

interTwin

D4.3 First version of the DTs capabilities for climate change and impact decision support tools

Status: APPROVED BY THE EC
Dissemination Level: public



Funded by the
European Union

Disclaimer: Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them

Abstract

Key Words digital twin, environment, climate change, impacts, tools

This deliverable describes the release of Digital Twin (DT) Applications that support the climate change use cases and the implementation of the related impact decision support tools. It details the capabilities, characteristics, and describes the functional specifications of the DT applications and the status of their integration into the DTE architecture. Finally, it provides the information about the next steps in the development and integration of those DTs.



Document Description			
D4.3 First version of the DTs capabilities for climate change and impact decision support tools			
Work Package number 4			
Document type	Deliverable		
Document status	APPROVED BY THE EC	Version	1
Dissemination Level	Public		
Copyright Status	 <p>This material by Parties of the interTwin Consortium is licensed under a Creative Commons Attribution 4.0 International License.</p>		
Lead Partner	CERFACS		
Document link	https://documents.egi.eu/document/3939		
DOI	https://zenodo.org/records/14974664		
Author(s)	<ul style="list-style-type: none"> • Christian Pagé (CERFACS) • Björn Backeberg (Deltares) • Donatello Elia (CMCC) • Emanuele Donno (CMCC) • Francesco Immorlano (CMCC) • Martin Schobben (TU Wien) • Bernhard Raml (TU Wien) • Massimiliano Fronza (UNITN) • Iacopo Ferrario (EURAC) • Alice Crespi (EURAC) • Suriyah Dhinakaran (EURAC) 		
Reviewers	<ul style="list-style-type: none"> • Raul Bardaji (EGI Foundation) • Fabrizio Antonio (CMCC) 		
Moderated by:	<ul style="list-style-type: none"> • Sjomara Specht (EGI Foundation) 		
Approved by	<ul style="list-style-type: none"> • Andrea Manzi (EGI Foundation) on behalf of TCB 		

Revision History			
Version	Date	Description	Contributors
V0.1	08/03/2024	ToC	Christian Pagé (CERFACS)
V0.2	06/05/2024	Intermediary version	Christian Pagé (CERFACS), Björn Backeberg (Deltares), Donatello Elia, Emanuele Donno (CMCC)
V0.3	14/05/2024	Version ready for internal review	Christian Pagé (CERFACS), Björn Backeberg (Deltares), Donatello Elia, Emanuele Donno (CMCC), Martin Schobben (TU Wien), Bernhard Raml (TU Wien) Massimiliano Fronza (UNITN), Iacopo Ferrario (EURAC), Alice Crespi (EURAC), Suriyah Dhinakaran (EURAC)
V0.4	21/05/2024	Version reviewed by internal reviewers	Raul Bardaji (EGI Foundation) Fabrizio Antonio (CMCC)
V0.5	27/05/2024	Version ready for TCB Review	Christian Pagé (CERFACS), Björn Backeberg (Deltares), Donatello Elia, Emanuele Donno (CMCC), Martin Schobben (TU Wien), Bernhard Raml (TU Wien), Massimiliano Fronza (UNITN), Iacopo Ferrario (EURAC), Alice Crespi (EURAC), Suriyah Dhinakaran (EURAC)
V0.6	27/05/2024	Version reviewed by TCB	Andrea Manzi (EGI Foundation)
V1.0	28/05/2024	Final	

Terminology / Acronyms	
Term/Acronym	Definition
DTE	Digital Twin Engine
DT	Digital Twin
CMIP6	Coupled Model Intercomparison Project Phase 6
SSP	Shared Socioeconomic Pathways (set of scenarios)
NetCDF	Network Common Data Form
ML	Machine Learning
DNN	Deep Neural Network
Wflow	hydrological modelling framework
CNN	Convolutional Neural Network
CGNN	Convolutional Graph Neural Network



D4.3 First version of the DTs capabilities for climate change and impact decision support tools

WP6	Work Package 6
ERA5	Fifth generation ECMWF reanalysis for the global climate and weather
CERRA	Copernicus regional reanalysis for Europe
ECMWF	European Centre for Medium-Range Weather Forecasts
CDS	Climate data store
MODIS	Moderate Resolution Imaging Spectroradiometer
LAI	Leaf Area Index
LSTM	Long Short-Term Memory
hPa	Hectopascal
SEAS5	Seasonal Ensemble Prediction System 5
IBTrACS	International Best Track Archive for Climate Stewardship
EO	Earth Observations
VAE	Variational Auto Encoder
CVAE	Convolutional Variational Auto Encoder
RCP	Representative Concentration Pathways (set of scenarios)
xtclim	Generic Climate Extreme characterization and detection AI-based tool

Terminology / Acronyms: <https://confluence.egi.eu/display/EGIG>



Table of Contents

1	Introduction	9
1.1	Aim of this deliverable	9
1.2	For whom is this document	9
1.3	Structure of the document	9
2	DTs Application	10
2.1	Generic Detection of Climate Extremes	10
2.2	Wildfire danger prediction in response to climate change	11
2.3	Tropical storms change in response to climate change	12
2.4	Flood early warning in coastal and inland regions	15
2.5	Alpine droughts early warning	16
2.6	Flood climate impact in coastal and inland regions	21
3	DTs Application First release	23
3.1	Generic Detection of Climate Extremes	23
3.1.1	DTs Application Integrations	23
3.1.2	Scope and limitations	24
3.2	Wildfire danger prediction in response to climate change	24
3.2.1	DTs Application Integrations	24
3.2.2	Scope and limitations	26
3.2.3	Preconditions	27
3.3	Tropical storms change in response to climate change	27
3.3.1	DTs Application Integrations	27
3.3.2	Scope and limitations	29
3.3.3	Preconditions	30
3.4	Flood early warning in coastal and inland regions	30
3.4.1	DTs Application Integrations	30
3.4.2	Scope and limitations	32
3.4.3	Preconditions	32
3.5	Alpine droughts early warning	33
3.5.1	DTs Application Integrations	33
3.5.2	Scope and limitations	37
3.6	Flood climate impact in coastal and inland regions	38
3.6.1	DTs Application Integrations	38
3.6.2	Data requirements	40
3.6.3	Scope and limitations	41
3.6.4	Preconditions	42
4	Conclusions	43
5	References	44



Table of Figures

FIGURE 1 - COMPARISON OF REAL (RIGHT) AND PREDICTED (LEFT) BURNED AREAS ON THE TEST DATASET (2019-2020): THE NEURAL NETWORK'S PREDICTED BURNED AREAS RESEMBLE THE REAL DATA IN TERMS OF SPATIAL PATTERN AS WELL AS THE AMOUNT OF HECTARES BURNT.....	12
FIGURE 2 - INFERENCE OF ML MODEL ON KEONI TC (FROM THE TEST SET). RED MARKERS REPRESENT THE POSITIONS IDENTIFIED BY THE ML MODEL THAT WAS FED WITH SIX CLIMATIC VARIABLE PATCHES FOR EACH TIME STEP OF THE TC TRAJECTORY. BLUE MARKERS DENOTE THE ACTUAL POSITIONS OF THE TC THROUGHOUT ITS TRAJECTORY EXTRACTED FROM IBTRACS.	14
FIGURE 3 - FOUR EXAMPLE PATCHES WERE TAKEN FROM THE TEST SET AND FED TO THE TRAINED MODEL. THE TOP ROW SHOWS THE ACTUAL POSITIONS OF THE CYCLONES, WHILE THE BOTTOM CONTAINS THE PROBABILITY ESTIMATES OF THE POSITIONS PRODUCED BY THE CGNN MODEL. THE COLOUR BARS REPRESENT THE CONFIDENCE OF THE PREDICTION FROM 0 TO 1.....	14
FIGURE 4 - BASE CONFIGURATION FOR THE HUMBER ESTUARY AND CATCHMENT OF THE DT'S PHYSICS-BASED MODELS. A) SFINCS BASEMAP SHOWING THE DEM USED, THE RIVERS IN THE AREA AND USER-DEFINED OBSERVATION POINTS. B) WFLOW BASEMAP SHOWING THE RIVERS AND THE DEM OF THE FULL CATCHMENT AND THE POINTS WHERE RIVER DISCHARGES ARE PROVIDED FOR SFINCS.	15
FIGURE 5 - FLOOD MAP FOR THESSALY (GREECE) ON 28TH OF FEBRUARY 2018 WITH OPENEO IMPLEMENTATION OF TU WIEN FLOOD MAPPING ALGORITHM.....	16
FIGURE 6 - INTERTWIN REGION AND RIVER BASINS IN RED.	17
FIGURE 7 - DT HYDROLOGICAL WORKFLOW LINKED COMPONENTS DIAGRAM.....	18
FIGURE 8 - SIMULATED EVAPOTRANSPIRATION AND SURFACE SOIL MOISTURE TIME SERIES AT THREE LOCATIONS OVER THE ALPINE REGION.....	20
FIGURE 9 - DELFT-FIAT BASEMAP FOR THE HUMBER ESTUARY BASED ON BUILDING FOOTPRINTS FETCHED FROM OPENSTREETMAP.....	22
FIGURE 10 - A) FLOODMAP PRODUCED BY SFINCS SHOWING WHICH AREAS ARE INUNDATED AND BY HOW MUCH. B) DAMAGE MAP PRODUCED BY DELFT-FIAT BASED ON A SFINCS FLOODMAP.	22
FIGURE 11 - OVERVIEW OF THE WORKFLOW FOR DT ON EXTREME EVENTS.....	23
FIGURE 12 - OVERVIEW OF THE WILDFIRES DT APPLICATION.	26
FIGURE 13 - OVERVIEW OF THE TC DT APPLICATION.	28
FIGURE 14 - HIGH-LEVEL WORKFLOW DIAGRAM FOR THE FLOOD EARLY WARNING DT.	31
FIGURE 15 - HIGH LEVEL DIAGRAM OF THE DT WORKFLOW'S APPLICATION COMPONENTS.	35
FIGURE 16 - HIGH LEVEL DIAGRAM THAT SHOWS HOW DT'S APPLICATION COMPONENTS ARE RUN.....	36
FIGURE 17 - HIGH-LEVEL WORKFLOW DIAGRAM FOR THE FLOOD EARLY WARNING DT.	39
FIGURE 18 - ARCHITECTURE DIAGRAM HIGHLIGHTING INTEGRATION AND INTERACTION BETWEEN THE JUPYTER NOTEBOOK (THE DT INTERFACE), THE CWL WORKFLOWS AND INTERTWIN'S RUCIO-BASED DATALAKE.....	40

Table of Tables

TABLE 1 - INITIAL SET OF VARIABLES CONSIDERED FOR TRAINING.	23
TABLE 2 - INITIAL SET OF VARIABLES CONSIDERED FOR TRAINING.	25
TABLE 3 - INPUT PARAMETERS TO THE ML MODEL.	28
TABLE 4 - INPUT PARAMETERS FROM OPEN SOURCE DATASETS	33
TABLE 5 - PROCESSED INPUT-OUTPUT OF WFLOW_SBM.	34
TABLE 6 - DATASETS USED BY THE DT	40



Executive summary

This document is deliverable 4.3 of the interTwin project, part of work package 4. It is a report collectively written by the partners of tasks 4.5, 4.6 and 4.7, who are directly involved in designing DT Applications for the environmental domain (climate projections & extreme events). This report aims to describe the Digital Twins (DTs) capabilities, as well as their integration status into the Digital Twin Engine (DTE), especially for the integration with the infrastructure and with the Core Components of the architecture.

First, each DT application is described to explain their specific capabilities that support the use cases, and which capabilities the user will be able to access in order to explore what-if scenarios. These DT applications will enable users to assess some specific aspects of the climate change impacts in various geographical regions by covering tropical storms, wildfires, floods, droughts and other extreme weather events.

In addition to the capabilities, the current status of the integration of each DT application with the main Core as well as with the infrastructure components is described. The next steps that are planned toward full integration are also discussed.

Overall, each DT is specifically designed and linked to address specific climate change impacts and provide valuable insights for assessing climate risks, identifying early warning signals, and implementing mitigation measures. It provides invaluable tools enabling users to explore several future possible specific impacts of climate change.

1 Introduction

1.1 Aim of this deliverable

The overall objective of deliverable 4.3 is to provide information about the capabilities of DT Applications related to climate change and impact decision support tools. Those DTs are from the environmental domain (T4.5, T4.6, T4.7). A **Digital Twin Application** is a user-facing implementation of a DT. DT applications are the consumers of the capabilities offered by the interTwin DTE, thus introducing use case-specific requirements.

1.2 For whom is this document

The deliverable 4.3 could be useful for both developers and end users as described below:

For **developers**: the current deliverable provides them with an opportunity to be informed on how the integration of DTs is being planned and implemented. This integration is also linked to the different capabilities and features of each specific DT. It gives insights on how those DTs will fit into the overall interTwin architecture by using specific core components.

For **end users**: the specified deliverable provides information on what are the capabilities of each specific DT in the environmental domain. It provides insights to the users on what can be achieved by using those DTs: what it can be used for, and eventually what are the parameters and configurations that can be set by the end users for tailoring those DTs to their specific needs. By establishing a common framework for communication, researchers and stakeholders will be able to exchange information, validate models, and collaboratively address climate change impacts and suggest mitigation measures.

1.3 Structure of the document

The structure of this deliverable is as follows. [Section 2](#) describes the capabilities of each digital twin. [Section 3](#) explains how the integration is planned and what the current status of the integration is. The reader should refer to D4.1¹ for a description of the architecture.

¹ D4.1 First Architecture design of the DTs capabilities for climate change and impact decision support tools
<https://doi.org/10.5281/zenodo.10417135>



2 DTs Application

2.1 Generic Detection of Climate Extremes

This DT is aimed at the detection and characterization of climate extremes to assess their impacts and provide useful information for the decision making process. The DT enables users to quantify and evaluate the changes of climate extremes, such as changes in the frequency of occurrence, as well as their spatial extent, duration, and intensity. In the first version of the DT, the maximum daily temperature will be the variable used to calculate extreme temperatures compared to the normal values, according to a specific season or monthly period. In future versions of the DT, users will be able to select other variables such as precipitation and wind speed.

