interTwin

D4.1 First Architecture design of the DTs capabilities for climate change and impact decision support tools

Status: FINAL
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 Architecture, design, capabilities, Digital Twin Engine, climate change, impact decision, support tools

This deliverable describes the capabilities that the architecture design of a Digital Twin Engine (DTE) has to provide in order to be able to support the climate change use cases and the implementation of the related impact decision support tools. It details the functional specifications and requirements analysis for these use cases. Finally, it provides insights into the architecture design decisions made when developing the blueprint architecture of the DTE to specifically address the needs of the climate change use cases.

Document Description

D4.1 First Architecture design of the DTs capabilities for climate change and impact decision support tools

Work Package number 4

Document type	Deliverable		
Document status	FINAL	**Version**	1
Dissemination Level	Public		
Copyright Status	This material by Parties of the interTwin Consortium is licensed under a [Creative Commons Attribution 4.0 International License]().		
Lead Partner	EURAC		
Document link	https://documents.egi.eu/document/3937		
DOI	**10.5281/zenodo.8321307**		
Author(s)	● Muhammad Usman Liaqat (EURAC) ● Mariapina Castelli (EURAC) ● Donatello Elia(CMCC) ● Gabriele Accarino (CMCC) ● Davide Donno (CMCC) ● Sandro Fiore (UNITN) ● Bjorn Backeberg (DELTARES) ● Matthias Schramm (TU Wien) ● Christian Pagé (CERFACS) ● Frederique de Groen (DELTARES) ● Albrecht Weerts (DELTARES) ● Kathryn Roscoe (DELTARES) ● Atef Ben Nasser (IPSL)		
Reviewers	• Brian Pondi (University of Münster) • Zdenek Sustr (CESNET)		
Moderated by:	• Andrea Manzi (EGI Foundation) • Levente Farkas (EGI Foundation)		
Approved by	Andrea Manzi (EGI Foundation) on behalf of TCB		

Revision History

Version	Date	Description	Contributors
V0.1	04/07/2023	ToC	Andrea Manzi(EGI), Levente Farkas(EGI), Muhammad Usman Liaqat (EURAC)
V0.2	04/08/2023	First version ready for internal review	Muhammad Usman Liaqat (EURAC) and the other authors.
v0.3	16/08/2023	Reviewed version	Brian Pondi (University of Münster), Zdenek Sustr (CESNET)
v0.4	29/08/2023	Version Ready for TCB approval	Muhammad Usman Liaqat (EURAC) and the other authors.
v0.5	31/08/2023	Version approved by TCB	Andrea Manzi(EGI)
V1.0	06/09/2023	**Final**	Andrea Manzi (EGI)

Terminology / Acronyms

Term/Acronym	Definition
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
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
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

Executive summary

This document is deliverable 4.1 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 explain the motivation behind the development of specific DT applications, the architecture design, who will be the beneficiaries, and what may be expected in the future from the specific digital twins.

Starting from the description of user stories as scientists, students, developers, decision makers and policy makers to use DT applications for climate change impacts in various geographical regions by covering tropical storms, wildfires, floods, droughts and extreme weather events. Each DT also provided a user interface as a visualisation tool and Jupyter notebook to enhance usability and accessibility for both technical and non-technical users.

Furthermore, each DT application follows a specific workflow from the user's perspective, which involves defining geographical extent, data selection and preprocessing, model selection/training, post-processing and visualising results. Various modules from different Work Packages (WPs) within the interTwin project are incorporated into the DTs for processing data, composing workflows, and visualising results. Depending on each DT application, the results are stored as output files or visualised using Python visualisation modules or user interface. Overall, these DTs are designed and linked to address specific climate change impacts and provide valuable insights for assessing climate risk, identifying early warning signals, and implementing mitigation measures.

1 Introduction

1.1 Aim of this deliverable

The overall objective of deliverable 4.1 is to provide an overview of DT Applications features and architecture design for the environmental domain (T4.5, T4.6, T4.7) and their key requirements in the interTwin project. 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 who is this document

The deliverable 4.1 could be useful for both developers and end users as described below:

For **developers**: the current deliverable provides them with an opportunity to gain insight into different components, data integration strategies, and computational models required to build an effective digital twin. It would allow them to incorporate new features, leverage components and workflow, improve scalability, support evolving problems over time, and ensure interoperability.

For **end users**: the specified deliverable facilitates data sharing, integration, and analysis among various stakeholders and scientists. 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 user interface and requirements for each digital twin. A detailed table is provided by each digital twin application where details are provided regarding user problems, their requirements (following the MoSCow method[1]), expectations, and timeframe for completing the tasks. Section 3 mainly explains architecture design which illustrates the workflow composition within each digital twin application. It depicts sequential or parallel steps involved in each DTs operations, highlighting the input, processing, and expected outcomes.

