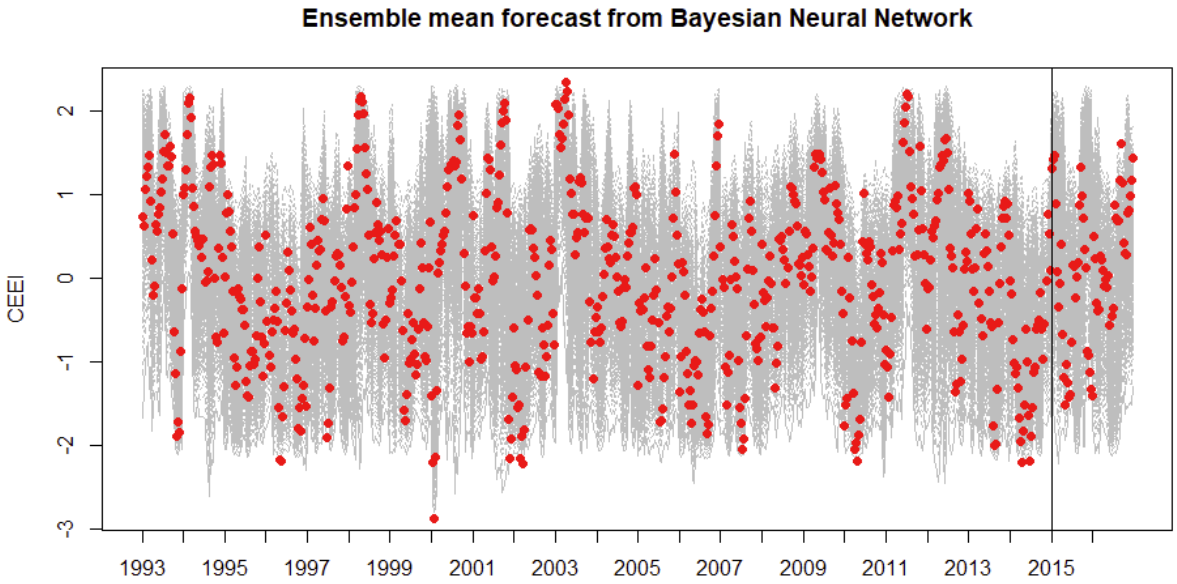

Figure 37 Forecast of the derived BNN model. Grey lines represent the generated set of quantiles by the BNN model evaluated at a grid of conditional quantiles q=0.01,0.02,...0.99. Red points are observations, and the vertical black lines represent the separation between the training and test sets.

Evaluating variable importance and approximate relationships

Understanding the interconnections and influences among input and output variables in the climate system is paramount, given its physical nature. To discern the variables significantly affecting the forecast and gauge their importance, we compute the ALE for each ensemble member. The standard deviation of ALE is utilised to rank the variables based on their importance. Subsequently, we construct the median and 95% credible intervals for conditional quantiles at q=0.10, 0.50, and 0.90. This approach enables us to investigate various aspects of the time series, encompassing very low, intermediate, and extreme positive values. Of particular interest are these extreme values, indicative of concurrent extremes. The numbers displayed in the bar plots represent the ranking of the features with respect to the median of the obtained distribution of the variable importance criteria, and whiskers correspond to a 95 % credible interval. For demonstration purposes, we use the five most important features.

Our analysis, depicted in Figure 38 underscores that the ranking of predictors varies depending on the conditional quantile. For instance, while the soil moisture in April emerges as the most crucial predictor across all quantiles, the 750 hPA relative humidity in June is the third most important predictor for the 90th percentile, while it does not at all occur for the 10th percentile. Another example is the relative humidity in April, which is only ranked within the five most important predictors for the 10th percentile. Finally, predictive powers of the input variables are also differently ranked throughout percentiles: For instance, the SST in June is the fifth most important predictor for the 90th percentile and the third most important for the median. This underscores the diverse effects of predictors across different segments of the output variable's distribution, demonstrating the efficacy of our method in identifying these nuanced relationships. This capability presents a distinct advantage over many regression methods, which typically infer solely on certain parts of the distribution, like the conditional mean. By allowing us to scrutinise various facets of the distribution, our approach offers a more comprehensive understanding of the dataset.

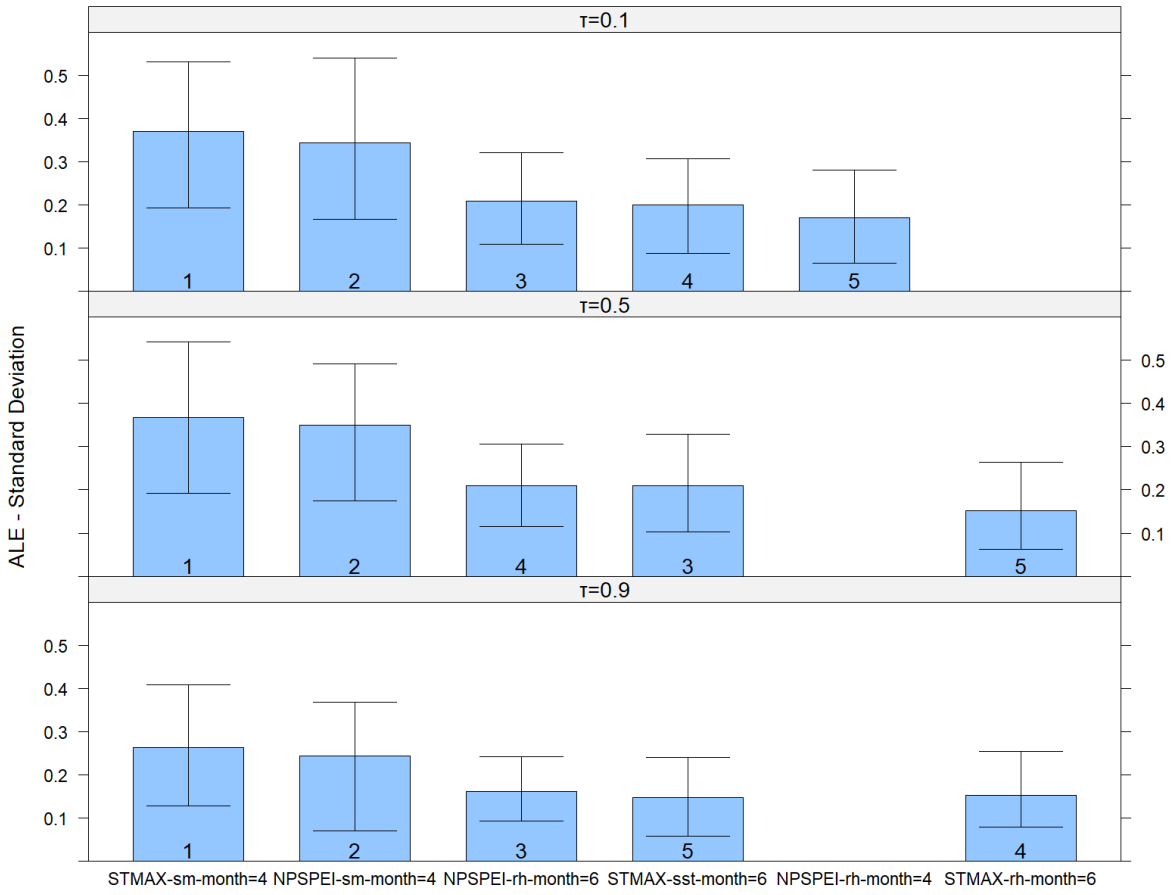

Figure 38 Variable importance plots for the five most important input features of the BNN model. Numbers indicate ranking due to the variable importance criterion with respect to the median of the distribution of the latter. Whiskers indicate 95 % credible intervals. A description of variable names can be found in the description of Figure 31.

Our initial investigation focuses on determining the significance of specific driver combinations for prediction, particularly targeting the 90th percentile, which corresponds to concurrent extremes revealing that several crucial variables manifest as early as in spring, like soil moisture in April. To underscore the importance of analysing the joint effects of predictors, we delve deeper into the individual impacts of soil moisture in April and SST in June, followed by an exploration of their combined effects through second-order ALE plots.

We initiate our analysis by examining the individual effects of the variables displayed in the ALE plot in Figure 39. The notation "-ncomp=1" is used to describe that it represents the first constructed component of KRGCCA. Notably, during negative phases, both variables hover ALEs values around zero, while the latter steadily increases as they shift towards positive values. This increase suggests a corresponding rise in the values of the CEEI for positive values of the variables. Intuitively, one might assume that negative values of the individual time series exert negligible influence on the CEEI, given the near-zero ALE. However, a deeper understanding emerges when exploring the joint effect, as depicted in Figure 40: here, we observe that increasing values of the CEEI can indeed manifest when the SSTs experience a negative phase and the soil moisture in April has already been in a positive state. Notably, the estimated ALEs in the upper left corner of Figure 40are almost of similar magnitude as the individual effects in Figure 39. Hence, if these compounding effects of precursors occur, the likelihood of observing a concurrent heatwave and drought in August seems to be significantly increased. Such nuanced interactions might be overlooked when solely considering the individual effects of the time series, as illustrated in Figure 39. Consequently, our

proposed method offers a comprehensive approach to evaluating the compounding nature of drivers, uncovering complex relationships critical for accurate predictions.

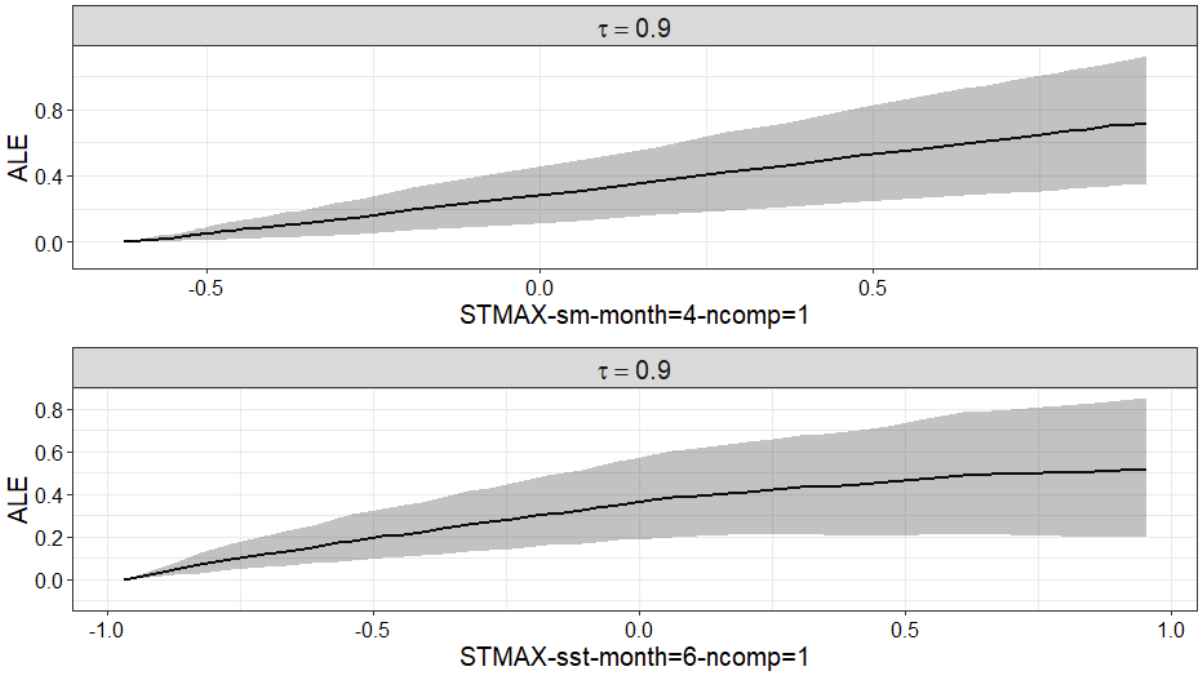

Figure 39 Estimate local accumulated effects of the soil moisture ("sm") in April and sea surface temperatures ("sst") in June for the 90% conditional Quantile obtained from the BNN model. The blacks line marks the mean of the ensemble and the grey tube corresponds to 95 % credible intervals. The abbreviation "ncomp=1" means, that the used variables correspond to the first constructed component of the KRGCCA.

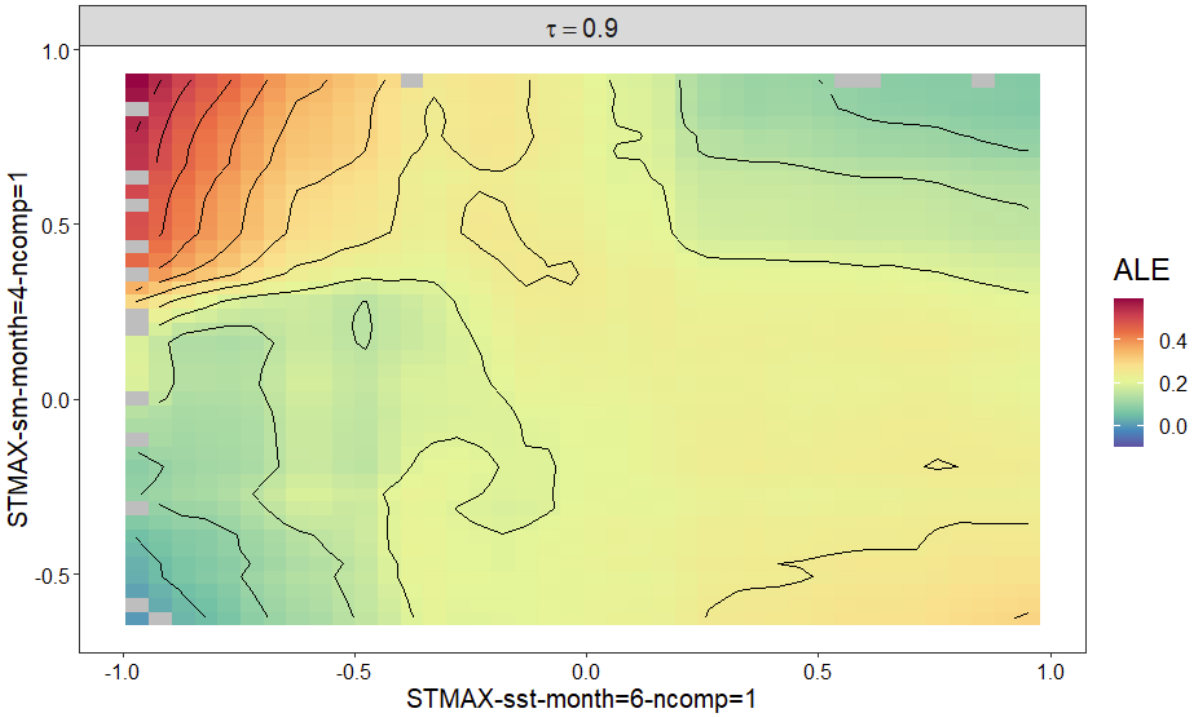

Figure 40 Second-order ALE interactions using soil moisture (SM) in April and sea surface temperatures (SST) in June. Both variables are drivers for STMAX-3 (Figure 33). Variables labels are described in Figure 39.

One should note that due to the properties of neural networks, the identified important variables do not necessarily have to be physically consistent drivers and can merely be statistical predictors that the neural network found useful for prediction. If one is interested in the underlying physical causality, more evaluations are necessary. It is possible, for instance, to obtain climatic patterns

corresponding to the constructed input features using methods such as pre-images, as outlined in Bakir et al. (2004). For this deliverable, we will not pursueis this latter path further, as the focus is on representing the newly developed statistical methods.