The end users will be able to select specific greenhouse gas scenarios to explore different impacts according to the evolution of those emissions that are driven by national and international policies. Users will also have the capability to select a region of interest to focus on specific areas and better evaluate regional and local impacts. It will also be possible to select a specific time period and a season (or a monthly period), as well as several datasets coming from different types of future climate simulations, such as global circulation coupled models, regional climate models, and any other relevant datasets.

To evaluate the changes, the end users will define a time period of reference to compare with. Typically, a period of 25 to 30 years is used as a climate reference, in order to better evaluate the needed adaptation compared to a specific state of the climate.

In more details, the Deep Learning method that is used consists of a Convolutional Variational Autoencoder (CVAE). Autoencoder and decoder are a pair of unsupervised trained neural networks where autoencoders are trained to compress the data and decoders are trained to decompress the compressed data with a minimal loss. This can be used for anomaly detection, hence also for climate extremes. When input data is compressed, the main features of the data are kept, and there is some loss since, after compression, data is stored in a lower dimensional space (called latent space) before being decompressed. Variational autoencoders (VAE) model the latent space as a probability distribution. CVAEs use Convolutional Neural Networks (CNN) for both encoder and decoder parts.

In this implementation, CVAE input data is a subset of CMIP6 data: a time sequence of 32 X 32 square images of a daily average of a specific atmospheric surface variable over a specific spatial subset region. Training is done by season, for each specific climate model separately, using a time period long enough to have sufficient samplings (typically 50 years or more) and in which the greenhouse gas emissions have a weak tendency (e.g. 1850-1950). The CVAE model can then be applied to any time period of interest to the user, using any CMIP6 simulation data of this specific global climate model. For each daily image reconstructed by the CVAE, there is a loss, corresponding to the error in the reconstructed image. This loss value is used for anomaly detection (climate extremes are anomalies).



This DT is based on artificial intelligence techniques, and implements a method based on unsupervised Deep Learning (CVAE). This unlocks the possibility for users to better quantify the uncertainties associated with different sources such as greenhouse gas emission scenarios, specific climate models and also inherent climate variability. This is possible thanks to the high performance of DL methods to process large datasets. Typically, using analytical methods, users would only use a small subset of all available simulations, leading to partial uncertainty assessments resulting in incomplete impact assessments. This could lead to less than optimal adaptation and mitigation decisions.

2.2 Wildfire danger prediction in response to climate change

The DT related to the wildfires application focuses on the generation of fire danger maps on a global domain that resemble the burned area spatial distribution learnt from historical records. Output fire danger maps are produced through ML models, in particular using the *UNET++*² architecture, but the evaluation will be spread to several Deep Neural Networks. The networks are trained to learn the non-linear spatial relationship between multiple environmental variables and the hectares of burned areas. Such variables include climatic, weather and potentially vegetation conditions and are provided as predictors for the trained models. The hectares of burned areas produced by wildfires events, instead, expressed as the percentage of burned area compared to the theoretical maximum burnable area per pixel (in the geographical domain of interest) are used as targets. Trained ML models will be applied on future projection scenarios to give an indication on how climate change is affecting geographical areas and the extent of wildfires.

Several architectures have been investigated for learning complex relationships between the chosen input variables and extracting global fire danger maps containing the percentage of wildfire burned areas for each pixel. In particular, the UNET++ architecture has been identified as a possible solution to the problem about the wildfire danger maps prediction. Other deep learning architectures could be considered during the remainder of the project to improve the prediction capabilities.

The data used came from SeasFire Cube³, a scientific datacube containing 21 years of data (2001-2021) with an 8-day temporal resolution and 0.25° spatial resolution, designed to forecast seasonal fires around the world. It contains 59 seasonal fire drivers, including atmospheric and climatic variables, vegetation, socioeconomic drivers, burned areas, fire radiative power, as well as wildfire-related CO2 emissions.

The proposed ML architecture takes as input and produces as output a stack of several climatic variables. The stack has a generic dimension of $H \times W \times C$, where H is the height and it is equal to 720, W is the width and it is equal to 1440, C is the number of climatic variables and it depends on whether the stack is made of drivers or targets: C is (currently)

² <https://arxiv.org/pdf/1807.10165>

³ <https://zenodo.org/records/8055879>



equal to 8 if the stack is composed of drivers (Leaf Area Index, Land-Sea mask, Land Surface temperature at day, Relative humidity, Surface net solar radiation, Sea Surface Temperature, Temperature at 2 metres - Min, Total precipitation), otherwise, C is equal to 1 if the stack is composed of targets (Burned Areas from FCCI).

The hyperparameter fine tuning has been carried on in order to calibrate the ML architecture to achieve good results. A set of hyperparameters (the base filter dimensions, losses and optimizers) has been chosen and evaluated, for a total of 9 different model configurations.

The workflow and the ML model are implemented in Python using PyTorch as ML framework. The trained model and the inference workflow will be made available to end-users (e.g., scientists and policy makers) through Jupyter notebooks interacting with the core and thematic modules from the project. In particular, the end-users can exploit the workflow for running what-if analyses about wildfire burned area predictions (**Figure 1**) made with climate projection data. Users will interact with the model to provide the projection scenarios from a given list, the temporal (and potentially geographical) extents and ML models from a set of pre-trained model architectures. The output of the DT will be visualised through the notebook and downloaded/saved as NetCDF files. Additionally, customised maps and charts will be made available.

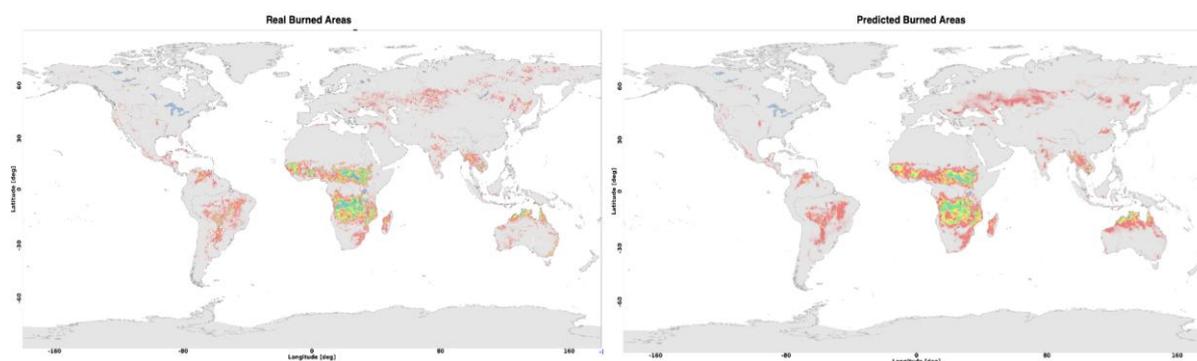


Figure 1 - Comparison of real (right) and predicted (left) burned areas on the test dataset (2019-2020): the neural network's predicted burned areas resemble the real data in terms of spatial pattern as well as the amount of hectares burnt.

2.3 Tropical storms change in response to climate change

This DT focuses on the detection of tropical cyclones (TCs) and consists in classifying the absence or presence of a cyclone in gridded climate fields and in a specific time instant and, if present, localising its centre (or “eye”) in terms of latitude/longitude geographical coordinates. ML models allow learning the mapping between climatic fields significant to the cyclogenesis and the positions and trajectories that storms follow during their lifetime in historical records. Trained models will be exploited to predict the occurrence of storms in future projection scenarios in order to give an indication, across both space and time, about the areas of the world that will be more susceptible to experience such phenomena, according to different levels of climate change.

Since the variable describing the cyclogenesis consists of a set of 2-dimensional (2D) data, each variable can be easily considered as a 2D image, where each pixel corresponds to a cell of the lat-lon grid. Starting from this consideration, two different approaches are being explored for the detection of TCs: Convolutional Neural Networks (CNNs) and Graph Convolutional Neural Networks (GCNNs).

Concerning the geographical domain, the joint North Atlantic-North Pacific formation basin was targeted because it is considered the one with the highest number of TCs occurrences every year. Six input climatic variables (i.e., mean sea level pressure, 10 m wind gust since previous post-processing, instantaneous 10 m wind gust, relative vorticity at 850 mb, and temperature at 300 and 500 mb) have been gathered from Copernicus Climate Change Service ERA5 reanalysis datasets for the region of interest and used as predictors of the presence of TCs: they were stacked together and treated as image channels. The temporal extent considered is 1980–2019 with a temporal resolution of 6 h. These ERA5 maps (size: 280×880 grid points) were preliminary divided into 7×22 non-overlapping patches of fixed size 40×40 . Each patch is fed to the CNN; thus, the final dimension of the input is $6 \times 40 \times 40$. This pre-processing step is performed to improve the model efficiency and, more importantly, to ensure that each patch contains at most one TC. The ML model architectures are configured to output the (row,col) coordinates of a TC within the patch. The model can output a (row,col) coordinate in the range $[[0-39], [0-39]]$ if a TC is detected (i.e., positive patch). Instead, negative labels (e.g., (-1, -1)) are predicted when no cyclone is found within the patch (i.e., negative patch). The IBTrACS historical dataset has been used to map each positive patch with the corresponding TC historical occurrence. Concerning the dataset split, the years 2010, 2011, 2012, 2013 were reserved for validation, the years 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021 for test and the remaining for training. By construction, the training set is imbalanced towards negative patches (only 20% of them contain a TC). A data augmentation procedure was applied to address the imbalance and reach a 50/50 ratio.

In the CNN approach, the ML model identified for the DT is a Visual Geometry Group (VGG)-like CNN [R1, R2]. In this case, an ensemble approach has been implemented to combine the predictions made by different models with the aim of improving the overall accuracy skills. To this extent, 13 VGG-like NNs were developed and trained for the TC centre localization task. They differ in convolutional block complexities (i.e., composition and number of parameters) and hyperparameters configurations (e.g., loss function, kernel size), thus each of them learns distinct characteristics and high-level features. At first, for each patch, the approach consists in evaluating how many models agree on classifying it as positive. To classify a specific patch as positive, a minimum number of 7 models that predict the presence of a TC are needed. Subsequently, the Interquartile Range (IQR) method is applied on the locations predicted by the models to keep only the estimates closer to their median value and remove the outliers. Finally, the position of the TC centre is computed as the ensemble average of the inliers estimates. In terms of software infrastructure, it has been coded in Python v3.11.2 based on Keras and using Tensorflow as a back-end.



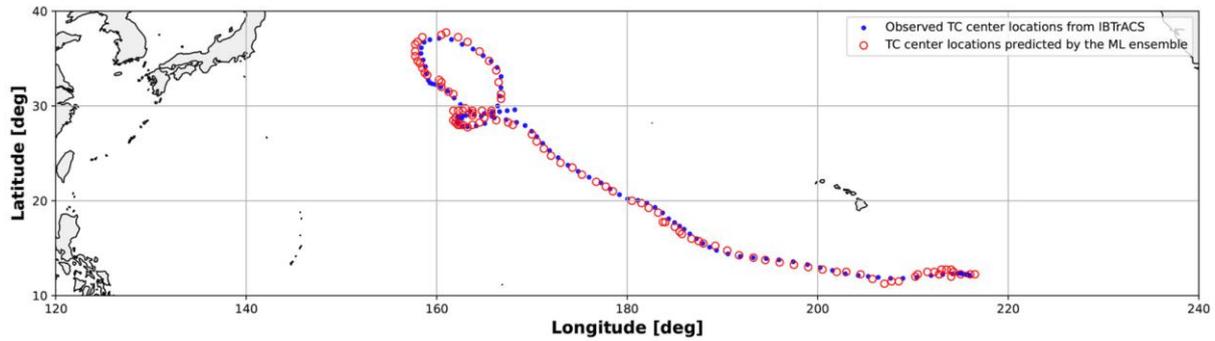


Figure 2 - Inference of ML model on Keoni TC (from the test set). Red markers represent the positions identified by the ML model that was fed with six climatic variable patches for each time step of the TC trajectory. Blue markers denote the actual positions of the TC throughout its trajectory extracted from IBTrACS.

In the second case a CGNN-based approach has been investigated. The pre-processing works in a similar way as for the CNNs. While the stacking of the atmospheric variables and the division into patches is identical to the CNNs case, the data at this stage is organised in grids, not in graphs. An additional step is hence needed to make the patches readable by the CGNN, and it consists of retrieving the adjacency information from these matrices, linking together the pixels as if they were nodes in a graph. Along with this information, the features are permuted from their current dimension, $C \times H \times W$, to $L \times C$, where L is the list of graph nodes with dimension $W \times H$, rather than a matrix.

Together with the coordinates (lat, lon) information of the cyclones, the dataset also provides probability density maps that are still derived from IBTrACS and range from 1.0 (the pixel with the cyclone) to 0.0 (the furthest locations from the eye of the turbulence). To maintain the compatibility of the pipeline with a previous version of the code, while in the CNN supervised training we make use of the matrix coordinates, in the CGNN learning we use the probability density maps instead. Therefore, the CGNN will be trained to find a non-linear mapping between 3D input graphs and the position of the corresponding maximum probability value in output.

Ground truth density maps(top) versus predictions of the model(bottom)

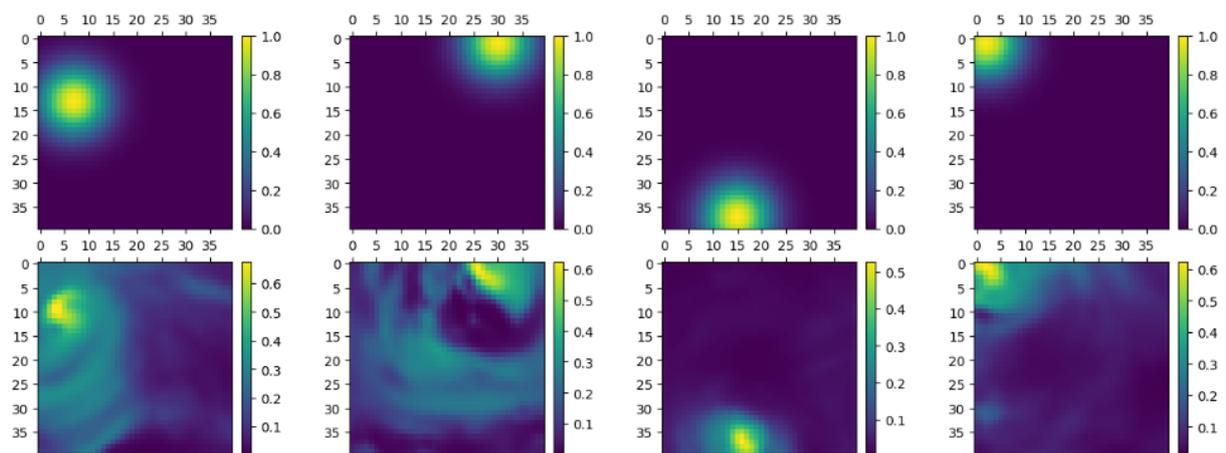


Figure 3 - Four example patches were taken from the test set and fed to the trained model. The top row shows the actual positions of the cyclones, while the bottom contains the probability estimates of the positions produced by the CGNN model. The colour bars represent the confidence of the prediction from 0 to 1.

End-users can interact with the DT use case through a notebook for running analysis concerning TCs on future projection data (e.g., trends about frequency of events). Similarly to the wildfire case, the workflow and the ML model are implemented in Python using PyTorch as ML framework. The user can run the workflow for what-if analyses about TC trends based on climate projection data, visualise the results through customizable maps and charts as well as download/save them as NetCDF files.

2.4 Flood early warning in coastal and inland regions

The DT for flood early warning in coastal and inland regions will focus on the generation of flood risk maps that can be used by early warning systems to trigger alerts when a flood is predicted. The DT enables users to monitor floods using satellite remote sensing and operationally predict compound floods using a flood inundation and a hydrological model providing river discharges.

End users will be able to specify a geographic region of interest and interTwin's core and thematic modules will enable setting up the flood inundation SFINCS and hydrological model Wflow (Figure 4), including necessary Earth Observation data processing pipelines to monitor and predict floods for the user-defined region of interest. The DT runs the models and processing pipelines operationally to generate flood risk forecasts and prepares and standardises the data for easy ingestion into an early warning system. Users will also have the capability to add their own local data to the model schematisations, thus enhancing model accuracy (e.g. Digital Elevation Models based on local LiDAR surveys).

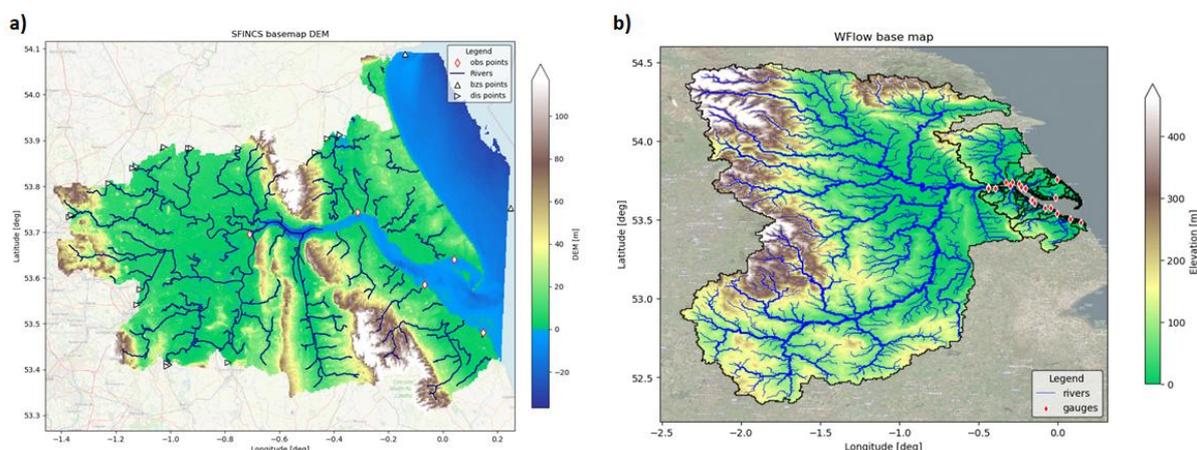


Figure 4 - Base configuration for the Humber estuary and catchment of the DT's physics-based models. a) SFINCS basemap showing the DEM used, the rivers in the area and user-defined observation points. b) Wflow basemap showing the rivers and the DEM of the full catchment and the points where river discharges are provided for SFINCS.

The DT's flood risk maps are produced by SFINCS⁴, a reduced-complexity model for super-fast dynamic modelling of compound flooding, which receives river discharge data from Wflow⁵, a hydrological model. Both SFINCS and Wflow are forced by weather forecasts. Additionally, the DT will combine the SFINCS flood maps with Sentinel-1 based flood maps

⁴ <https://github.com/Deltares/SFINCS>

⁵ <https://github.com/Deltares/Wflow.jl>

generated by the openEO implementation of the TU Wien flood mapping algorithm [R7]. This algorithm enables near real-time mapping of the flood extent through Bayesian inference from Sentinel-1 Synthetic-Aperture Radar (SAR) microwave backscattering. Globally applicable flood signatures are obtained through establishing predefined probability distributions of pixels under flood and non-flood conditions. These conditions are inferred on generalised backscattering characteristics of water and land. The end user can run the openEO implementation and derive flood maps for the targeted region (Figure 5). An example of the openEO workflow for Sentinel-1 based flood mapping in the Python syntax has been published under the following link: <https://intertwin-eu.github.io/openeo-flood-mapper-local/>.

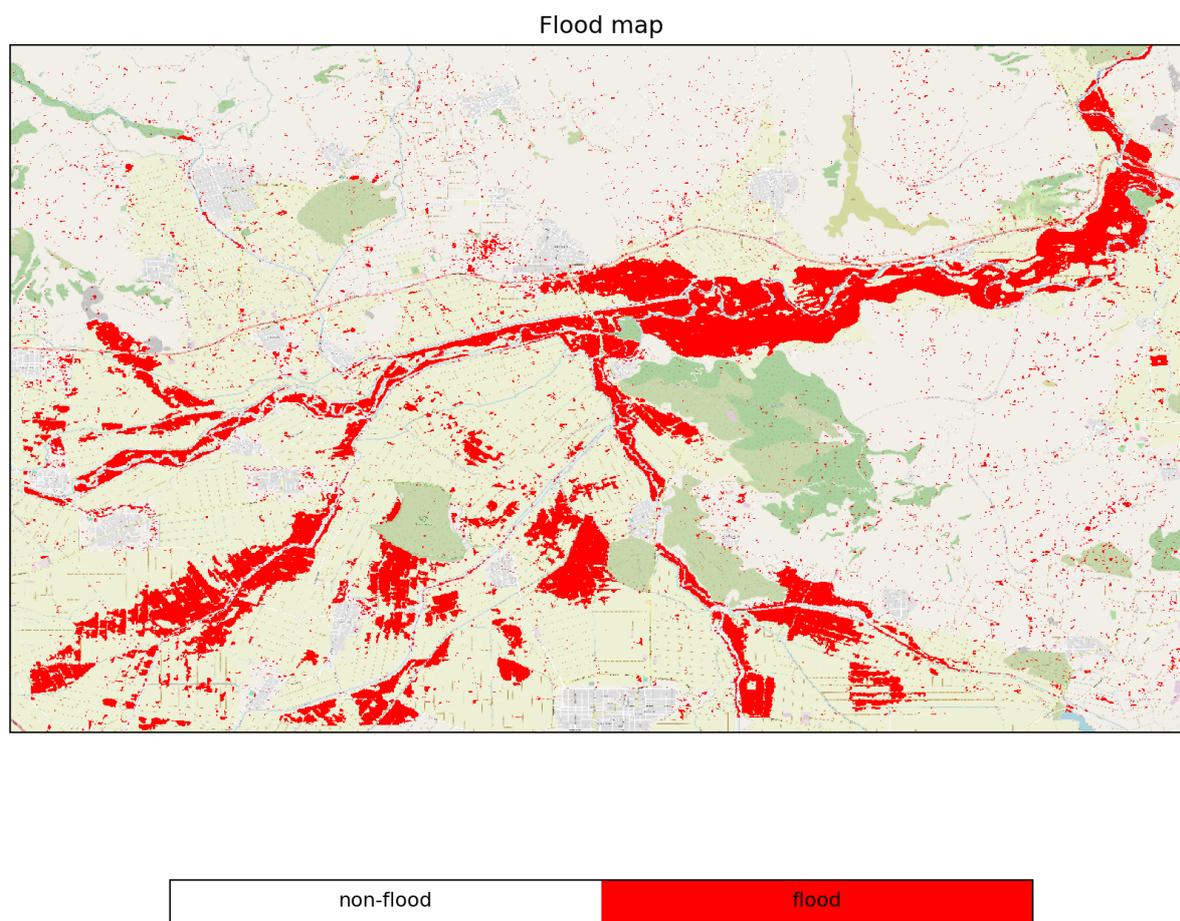


Figure 5 - Flood map for Thessaly (Greece) on 28th of February 2018 with openEO implementation of TU Wien flood mapping algorithm.

The core and thematic modules of interTwin will support the implementation of flood early warning systems anywhere on Earth and enable easy, rapid, and streamlined deployment of flood early warning digital twins.

2.5 Alpine droughts early warning

This DT aims at developing a drought early warning system for the Alpine region, providing seasonal forecasts of daily time series and maps of key hydrological variables such as surface soil moisture (SSM, %), actual evapotranspiration (ET, mm) and

streamflow (Q, m³/s). SSM and ET are produced at 1 km over the entire Alpine region whereas Q is produced at the outlets of the Alpine River basins (Figure 6).

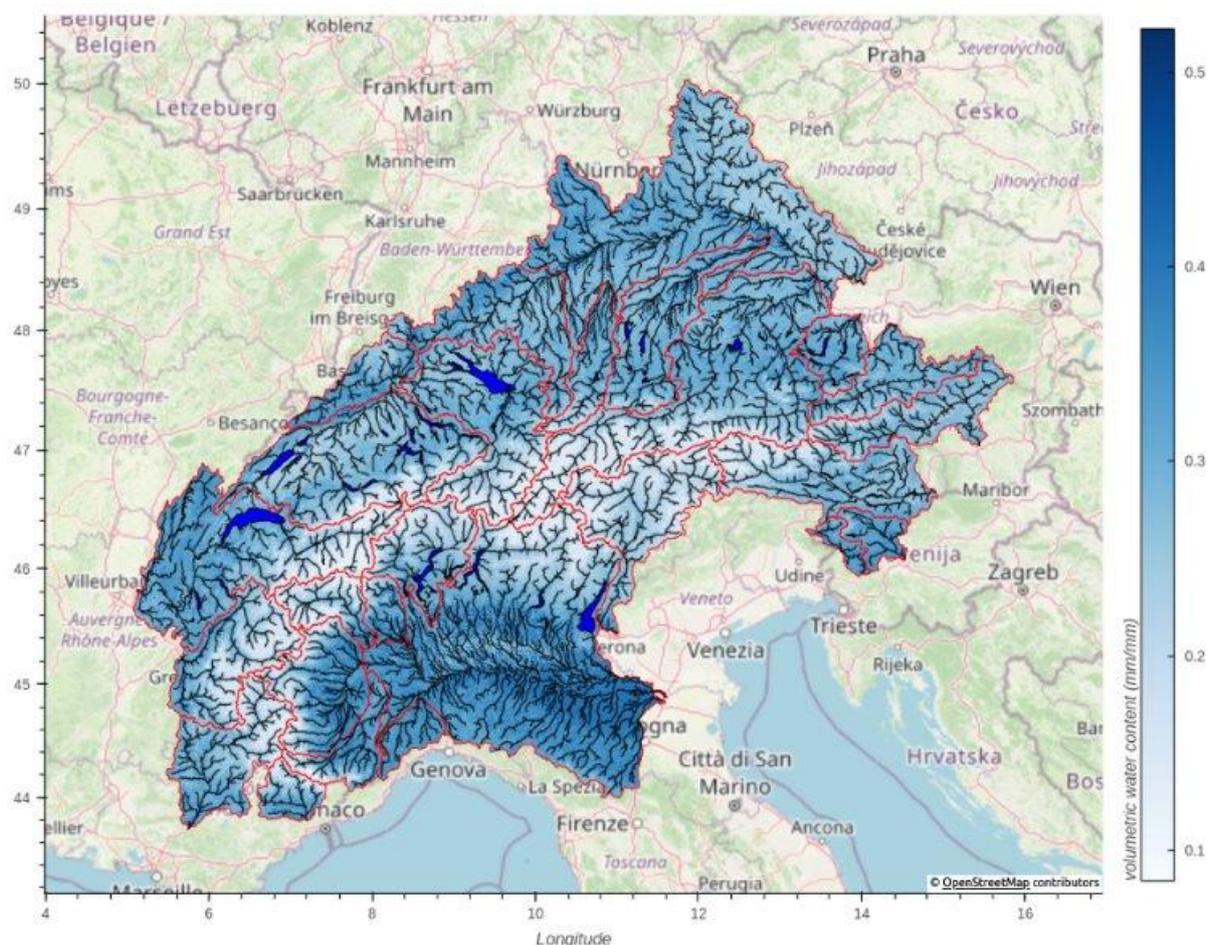


Figure 6 - interTwin region and river basins in red.

In summary, the DT receives meteorological variables from the ECMWF's SEAS5 seasonal forecast system [R3] and feeds them into a hydrological workflow to predict hydrological variables up to 6 months in the future. New seasonal forecasts are produced every beginning of the month, enabling the monitoring of the onset, propagation and termination of hydrological droughts.

The users will be able to set up and customise the DT's hydrological workflow through the openEO interface and run experiments and visualise the outputs by means of Jupyter Notebooks.

The DT draws inspiration from the recent progress in hybrid modelling, where the relative strengths of data-driven algorithms and physical knowledge are combined and complemented. In the current design, the injection of physical knowledge occurs in the initialisation of the data-driven model parameters, an approach that is known as physics-guided machine learning [R4]. The data-driven surrogate is trained to emulate the input-output mappings of a distributed hydrological model, wflow_sbm. The

surrogate inputs are wflow_sbm's effective parameters and meteorological variables. Once the surrogate successfully emulates wflow_sbm, it can be further fine-tuned by means of observations or, as it is currently done in the DT, it can replace wflow_sbm for performing extensive calibration of its parameters.