2.8 Post-processing of hydrological model predictions using ML methods

Hydrological modelling has greatly enhanced our understanding of the water cycle, providing valuable perspectives on water flow, distribution, and quality (Guse et al., 2021; Yang et al., 2021). However, implementing large-scale hydrological models from national to global levels presents a series of challenges. These challenges arise from uncertainties in identifying the correct structure and parameters of the model, which can reduce model accuracy and create gaps in our understanding of water cycle dynamics (Kraft et al. 2022; Chevuturi et al. 2023). The variability in hydrological responses, influenced by a range of factors (e.g. climatic conditions, soil types, topographical variations, and human interventions; Du et al., 2023; Pechlivanidis et al., 2020), further complicates the modelling process, especially in areas with limited measuring stations, where conventional data collection methods are inadequate. The lack of data hinders the calibration and validation of large-scale hydrological models, therefore limiting the models' reliability and effectiveness.

Recently, machine learning methods have significantly improved hydrological modelling by enabling the analysis of complex patterns in hydrological data (Kraft et al. 2022; Bézenac, Pajot, and Gallinari 2019; Geer 2021; Moradkhani et al. 2005; Xu and Liang 2021). Advanced algorithms, including neural networks, decision trees, and ensemble learning, have enabled a better understanding of nonlinear relationships between hydrological processes and environmental factors, leading to notable advancements in predicting water distribution, flow, and quality. These methods excel in managing large datasets, improving model precision, and reducing uncertainties, consequently offering critical support for water resource management and environmental planning. ML and statistical approaches add significant improvements to physics-based models, especially for extreme events.

Consequently, this research investigates two ML-based post-processing approaches to enhance streamflow prediction. We tailored the streamflow outputs from the hydrological model to better reflect local dynamics, aiming to effectively narrow the discrepancy between large-scale hydrological model outputs and local conditions.

2.8.1 Methodology

Post-processing workflow

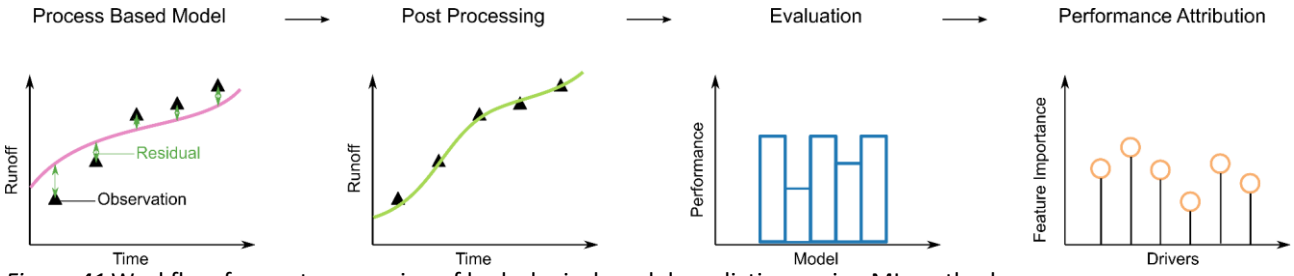

Figure 41 Workflow for post-processing of hydrological model predictions using ML methods.

In Figure 41, we illustrate the structure of our post-processing framework, and results are compared against the process-based hydrological model. To address the discrepancies between the model outputs and the observations, our post-processing phase applies specific algorithms, including machine learning methods, to reduce these discrepancies. In our study, we use machine learning approaches, Random Forest (RF) and Long Short-Term Memory networks (LSTM). We assess the

performance of these post-processors through various metrics focusing on both volume and extremes. Moreover, we explore the importance of different features by examining how the post-processors' performance correlates with factors like climatology, topography, human activity, and hydrological regimes.
The analysis in this deliverable focuses on method verification, therefore in the evaluation phase. An extended experiment will be carried out in WP6, with spatial extension to pan European region, and therefore, the performance attribution analysis will then be conducted, to include diverse climate conditions and hydrological dynamics.

Random Forest

Random Forest (RF) is a supervised, non-parametric algorithm, where an ensemble of uncorrelated trees yields a prediction for classification or regression, instead of a single decision tree. The method is employed to model observed runoff by capturing complex relationships and interactions within the simulations. Multiple trees are built based on bootstrapping samples from the training data. At each split node, a random subset of predictors is considered, and the split is made to minimise the sum variances of the target variable in the two resulting branches. In the case of regression, after all the trees are grown, the forests produce the final results by averaging predictions from the trees (Pham, Luo, and Finley 2021). The Random Forest package in R was used for model training and testing. This approach is applied individually to each station, with observed runoff serving as the target and simulated runoff as the input variable. The same model configuration, regarding maximum node numbers (10) and minimum node size, is maintained across all stations. This ensures comparability throughout the study domain, facilitating the analysis of potential influencing factors.

Long Short-Term Memory model

Long Short-Term Memory (LSTM), initially introduced by Hochreiter and Schmidhuber (1997), is a state-of-art model for time series (Kratzert et al. 2018), which is capable of learning long-term dependencies. For post-processing purposes, the LSTM in our framework is specifically designed with a 3-day lookback period, which has confirmed its capability of capturing temporal dependencies present in runoff data, by our experiments between 1 to 215 look-back days. This model is structured with three layers containing different numbers of cells (100-50-20), which allows the LSTM to effectively process and remember information over extended periods, making it exceptionally suitable for modelling complex hydrological sequences.
For model training, the dataset was subsequently divided into training and testing periods, by 80%/20% split. To ensure the model's generalizability and prevent overfitting, a portion of the training set (10%) is reserved as a validation set. The model training includes a monitoring mechanism where if the validation loss does not decrease over 10 consecutive steps, an early stopping criterion is triggered. This strategy ensures the model against overfitting by interrupting the training process when no further improvement is observed in the validation dataset, thereby allowing the model's performance to be optimized without compromising its ability to generalize to new, unseen data.
Normalization is applied to the input data to standardize the range of data points, facilitating smoother training dynamics and more stable convergence. The target variable, representing the relative residual between observed and simulated runoff, is calculated as below:

$$target = (y_{obs} - y_{sim})/(y_{sim} + \varepsilon) + 1 \quad (4)$$

ϵ is a small constant introduced to prevent division by zero, particularly in scenarios of low flow, ensuring the target remains within a reasonable range.
To address data imbalances, particularly concerning extreme values critical for hydrological services, the sample weight technique is implemented. This method assigns weights to samples, emphasizing

the importance of accurately predicting extreme events, which are often underrepresented in the dataset but hold significant importance for hydrological analyses and applications. Through this weighted approach, the model is better equipped to focus on and learn from these extremes. Weights are assigned by percentiles in the observation runoff as in Table 12, where 10th, 33rd, 66th, 90th percentiles were included for dividing the groups, representing low extremes, lower than normal, higher than normal, and high extremes. Thus, the extreme categories are effectively weighted three times more strongly than the central categories.

Table 12 Sample weights assigned for LSTM based on observed runoff quantiles.

Range	Weight
> 90th	0.2
66th to 90th	0.2
33rd to 66th	0.2
10th to 33rd	0.2
< 10th	0.2

Metrics for performance assessment

The Kling–Gupta efficiency (KGE) is used to assess the performance of ML based post processing methods. KGE is a goodness-of-fit indicator widely used in the hydrologic model evaluation for comparing simulations to observations, following the formula:

$$KGE = 1 - \sqrt{(r-1)^2 + (\alpha-1)^2 - (\beta-1)^2} \quad (5)$$

Where r is the Pearson correlation coefficient, α is a term representing the variability of prediction errors calculated from variance, β is a bias term calculated from the means. Details of the metric can be found in (Gupta et al. 2009).

Enhancement at each station is represented by the computation of skill scores, which measure the effectiveness of post-processing methods compared to unprocessed simulations, as below:

$$Skill = \frac{Score_{ml} - Score_{raw}}{Score_{perfect} - Score_{raw}} \quad (6)$$

In this context, skill scores below zero suggest a decline in model performance, whereas positive scores indicate enhancements. A skill score nearing 1 indicates a significant improvement in predictive accuracy, underscoring the success of the post-processing strategies in improving hydrological predictions. This method of quantification allows for a detailed assessment of how post-processing contributes to the accuracy and reliability of hydrological forecasts, distinguishing between instances of model refinement and instances where the model's performance may have been compromised.

2.8.2 Implementation

The post-processing framework is applied to stations in Sweden, where simulated runoff data from the EHYPE model covering the period from 1961 to 2023 were obtained, while observations from in situ stations were collected and stations with at least 10 years of records were selected. An

extended experiment is carried out in WP6, where the post-processing framework is applied to the Pan-European region.

2.8.3 Results

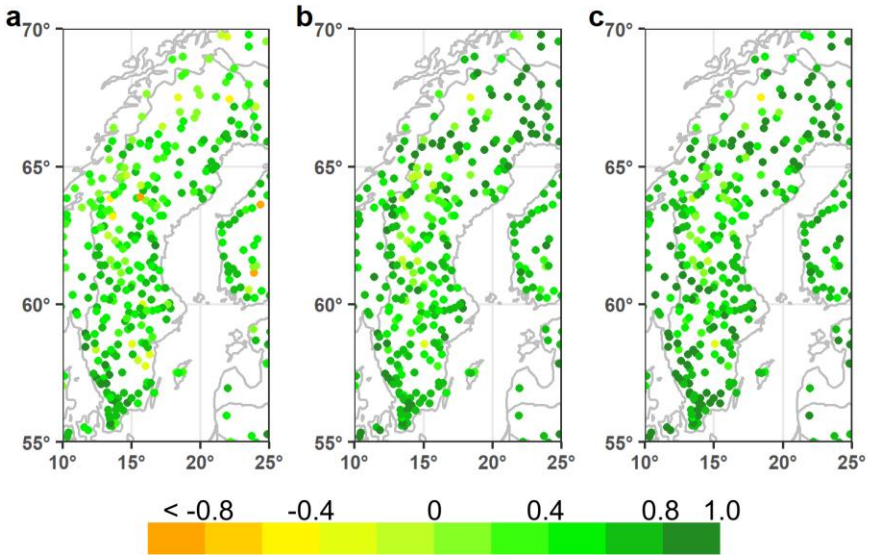

Figure 42 Performance of raw model, and post-processing methods (measured by KGE): a, raw model; b, RF; c, LSTM.

In Figure 42, the performance of raw model and ML-based post-processing methods are depicted, with orange representing unsatisfied performance (low KGE), and green representing good performance (high KGE). From the figure, we see a change towards darker green in both post-processed results, indicating improvement in terms of KGE. Similar results can be achieved by RF and LSTM, whereas LSTM shows higher KGE in southern Sweden.

In Figure 43, the improvements gained from post-processing are presented as skills, where the grey colour denotes no improvement compared with the raw model performance. For both methods, higher skills are achieved in northern Sweden, and smaller skills in the south. LSTM is more robust since it achieves improvement in most of the stations, while more stations receive no improvement from RF (grey dots). Overall, both methods have the capability of improving the raw model simulation, adapting it to local dynamics. Experiments on an extended spatial domain will further validate their performance in more diverse climate and hydrological conditions.

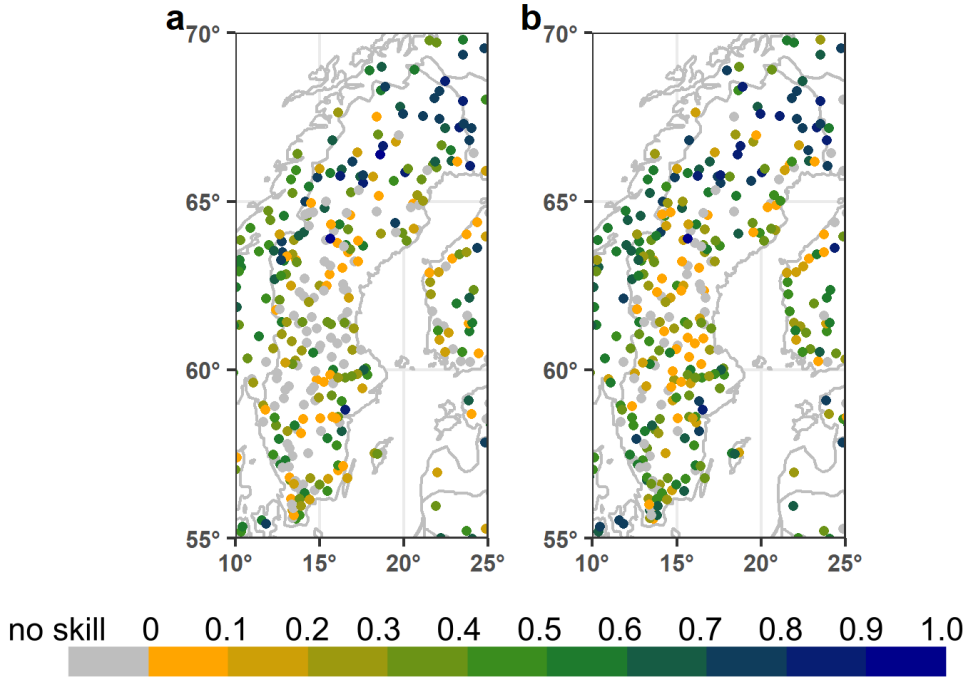

Figure 43 Improvement of post processing against the raw hydrological model (measured by skills): a, RF; b, LSTM.

2.9 Extraction of valuable information from multi-timescale forecasts via Reinforcement Learning

This method aims to jointly extract of the most valuable information for informing multipurpose reservoir operations from a set of multi-timescale forecast products and use this information in the design of improved operating policies. The approach is based on the Direct Policy Search method, a Reinforcement Learning (RL) approach where the operating policy, as well as the information extraction procedure, are parameterized; the search for their optimal parameters is performed by an optimization algorithm with respect to the operating objectives associated with the reservoir operation. The approach is demonstrated using the CLINT case study of Lake Como (Italy), where three diverse forecast products are available, including the short-term forecasts currently used by the lake operator and the sub-seasonal and seasonal reforecasts- produced by Copernicus EFAS. The novelty of our RL method lies in making a selection not based on forecast skill, but rather on forecast value. This allows us, for example, to potentially select a less skilful forecast over a longer lead time if this provides more valuable information for drought management than a short-term product characterized by a higher accuracy.

2.9.1 Methodology

Problem formulation

We consider a general multi-objective policy design problem of the following form (Castelletti et al., 2008):

$$p^* = argJ(p, s_0, q_{[1,H]}) \quad (7)$$

where ***J*** is the vector of operating objectives, s_0 the initial storage (given), and $q_{[1,H]}$ is the trajectory of the external drivers (e.g., reservoir inflows) over the evaluation horizon *H*.

The policy p is defined as a closed-loop control law that determines the daily release decision at each time step as a function of the year (d_t), the reservoir storage (s_t), and, potentially, a vector of forecast information (I_t), i.e.

$$u_t = p(d_t, s_t, I_t) \tag{8}$$

The dynamics of the reservoir system evolves according to a state transition function that describes reservoir storage variations using a mass balance equation

$$s_{t+1} = s_t + q_{t+1} - R(s_t, u_t, q_{t+1}) \tag{9}$$

where q_{t+1} is the net inflow entering the reservoir (that is, the inflow and precipitation minus evaporation and other losses) and *R()* the actual release, which is determined by the decision u_t. This release generally coincides with u_t corrected, when necessary, to respect physical and legal constraints specifying the minimum and maximum volume that can be released over the time interval *[t,t+1)*. In the adopted notation, the time subscript of a variable indicates the time instant when its value is deterministically known: the reservoir storage is measured at time t and thus is denoted as s_t, while the net inflow is denoted as q_{t+1} because it can be known only at the end of the time interval.