There are two properties of the physics-guided approach that are particularly advantageous: the computational performance and the flexible nature of deep learning architectures. The surrogate can run, in prediction, several orders of magnitudes faster than wflow_sbm, increasing the possibility of exploring larger regions of the parameter space to find optimal parameter sets. The deep learning models are also flexible as they can be adapted to different types and numbers of inputs, and they can be (relatively easily) composed and coupled with other neural networks or differentiable models [R5].

The DT hydrological workflow consists of several logically linked components (Figure 7), which are to some degree customisable by the users. The workflow starts with the ingestion of SEAS5 and ends with the prediction of hydrological droughts.

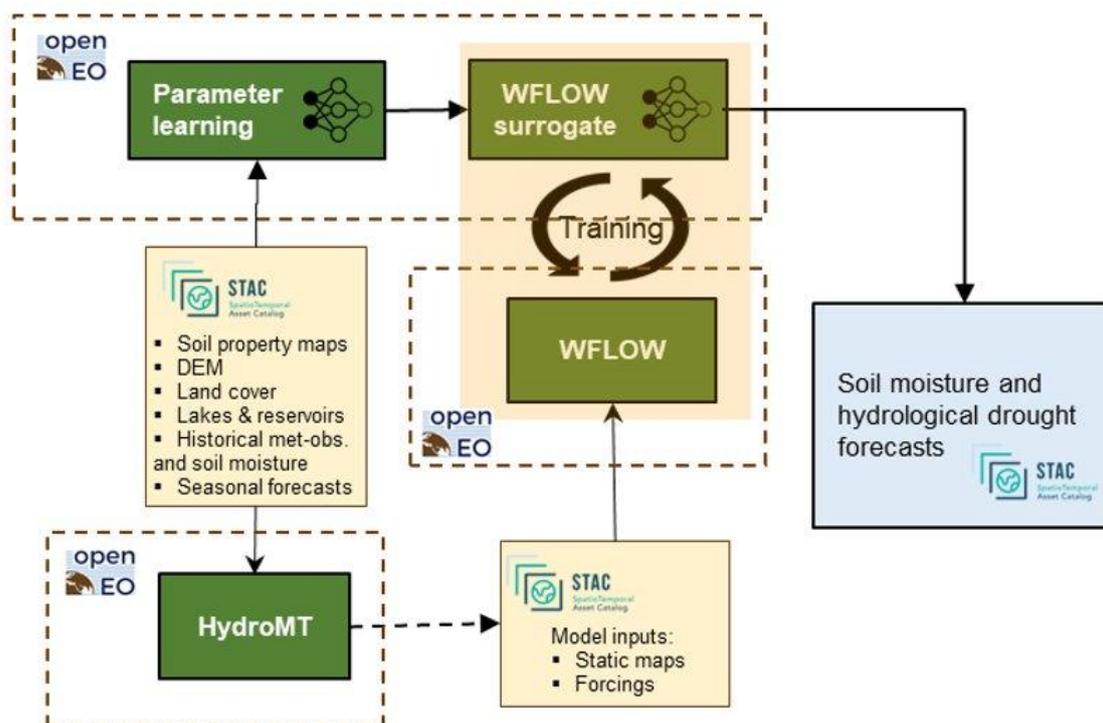


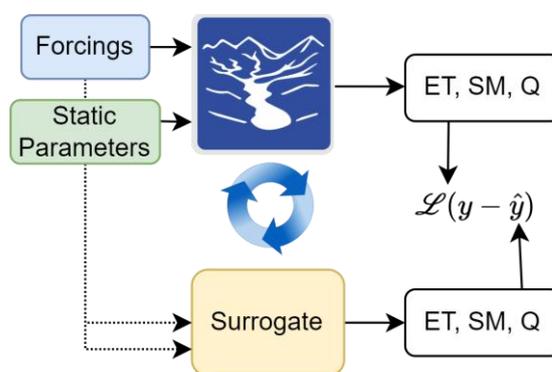
Figure 7 - DT hydrological workflow linked components diagram.

The *first component* of the workflow initially downscales meteorological variables from the SEAS5 forecast for better skills in the Alpine region using the 'downscaleML' python package. This package, stemming from WP7.4, currently includes routines for data preprocessing and for building statistical downscaling models for temperature and precipitation fields using a predefined dataset as a reference.

The *second component* is responsible for setting up and building the hydrological model, `wflow_sbm`, leveraging on the HydroMT⁶ application, a software for automating model building based on configuration files. The current release offers the option to select two precipitation inputs (E-OBS⁷ and CERRA-LAND⁸), to change model resolution, and to filter input data temporally and spatially.

The *third component* deals with running `wflow_sbm`, which is a semi-distributed hydrological model developed by Deltares and recently rewritten from Python to the Julia language, to boost performance. One of the strengths of the model is that most of the effective parameters can be estimated by pedo-transfer functions using readily available global datasets. The effective parameters of the model are conceptual constructs that can't be observed. Nonetheless they can be functionally derived from properties and attributes of the basin (i.e. topography, vegetation structure, etc.).

The *fourth component* is responsible for training the surrogate model. The surrogate is a Long Short-Term Memory (LSTM) Neural Network which has been outperforming established physically based hydrological models in streamflow prediction tasks [R6]. As `wflow_sbm` spatial support is a grid of 1 km resolution, the LSTM model is trained over a representative subsample of its grid cells, to improve the training efficiency (maximise learned information and minimise training time). In every training loop, LSTM learns both SSM and ET at each grid cell (Figure 8) and Q at the outlet of the river basin. The loss function is a weighted sum of the individual losses of SSM, ET and Q.



⁶ <https://github.com/Deltares/hydromt>

⁷ <https://cds.climate.copernicus.eu/cdsapp#!/dataset/insitu-gridded-observations-europe>

⁸ <https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-cerra-land>

D4.3 First version of the DTs capabilities for climate change and impact decision support tools

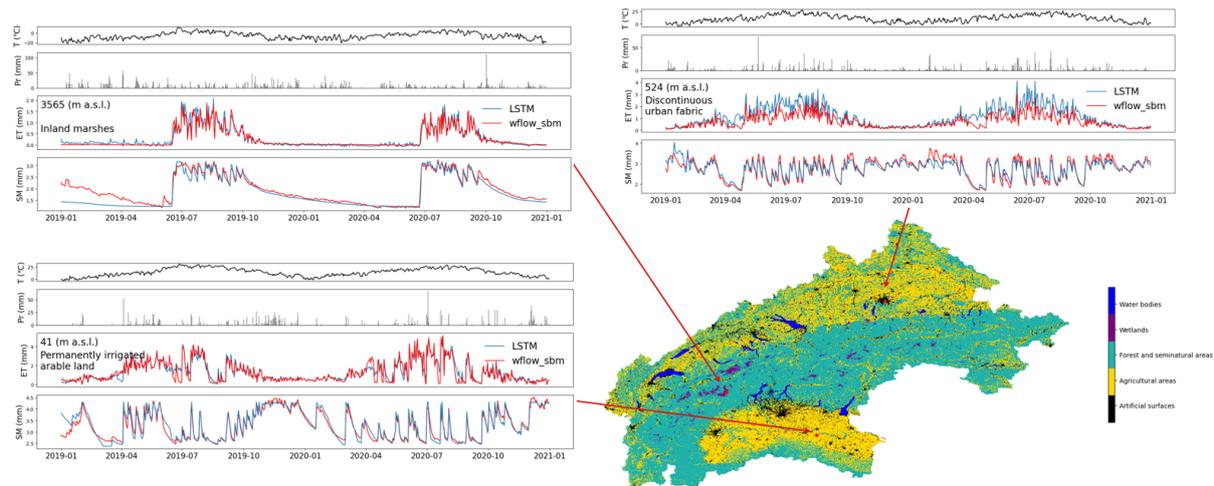
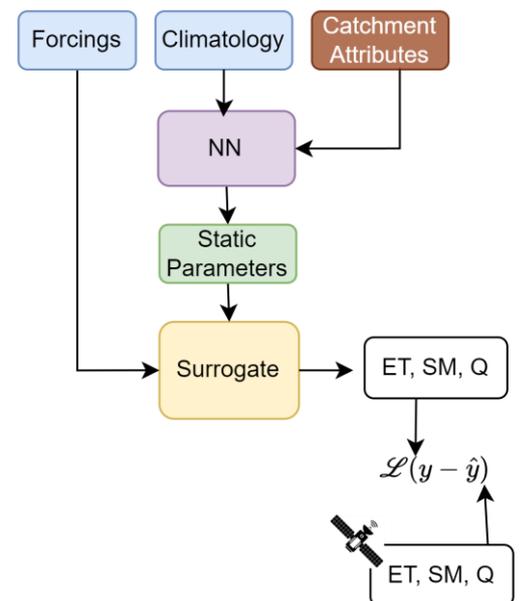
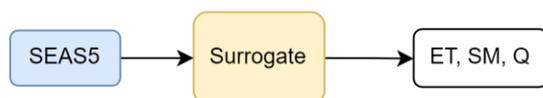


Figure 8 - Simulated Evapotranspiration and Surface Soil Moisture time series at three locations over the Alpine region.

The *fifth component* deals with the parameter learning task. This task enables the calibration of the wflow_sbm parameters by training the surrogate model coupled with a NN that is responsible for learning the transfer function between the catchment attributes and the effective parameters. In this task the surrogate model weights are frozen and a CNN encoder is trained end-to-end to produce the optimal set of effective parameters. The CNN encoder is learning a transfer function from observations to effective model parameters. The loss function is based on the MSE between the simulated SM and the satellite based SM retrievals (produced by TU Wien), and the simulated Q and the observed Q from the Alpine Drought Observatory (ADO)⁹ database.



Finally, the set of optimal parameters can be used with the surrogate or with wflow_sbm to predict the seasonal forecasts of SSM, ET and Q.



⁹ <https://ado.eurac.edu/>

2.6 Flood climate impact in coastal and inland regions

The DT for Flood Climate Impact in Coastal and Inland regions focus on simulating (compound¹⁰) flood events by producing their flood hazard and risk maps and assessing their impact on building, utilities, roads, and accessibility. Additionally, the DT enables end users to assess the efficacy of flood adaptation and mitigation measures for historic events and for future climate and socio-economic scenarios, such as sea level rise and population growth.

End users can define a geographic region of interest which will configure the necessary hazard and impact models and processing workflows using interTwin's core and thematic modules. Additionally, users can:

- Select weather events to be simulated
- Select future changes to scenarios including physical projections (such as sea level rise or land subsidence)
- Select socio-economic projections (such as population growth and economic growth),
- Select adaptation and mitigation measures to be implemented for a given scenario (e.g. Flood walls, pumps, levees, culverts, buyouts, flood proofing, and/or raising properties).

Aside from using data provided through interTwin's data lake, users also have the capability to add their own local datasets (e.g. local DEM, building footprints, critical infrastructure) into the DT workflow, enhancing the models' accuracy.

The DT model chain comprises four different models. Firstly, the flood hazard maps are produced by SFINCS, a reduced-complexity hydrodynamic model calculating flood extents. The user describes an event to simulate which determines the meteorological data used as boundary conditions for SFINCS and for the second model Wflow, a hydrological model calculating river discharges for SFINCS to use. The flood maps produced by SFINCS are then ingested into the third and fourth models, Delft-FIAT¹¹ and RA2CE¹², which for example assess the direct damage to buildings and roads and the compounding impact on infrastructure networks respectively. The climate and/or socio-economic scenarios and flood adaptation measures described by the user are translated into updated boundary conditions for these models by the FloodAdapt backend.

The end user is provided with Jupyter notebooks as an interface to the DT, which guides a user through the steps for configuring and running the DT. First, the user provides an

¹⁰ Compound flooding refers to a situation where multiple flooding sources combine to exacerbate the overall flood impact. These sources can include:

- Fluvial Flooding: Flooding from rivers and streams due to heavy rainfall or snowmelt.
- Coastal Flooding: Flooding from storm surges, high tides, or sea-level rise affecting coastal areas.
- Pluvial Flooding: Flooding caused by intense rainfall overwhelming drainage systems, not necessarily linked to a body of water like a river or sea.

When these sources coincide, their combined effects can lead to more severe flooding than would occur from any single source alone.

¹¹ <https://www.deltares.nl/en/software-and-data/products/delft-fiat-flood-impact-assessment-tool>

¹² <https://github.com/Deltares/ra2ce>



D4.3 First version of the DTs capabilities for climate change and impact decision support tools

area of interest based on which the models are configured (Figure 3 and Figure 9). Next, the user describes the event to run by selecting meteorological data and providing a description of a climate and/or socio-economic scenario. The event will then be run by the model chain. Finally, the user can visualise and interact with the output data, either through the provided visualisation notebook, examples of which are shown in Figure 10, or by accessing the output data directly.

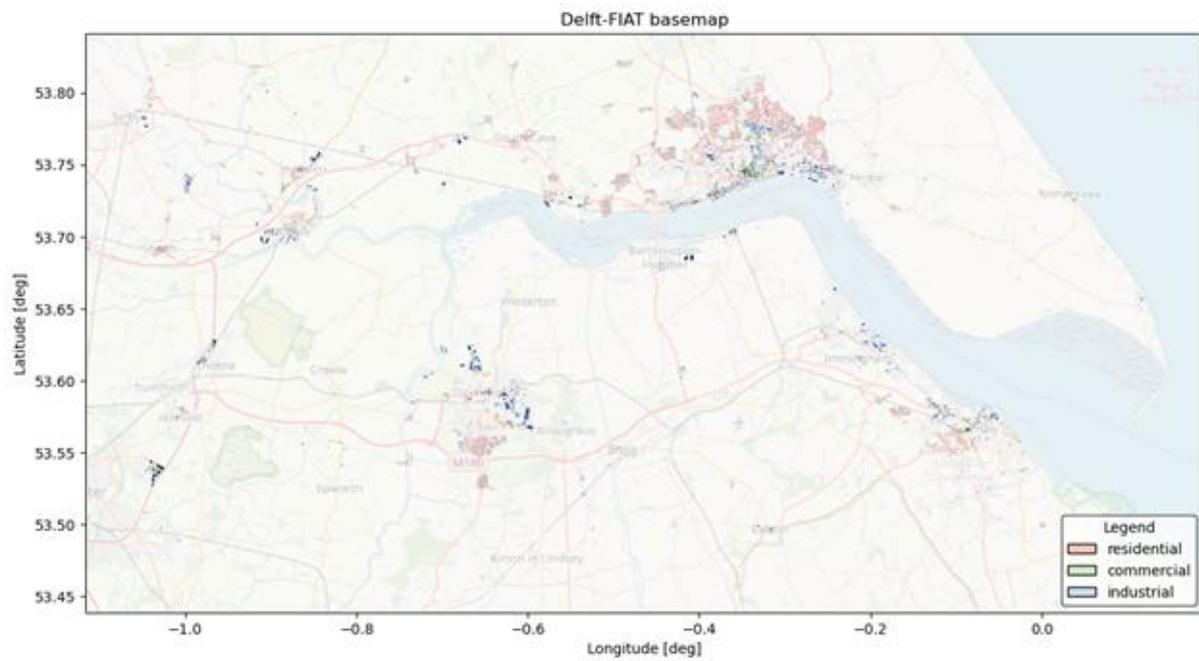


Figure 9 - Delft-FIAT basemap for the Humber estuary based on building footprints fetched from OpenStreetMap

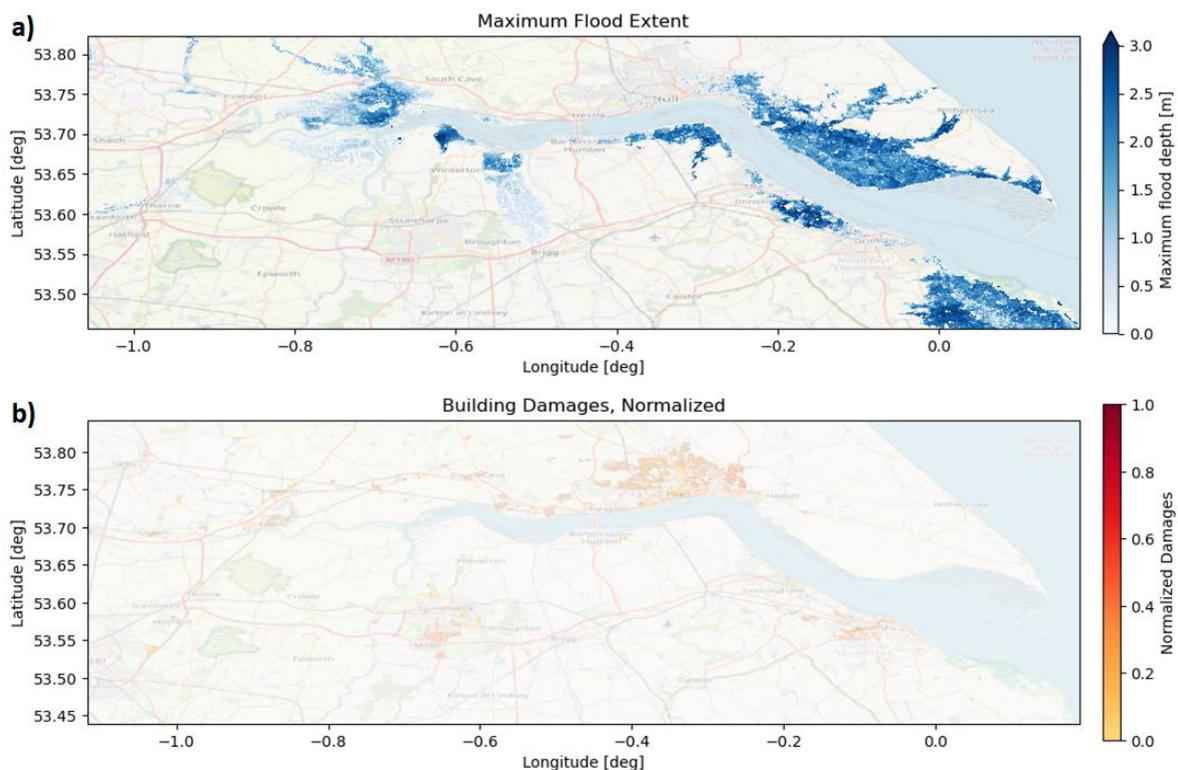


Figure 10 - a) Floodmap produced by SFINCS showing which areas are inundated and by how much. b) Damage map produced by Delft-FIAT based on a SFINCS floodmap.

3 DTs Application First release

3.1 Generic Detection of Climate Extremes

3.1.1 DTs Application Integrations

Model requirements

The ML model uses as input data a subset of daily climate variables from the CMIP6 dataset. The current implementation is for the maximum daily surface temperature, in order to address heatwaves. It will be extended to other temperature, precipitation and wind variables in the next implementation steps.

Table 1 - Initial set of variables considered for training.

Full name	CMIP6 variable name	Unit
Daily Maximum Surface Temperature	tasmax	Kelvin
Daily Total Precipitation	prtot	kg/m ² /s
Daily Near Surface Wind Speed	uvas	m/s

Workflow

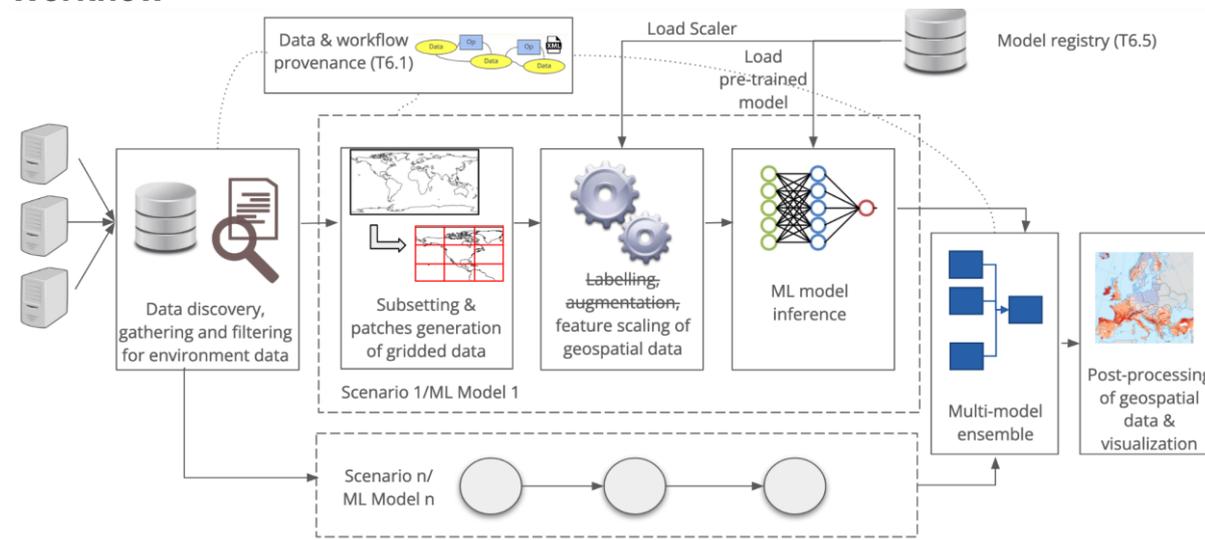


Figure 11 - Overview of the workflow for DT on extreme events.

The workflow about the Generic Detection of Climate Extremes DT will exploit the following components from the project:

- WP7: The workflow depends on the modules available in the *xtclim* thematic module¹³. The Jupyter notebook implementation will directly use the functions made available by the library.
- WP6: The integration has not started yet with the *itwinai* framework. Currently, only the preliminary work to prepare the integration has been done. The integration work will be gradually done during the remainder of the project;
- WP5: Data for training and inference will be accessed from the interTwin Data Lake based on RUCIO, possibly by using a data cataloguing tool (e.g., STAC or intake). Initial feasibility tests have been successfully performed to evaluate the access to a sample set of CMIP6 data using the RUCIO and Xarray Python modules.

3.1.2 Scope and limitations

The current implementation of this DT targets climate extremes and impacts that can be calculated daily, and that are related to climate indices and percentile-based indicators. In the future implementations it can be extended to climate extremes that involve a time dimension, such as droughts, or also to compound extreme events.

Main Limitations:

- Hyperparameters need to be optimised and adjusted for each atmospheric variable.
- Training must be over a region in which atmospheric variables are relatively homogeneous and must be done season by season and separately for each climate model, using a long enough time period in which the climate change signal is relatively low.
- Current configuration of the implementation is for a 32x32 input data array only.

3.2 Wildfire danger prediction in response to climate change

3.2.1 DTs Application Integrations

Model requirements

The ML model uses as input data a subset of climate and environmental variables from the SeasFire Cube datacube. The SeasFire Cube contains 59 seasonal fire drivers on a temporal extent of 21 years (2001-2021), with a 8-day temporal resolution and 0.25° spatial resolution. The initial set of variables considered for training is reported in **Table 2**. The set is still susceptible to change as the model evaluation progresses.

¹³ <https://github.com/cerfacs-globc/xtclim>



Table 2 - Initial set of variables considered for training.

Full name	SeasFire Cube name	Unit
Leaf Area Index	lai	unitless
Land-Sea mask	lsm	0-1
Land Surface temperature at day	lst_day	K
Relative humidity	rel_hum	%
Surface net solar radiation	ssr	MJ m-2
Sea Surface Temperature	sst	K
Temperature at 2 metres - Min	t2m_min	K
Total precipitation	tp	m
Burned Areas from FCCI	fcci_ba	ha

Workflow

Figure 12 depicts the high level workflow related to the digital twin use case on wildfires prediction and links with the project components/infrastructure.

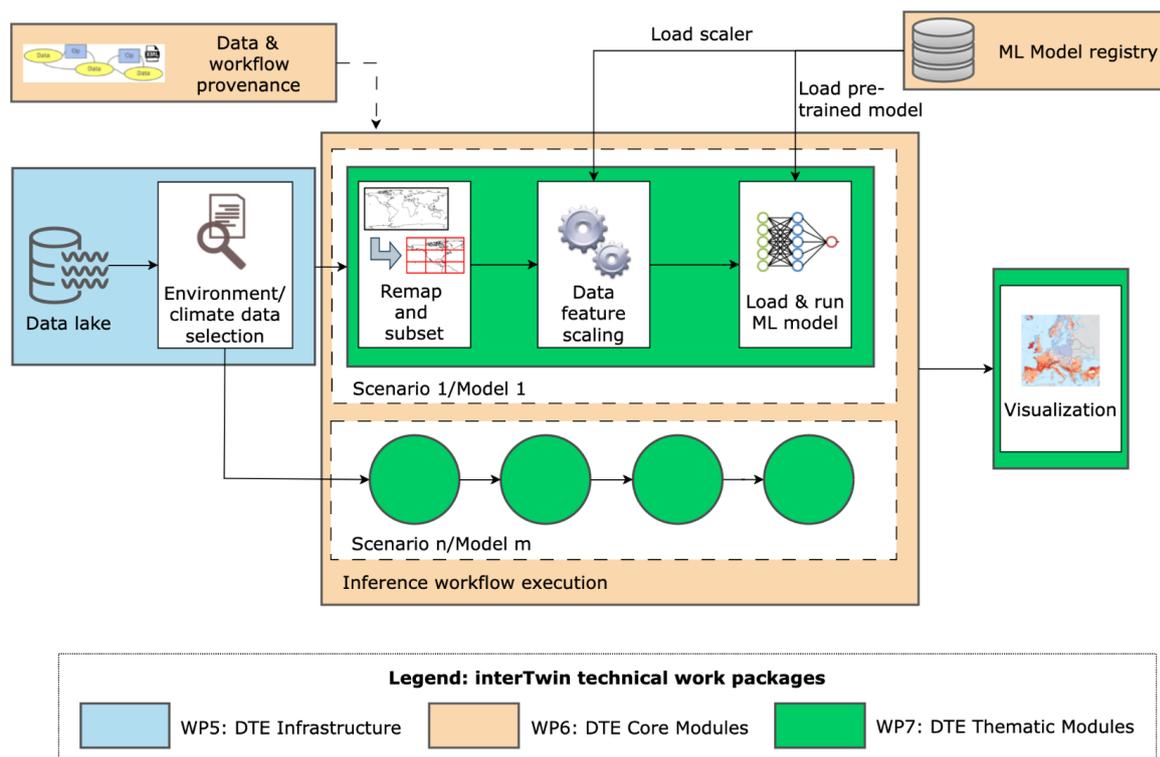


Figure 12 - Overview of the wildfires DT application.

In particular, the workflow about the wildfires DT has been exploiting (or will exploit) the following components from the project:

- WP7: The workflow depends on the functionalities from the *ML4Fires* thematic module¹⁴. The current implementation of the notebooks is strictly linked with the library from the Python module¹⁵.
- WP6:
 - Integration with a workflow management system (e.g., PyOphidia) and support for provenance will be added in the upcoming months;
 - Model training will also be integrated with the *itwina* framework;
 - Integration with the *SQaaS platform* will be evaluated in order to validate the workflow and/or the trained model quality;
- WP5: Data for training and inference will be accessed from the interTwin Data Lake based on RUCIO. Initial feasibility tests have been performed to evaluate access to the data via Python modules.

3.2.2 Scope and limitations

The goal is to provide Jupyter Notebooks for expert users (climate/environmental scientists) in order to:

1. Select:

¹⁴ <https://github.com/CMCC-Foundation/ML4Fires>

¹⁵ https://github.com/CMCC-Foundation/ML4Fires/tree/main/digital_twin_notebooks

- a. future projection scenarios from a given list;
 - b. temporal extents;
 - c. ML models from a set of pre-trained models;
2. Run the DT workflows on the selected input data and pre-trained models;
 3. Download/save as NetCDF and/or visualise the results. Maps and charts can be potentially customised through widgets. Different indicators can be provided, for example:
 - a. Average fire danger maps on a monthly/seasonal/annual basis,
 - b. Areas more affected by wildfires seasonally/annually (by threshold),
 - c. The trend of wildfire annually.