Finally, we assume the availability of a set of different forecast products ($\hat{Q}_t$). Formally, for each product ($\gamma \in \Gamma$) and ensemble member ($i \in [1, \dots, n_e^\gamma$), there is a vector of hydrological forecasts $\hat{q}_\tau^{\gamma,i}$, of size equal to the product's maximum lead time (LT^γ), predicting the net inflow from $q_{\tau+1}$ to $q_{\tau+1+LT^\gamma}$. In this notation, the subscript $\tau \in T^\gamma$ represents the time instant when the forecast has been issued, as each product has a different update frequency that is not necessarily synchronized with the daily simulation time step.

Joint learning of forecast information and operating policy

In this section, we illustrate the proposed RL approach (Figure 44) for the design of the optimal operations of a multipurpose reservoir leveraging the most valuable information (I_t) to be extracted from a set of a candidate forecast ($\hat{Q}_t$). Specifically, we introduce a generic parametric function representing the extraction of information from available forecasts:

$$I_t = F_\zeta(\hat{Q}_t) \tag{10}$$

This function can include the following operations:

- selection of the best forecast product ($\gamma^* \in \Gamma$)
- selection of the best effective forecast lead time ($\lambda^* \in LT^\gamma$), here also called Aggregation Time (*AT*) as forecasts are aggregated over it;
- selection of the best temporal aggregation operator of the forecasts over the selected lead time (ψ_γ^*);
- selection of the operator to deal with the forecast uncertainty ($\psi_{n_e}^*$, where n_e is the dimension of the forecast ensemble).

Moreover, an implicit operation is always performed to use only the most recent forecast between those available at time t.

The formulation of such a parametric information extraction function is then coupled with a Direct Policy Search (DPS) formulation of the operating policy design problem. DPS is based on the

parameterization of the operating policy (p_θ) within a given family of functions and the exploration of the parameter space ($\theta \in \Theta$) to find a parameterized policy that is optimal with respect to the operating objectives (Ruckstiess et al., 2010). Given the presence of multiple competing objectives, we used the Evolutionary Multi-Objective Direct Policy Search method (Giuliani et al., 2016) that allows an efficient search of the optimal parameters with respect to a multidimensional objective space.
Combining these two formulations, the daily release decision of Lake Como is now determined as

$$u_t = p_\theta(d_t, s_t, F_\zeta(\hat{Q}_t)) \tag{11}$$

The multi-objective optimal control problem introduced in the previous section can be then reformulated as finding the best parameters of an Extended Operating Policy (EOP) that will specify both forecast information extraction (ζ^*) and reservoir operation (θ^*):

$$[\zeta^*, \theta^*] = arg\ arg\ J \qquad s.t. \zeta \in Z, \theta \in \Theta \tag{12}$$

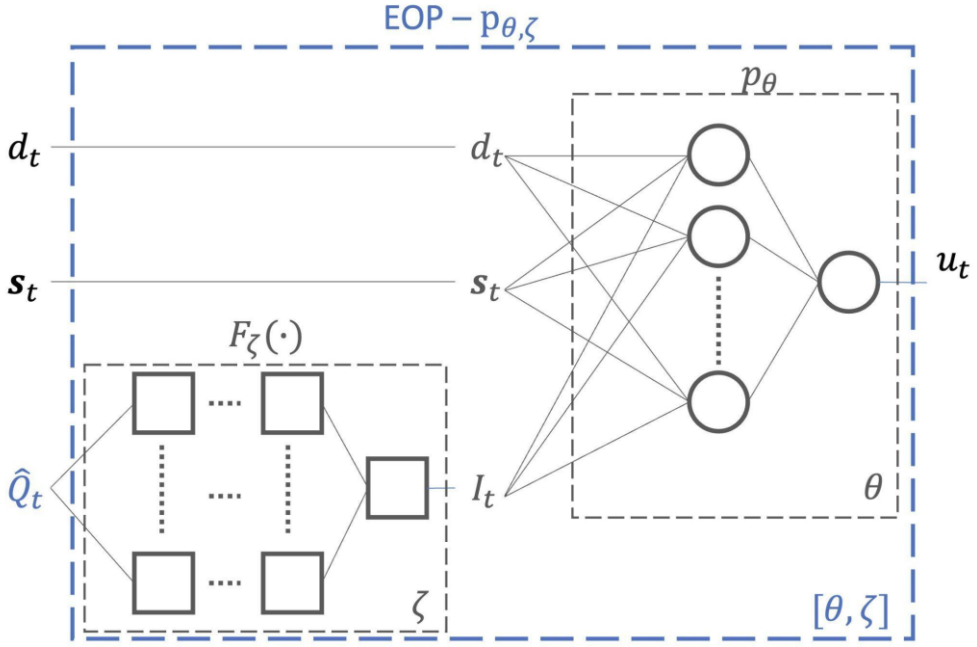

Figure 44 Internal structure of the Extended Operating Policy optimized by the joint learning of forecast information and operating policy. In the scheme, the circles represent the activation functions in the non-linear approximating network used to parameterize the operating policy, and the squares represent the operations that the EOP can perform on the forecasts (e.g., selection, temporal aggregation, or post-processing).

The multi-objective nature of this problem prevents us from using a single metric to describe the value of the selected forecast information. We instead consider multiple metrics (for a review, see Maier et al., 2004 and references therein) accounting for the convergence of the final solutions to the Pareto front associated with the POPs, the coverage of the non-dominated space (diversity), and the extent of the non-dominated front. Following the approach proposed in Giuliani et al. (2015), here we use the following three metrics

- the hypervolume indicator (HV), which captures both convergence and diversity. This metric allows for set-to-set evaluations, where the Pareto front with the higher HV is deemed the better.
- the minimum distance from a target solution (Dmin), which measures the proximity between the target POP solution and the closest point of the Pareto front under exam. Since Dmin is a point-to-point metric, achieving a good (small) value of Dmin requires only a single solution in the Pareto optimal set close to the target solution.
- the average distance (Davg), which measures the average distance of the entire Pareto front under exam from the target POP. The underlying idea is obtaining not only a single solution very close to the target POP but, rather, a set of solutions around the selected target so to explore the trade-offs between the competing objectives.

Benchmark Framework

The Information Selection and Assessment (ISA) framework (Giuliani et al., 2015) is used as a benchmark for the proposed RL approach. In the ISA framework, a set of Perfect Operating Policies are first designed via Deterministic Dynamic Programming (Bellman, 1957) assuming a deterministic knowledge of future inflows. These policies are used to estimate the Expected Value of Perfect Information, i.e., the value of completely removing uncertainty, by contrasting their performance and the performance of a set of poorly informed baseline solutions relying on a basic set of information (Basic Operating Policies). The gap between the BOPs and POPs reveals the performance improvement that could be achieved under the ideal assumption of perfect foresight of future system conditions, indicating the potential benefit of collecting accurate forecast information.

To identify which information can act as a surrogate of the sequence of future inflows used in the design of the POPs, we used the Iterative Input Selection (IIS) algorithm (Galelli and Castelletti, 2013) trying to model the optimal sequence of release decisions of a selected POP using Extremely Randomized Trees (Geurts et al., 2006). The candidate set of information for running the IIS algorithm is built from the forecast products in a pre-processing phase applying the same pre-processing function of the joint learning framework. However, this would result in a set with many candidate variables with redundant information that would confuse the IIS algorithm. Therefore, we selected only some values of the temporal aggregation hyperparameter (e.g., 1,3,5,7,14,21,… days).

Finally, once the IIS has selected the best forecast surrogating the perfect knowledge of future inflows, the final step of the ISA framework is the design of a set of Improved Operating Policies (IOPs) that use the selected forecast to inform operational decisions. Specifically, we use the same Evolutionary Multi-Objective Direct Policy Search method (Giuliani et al., 2016) that is implemented also in the joint learning framework.

2.9.2 Implementation

The methodological framework described in the previous section was tested on the Lake Como case study (Italy), aiming to improve the lake regulation in addressing challenges associated with both floods and droughts. Historically, two primary competing objectives have driven the lake regulation: (i) flood control to avoid flooding that affects Como and other populated areas on its shoreline, and (ii) water supply to satisfy the demand of downstream agricultural districts and run-of-the-river hydropower plants. Recently, the lake regulation is also considering the control of low lake levels that are detrimental to several users, including navigation, tourism, and the environment.

According to previous studies and interactions with the local stakeholders (see Deliverable D7.1), these objectives are formulated as follows:

- **Flood days**: the average annual number of days when the lake level (h_t) is above the threshold h^{flo}=1.1 m:

$$J^{flo} = \frac{1}{H/T}\sum_{t=0}^{H-1} g_{t+1}^{flo}; \qquad g_{t+1}^{flo} = \{1 \ if \ h_{t+1} > h^{flo} \ 0 \ otherwise \tag{13}$$

where H is the simulation horizon (days), and T is the annual period of the year (days).

- **Water supply deficit**: the daily mean deficit considering the water released from the lake (r_{t+1}) and the water demand of the downstream users (w_t):

$$J^{def} = \frac{1}{H}\sum_{t=0}^{H-1} g_{t+1}^{def}; \qquad g_{t+1}^{def} = [(w_t - (r_{t+1} - q^{MEF}), 0)]^{\beta_t} \tag{14}$$

where q^{MEF} = 22 m^3/s is the Minimum Environmental Flow constraint ensuring adequate environmental conditions in the Adda River, and β_t is a time-varying exponent that penalises with different importance the deficit during summer and winter. This parameter was tuned to mimic the decision-making preferences of the operator, with the deficit squared during the summer (1 April to 10 October), while the unitary value is taken during winter.

- **Low lake levels days**: the average annual number of days when the lake level (h_t) is below the threshold h^{low}=-0.2 m:

$$J^{low} = \frac{1}{H/T}\sum_{t=0}^{H-1} g_{t+1}^{low}; \qquad g_{t+1}^{low} = \{1 \ if \ h_{t+1} < h^{low} \ 0 \ otherwise \tag{15}$$

Given these three objectives, the policy design problem introduced in the previous section can be specialised for the Lake Como application as follows:

$$p^* = argJ(p) = |J^{flo}, J^{def}, J^{low}| \tag{16}$$

The design of the Extended Operating Policies includes an operating policy which is parameterized as a non-linear approximating network through a combination of Gaussian Radial Basis Function (Busoniu et al., 2011), while the information extraction function specifies the forecast product selection and its temporal aggregation. In this study, we instead fixed a priori the operators for the forecasts' temporal aggregation over the selected lead time and for its ensemble (uncertainty) processing using the average. The same RBF parameterisation is used for the definition of BOPs and IOPs in the ISA framework.

The search for the optimal parameters of the Extended Operating Policies (and also of the BOPs and IOPs of the ISA framework) is performed using the self-adaptive Borg Multi-Objective Evolutionary Algorithm (Hadka and Reed, 2013). This algorithm was shown to be effective in solving multi-objective optimal control problems, outperforming other state-of-the-art MOEAs (Zatarain Salazar et al., 2016). Each optimisation was run for 2 million function evaluations over the horizon 1999-2018, with 20 random initialisations (initial populations).

The multi-timescale forecasts available to inform the Lake Como operations are the following:

- Short-term deterministic forecasts (PRO), provided by the local company PROGEA and obtained by feeding a locally-calibrated hydrological model with short-term weather

forecasts from COSMO[1]. They are single trajectories with an hourly time step and update frequency, a lead time of up to 60 hours, and initially available between 2014 and 2022.

- Sub-seasonal probabilistic re-forecasts (EFRF) produced by Copernicus EFAS over the whole European domain by forcing the LISFLOOD (Knijff et al., 2010) hydrological model (uncalibrated for the Lake Como basin) with extended-range ensemble forecasts. These ensemble forecasts comprise 11 members with a 6-hour time step, a twice-weekly update frequency, a 46-day lead time, and availability over the period 1999-2018 (Barnard et al., 2020).
- Seasonal probabilistic re-forecasts (EFSR) are produced by EFAS, too. Similarly to the sub-seasonal product, these ensemble forecasts are obtained by LISFLOOD but are forced here with seasonal meteorological forecasts from the SEAS5 model. Their characteristics are 25 ensemble members, daily time step, issued on the first day of each month, up to 6 months lead time, and availability over in the period 1999-2019 (Wetterhall et al., 2020).

2.9.3 Benchmarking

Building on the forecast skill assessment reported in Deliverable D7.2, we perform a first experiment to verify the learning of the best Aggregation Time (AT) of a single fictitious seamless product (i.e., $\zeta = \lambda$). This is called the best-skill product and combines the 3 available forecast products by selecting the forecast with the best KGE score for each AT. This means using PROGEA for the first 3 days, EFAS EFRF between 4 and 42 days, and EFAS EFSR from 43 days onwards. The performance of the EOPs is benchmarked against a set of Basic Operating Policies (BOPs) not informed by any forecast and a set of Perfect Operating Policies (POPs) representing an upper-bound solution obtained by solving a deterministic problem with perfect knowledge on the future inflow trajectory. Results in Figure 45 show that the EOPs successfully improve the performance of the BOPs, especially for solutions with less than 5.5 flood days per year (top-left panel). The policy design consistently selects 3 days as AT, corresponding to using the PROGEA forecasts. This choice can be explained by the substantially higher accuracy of the PROGEA short-term forecasts with respect to the sub-seasonal and seasonal EFAS products. Using more skilful, shorter-term forecasts results in policies that outperform those using lower-skill, longer-term products. However, it is interesting to observe that the EOPs select the PROGEA forecasts at their maximum AT, although their KGE is lower than the one of PROGEA forecasts with 1 day as AT (0.806 vs 0.828).
Figure 45 also shows that it is possible to further improve the Water Deficit and Low-Level objectives by accepting more flood events. However, the forecast value decreases when moving to solutions with higher numbers of flood days because, in this case, the knowledge of future inflows is less critical for the lake operation, which can store water in favour of the other objectives without being limited by the increasing flood risk.