At the time of writing this document, testing of the model on climate projection data has not started yet. However, the model's successful prediction of burned area maps using 2019-2020 test data suggests that promising results could be achieved when applied to future projection data. Additionally, correlation analysis using the Pearson coefficient indicates that incorporating additional climate and atmospheric variables and exhibiting stronger correlations with the output variable, might be beneficial for future iterations of the model.

3.2.3 Preconditions

Users have access to DT data, pre-trained ML models, thematic components and Jupyter notebook, as well as the interTwin platform.

3.3 Tropical storms change in response to climate change

3.3.1 DTs Application Integrations

Model requirements

The ML model uses as input data a subset of climate and environmental variables from ERA5, combined with IBTrACS information to be used during the supervised training. The portion of ERA5 data is divided as follows:

- Training: 30 years — from 1980 to 2009 — 70% of the data
- Validation: 4 years — from 2010 to 2013 — 10% of the data
- Testing: 8 years — from 2014 to 2021 — 20% of the data

As reported in [Table 3](#), this data occurs with a 6-hour temporal resolution and a 0.25 degree spatial resolution. The set of variables considered for training is reported in [Table 3](#). The set is still susceptible to change as the model evaluation progresses.



Table 3 - Input parameters to the ML model.

Full name	Temporal Resolution	Spatial Resolution	ERA5 name	Unit
10 m wind gust since previous post-processing	6-hourly	0.25°x0.25°	fg10	m/s
instantaneous 10 m wind gust	6-hourly	0.25°x0.25°	i10fg	m/s
temperature at 500 mb	6-hourly	0.25°x0.25°	t_500	K
temperature at 300 mb	6-hourly	0.25°x0.25°	t_300	K
relative vorticity at 850 mb	6-hourly	0.25°x0.25°	vo_850	1/s
mean sea level pressure	6-hourly	0.25°x0.25°	msl	Pa

Workflow

Figure 13 shows the high level view of the workflow related to the digital twin use case on TCs detection and links with the project components/infrastructure.

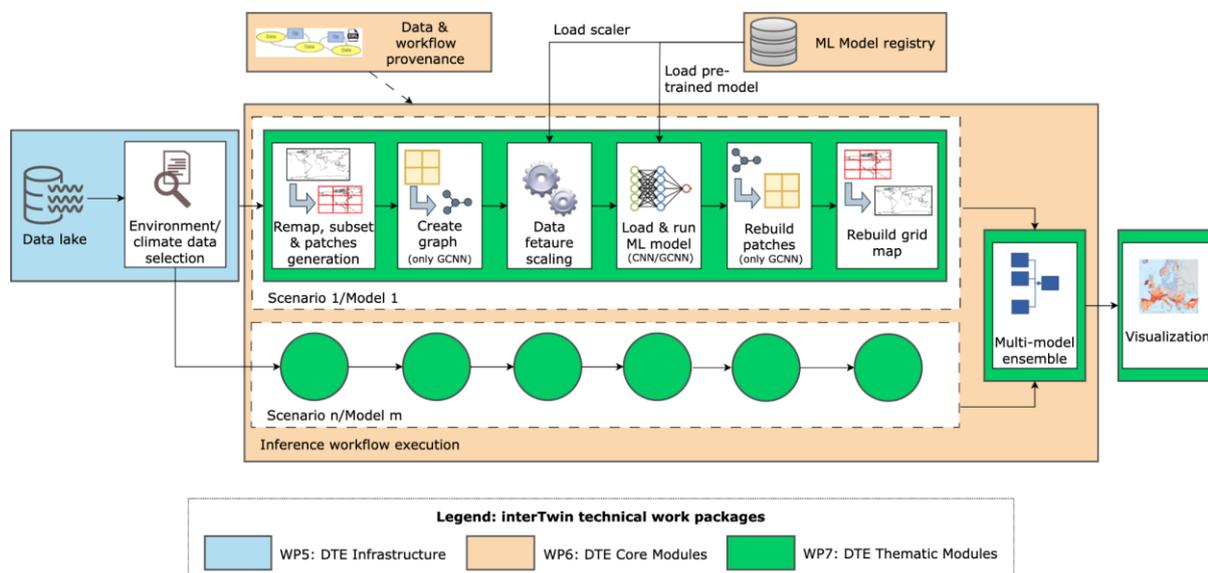


Figure 13 - Overview of the TC DT application.

In particular, the workflow about the tropical cyclones DT exploits (or will exploit) the following components from the project:

- WP7: The workflow depends on the modules available in the *ML TC Detection* thematic module¹⁶. The Jupyter notebook implementation uses directly the functions made available by the library.
- WP6:
 - Preliminary integration of the ML for training with the *itwinai* framework has been successfully performed¹⁷. The integration will be further strengthened during the remainder of the project;
 - Similarly to the wildfires use case, the integration with a workflow management system (e.g., PyOphidia) and support for provenance will be added and integration with the *SQaaS platform* will be evaluated;
 - For the CGNN workflow, the preliminary version of a provenance tracker developed at UNITN has been already integrated within the TC detection modules to track information about the ML process (i.e., loss values, system metrics, and more). This contribution is built on top of the yProv service developed in WP6¹⁸;
- WP5: Data for training and inference will be accessed from the interTwin Data Lake based on RUCIO, possibly by using a data cataloguing tool (e.g., STAC or intake). Initial feasibility tests have been performed to evaluate access to a sample set of ERA5 data using the RUCIO and Xarray Python modules.

3.3.2 Scope and limitations

The goal is to provide Jupyter Notebooks for expert users (climate/environmental scientists) in order to:

1. Select:
 - a. future projection scenarios from a given list;
 - b. temporal and geographical extents;
 - c. ML models from a set of pre-trained models, and when available the configuration of the ML ensemble (number of models to be involved);
2. Run the DT workflows on the selected input data and pre-trained models;
3. Download/save as NetCDF and/or visualise the results. Maps and charts can be potentially customised through widgets. Different indicators can be provided, for example:
 - a. Frequency of TCs occurrences on a seasonal/annual basis;
 - b. Trend of TCs per basin/year.

At the current time, the DT application had limited testing with climate projection data. Preliminary testing has been carried out on the pre-trained CNN models on a few models

¹⁶ <https://github.com/CMCC-Foundation/ml-tropical-cyclones-detection/tree/torch>

¹⁷ https://github.com/interTwin-eu/itwinai/tree/cyclone_tf_dist/use-cases/cyclones

¹⁸ <https://github.com/HPCI-Lab/yProv>



from the CMIP6 experiments. This activity will continue in the following months and will also be extended to the graph-based GCNN model.

3.3.3 Preconditions

Users have access to DT data, pre-trained ML models, thematic components and Jupyter notebook, as well as the interTwin platform.

3.4 Flood early warning in coastal and inland regions

This DT application leverages on the models and tools listed in the next section. These are open source and can be found at the respective repositories listed in D7.1¹⁹.

Jupyter Notebooks and CWL workflows are available at the project's repositories²⁰ and describe how to set up the necessary models and run them for the DT application.

The notebooks and workflows are still in development, with the main objective to streamline and simplify them for end-users by leveraging on the interTwin DTE.

3.4.1 DTs Application Integrations

Model requirements

The flood early warning DT for coastal and inland regions relies on two process-based models combined with satellite observations of floods:

1. Super-Fast INundation of CoastS (SFINCS): a reduced-complexity model designed for super-fast modelling of compound flooding in a dynamic way.
2. Wflow: a framework for modelling hydrological processes, allowing users to account for precipitation, interception, snow accumulation and melt, evapotranspiration, soil water, surface water and groundwater recharge in a fully distributed environment.
3. HydroMT: an open-source Python package that facilitates the process of building and analysing spatial geoscientific models with a focus on water system models. It does so by automating the workflow to go from raw data to a complete model instance which is ready to run and to analyse model results once the simulation has finished.
4. openEO satellite-based flood monitoring: an existing workflow for flood monitoring has been re-developed in the openEO syntax ensuring interoperability on several platform backends. Publicly available datasets stored at the EODC facilitate the openEO implementation of Sentinel-1based flood maps. These datasets are published as SpatioTemporal Asset Catalogues (STAC) — a common language to describe geospatial information, which ensures discoverability and interoperability of the data. The STAC catalogues and openEO platform at EODC

¹⁹ D7.1 Report on requirements and thematic modules definition for the environment domain
<https://doi.org/10.5281/zenodo.10417158>

²⁰ <https://github.com/interTwin-eu/DT-flood/tree/main/Notebooks>;
https://github.com/interTwin-eu/DT-flood/tree/main/DT_flood/workflows



can be accessed with common HTTP verbs, and clients are available for Python, R, and JavaScript. The openEO implementation consists of three datasets (or "collections" in openEO terminology): 1) the σ_0 backscatter data projected in Equi7Grid at 20 meter pixel spacing, 2) the projected local incidence angle (PLIA) values of those measurements, and 3) the harmonic parameters of a model fit on the pixel's backscatter time series over land. This harmonic model fitted on historical data describes typical seasonal Sentinel-1 σ_0 backscatter variation on a 20 meter pixel level. These three collections of the STAC catalogue are respectively designated as: "SENTINEL1_SIG0_20M", "SENTINEL1_HPAR", and "SENTINEL1_MPLIA".

Workflow

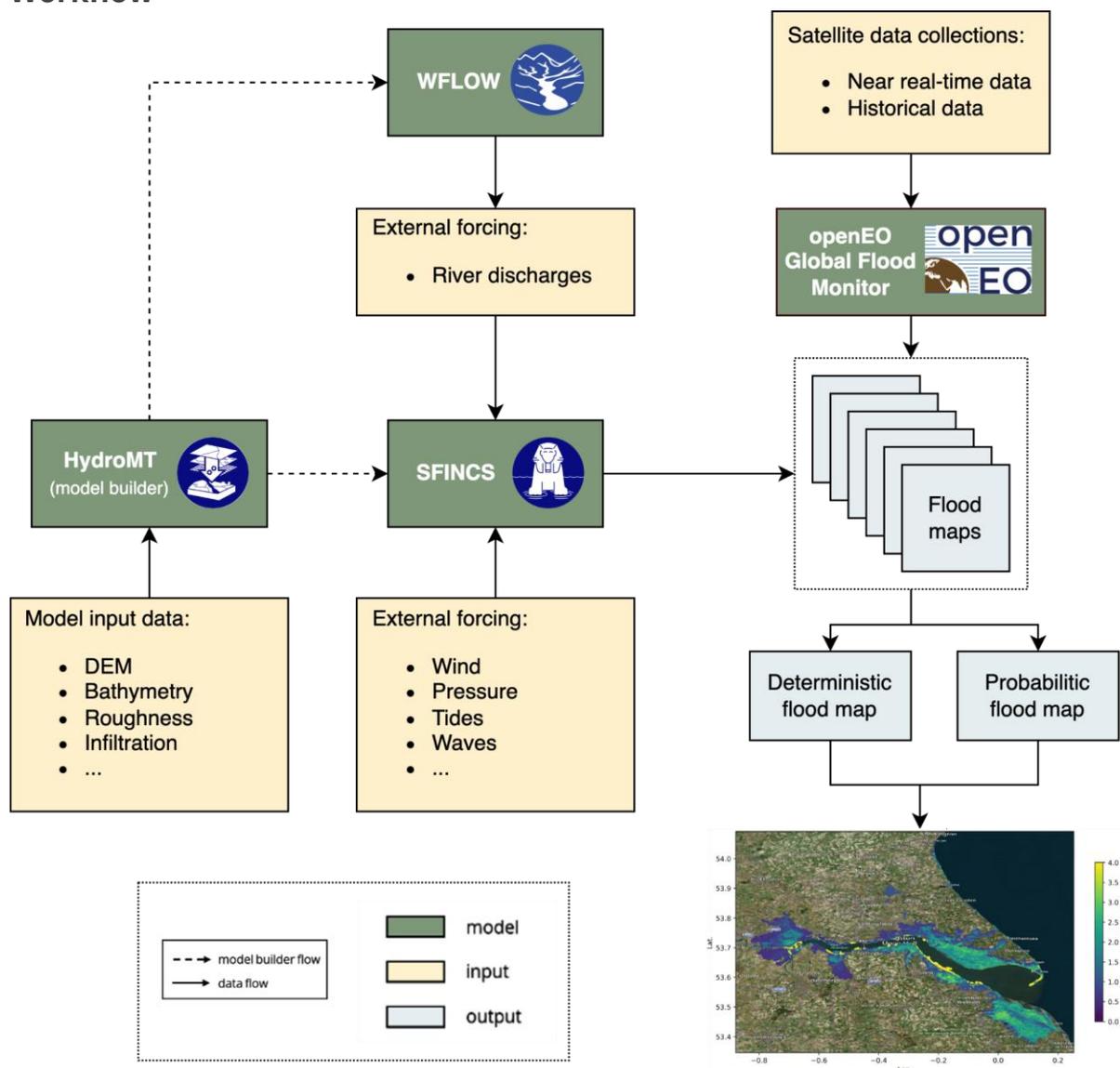


Figure 14 - High-level workflow diagram for the flood early warning DT.

The workflow for the flood early warning DT is implemented in a Jupyter Notebook ([Figure 18](#) in the [Section on the flood climate impact DT](#)) and has been exploiting / will exploit the following components from the project:

- WP7: The workflow depends on functionalities from the hydrological model data processing thematic module dependent on HydroMT²¹, SFINCS²², and Wflow²³. Upcoming developments include:
 - Addition of a workflow for probabilistic flood maps from SFINCS.
 - Improved downscaling approach for SFINCS.
 - Addition of validation methods for SFINCS.
 - Examples and documentation for using local data in SFINCS.
- WP6:
 - CWL workflows have been developed integrating HydroMT, SFINCS and Wflow to preprocess and run WFLOW and SFINCS to generate flood maps.
 - Discussions are ongoing regarding interTwin's workflow composition interface and execution engine; integration of the CWL workflows will occur after the conclusion of these discussions.
 - An exploratory activity is underway to understand how to exploit OSCAR, an open-source platform for serverless event-driven data processing of containerized applications for this DT.
- WP5: Data for model building and preprocessing will be accessed from the interTwin Data Lake based on RUCIO, preferably by using a data cataloguing tool such as STAC. Example datasets have been uploaded for SFINCS and Delft-FIAT and tests are ongoing to access data for workflows.

3.4.2 Scope and limitations

The goal is to provide Jupyter Notebooks for expert users in order to:

1. Set up the necessary models for a user-defined region of interest.
2. Run the necessary models and Earth Observation processing pipelines to produce deterministic and probabilistic flood maps for a user-defined region of interest and validate the resultant output data against observations.
3. Prepare the data for easy ingestion into an early warning system.

A demonstration will be provided for Humber, UK.

The main limitation for Sentinel-1 based flood maps are the availability of suitable overpasses of the satellite. This means that flood events, or the peak flooding extent of an event, can be missed.

3.4.3 Preconditions

Users have access to DT data, models, thematic components and Jupyter Notebooks.

1. Users can:

²¹ <https://github.com/Deltares/hydromt>

²² <https://github.com/Deltares/SFINCS>

²³ <https://github.com/Deltares/Wflow.jl>



- a. specify a region of interest,
 - b. specify a temporal period to simulate,
 - c. select local data for the models if available.
2. User runs the DT workflows for the specified region and period using default global data or selected local data if available.
 3. The output of the DT can be visualised in the Jupyter Notebooks and the data can be downloaded/saved as NetCDF data.

3.5 Alpine droughts early warning

3.5.1 DTs Application Integrations

The datasets required by the DT should be available on STAC.

The inputs of the wflow_sbm hydrological model are produced and processed from dynamical meteorological forcings and static parameter maps by running the HydroMT component. These inputs are open source datasets (**Table 4**) that are collected and registered into a Spatio-Temporal Assets Catalog.

Table 4 - Input Parameters from Open Source Datasets

Collection/Dataset	Parameter	Source	STAC
CERRA Land	total_precipitation	https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-cerra-land?tab=overview	stac.eurac.edu/collections/CERRA_LAND
	surface_solar_radiation_downwards		
	orography		
CERRA	2m_temperature	https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-cerra-single-levels?tab=overview	stac.eurac.edu/collections/CERRA
	total_precipitation		
	surface_pressure		
E-OBS	precipitation	https://cds.climate.copernicus.eu/cdsapp#!/dataset/insitu-gridded-observations-europe?tab=form	stac.eurac.edu/collections/EOBSv28
SoilGrid 2000	Bulk density	https://files.isric.org/soilgrids/lat/est/data/bdod	stac.eurac.edu:8080/collections/SOILGRIDS
	Organic carbon	https://files.isric.org/soilgrids/lat/est/data/soc	
	Clay	https://files.isric.org/soilgrids/lat/est/data/clay	
	Silt	https://files.isric.org/soilgrids/lat/est/data/silt	
	Sand	https://files.isric.org/soilgrids/lat/est/data/sand	
	pH	https://files.isric.org/soilgrids/lat/est/data/phh2o	



	Soil thickness	https://files.isric.org/soilgrids/for-mer/2017-03-10/data/BDTICM_M_250m_ll.tif	
Ecodatacube	land cover class	http://s3.eu-central-1.amazonaws.com/eumap/lcv/lcv_l_andcover.hcl_lucas.corine.rf_p_30m_0..0cm_2018_eumap_epsg3035_v0.1.tif	No
Merit hydro	flow direction	https://hydro.iis.u-tokyo.ac.jp/~yamadai/MERIT_Hydro/	stac.eurac.edu/collections/MERIT_HYDRO
	elevation		
	HAND		
	upstream area		
	river width		
HydroLakes	-	https://www.hydrosheds.org/products/hydrolakes	No
Grand v1.3	-	https://ln.sync.com/dl/bd47eb6b0/anhxaikr-62pmrgtq-k44xf84f-pyz4atkm/view/default/447819520013	No
Randolph Glaciers inventory	-	https://cds.climate.copernicus.eu/cdsapp#!/dataset/insitu-glaciers-extent?tab=overview	No
Roughness river mapping	-	https://github.com/Deltares/hydromt_wflow/tree/main/hydromt_wflow/data/wflow	No
Corine mapping	-	https://github.com/Deltares/hydromt_wflow/tree/main/hydromt_wflow/data/lulc	No
river_ge30m	-	https://zenodo.org/records/3552776#.YVbOrppByUk	No

The processed input-output of wflow_sbm is required for training the surrogate model (Table 5).

Table 5 - Processed input-output of wflow_sbm.

Collection/ Dataset	Parameter	Source	STAC
Wflow sbm forcings	Precipitation	Processed by HydroMT	No
	PET	Processed by HydroMT	No
	Temperature	Processed by HydroMT	No
Wflow sbm static maps	effective parameters²⁴	Processed by HydroMT	No

²⁴ https://deltares.github.io/Wflow.jl/stable/model_docs/params_vertical/



Wflow sbm outputs	Actual evapotranspiration (ET)		No
	Surface Soil Moisture (SSM)		No
	River Discharge (Q)		No
	Runoff (R)		No

The datasets for the parameter learning task are currently under review and will be documented in the next release.

Workflow

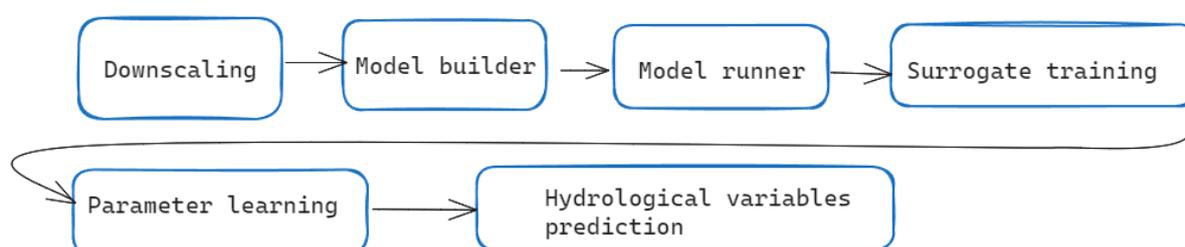


Figure 15 - High level diagram of the DT workflow's application components.

Downscaling component (WP 7.4). DownscaleML

To streamline the downscaling process using downScaleML²⁵ version 1.0.0 (WP 7.4), we've developed preprocessing and downscaling methodologies that use reanalysis datasets as both predictors and references. The current version introduces grid cell-wise statistical downscaling, employing ERA5 predictor fields and incorporating CERRA and CERRA-LAND reanalysis data as targets for 2m temperature and precipitation, respectively. This approach utilises the LGBM regressor and grid search capabilities for fine-tuning hyperparameters. Furthermore, we are currently refining and assessing seasonal forecast downscaling for upcoming releases. Our upcoming efforts are focused on seamlessly integrating with itwinAI, STAC, CWL, and OpenEO.

²⁵ <https://github.com/interTwin-eu/downScaleML>



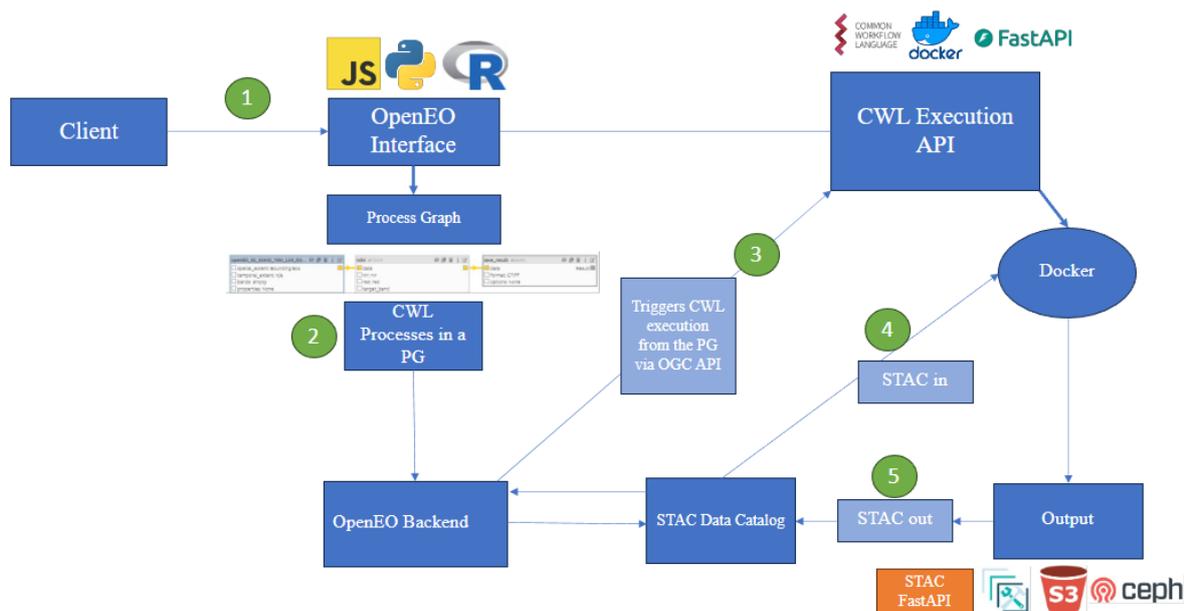


Figure 16 - High level diagram that shows how DT's application components are run.

Model builder and model runner components (WP 7.6).

The model builder (HydroMT) and model runner (wflow_sbm) are available as an OGC Application Package in the HyDroForm²⁶ repository. The integration of HydroMT with STAC has been developed in a forked development branch²⁷ waiting to be tested and integrated officially in the version 1.0.0 of HydroMT. The implementation currently allows the generation of wflow_sbm model with different spatial resolutions and precipitation inputs available on the STAC catalog.

The plans for the future release are: 1) work with Deltares for the official implementation of STAC support in HydroMT 1.0.0, 2) work on the OpenEO integration (WP 6.1, 6.3), consisting in the definition of OpenEO processes that call the OGC Application Package.

Surrogate training component.

The deep learning surrogate has been successfully developed and it is available in the project repository Hython²⁸, where demo notebooks are available for the interested users. The integration with ItwinAI (WP 6.5) framework is completed for the preprocessing and training steps.

The plans for the future release are, 1) to additionally emulate discharge on top of SSM and ET, 2) to work on the integration with CWL, STAC and OpenEO, 3) to work on the integration of the prediction step in ItwinAI.

Parameter learning component.

The requirements for the production of surface soil moisture (SSM) data to perform the parameter learning task have been shared with TU Wien. TU Wien then produced a first version of SSM over the Alpine region (WP 7.5).

²⁶ <https://github.com/interTwin-eu/HyDroFormM>

²⁷ <https://github.com/iacopoff/hydromt>

²⁸ <https://github.com/interTwin-eu/hython>

This initial version uses a radiative transfer model to derive soil moisture data from 500 m microwave backscatter and Leaf Area Index (LAI). For enhancing the accuracy of the soil moisture retrievals high-resolution information has been included from Sentinel-1 VV backscatter data (20 m). A novel soil moisture sensitivity dataset [R8] was used for a static spatial filtering technique at the 20 m scale. Also, extreme backscatter values (higher than -5dB or lower than -19dB) have been filtered out, assuming that they are unlikely to contain reliable soil moisture information. Both strategies amplify the soil moisture signal of the 500 m backscatter information, to which the 20m sub-pixels have been aggregated subsequently. Only those 500m target pixel were calculated that consisted of more than 1% valid 20m pixels after the described filter processes.

Based on this 500 m backscatter datacube and LAI data, the radiative transfer model parameters have been calibrated using 4 years of data from 2016 up to 2020 with ERA5-Land swvl1 as reference data. During calibration, the frozen soil conditions indicated by ERA5-Land soil temperature layer 1 (stl1) and snow depth (sd) have been masked out to make sure only valid data is used during model inversion.

After calibration, the parameters were used to estimate SSM at a 500 m resolution, without further masking to avoid losing valid data, especially over alpine valleys. This decision requires users to manually mask out data from frozen soils, and over steep and complex terrain. Despite its limitations in handling complex terrains, the initial version shows promising results, particularly in alpine valleys. By incorporating volume scattering using LAI and masking soil moisture insensitive pixels at high 20 m resolution, we anticipate significant improvements in SSM retrievals compared to earlier models.

The plans for the future release are: 1) Select input datasets, 2) design and develop the parameter learning workflow

Seasonal forecast component.

The component is planned to be developed in future releases.

3.5.2 Scope and limitations

The goal is to provide a user interface, consisting in Jupyter Notebooks or others, that allows:

1. Set up the wflow_sbm model, selecting the model resolution, choosing different data inputs, time period and spatial domain.
2. Training the surrogate model.
3. Perform the parameter learning.
4. Run seasonal forecasts of hydrological variables.

The DT presents three main limitations:

- The DT predicts hydrological variables based on the type and level of detail of hydrological processes represented in the wflow_sbm model. This fact limits the variety of hydrological processes that could instead be better represented by running an ensemble of hydrological models.

- The Alpine region, with its complex terrain, vegetation cover and diverse climatic conditions, poses a great challenge for the retrieval of surface soil moisture and actual evapotranspiration from satellites. It is no surprise then that there are currently no products that satisfactorily cover the region. The products generated for the DT should therefore be considered as prototypes that require further research, developments and validation. The parameter learning task's loss function is based on these satellite products and therefore depends on their quality.
- The DT does not represent human processes, such as irrigation and hydropower generation, that affects the hydrologic cycle.

3.6 Flood climate impact in coastal and inland regions

3.6.1 DTs Application Integrations

Model requirements

The flood climate impact DT consists of two physics-based models, two impact models, and a python package facilitating processing the models:

1. Super-Fast INundation of CoastS (SFINCS): a reduced-complexity model designed for super-fast modelling of compound flooding in a dynamic way.
2. Wflow: a framework for modelling hydrological processes, allowing users to account for precipitation, interception, snow accumulation and melt, evapotranspiration, soil water, surface water and groundwater recharge in a fully distributed environment.
3. Delft-FIAT: a fast, flexible, Python-based tool to rapidly assess direct economic impacts to buildings, utilities, and roads for user-input flood maps.
4. RA2CE: a Resilience Assessment and Action perspective for Critical infrastructureE – model for mapping the exposure, criticality, and vulnerability as well as the forthcoming prioritisation of locations to take actions based on cost benefit assessment.
5. HydroMT: an open-source Python package that facilitates the process of building and analysing spatial geoscientific models with a focus on water system models. It does so by automating the workflow to go from raw data to a complete model instance which is ready to run and to analyse model results once the simulation has finished.