[1] https://www.cosmo-model.org

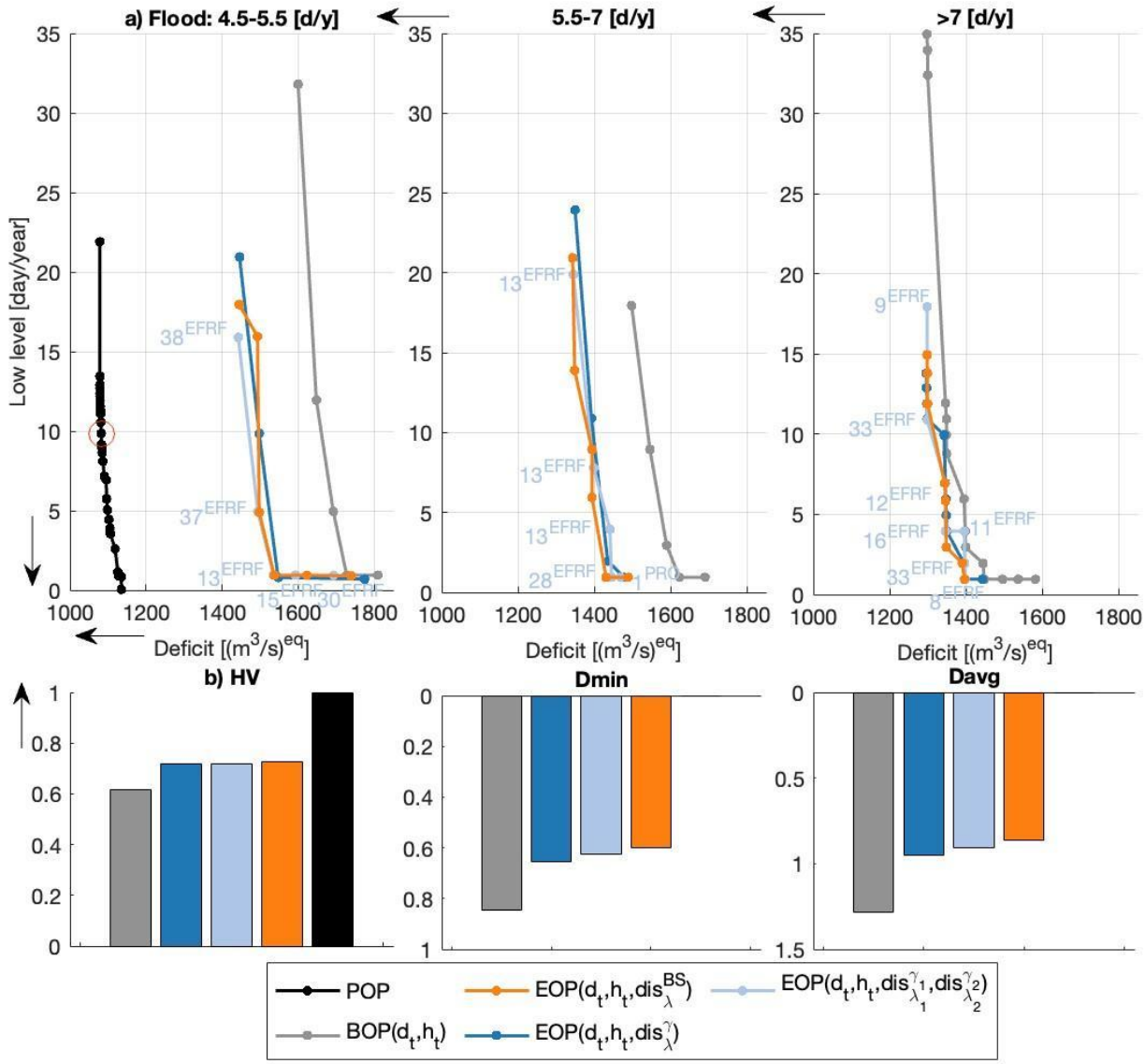

Figure 45 Performance of three sets of Extended Operating Policies informed by the best-skill product (orange), all products processed in one input (blue), and all products processed in two inputs (light blue). Panel (a) shows the solutions' performance in three projections of the objective space, grouped for different levels of flood days. The arrows indicate the direction of preference, with the best solutions located in the bottom-left corner of the leftmost plot. The numbers close to the light blue solution indicate the hyperparameters learned by the RL algorithm when extracting the forecast information from the second product (all other solutions use the PROGEA forecast with a 3 days AT). Panel (b) illustrates the forecast value quantified by the three metrics (hypervolume, minimum and average distance from a target solution) using the target POP marked by the red circle.

Given these promising results in learning the best AT, we run a second experiment in which the policy design simultaneously learns the best AT and the best forecast product (i.e., $\zeta = [\lambda, \gamma]$). The performance of the resulting EOPs (Figure 45) is very similar to the solutions informed by the best-skill product. This result is not surprising, as the EOP design selects again the PROGEA forecast at 3 days AT to inform the lake operation.

As a third experiment, we now solve the EOP design problem while selecting two different forecast products with their respective ATs (i.e., $\zeta = [\lambda_1, \gamma_1, \lambda_2, \gamma_2]$). In this case, the performance of the EOPs improves slightly from that of the EOPs informed by a single product but does not outperform the solutions informed by the best-skill product (Figure 45) especially in terms of Dmin and Davg.

However, it is worth analysing the selected combinations of forecast products and ATs: all solutions rely on the PROGEA forecasts aggregated over 3 days, along with medium- to extended-range information (i.e., AT between 9 and 38 days for the EFRF product). Notably, the EOPs attaining the

best performance in terms of flood control (top-left panel) and located in the compromise region of the Pareto front between Deficit and Low Level select an AT of 13 and 15 days, while the extreme solutions of the same Pareto front are associated to an AT of 38 and 30 days for the best Deficit and Best Low-Level solutions, respectively.

This result differs from the findings of previous studies (e.g., Denaro et al., 2017; Zaniolo et al., 2021) that used perfect forecasts, which selected much longer ATs. In addition to the influence of larger errors in real forecast products at longer lead times, this difference is due to the reduction of the lake's active capacity due to the subsidence of the city of Como and is in line with other studies that linked reservoir capacity with forecast horizon (e.g., Zhao et al. 2019; Turner et al., 2020). Yet, the added value of this second medium- to extended-range information is marginal, and the forecast value is dominated by the information provided by the PROGEA product.

To isolate the EFRF product's value and better understand their bias's impact, we run a final experiment allowing the algorithm to search only for the best AT of those forecasts (i.e., $\zeta = [\lambda; \gamma = EFRF]$). Results show that the performance of the resulting EOPs is lower than the one using the PROGEA forecasts (see Figure 46), confirming that our approach effectively learned the most valuable forecast information. Still, it is interesting to notice that the algorithm selects again an extended-range AT (i.e., 18 or 22 days), which is close to the 14 days that represent the maximum forecast accuracy in terms of KGE and correlation, despite the bias increasing with lead time.

A similar AT is also selected when running the EOP design with perfect forecasts (i.e., future observed inflows), with a maximum AT equal to the seasonal forecasts (i.e., 6 months). In this case, the preferred AT is between 20 and 25 days, and the resulting EOP performance is substantially improved compared to all other EOPs due to the more accurate (perfect) information used to condition the lake operation. The move towards longer selected ATs (20-25 days) with perfect forecasts with respect to those chosen using EFRF (18-22 days) suggests that with real forecasts, the RL algorithm finds a trade-off between more skilful shorter-range forecasts and, ideally, more informative longer-range ones.

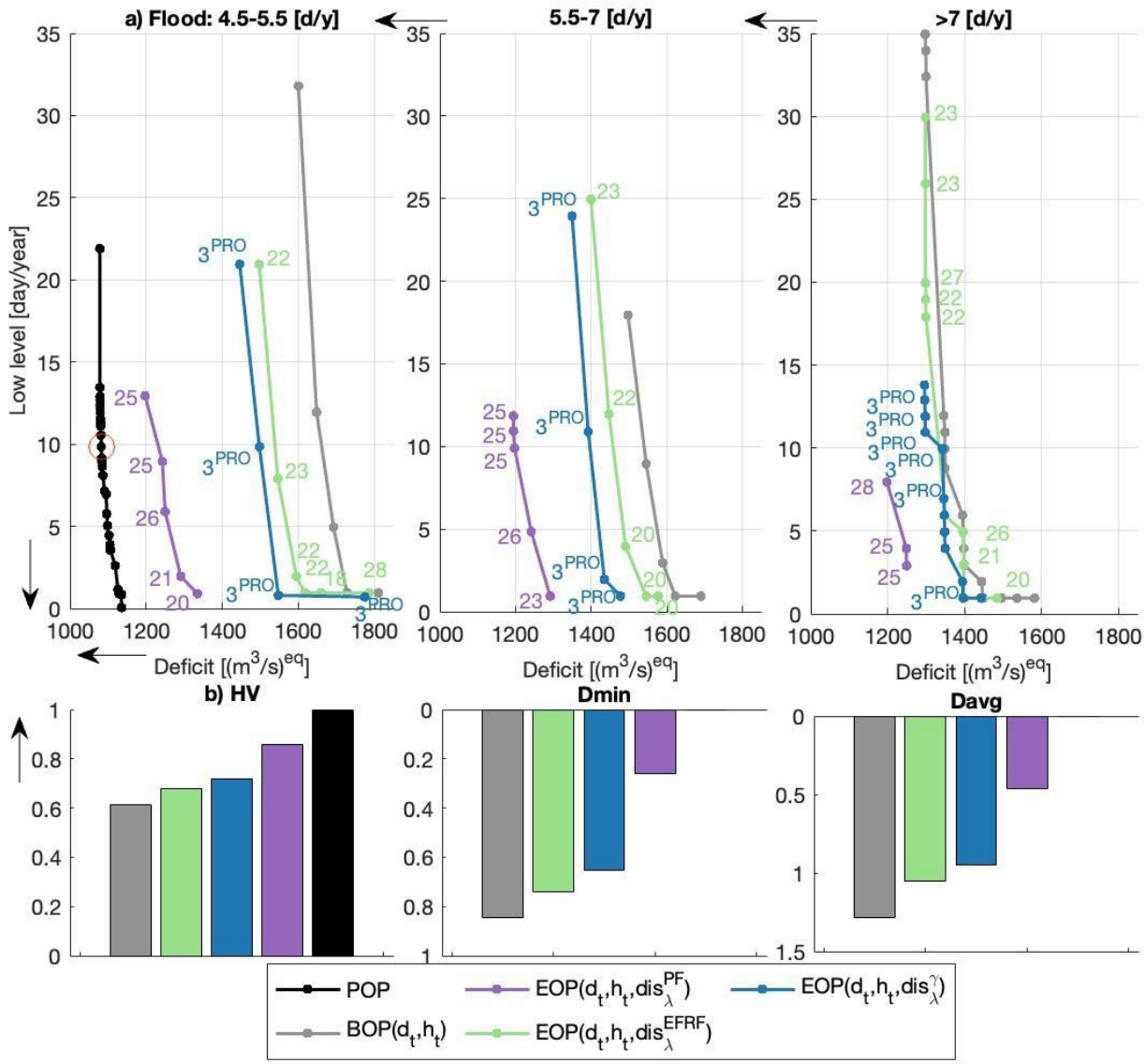

Figure 46 Performance of three sets of Extended Operating Policies informed by all products processed in one input (blue, as in Figure 45), perf forecasts (purple), EFRF forecasts at their best AT (green). Panel (a) shows the solutions' performance in three projections of the objective space, grouped for different levels of flood days. The arrows indicate the direction of preference, with the best solutions located in the bottom-left corner of the leftmost plot. The numbers close to the solutions indicate the hyperparameters learned by the RL algorithm when extracting the forecast information. Panel (b) illustrates the forecast value quantified by the three metrics (hypervolume, minimum and average distance from a target solution) using the target POP marked by the red circle.

Lastly, we run the ISA framework to design a set of Improved Operating Policies that use the forecast information selected by the IIS algorithm; these solutions are then contrasted against the EOP to validate the potential of our method. Interestingly, the IIS algorithm selects the extended-range forecast information with AT equal to 14 days or 21 days for EFRF or perfect forecasts, thus confirming the outputs of the EOP design. Results indicate that the policies based on perfect forecasts and designed using the information selected through IIS are almost as effective as the EOPs (purple solutions in Figure 47). This is not surprising because ISA-selected ATs are similar to those identified by the joint learning method. However, when dealing with real forecasts, the IIS prefers the EFRF to the PROGEA product, although the latter results are more valuable for informing the operating policy with the EOP solutions outperforming the IOPs (green and blue solutions in Figure 47). These findings demonstrate the added value of jointly learning the forecast information and operating policy with respect to the ISA framework that separately selects the information to be used in the subsequent policy design problem.

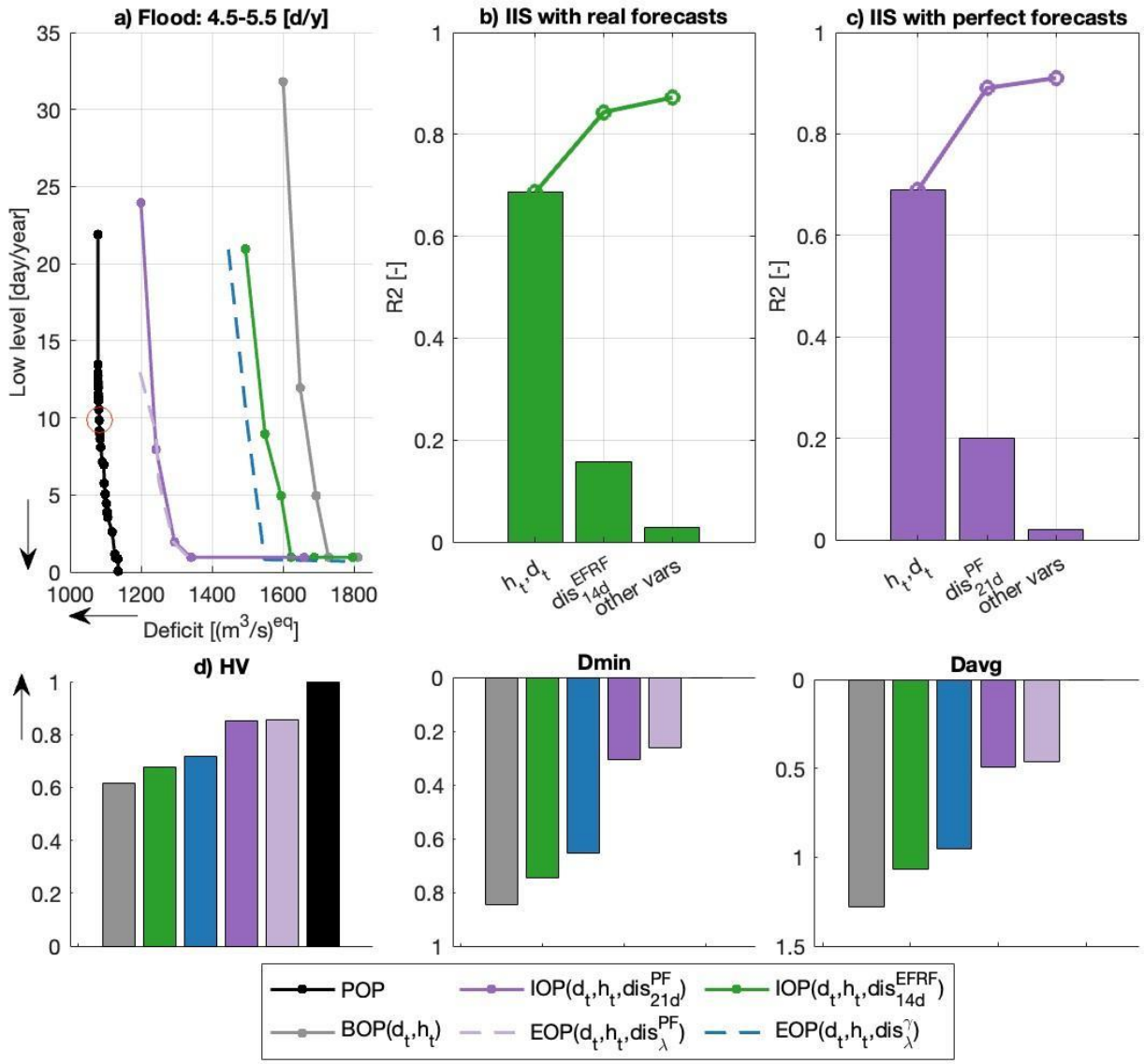

Figure 47 Performance of two sets of Extended Operating Policies - designed via joint learning of forecast information and operating policy - and Improved Operating Policies – designed with the ISA framework. The two set of solutions rely either on perfect or real forecast products. Panel (a) shows the solutions' performance in three projections of the objective space, grouped for different levels of flood days. The arrows indicate the direction of preference, with the best solutions located in the bottom-left corner of the leftmost plot. Panel (b) illustrates the forecast value quantified by the three metrics (hypervolume, minimum and average distance from a target solution) using the target POP marked by the red circle.