Workflow

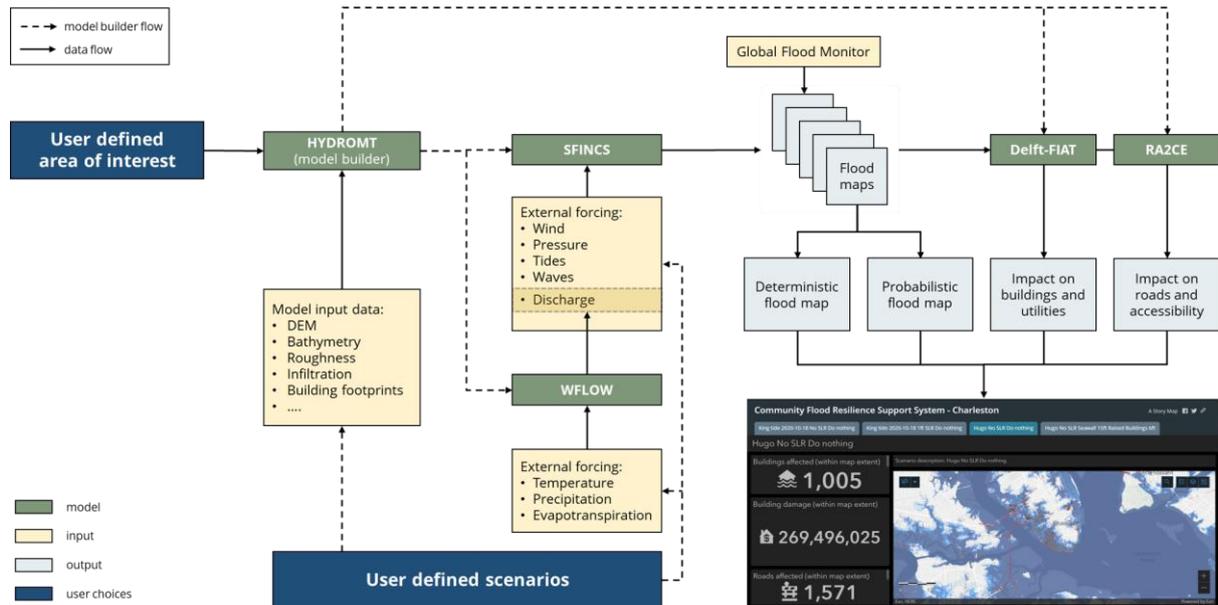


Figure 17 - High-level workflow diagram for the flood early warning DT.

The workflow for the flood climate impacts DT (Figure 17) is implemented in a Jupyter Notebook (Figure 18) and is exploiting / will exploit the following components from the project:

- WP7: The workflow depends on functionalities from the hydrological model data processing thematic module dependent on HydroMT²⁹, SFINCS³⁰, and Wflow³¹. Upcoming developments include:
 - Addition of RA2CE³² to the workflow.
 - Addition of options for green infrastructure for SFINCS.
 - Examples and documentation for using local data in Delft-FIAT.
 - Docker container for Delft-FIAT.
 - Improved quantification of Nature-based Solutions for Wflow.
- WP6:
 - CWL workflows have been developed integrating HydroMT, SFINCS and Wflow to preprocess and run WFLOW and SFINCS to generate flood maps.
 - Discussions are ongoing regarding interTwin's workflow composition interface and execution engine; integration of the CWL workflows will occur after the conclusion of these discussions.
 - An exploratory activity is underway to understand how to exploit OSCAR, an open-source platform for serverless event-driven data processing of containerized applications for this DT.

²⁹ <https://github.com/Deltares/hydromt>

³⁰ <https://github.com/Deltares/SFINCS>

³¹ <https://github.com/Deltares/Wflow.il>

³² <https://github.com/Deltares/ra2ce>

D4.3 First version of the DTs capabilities for climate change and impact decision support tools

- WP5: Data for model building and preprocessing will be accessed from the interTwin Data Lake based on RUCIO, preferably by using a data cataloguing tool such as STAC. Example datasets have been uploaded for SFINCS and Delft-FIAT and tests are ongoing to access data for workflows.

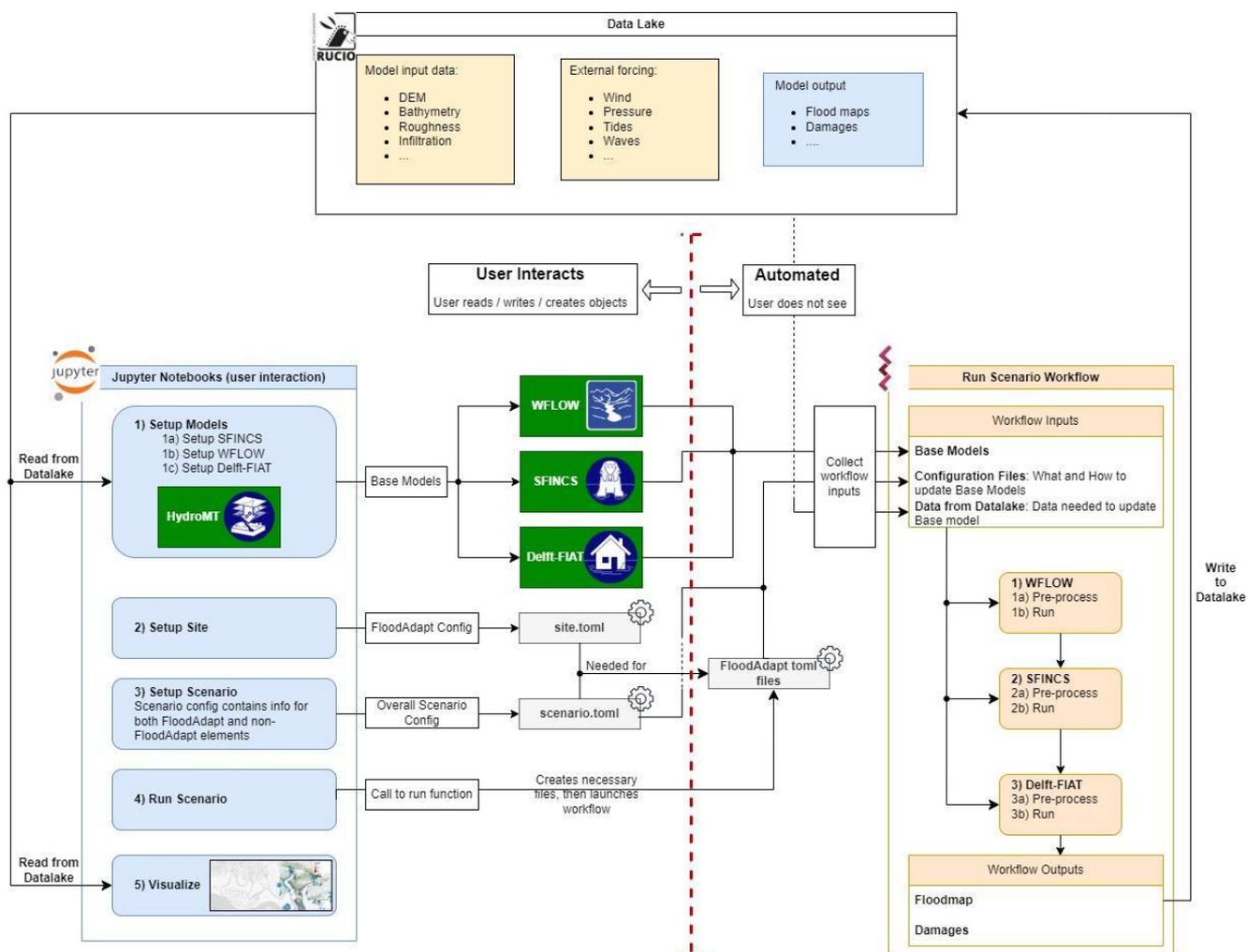


Figure 18 - Architecture diagram highlighting integration and interaction between the Jupyter Notebook (the DT interface), the CWL workflows and interTwin's RUCIO-based datalake.

3.6.2 Data requirements

The datasets currently used by the DT are listed in **Table 6**. These include both static parameters maps for SFINCS, WFLOW, Delft-FIAT and dynamic meteorological forcing data for SFINCS, WFLOW.

Table 6 - Datasets used by the DT

Dataset Name	Source	Parameter Name
ERA5	https://doi.org/10.24381/cds.bd0915c6	2m Temperature
		Mean Sea Level Pressure
		10m Wind u-component



		10m Wind v-component
		Total Precipitation
		Surface net Solar Radiation
		Geopotential
Copernicus 30m DEM	https://spacedata.copernicus.eu/documents/20123/121286/Copernicus+DEM+Open+HTTPS+Access.pdf	Topography
GEBCO	https://www.gebco.net/	Bathymetry
GTSM reanalysis	https://doi.org/10.24381/cds.8c59054f	Waterlevel
		Surge
Hydro Atlas	https://www.hydrosheds.org/hydroatlas	Basin atlas
ESA worldcover	https://doi.org/10.5281/zenodo.5571936	Landclass
GCN250 Infiltration curves	https://doi.org/10.6084/m9.figshare.7756202.v1	Average antecedent
		Dry Antecedent
		Wet Antecedent
MERIT_HYDRO	http://hydro.iis.u-tokyo.ac.jp/~yamadai/MERIT_Hydro	flwdir
		strord
		uparea
MODIS Leaf Area Index	https://lpdaac.usgs.gov/products/mcd15a3hv006/	lai
Soilgrids	https://www.isric.org/explore/soilgrids/faq-soilgrids-2017	soilmaps
OpenStreetMap	https://osmnx.readthedocs.io/en/stable/	Building footprints
GADM	https://gadm.org/data.html	level 3

3.6.3 Scope and limitations

This use case will provide Jupyter Notebooks for expert users in order to

1. Set up the necessary models for a user-defined region of interest.
2. Run the necessary models to produce baseline flood maps for a user-defined region of interest and quantify impacts and damages to buildings, utilities, roads and accessibility.
3. Select flood mitigation and adaptation measures and re-run flood scenarios to test their effectiveness at reducing flood-related impacts.

A demonstration will be provided for Humber, UK.



3.6.4 Preconditions

The user has access to DT data, models, thematic components and Jupyter Notebooks.

Users can:

- specify a region of interest,
- specify a temporal period to simulate,
- select local data for the models, if available,
- select and specify mitigation and adaptation measures.

The user runs the DT workflows for the specified region and scenario using default global data or selected local data if available

The output of the DT can be visualised in the Jupyter Notebooks and the data can be downloaded/saved as NetCDF and GeoPackage data.



4 Conclusions

The first release of the interTwin DTs applications for WP4 concerned with the environmental domain was developed during the first 18 months of the project. In the current deliverable D4.3, the main focus is on specifying the capabilities of each specific DT and describing the current integration with the Infrastructure Components provided by WP5 and the Core Components provided by WP6, along with schematic high-level workflows. The current scope and limitations of this first release are also identified.

The next step will be to complete the development and the integration of this first release of each DT Application. This deliverable is a first overview of the future prototypes of the DT applications for the environmental domain in the DTE architecture.

5 References

Reference	
No	Description / Link
R1	An ensemble machine learning approach for tropical cyclone localization and tracking from ERA5 reanalysis data. G. Accarino et al. In Earth and Space Science 2023, vol 10, e2023EA003106 DOI: 10.1029/2023EA003106
R2	Very deep convolutional networks for large-scale image recognition. K. Simonyan and A. Zisserman. arXiv preprint, 2014 DOI: arXiv:1409.1556
R3	SEAS5: The new ECMWF seasonal forecast system. Johnson, S. J., Stockdale, T. N., Ferranti, L., Balmaseda, M. A., Molteni, F., Magnusson, L., Tietsche, S., Decremmer, D., Weisheimer, A., Balsamo, G., Keeley, S. P. E., Mogensen, K., Zuo, H., & Monge-Sanz, B. M. (2019). <i>Geoscientific Model Development</i> , 12(3), 1087–1117. DOI: 10.5194/gmd-12-1087-2019
R4	Integrating Scientific Knowledge with Machine Learning for Engineering and Environmental Systems. Willard, J., Jia, X., Xu, S., Steinbach, M., & Kumar, V. (2022). <i>ACM Computing Surveys</i> , 55(4). DOI: 10.1145/3514228
R5	Differentiable modeling to unify machine learning and physical models and advance Geosciences. Shen, C., Appling, A. P., Gentine, P., Bandai, T., Gupta, H., Tartakovsky, A., Baity-Jesi, M., Fencia, F., Kifer, D., Li, L., Liu, X., Ren, W., Zheng, Y., Harman, C. J., Clark, M., Farthing, M., Feng, D., Kumar, P., Aboelyzeed, D., ... Lawson, K. (2023). DOI: 10.1038/s43017-023-00450-9
R6	Continuous streamflow prediction in ungauged basins: Long Short-Term memory neural networks clearly outperform traditional hydrological models. Arsenault, R., Martel, J. L., Brunet, F., Brissette, F., & Mai, J. (2023). <i>Hydrology and Earth System Sciences</i> , 27(1), 139–157. DOI: 10.5194/hess-27-139-2023
R7	Satellite-Based Flood Mapping Through Bayesian Inference from a Sentinel-1 SAR Datacube. Bauer-Marschallinger, Bernhard, Senmao Cao, Mark Edwin Tupas, Florian Roth, Claudio Navacchi, Thomas Melzer, Vahid Freeman, and Wolfgang Wagner. (2022). <i>Remote Sensing</i> , 14 (15): 3673. DOI: 10.3390/rs14153673
R8	Enabling global scale Sentinel-1 time series analysis through streaming. Raml, B., Vreugdenhil, M., Massart, S. J. A., Navacchi, C., & Wagner, W. (2023). Enabling global scale Sentinel-1 time series analysis through streaming. In P. Soille, S. Lumnitz, & S. Albani (Eds.), <i>Proceedings of the 2023 conference on Big Data from Space (BiDS'23): From foresight to impact</i> (pp. 29–32). Publications Office of the European Union. DOI: 10.34726/5308