3 EXTREME EVENTS RECONSTRUCTION WITH MACHINE LEARNING

1.1 Methodology

The method used for the reconstruction of EE datasets is a deep-learning-based inpainting technique. It makes use of the U-Net model and was developed initially in the field of Computer vision (Pathak et al. 2016).

U-Net (Ronneberger et al. 2015) is a convolutional neural network (CNN) fed with spatial data (traditionally RGB images). As shown in Figure 48 it consists of two paths:

- a path which contracts the spatial information of the input data (downsampling), while increasing the feature information, through a series of convolutional layers
- a subsequent path that expands back the spatial information (upsampling) by applying a combination of convolutional filters and interpolations until reaching the original spatial resolution of the input data

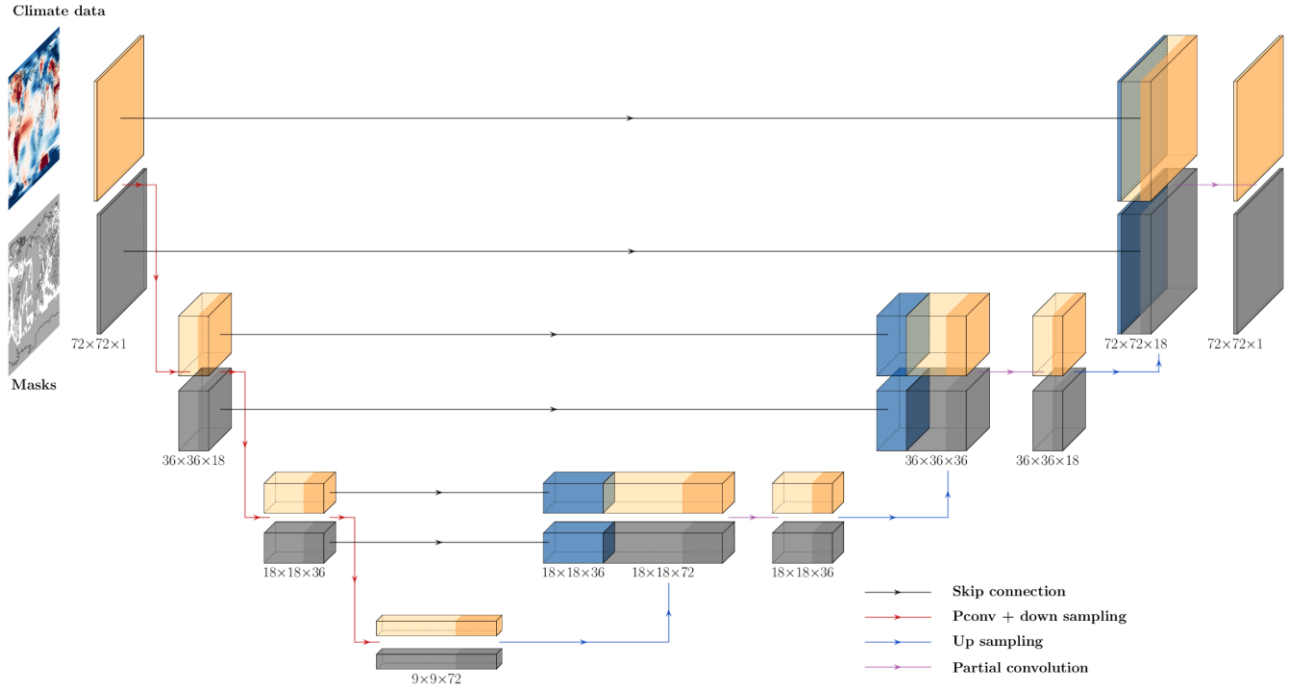

Figure 48 *Architecture of the U-Net used for the reconstruction of the HadCRUT5 dataset*

Unlike regular autoencoders, the two paths of the U-Net are not fully decoupled, as skip connections transfer information from the layers in the left branch to the corresponding ones in the right branch. Skip connections are used here to recover fine-grained details in the prediction and alleviate the vanishing gradients problem.

U-Net is particularly suitable to capture high-level semantic features from the input data and is commonly used for image segmentation (Helhamer et al. 2017). As shown in (Pathak et al. 2016), it can be applied to image inpainting by masking the input data to create artificial missing values. By minimising the difference between the predictions and the original input data (without missing values), it is then possible to train an inpainting model.

When using masked input data with deep learning-based methods, initial values of the missing values (typically by using the mean value of the training dataset) must be defined. Because they are part of the input data, these placeholder values are propagated through the network and lead to artefacts in the output (e.g., lack of texture in the regions of missing values). To overcome the effects

of this conditioning, several studies have implemented sophisticated techniques that require expensive post-processing, e.g., by using a second-stage refinement NN (Song et al. 2017, Yu et al. 2018). Although these techniques yield improved outputs, they fall short of completely eliminating the artefacts.

An alternative and efficient approach has been suggested by (Liu et al. 2018), where the authors overcame the issues related to the conditioning of the missing values by replacing the standard convolutional layers with partial convolutional layers. As shown in Figure 49, the partial convolution layer is made of two parts:

- a partial convolution operation, in which the kernel is applied to a subset of the input data if this subset contains at least one valid value (denoted by the value 1 in the mask). Otherwise, zero is returned
- a mask update, in which a new mask is created that contains the values 1 for the subsets with at least one valid value, zeros otherwise.

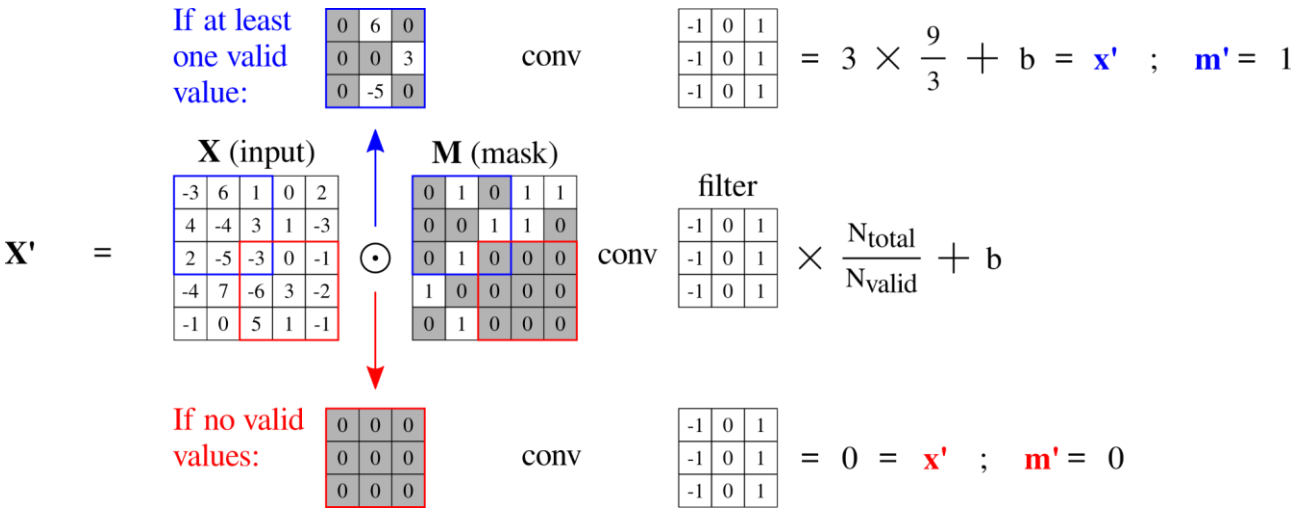

Figure 49 Illustration of the convolution operation and mask update in a partial convolutional layer.

The successive applications of the partial convolutional layers lead to the progressive infilling of the missing values. If the input data contains enough valid values and the network has enough layers, the masks will only contain ones at the end of the forward propagation, and the output only valid values.

In addition to the resolution of the conditioning problem, this technique also has the advantage of being relatively fast and having the capacity to reconstruct large and irregular regions of missing values. A further improvement to the original U-Net is the possibility to use a state-of-the-art loss function (called “inpainting loss”) composed of five terms:

- a hole term L_{hole}, computing the mean absolute error between the original masked values and the corresponding predicted values
- a valid term L_{valid}, computing the mean absolute error between the original valid values and the corresponding predicted values
- a total variation term L_{TV}, computing the mean absolute error between the original valid values and the predicted missing values at the border of the regions of missing values. This term ensures a smooth transition between the infilled regions and the regions of valid values
- a perceptual term $L_{perceptual}$ computing the mean absolute error between the inputs and outputs data projected into higher level feature spaces using a pretrained VGG-16 network. This term reduces some artefacts in the output such as the grid-shaped artefacts

- a style term L_{style}, computing the mean absolute error between the autocorrelation of the input and output feature maps calculated for the perceptual loss. This term accounts for realistic results by learning high-level features.

The total loss function L_{total} is the weighted sum of these terms:

$$L_{total} = 6L_{hole} + L_{valid} + 0.1L_{TV} + 0.05L_{perceptual} + 120L_{style} \quad (17)$$

The weighting coefficients have been determined via hyperparameter tuning.

Depending on the task, it is possible to opt for the inpainting loss or a more standard loss function such as the Mean Absolute Error (MAE) loss. The choice of the loss function mostly depends on whether pixel-level accuracy (MAE loss) or physical realism (inpainting loss) is more relevant for the task.

U-Nets closely resemble autoencoders and share, by extension, many characteristics with variational autoencoders (VAEs, Kingma et al. 2022). Consequently, it is feasible to adapt a U-Net code with minimal effort to utilise it for training generative models. VAEs extend the traditional autoencoder framework by introducing a probabilistic approach to the encoding process. Contrarily to U-Nets or autoencoders that encode an input as a single point in the latent space, the VAE encoders map it to a probability distribution in the latent space, typically a Gaussian distribution. By sampling this distribution using the so-called "reparameterisation trick", it is possible to generate diverse reconstructions during the decoding phase. This generative capability can be used to quantify part of the uncertainties associated with the reconstruction of the input data. It can potentially be used as well in Task 5.2 as an anomaly detector to perform attribution of trends in extreme events.

1.2 Implementation

The partial convolutional inpainting method described in the previous section has been implemented as a Python code using the PyTorch framework (https://pytorch.org/).

The basic structure of the code was taken from an existing repository (under MIT licence) (https://github.com/naoto0804/pytorch-inpainting-with-partial-conv) and adapted for the specific tasks of the CLINT project (essentially Task 3.5, 5.2 and 8.4).

The code is separated in two parts: the training of the model and the evaluation (or reconstruction) of the observational dataset. These two parts share many functionalities (such as the data loading, the definition of the model structure, etc.) but can be executed separately, even in different computing systems. In both cases, the input data must contain:

- multiple timesteps of the spatial field (in a uniform longitude/latitude grid) corresponding to the climate variable to be reconstructed
- one or several masks representing the missing values of the observational dataset.

The training process requires considerable computational resources and is generally performed on a HPC using GPUs. The input fields are taken from simulation models or reanalysis datasets which contain only valid values. The dataset is split into training, validation and test sets to diagnose problems and evaluate the trained model. A standard split distribution is shown in Figure 50.

Figure 50 Example of the split of an input dataset into a training, a validation and a test set.

For a better result, the training dataset should contain a large number of timesteps and possibly multiple ensemble members.
During the training, the training and the validation data are further split into batches (typically with a batch size of 16) and are fed to the U-Net with randomly selected masks. For each iteration, a batch of samples from the training set is propagated through the network, the loss function is calculated, and the gradients are obtained by backpropagation. The weights and biases are then updated by using stochastic gradient descent. The optimal model can be assessed by looking at the learning curves, i.e., the evolution of the training and validation losses with respect to the number of iterations. To avoid overfitting, one typically selects the model that achieves the lowest validation loss (as shown in Figure 51).

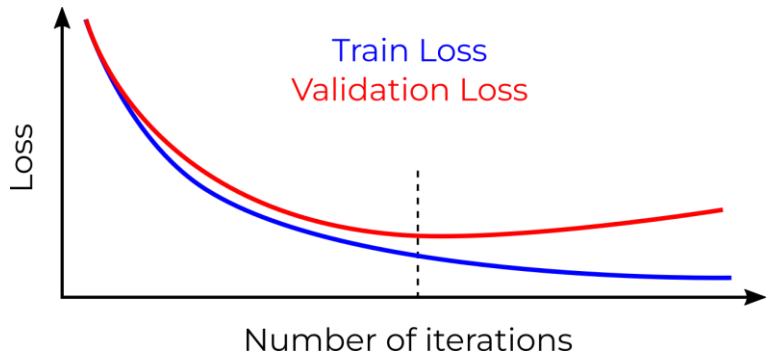

Figure 51 Illustration of the evolution of the training and validation losses as a function of the number of iterations. The vertical dashed line denotes the iteration at which the model begins to overfit the training data

In addition to the partial convolutional layers described previously, it is also common to use batch normalisation that alleviates the so-called “internal covariate shift” and speeds up the training.
The evaluation process requires much less computational resources compared to the training process and can run even on a conventional laptop, typically in a few minutes. This is a major advantage of the method because the evaluation process can be deployed on a virtual machine (as for the Web Processing Service (WPS) prototype created in Task 8.4). The evaluation process makes use of the trained model and takes the observational dataset as input. The masks of missing values can be given as a separate input or extracted automatically from the input observational data as long as the missing values are encoded as not a number (NaNs).
Numerous developments were required to reconstruct the missing values in the datasets of interest for the detection, causation, and attribution of EE. In total, more than 300 commits were pushed to the repository (https://github.com/FREVA-CLINT/climatereconstructionAI). A summary of the main modifications are listed here:

- Refactoring of the code into a Python package for better integration with the WPS deployed in Task 8.4 and easier dissemination within the scientific community (WP9)
- Cleaning of the code for better readability and compliance with PEP8 style conventions
- Creation of continuous integration workflows to test the main features of the code and PEP8 compliance
- Extension of the documentation regarding the installation process and usage
- Creation of a demo example to illustrate the evaluation process and to check the installation
- Implementation of an I/O interface using the xarray library to facilitate the reading/writing of climate data from/to netCDF (Rew et al., 1990) files (with the ability to conserve the attributes and add comments to the history attribute)
- Creation of a module to check the format of the netCDF file when using a predefined type of datasets
- Computation of the validation loss function to diagnose training issues such as overfitting

- Introduction of a learning rate scheduler that automatically reduces the learning rate parameter based on the gradient of the validation loss
- Implementation of multi-GPUs parallelization by means of the data parallelism module from PyTorch. In this approach, the batches are split into sub-batches which are distributed to the various GPUs and then fed separately to the NN. The outputs of the multi-forward propagation is then aggregated on the primary GPU
- Implementation of a custom padding that fulfils the spherical boundary conditions of global datasets (such as HadEX3 in Task 5.2). This custom padding makes use of a circular padding on the vertical edges of the data and a zero padding on the horizontal ones.
- Creation of an option to load steady masks, i.e., masks which determine the spatial region where the loss function should be calculated. It is particularly useful when dealing with land-based datasets (such as HadEX3 in Task 5.2) for which it is only required to infill the inland regions
- Creation of an option to normalize automatically the data before the training/evaluation by using the mean and the standard deviation of the training dataset
- Implementation of a monitoring function to show the progress status during the evaluation with the deployed WPS (Task 8.4)
- Improvement of exceptions handling
- Implementation of a multi-models reading/writing feature to perform model ensemble calculations and estimate the uncertainties
- Integration of a profiler to check the performance of the code on different systems
- Implementation of a feature to bind the predictions to a user-defined range of values (necessary to reconstruct some variables such as the percentile variables of HadEX3 in Task 5.2)
- Development of features to use multiple input channels (necessary for multivariate reconstruction), calculate the loss on multiple output channels and infill multiple channels
- Improvement of the memory usage in the data loading module and in the writing module of the evaluation step that gives the ability to use datasets which size is larger than the RAM of the system.
- Contribution to the development of a hyperparameter tuning scheme integrated with TensorBoard for the efficient selection of optimal hyperparameter configurations
- Concatenation of multi-model predictions into an extra xarray coordinate
- Extension of the code to enable the utilisation of a variational autoencoder as an alternative to the U-Net.

1.3 Benchmarking

To test our algorithm and the numerous implementations done in the framework of the CLINT project, we have used a simple test case: the reconstruction of the HadCRUT5 dataset (https://www.metoffice.gov.uk/hadobs/hadcrut5) using the historical monthly resolved MPI Earth System Model (Low Resolution) dataset from the CMIP6 archive (https://esgf-data.dkrz.de). This dataset contains global data in a 5º longitude/latitude grid that span the 1850-2014 period.
For this benchmark, a dataset of near-surface temperature anomalies was created and split into a training (47.520 samples), a validation (9.900 samples), and a test (1.980 samples) set.
The three sets were created by randomly selecting one or several members from the original MPI-ESM dataset for each date: 1 member/month for the test set, 5/month for the validation set, and 24/month for the training set. Great care was taken to ensure that there were no duplicate samples across the three sets.
2,076 masks corresponding to the missing values of the HadCRUT5 dataset were extracted and used as input for the training. The training process lasted 1 million iterations using a global padding, a

batch size of 16 and a learning rate of 2e-4. After 500,000 iterations, the learning rate was reduced to 5e-5 and the batch normalisation deactivated. The main parameters regarding the architecture of the network are given in Figure 48.

Twenty models were trained in order to compute the uncertainties related to the initial conditions (weight initialisation and combinations of masks/temperatures). These trained models were used to reconstruct the test dataset in which missing values had been artificially created by using the same dataset of masks used during the training. Some randomly selected examples of reconstruction for January 1850, July 1862, and March 1873 are shown in Figure 52 to Figure 54. In all cases, the trained models lead to good qualitative reconstruction when compared to the ground truth (original test set).

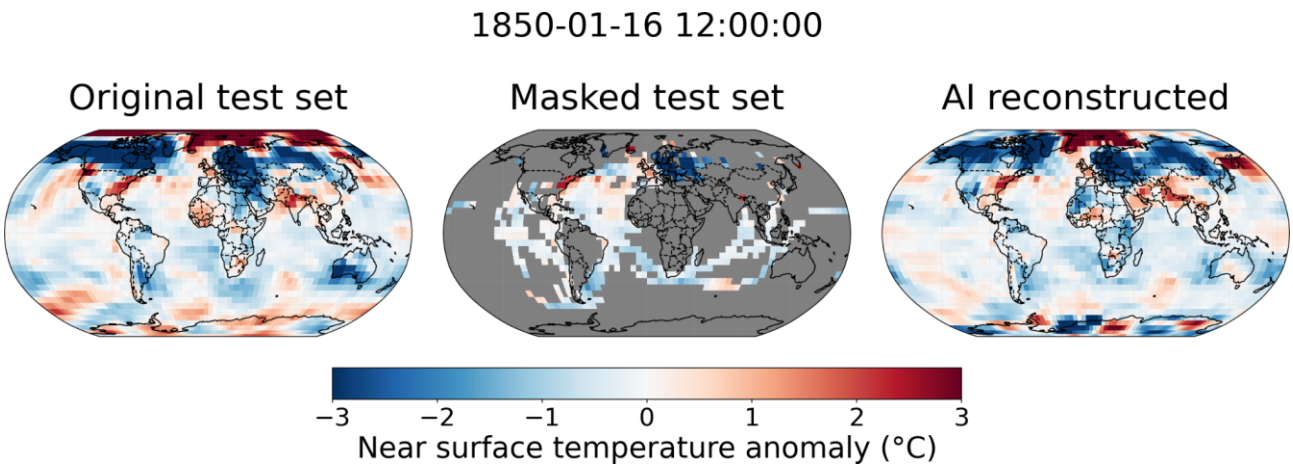

Figure 52 Original, masked and reconstructed test set for one of the trained models for January 1850.

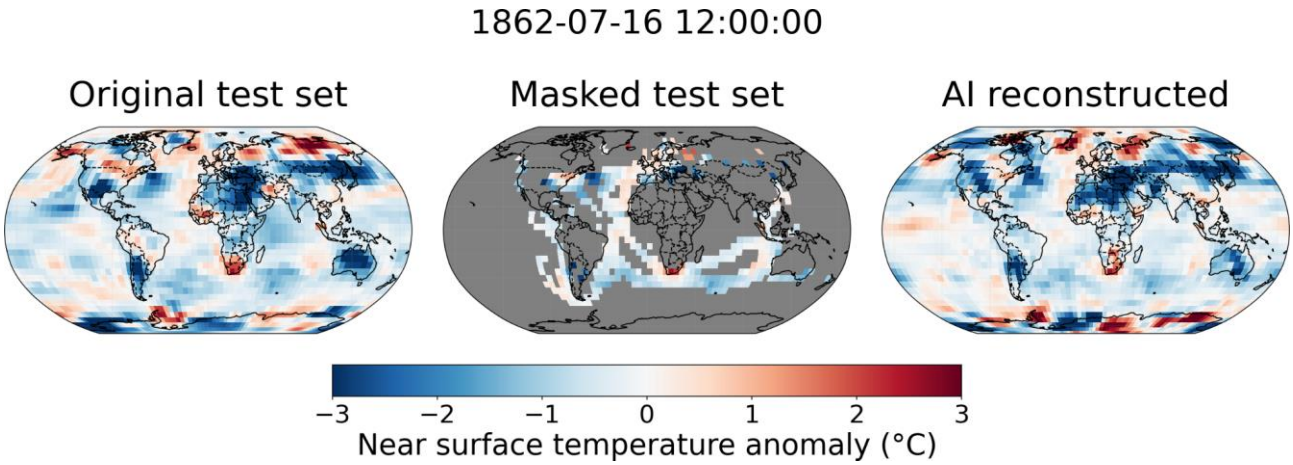

Figure 53 Original, masked and reconstructed test set for one of the trained models for July 1862.

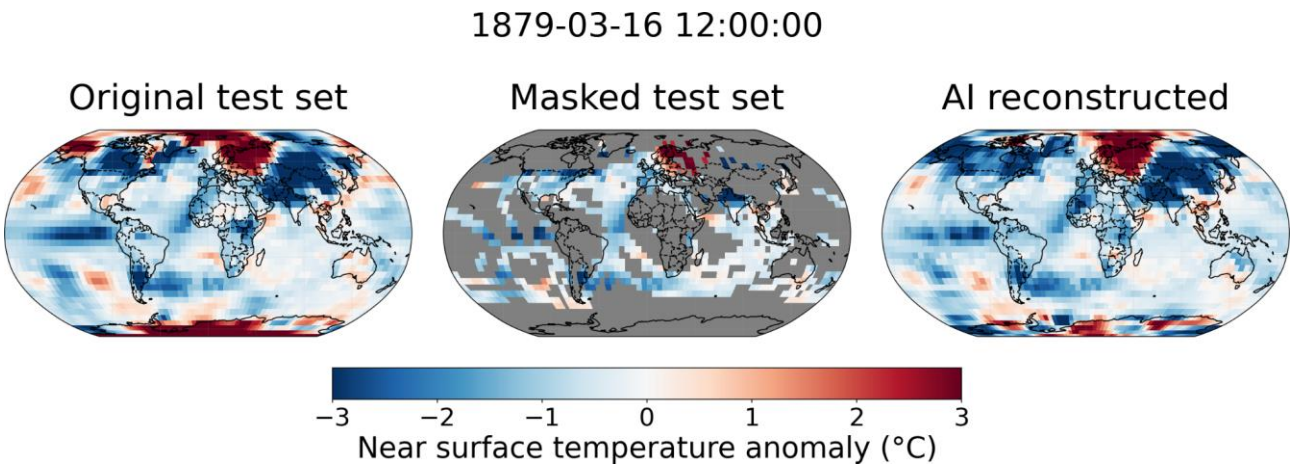

Figure 54 Original, masked and reconstructed test set for one of the trained models for March 1879.

The variations between the different models can be observed by calculating the annual global mean of the reconstructed data. The corresponding min/max spread of the ensemble is plotted in pale red in Figure 55, together with the ensemble mean (red), the original test set annual global mean (black), and the masked version (blue).

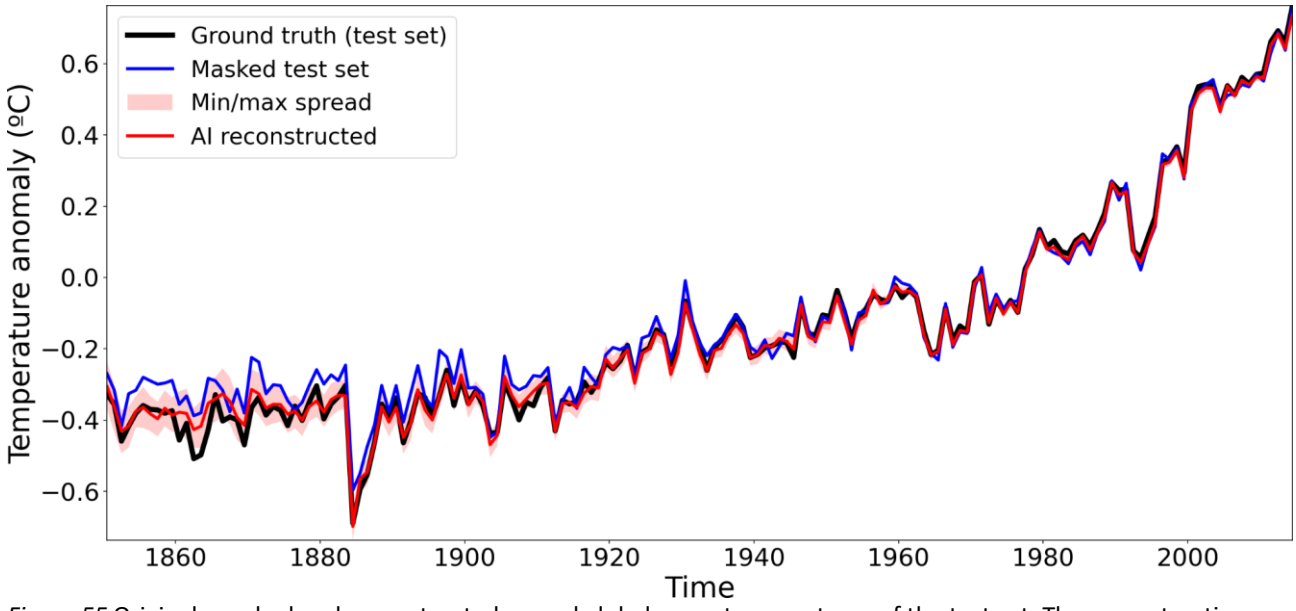

Figure 55 Original, masked and reconstructed annual global mean temperatures of the test set. The reconstructions were obtained by training twenty models using the same architecture (U-Net) and hyperparameters.

Similar results can be obtained by training a single VAE with the same hyperparameters and sampling the latent distribution twenty times for each input sample of the test set. As shown in Figure 56, the resulting annual global mean of the reconstructed data is similar to the previous result shown in Figure 55. The min/max spread, however, is slightly reduced, suggesting that the VAE may not encompass all associated uncertainties.

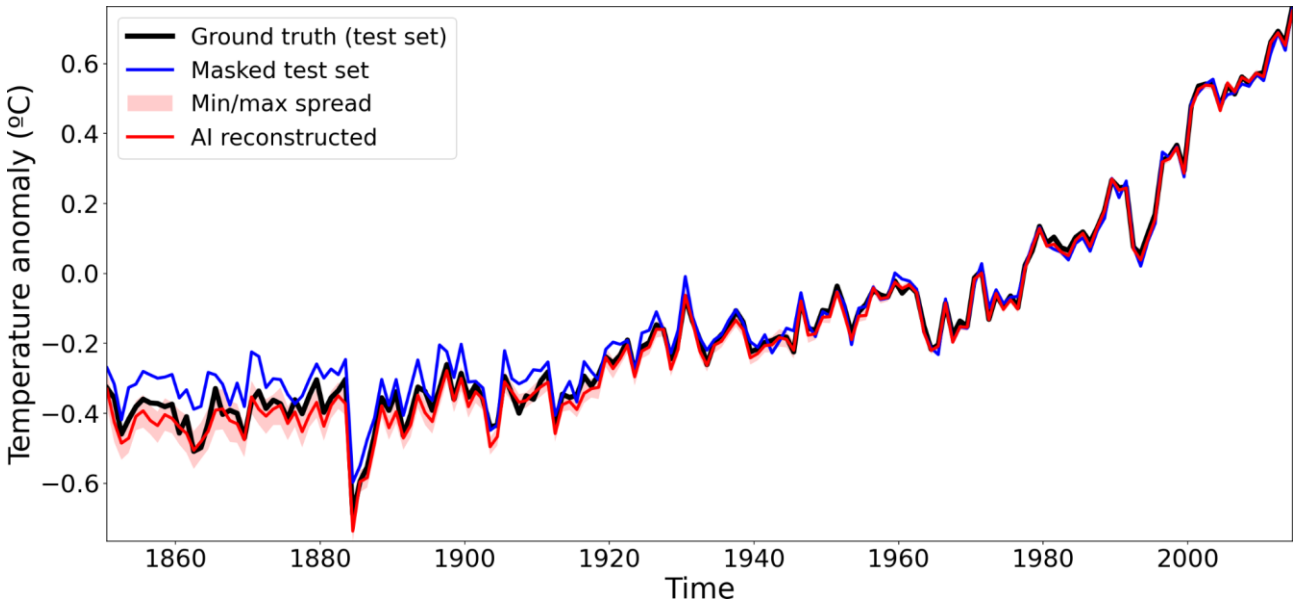

Figure 56 Original, masked and reconstructed annual global mean temperatures of the test set. The reconstructions were obtained by training a VAE with the same hyperparameters as the U-Net and sampling the latent distribution twenty times.

As observed in Figure 55 and Figure 56, the reconstructed ensemble mean and the ground truth are in good agreement for both reconstructions. To quantify this agreement, it is convenient to apply some evaluation metrics, such as the Root Mean Square Error (RMSE) and the temporal correlation on the annual global mean, using the complete test set as the ground truth. As shown in Table 13 the reconstructed temperatures obtained with our method lead to better results for all metrics compared to the ones obtained using the IWD method. Interestingly, a better RMSE is achieved by using the U-Net while the VAE gives a slightly better correlation value.

Table 13 Performance metrics for reconstructed temperature.

Evaluation metric	IWD	U-Net	VAE
RMSE	0.03	0.02	0.02
Correlation	0.99	0.99	0.99

4 CONCLUSIONS

This deliverable presented the methods developed by WP2 to forecast and reconstruct EE with Machine Learning techniques. Fully data-driven and hybrid approaches based on state-of-the-art or newly developed algorithms have been developed to address the forecast and reconstruction of droughts, tropical cyclones, heatwaves and warm nights, and compound events. These methods address EE either directly, i.e. forecasting the occurrence of an event (e.g. tropical cyclone rainfall enhancement), or indirectly, i.e. forecasting and reconstructing the hydrological or atmospheric variable that is then used to detect EE with specific indicators (e.g. reconstruction of temperature anomalies, post-processing of streamflow). Different lead-times have been addressed as well, ranging from short-term to seasonal horizons.

More specifically, this report presented a method to forecast total precipitation in a specific location, which leveraged on detailed climate information. Meteorological drought can then be forecasted using common indicators, such as SPI.

Three different methods have been presented to predict streamflow, one of which is purely data-driven and two of which are hybrid, i.e. use observational data and model predictions. Among these, higher potential is provided by the post-processing framework and the direct forecasting in large river basins, while challenges were found for a smaller basin. All these approaches can be further employed to forecast both EE drought and flood events. To predict these, an approach based on reinforcement learning has been developed as well, leveraging on the value of information contained in existing forecasts to make better decisions for reservoir management.

Two methods to forecast tropical cyclones were presented as well, one hybrid, to post-process forecasted rainfall, and one fully data-driven, to forecast the occurrence of the event. In contrast to hydrological flow predictions, for tropical cyclones both methods performed well at the short-term lead-times, and both can match or even outperform existing NWP forecasts.

To improve heatwaves forecasting, a method to improve drivers detection has been developed. Once the heatwave drivers are found, they can be further used to predict heatwave occurrence in a specific location. Moreover, heatwaves and droughts have been further analysed in the case of concurrent events, for which an approach to forecast the probability of their occurrence has been developed employing AI, also identifying relevant drivers and indicators.

Finally, a method to reconstruct past extreme events has been developed, leveraging on computer vision techniques. The results of this method have been shown here to reconstruct missing temperature anomalies from the 19th century onwards, and when tested, being able to reproduce values very close to the ground truth.

Some of the methods here presented have already been tested or are currently under testing in some of the CLINT climate change hotspots (e.g. meteorological drought forecasting in Rijnland, tropical cyclone rainfall forecasting in the Zambezi River Basin) and at the pan-European scale (e.g. post-processing for hydrological prediction), while for others the application is planned for the months M36-M46. Moreover, plans are already in place to apply some of the methods here presented to other climate change hotspots (e.g. meteorological drought forecasting in Douro river basin) or at the pan-European scale (e.g. hydrological forecasting with LSTM). The outcomes of these further applications will be shown in a later stage in deliverables D6.3 and D7.3.

Moreover, some of the approaches shown in this deliverable can be used or have already been employed with some modifications to detect EE. Therefore, further results will be shown in D3.3.

References

Apley, D. W., & Zhu, J. (2020). Visualizing the effects of predictor variables in black box supervised learning models. Journal of the Royal Statistical Society Series B: Statistical Methodology, 82(4), 1059-1086.

Ardilouze, C., Specq, D., Batté, L., & Cassou, C. (2021). Flow dependence of wintertime subseasonal prediction skill over Europe. Weather and Climate Dynamics, 2(4), 1033-1049.

Arsenault, R., Martel, J. L., Brunet, F., Brissette, F., & Mai, J. (2023). Continuous streamflow prediction in ungauged basins: long short-term memory neural networks clearly outperform traditional hydrological models. Hydrology and Earth System Sciences, 27(1), 139-157.

Ascenso, G., Ficchì, A., Giuliani, M., Scoccimarro, E., Castelletti, A. (under review). Downscaling, Bias Correction, and Spatial Adjustment of Extreme Tropical Cyclone Rainfall in ERA5 Using Deep Learning. Submitted to Weather and Climate Extremes.

Bakır, G. H., Weston, J., & Schölkopf, B. (2004). Learning to find pre-images. Advances in neural information processing systems, 16, 449-456.

Baldi, P. (2012, June). Autoencoders, unsupervised learning, and deep architectures. In Proceedings of ICML workshop on unsupervised and transfer learning (pp. 37-49). JMLR Workshop and Conference Proceedings.

Barnard, C., Krzeminski, B., Mazzetti, C., Decremer, D., Carton de Wiart, C., Harrigan, S., ... & Prudhomme, C. (2020). Reforecasts of river discharge and related data by the European Flood Awareness System, version 4.0. Copernicus Climate Change Service (C3S) Climate Data Store (CDS).(4th March 2021), 10.

Beck, H. E., Pan, M., Roy, T., Weedon, G. P., Pappenberger, F., Van Dijk, A. I., ... & Wood, E. F. (2019). Daily evaluation of 26 precipitation datasets using Stage-IV gauge-radar data for the CONUS. Hydrology and Earth System Sciences, 23(1), 207-224.

Beck, H. E., Wood, E. F., Pan, M., Fisher, C. K., Miralles, D. G., Van Dijk, A. I., ... & Adler, R. F. (2019). MSWEP V2 global 3-hourly 0.1 precipitation: methodology and quantitative assessment. Bulletin of the American Meteorological Society, 100(3), 473-500.

Beckers, J. M., & Rixen, M. (2003). EOF calculations and data filling from incomplete oceanographic datasets. Journal of Atmospheric and oceanic technology, 20(12), 1839-1856.

Bellman, R. (1957). Dynamic programming. Princeton University Press.

Bi, K., Xie, L., Zhang, H., Chen, X., Gu, X., & Tian, Q. (2023). Accurate medium-range global weather forecasting with 3D neural networks. Nature, 619(7970), 533-538.

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. Monthly weather review, 78(1), 1-3.

Buizza, R., & Leutbecher, M. (2015). The forecast skill horizon. Quarterly Journal of the Royal Meteorological Society, 141(693), 3366-3382.

Buontempo, C., Burgess, S. N., Dee, D., Pinty, B., Thépaut, J. N., Rixen, M., ... & de Marcilla, J. G. (2022). The Copernicus climate change service: climate science in action. Bulletin of the American Meteorological Society, 103(12), E2669-E2687.

Busoniu, L., Ernst, D., De Schutter, B., & Babuska, R. (2010). Cross-entropy optimization of control policies with adaptive basis functions. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 41(1), 196-209.

Cai, Z., Li, R., & Zhang, Y. (2022). A distribution free conditional independence test with applications to causal discovery. Journal of Machine Learning Research, 23(85), 1-41.

Castelletti, A., Pianosi, F., & Soncini-Sessa, R. (2008). Water reservoir control under economic, social and environmental constraints. Automatica, 44(6), 1595-1607.

Ceglar, A., & Toreti, A. (2021). Seasonal climate forecast can inform the European agricultural sector well in advance of harvesting. Npj Climate and Atmospheric Science, 4(1), 42.

Ceglar, A., Toreti, A., Zampieri, M., Manstretta, V., Bettati, T., & Bratu, M. (2020). Clisagri: An R package for agro-climate services. Climate services, 20, 100197.

Chantry, M. (2023). Retrieved from https://www.ecmwf.int/en/about/media-centre/science-blog/2023/rise-machine-learning-weather-forecasting.

Chattopadhyay, A., Nabizadeh, E., & Hassanzadeh, P. (2020). Analog forecasting of extreme-causing weather patterns using deep learning. Journal of Advances in Modeling Earth Systems, 12(2), e2019MS001958.

Chaudhuri, A., Kakde, D., Sadek, C., Gonzalez, L., & Kong, S. (2017, November). The mean and median criteria for kernel bandwidth selection for support vector data description. In 2017 IEEE International Conference on Data Mining Workshops (ICDMW) (pp. 842-849). IEEE.

Chawla, N. V. (2010). Data mining for imbalanced datasets: An overview. Data mining and knowledge discovery handbook, 875-886.

Cheung, K., Yu, Z., Elsberry, R. L., Bell, M., Jiang, H., Lee, T. C., ... & Tsuboki, K. (2018). Recent advances in research and forecasting of tropical cyclone rainfall. Tropical Cyclone Research and Review, 7(2), 106-127.

Chevuturi, A., Tanguy, M., Facer-Childs, K., Martínez-de la Torre, A., Sarkar, S., Thober, S., ... & Blyth, E. (2023). Improving global hydrological simulations through bias-correction and multi-model blending. Journal of Hydrology, 621, 129607.

Clark, R. A., Gourley, J. J., Flamig, Z. L., Hong, Y., & Clark, E. (2014). CONUS-wide evaluation of National Weather Service flash flood guidance products. Weather and Forecasting, 29(2), 377-392.

Cowtan, K., & Way, R. G. (2014). Coverage bias in the HadCRUT4 temperature series and its impact on recent temperature trends. Quarterly Journal of the Royal Meteorological Society, 140(683), 1935-1944.

De Bézenac, E., Pajot, A., & Gallinari, P. (2019). Deep learning for physical processes: Incorporating prior scientific knowledge. Journal of Statistical Mechanics: Theory and Experiment, 2019(12), 124009.

Denaro, S., Anghileri, D., Giuliani, M., & Castelletti, A. (2017). Informing the operations of water reservoirs over multiple temporal scales by direct use of hydro-meteorological data. Advances in water resources, 103, 51-63.

Deng, J., Couasnon, A., Dahm, R., Hrachowitz, M., van Heeringen, K. J., Korving, H., ... & Taormina, R. (2024). Operational low-flow forecasting using LSTMs. Frontiers in Water, 5, 1332678.

Domeisen, D. I., Eltahir, E. A., Fischer, E. M., Knutti, R., Perkins-Kirkpatrick, S. E., Schär, C., ... & Wernli, H. (2023). Prediction and projection of heatwaves. Nature Reviews Earth & Environment, 4(1), 36-50.

Du, Y., Clemenzi, I., & Pechlivanidis, I. G. (2023). Hydrological regimes explain the seasonal predictability of streamflow extremes. Environmental Research Letters, 18(9), 094060.

EEA, Economic losses from weather- and climate-related extremes in Europe. (2023). Retrieved from https://www.eea.europa.eu/en/analysis/indicators/economic-losses-from-climate-related?activeAccordion=ecdb3bcf-bbe9-4978-b5cf-0b136399d9f8.

Fan, J., & Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space. Journal of the Royal Statistical Society Series B: Statistical Methodology, 70(5), 849-911.

Fan, J., Li, R., Zhang, C. H., & Zou, H. (2020). Statistical foundations of data science. Chapman and Hall/CRC.

Fan, M., Liu, S., & Lu, D. (2023). Advancing subseasonal reservoir inflow forecasts using an explainable machine learning method. Journal of Hydrology: Regional Studies, 50, 101584.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences, 55(1), 119-139.

Galelli, S., & Castelletti, A. (2013). Tree-based iterative input variable selection for hydrological modeling. Water Resources Research, 49(7), 4295-4310.

Ganganwar, V. (2012). An overview of classification algorithms for imbalanced datasets. International Journal of Emerging Technology and Advanced Engineering, 2(4), 42-47.

Garali, I., Adanyeguh, I. M., Ichou, F., Perlbarg, V., Seyer, A., Colsch, B., ... & Tenenhaus, A. (2018). A strategy for multimodal data integration: application to biomarkers identification in spinocerebellar ataxia. Briefings in bioinformatics, 19(6), 1356-1369.

García-Franco, J. L., Lee, C. Y., Camargo, S. J., Tippett, M. K., Kim, D., Molod, A., & Lim, Y. K. (2023). Climatology of tropical cyclone precipitation in the S2S models. Weather and Forecasting, 38(9), 1759-1776.

Geenens, G., Charpentier, A., & Paindaveine, D. (2017). Probit transformation for nonparametric kernel estimation of the copula density.

Geer, A. J. (2021). Learning earth system models from observations: machine learning or data assimilation?. Philosophical Transactions of the Royal Society A, 379(2194), 20200089.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. Machine learning, 63, 3-42.

Ghimire, S., Yaseen, Z. M., Farooque, A. A., Deo, R. C., Zhang, J., & Tao, X. (2021). Streamflow prediction using an integrated methodology based on convolutional neural network and long short-term memory networks. Scientific Reports, 11(1), 17497.

Giuliani, M., Castelletti, A., Pianosi, F., Mason, E., & Reed, P. M. (2016). Curses, tradeoffs, and scalable management: Advancing evolutionary multiobjective direct policy search to improve water reservoir operations. Journal of Water Resources Planning and Management, 142(2), 04015050.

Giuliani, M., Pianosi, F., & Castelletti, A. (2015). Making the most of data: An information selection and assessment framework to improve water systems operations. Water Resources Research, 51(11), 9073-9093.

Giuliani, M., Zaniolo, M., Castelletti, A., Davoli, G., & Block, P. (2019). Detecting the state of the climate system via artificial intelligence to improve seasonal forecasts and inform reservoir operations. Water Resources Research, 55(11), 9133-9147.

Guo, X., Ren, H., Zou, C., & Li, R. (2023). Threshold selection in feature screening for error rate control. Journal of the American Statistical Association, 118(543), 1773-1785.

Gupta, H. V., Kling, H., Yilmaz, K. K., & Martinez, G. F. (2009). Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling. Journal of hydrology, 377(1-2), 80-91.

Guse, B., Fatichi, S., Gharari, S., & Melsen, L. A. (2021). Advancing process representation in hydrological models: Integrating new concepts, knowledge, and data. Water Resources Research, 57(11), e2021WR030661.

Hadka, D., & Reed, P. (2013). Borg: An auto-adaptive many-objective evolutionary computing framework. Evolutionary computation, 21(2), 231-259.

Hao, Z., Hao, F., Xia, Y., Feng, S., Sun, C., Zhang, X., ... & Meng, Y. (2022). Compound droughts and hot extremes: Characteristics, drivers, changes, and impacts. Earth-Science Reviews, 235, 104241.

Haupt, S. E., Chapman, W., Adams, S. V., Kirkwood, C., Hosking, J. S., Robinson, N. H., ... & Subramanian, A. C. (2021). Towards implementing artificial intelligence post-processing in weather and climate: Proposed actions from the Oxford 2019 workshop. Philosophical Transactions of the Royal Society A, 379(2194), 20200091.

He, B., Liu, P., Zhu, Y., & Hu, W. (2019). Prediction and predictability of Northern Hemisphere persistent maxima of 500-hPa geopotential height eddies in the GEFS. Climate dynamics, 52, 3773-3789.

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., ... & Thépaut, J. N. (2020). The ERA5 global reanalysis. Quarterly Journal of the Royal Meteorological Society, 146(730), 1999-2049.

Hess, P., & Boers, N. (2022). Deep learning for improving numerical weather prediction of heavy rainfall. Journal of Advances in Modeling Earth Systems, 14(3), e2021MS002765.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

Hoffman, M. D., & Gelman, A. (2014). The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. J. Mach. Learn. Res., 15(1), 1593-1623.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. Neural networks, 2(5), 359-366.

Huang, B., Liu, C., Banzon, V., Freeman, E., Graham, G., Hankins, B., ... & Zhang, H. M. (2021). Improvements of the daily optimum interpolation sea surface temperature (DOISST) version 2.1. Journal of Climate, 34(8), 2923-2939.

Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine: theory and applications. Neurocomputing, 70(1-3), 489-501.

IPCC (2021), Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change, In Press, Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA.

Jin, Q., Meng, Z., Sun, C., Cui, H., & Su, R. (2020). RA-UNet: A hybrid deep attention-aware network to extract liver and tumor in CT scans. Frontiers in Bioengineering and Biotechnology, 8, 605132.

Jones, C., & Dudhia, J. (2017). Potential predictability during a Madden–Julian Oscillation event. Journal of Climate, 30(14), 5345-5360.

Jones, C., & Dudhia, J. (2017). Potential predictability during a Madden–Julian Oscillation event. Journal of Climate, 30(14), 5345-5360.

Jones, P. W. (1999). First-and second-order conservative remapping schemes for grids in spherical coordinates. Monthly Weather Review, 127(9), 2204-2210.

Kadow, C., Hall, D. M., & Ulbrich, U. (2020). Artificial intelligence reconstructs missing climate information. Nature Geoscience, 13(6), 408-413.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Kleinbaum, D. G., Dietz, K., Gail, M., Klein, M., & Klein, M. (2002). Logistic regression (p. 536). New York: Springer-Verlag.

Knapp, K. R., Kruk, M. C., Levinson, D. H., Diamond, H. J., & Neumann, C. J. (2010). The international best track archive for climate stewardship (IBTrACS) unifying tropical cyclone data. Bulletin of the American Meteorological Society, 91(3), 363-376.

Kraft, B., Jung, M., Körner, M., Koirala, S., & Reichstein, M. (2021). Towards hybrid modeling of the global hydrological cycle. Hydrology and Earth System Sciences Discussions, 2021, 1-40.

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., & Herrnegger, M. (2018). Rainfall–runoff modelling using long short-term memory (LSTM) networks. Hydrology and Earth System Sciences, 22(11), 6005-6022.

Kratzert, F., Klotz, D., Shalev, G., Klambauer, G., Hochreiter, S., & Nearing, G. (2019). Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. Hydrology and Earth System Sciences, 23(12), 5089-5110.

Lagerquist, R., & Ebert-Uphoff, I. (2022). Can we integrate spatial verification methods into neural network loss functions for atmospheric science?. Artificial Intelligence for the Earth Systems, 1(4), e220021.

Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Alet, F., ... & Battaglia, P. (2022). GraphCast: Learning skillful medium-range global weather forecasting. arXiv preprint arXiv:2212.12794.

Lamers, A., Sharma, M., Berg, R., Gálvez, J. M., Yu, Z., Kriat, T., ... & Moron, L. A. (2023). Forecasting tropical cyclone rainfall and flooding hazards and impacts. Tropical Cyclone Research and Review, 12(2), 100-112.

Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., & Müller, K. R. (2019). Unmasking Clever Hans predictors and assessing what machines really learn. Nature communications, 10(1), 1096.

Lazer, D., Kennedy, R., King, G., & Vespignani, A. (2014). The parable of Google Flu: traps in big data analysis. science, 343(6176), 1203-1205.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

Lee, R. W., Woolnough, S. J., Charlton-Perez, A. J., & Vitart, F. (2019). ENSO modulation of MJO teleconnections to the North Atlantic and Europe. Geophysical Research Letters, 46(22), 13535-13545.

Li, J., Yuan, X., & Ji, P. (2023). Long-lead daily streamflow forecasting using Long Short-Term Memory model with different predictors. Journal of Hydrology: Regional Studies, 48, 101471.

Ling, F., Li, Y., Luo, J. J., Zhong, X., & Wang, Z. (2022). Two deep learning-based bias-correction pathways improve summer precipitation prediction over China. Environmental Research Letters, 17(12), 124025.

Liu, G., Reda, F. A., Shih, K. J., Wang, T. C., Tao, A., & Catanzaro, B. (2018). Image inpainting for irregular holes using partial convolutions. In Proceedings of the European conference on computer vision (ECCV) (pp. 85-100).

Loader, C. (1999), Local Regression and Likelihood, Statistics and Computing Ser, Springer New York, New York, NY.

Loader, C. R. (1996). Local likelihood density estimation. The Annals of Statistics, 24(4), 1602-1618.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).

Maier, H. R., Kapelan, Z., Kasprzyk, J., Kollat, J., Matott, L. S., Cunha, M. C., ... & Reed, P. M. (2014). Evolutionary algorithms and other metaheuristics in water resources: Current status, research challenges and future directions. Environmental Modelling & Software, 62, 271-299.

McElreath, R. (2020), Statistical Rethinking, 2nd Edition: A Bayesian course with examples in R and Stan, Chapman & Hall/CRC texts in statistical science, 2nd edition, Chapman and Hall/CRC; Safari, Erscheinungsort nicht ermittelbar, Boston, MA.

McKee, T. B., Doesken, N. J., & Kleist, J. (1993, January). The relationship of drought frequency and duration to time scales. In Proceedings of the 8th Conference on Applied Climatology (Vol. 17, No. 22, pp. 179-183).

Miralles, D. G., Gentine, P., Seneviratne, S. I., & Teuling, A. J. (2019). Land–atmospheric feedbacks during droughts and heatwaves: state of the science and current challenges. Annals of the New York Academy of Sciences, 1436(1), 19-35.

Moradkhani, H., Sorooshian, S., Gupta, H. V., & Houser, P. R. (2005). Dual state–parameter estimation of hydrological models using ensemble Kalman filter. Advances in water resources, 28(2), 135-147.

Neal, R. M. (2012). Bayesian learning for neural networks (Vol. 118). Springer Science & Business Media.

Nearing, G., Cohen, D., Dube, V., Gauch, M., Gilon, O., Harrigan, S., ... & Matias, Y. (2024). Global prediction of extreme floods in ungauged watersheds. Nature, 627(8004), 559-563.

Nederhoff, K., van Ormondt, M., Veeramony, J., van Dongeren, A., Antolínez, J. A. Á., Leijnse, T., & Roelvink, D. (2024). Accounting for uncertainties in forecasting tropical-cyclone-induced compound flooding. Geoscientific Model Development, 17(4), 1789-1811.

North, G. R., Bell, T. L., Cahalan, R. F., & Moeng, F. J. (1982). Sampling errors in the estimation of empirical orthogonal functions. Monthly weather review, 110(7), 699-706.

Oddo, P. C., Bolten, J. D., Kumar, S. V., & Cleary, B. (2024). Deep Convolutional LSTM for improved flash flood prediction. Frontiers in Water, 6, 1346104.

Olivetti, L., & Messori, G. (2024). Advances and prospects of deep learning for medium-range extreme weather forecasting. Geoscientific Model Development, 17(6), 2347-2358.

Otero, N., & Horton, P. (2023). Intercomparison of deep learning architectures for the prediction of precipitation fields with a focus on extremes. Water Resources Research, 59(11), e2023WR035088. Rasp, S., Dueben, P. D., Scher, S., Weyn, J. A., Mouatadid, S., & Thuerey, N. (2020). WeatherBench: a benchmark data set for data-driven weather forecasting. Journal of Advances in Modeling Earth Systems, 12(11), e2020MS002203.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., & Efros, A. A. (2016). Context encoders: Feature learning by inpainting. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2536-2544).

Pathak, J., Subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., ... & Anandkumar, A. (2022). Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. arXiv preprint arXiv:2202.11214.

Pechlivanidis, I. G., Crochemore, L., Rosberg, J., & Bosshard, T. (2020). What are the key drivers controlling the quality of seasonal streamflow forecasts?. Water Resources Research, 56(6), e2019WR026987.

Perkins, S. E. (2015). A review on the scientific understanding of heatwaves—Their measurement, driving mechanisms, and changes at the global scale. Atmospheric Research, 164, 242-267.

Pham, L. T., Luo, L., & Finley, A. (2021). Evaluation of random forests for short-term daily streamflow forecasting in rainfall-and snowmelt-driven watersheds. Hydrology and Earth System Sciences, 25(6), 2997-3015.

Prodhan, F. A., Zhang, J., Hasan, S. S., Sharma, T. P. P., & Mohana, H. P. (2022). A review of machine learning methods for drought hazard monitoring and forecasting: Current research trends, challenges, and future research directions. Environmental modelling & software, 149, 105327.

Ramsay, J. O. (1988). Monotone regression splines in action. Statistical science, 425-441.

Roberts, N. M., & Lean, H. W. (2008). Scale-selective verification of rainfall accumulations from high-resolution forecasts of convective events. Monthly Weather Review, 136(1), 78-97.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18 (pp. 234-241). Springer International Publishing.

Rückstiess, T., Sehnke, F., Schaul, T., Wierstra, D., Sun, Y., & Schmidhuber, J. (2010). Exploring parameter space in reinforcement learning. Paladyn, 1, 14-24.

Rufibach, K. (2010). Use of Brier score to assess binary predictions. Journal of clinical epidemiology, 63(8), 938-939.

Sahin, C. (2022, May 11). Extended-range forecasts [Text]. ECMWF. https://www.ecmwf.int/en/forecasts/documentation-and-support/extended-range-forecasts.

Salazar, J. Z., Reed, P. M., Herman, J. D., Giuliani, M., & Castelletti, A. (2016). A diagnostic assessment of evolutionary algorithms for multi-objective surface water reservoir control. Advances in water resources, 92, 172-185.

Salcedo-Sanz, S., Pérez-Aracil, J., Ascenso, G., Del Ser, J., Casillas-Pérez, D., Kadow, C., ... & Castelletti, A. (2024). Analysis, characterization, prediction, and attribution of extreme atmospheric events with machine learning and deep learning techniques: a review. Theoretical and Applied Climatology, 155(1), 1-44.

Schäfer, J., & Strimmer, K. (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. Statistical applications in genetics and molecular biology, 4(1).

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural networks, 61, 85-117.

Schölkopf, B., & Smola, A. J. (2002). Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press.

Seneviratne et al_2021_Weather and climate extreme events in a changing climate.pdf, (https://www.ipcc.ch/report/ar6/wg1/downloads/report/IPCC_AR6_WGI_Chapter_11.pdf) accessed November 9, 2021.

Sha, Y., Gagne II, D. J., West, G., & Stull, R. (2020). Deep-learning-based gridded downscaling of surface meteorological variables in complex terrain. Part II: Daily precipitation. Journal of Applied Meteorology and Climatology, 59(12), 2075-2092.

Sharifi, E., Eitzinger, J., & Dorigo, W. (2019). Performance of the state-of-the-art gridded precipitation products over mountainous terrain: A regional study over Austria. Remote Sensing, 11(17), 2018.

Shi, X., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. Advances in neural information processing systems, 28.

Singh, O. P., Ali Khan, T. M., & Rahman, M. S. (2000). Changes in the frequency of tropical cyclones over the North Indian Ocean. Meteorology and Atmospheric physics, 75(1), 11-20.

Song, Y., Yang, C., Lin, Z., Liu, X., Huang, Q., Li, H., & Kuo, C. C. J. (2018). Contextual-based image inpainting: Infer, match, and translate. In Proceedings of the European conference on computer vision (ECCV) (pp. 3-19).

Specq, D., & Batté, L. (2022). Do subseasonal forecasts take advantage of Madden–Julian oscillation windows of opportunity?. Atmospheric Science Letters, 23(4), e1078.

Teegavarapu, R. S., & Chandramouli, V. (2005). Improved weighting methods, deterministic and stochastic data-driven models for estimation of missing precipitation records. Journal of hydrology, 312(1-4), 191-206.

Tenenhaus, A., & Tenenhaus, M. (2011). Regularized generalized canonical correlation analysis. Psychometrika, 76, 257-284.

Tenenhaus, A., Philippe, C., & Frouin, V. (2015). Kernel generalized canonical correlation analysis. Computational Statistics & Data Analysis, 90, 114-131.

Tenenhaus, M., Tenenhaus, A., & Groenen, P. J. (2017). Regularized generalized canonical correlation analysis: a framework for sequential multiblock component methods. Psychometrika, 82(3), 737-777.

Tong, Z., Cai, Z., Yang, S., & Li, R. (2023). Model-free conditional feature screening with FDR control. Journal of the American Statistical Association, 118(544), 2575-2587.

Turner, S. W., Xu, W., & Voisin, N. (2020). Inferred inflow forecast horizons guiding reservoir release decisions across the United States. Hydrology and Earth System Sciences, 24(3), 1275-1291.

Van Der Knijff, J. M., Younis, J., & De Roo, A. P. J. (2010). LISFLOOD: a GIS-based distributed model for river basin scale water balance and flood simulation. International Journal of Geographical Information Science, 24(2), 189-212.

Vehtari, A., Gelman, A., & Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. Statistics and computing, 27, 1413-1432.

Vicente-Serrano, S. M., Beguería, S., & López-Moreno, J. I. (2010). A multiscalar drought index sensitive to global warming: the standardized precipitation evapotranspiration index. Journal of climate, 23(7), 1696-1718.

Vitart, F., & Robertson, A. W. (2019). Introduction: Why sub-seasonal to seasonal prediction (S2S)?. In Sub-seasonal to seasonal prediction (pp. 3-15). Elsevier.

Wahl, T., Jain, S., Bender, J., Meyers, S. D., & Luther, M. E. (2015). Increasing risk of compound flooding from storm surge and rainfall for major US cities. Nature Climate Change, 5(12), 1093-1097.

Wang, Z., Zhao, J., Huang, H., & Wang, X. (2022). A review on the application of machine learning methods in tropical cyclone forecasting. Frontiers in Earth Science, 10, 902596.

Watanabe, S., & Opper, M. (2010). Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. Journal of machine learning research, 11(12).

Wilks, D. S. (2011). Statistical methods in the atmospheric sciences (Vol. 100). Academic press.

Xu, S. G., & Reich, B. J. (2023). Bayesian nonparametric quantile process regression and estimation of marginal quantile effects. Biometrics, 79(1), 151-164.

Xu, S. G., Majumder, R., & Reich, B. J. (2022). SPQR: An R Package for Semi-Parametric Density and Quantile Regression. arXiv preprint arXiv:2210.14482.

Xu, T., & Liang, F. (2021). Machine learning for hydrologic sciences: An introductory overview. Wiley Interdisciplinary Reviews: Water, 8(5), e1533.

Yang, D., Yang, Y., & Xia, J. (2021). Hydrological cycle and water resources in a changing world: A review. Geography and Sustainability, 2(2), 115-122.

Yang, T. H., Yang, S. C., Ho, J. Y., Lin, G. F., Hwang, G. D., & Lee, C. S. (2015). Flash flood warnings using the ensemble precipitation forecasting technique: A case study on forecasting floods in Taiwan caused by typhoons. Journal of Hydrology, 520, 367-378.

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., & Huang, T. S. (2018). Generative image inpainting with contextual attention. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5505-5514).

Zaniolo, M., Giuliani, M., & Castelletti, A. (2021). Policy representation learning for multiobjective reservoir policy design with different objective dynamics. Water Resources Research, 57(12), e2020WR029329.

Zhao, Q., Cai, X., & Li, Y. (2019). Determining inflow forecast horizon for reservoir operation. Water Resources Research, 55(5), 4066-4081.

Zhao, Y., Liu, C., Di, D., Ma, Z., & Tang, S. (2022). High-resolution typhoon precipitation integrations using satellite infrared observations and multisource data. Atmospheric Measurement Techniques, 15(9), 2791-2805.

Zimmerman, B. G., Vimont, D. J., & Block, P. J. (2016). Utilizing the state of ENSO as a means for season-ahead predictor selection. Water resources research, 52(5), 3761-3774.

Zscheischler, J., & Seneviratne, S. I. (2017). Dependence of drivers affects risks associated with compound events. Science advances, 3(6), e1700263.

Zscheischler, J., Martius, O., Westra, S., Bevacqua, E., Raymond, C., Horton, R. M., ... & Vignotto, E. (2020). A typology of compound weather and climate events. Nature reviews earth & environment, 1(7), 333-347.

Zscheischler, J., Westra, S., Van Den Hurk, B. J., Seneviratne, S. I., Ward, P. J., Pitman, A., ... & Zhang, X. (2018). Future climate risk from compound events. Nature climate change, 8(6), 469-477.

This project is part of the H2020 Programme supported by the European Union, having received funding from it under Grant Agreement No 101003876