[1] https://en.wikipedia.org/wiki/MoSCoW_method

2 DT Applications User Stories

2.1 DT Application: Tropical storms change in response to climate change

Geographical region of interest: Global

Summary
This DT will focus on the detection of tropical cyclones that consists in classifying, given some input drivers, the absence or presence of a cyclone in input data and, if present, subsequently localising its centre (or "eye") in terms of latitude/longitude coordinates. Machine Learning models allow learning the mapping between climatic variables 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.

Use case
The goal is to provide notebooks for scientists and policy makers for running analysis on tropical cyclones on future projection data:

Preconditions

User has access to DT data, models, thematic components and notebook

1. Users can select:
   a. future projection scenarios from a given list (e.g., CMIP6 scenarios – SSP)
   b. temporal and geographical extents;
   c. ML models in the ensemble from a set of pre-trained models (types and number of models);
2. The user runs the DT workflows on the selected input data and trained models
3. The output of the DT can be downloaded/saved as NetCDF and/or visualisations. Maps and charts can be customised through widgets.
   a. What-if analyses can be performed according to the SSPs selected
      i. List of maps/charts of TC position on the different scenarios
      ii. Ensemble of the model scenarios (average)
   b. Different indicators can be provided:
      i. Frequency of TC occurrences on a seasonal/annual basis
      ii. Number of TCs per basin/year
      iii. The trend of TCs per basin/year

Table 1 – User stories for DT Application: Tropical storms change in response to climate change

Ref N	As a Stakeholder	I want to	So that	And it's considered done when	MoSCoW
4.5-1	Policy maker with little technical expertise	Specify a region and period of interest, the CMIP scenarios to be used for TC analysis	The pre-trained ML model can perform TC detection and tracking on the input climate data	The resulting datasets with TC detections and plots are generated	Must have: - Access to Jupyter Notebook as a service - Access to necessary global data - Access to the ML models Could have: - Possibility to adjust the visualisation interactively - Possibility to download the results
4.5-2	Scientists with some technical expertise	Specify the number and types of pre-trained ML models to be used together with the extent and scenarios	Can run complex analysis with one or more ML models for TC detection and tracking	The resulting datasets with TC detections and plots are generated	Must have: - Access to Jupyter Notebook as a service - Access to necessary global data -Access to the ML models Could have: - Interfaces to adapt the visualisation - Possibility to download the results

2.2 DT Application: Wildfire risk assessment in response to climate change

Geographical region of interest:_Europe

Summary

The DT related to the wildfires application will focus on the generation of fire risk maps that closely resemble the original fire risk distribution in historical records. Output fire risk maps are produced through Machine Learning models, such as DNN, that are trained to learn the non-linear relationship between different climatic, weather, and potentially vegetation conditions provided as input and the likelihood of wildfires in the geographical domain of interest. Trained ML models will be applied to future projection scenarios to give an indication of the areas that are more likely to experience wildfires and the impact of climate change.

Use case

The goal is to provide notebooks for scientists and policy makers for running analysis on tropical cyclones on future projection data:

Preconditions

User has access to DT data, models, thematic components and notebook

1. Users can select:
   a. future projection scenarios from a given list (e.g., CMIP6 scenarios - SSP)
   b. temporal and geographical extents;
   c. ML models from a set of pre-trained models (types of models);
2. The user runs the DT workflows on the selected input data and trained models
3. The output of the DT can be downloaded/saved as NetCDF and/or visualisations. Maps and charts can be customised through widgets.
   a. What-if analyses can be performed according to the SSPs selected
      i. List of maps/charts of wildfire risk on the different scenarios
      ii. Ensemble of the model scenarios (average)
   b. Different indicators can be provided:
      i. Average fire risk maps on a monthly/seasonal/annual basis
      ii. Areas more affected by wildfires seasonally/annually (by threshold)
      iii. The trend of fire risk annually globally or region
   c. Outputs can be filtered by bounding box or geographical domains (e.g., Cordex)

Table 2 – User stories for DT Application: Wildfire risk assessment in response to climate change

Ref N	As a Stakeholder	I want to	So that	And it's considered done when	MoSCoW
4.5-3	Policy maker with little	Specify a region and period of	The pre-trained ML model can	The resulting maps with the wildfire	Must have: - Access to Jupyter Notebook as a service

	technical expertise	interest, the CMIP scenarios to be used for the wildfire risk area	predict wildfires risk maps	risk probabilities are produced	- Access to necessary data -Access to the ML models Could have: - Possibility to adjust the visualisation interactively - Possibility to download the results
4.5-4	Scientists with some technical expertise	Specify the type of pre-trained ML models to be used together with the extent and scenarios	Can run complex analysis with the ML models for wildfires risk maps prediction	The resulting datasets with wildfire risk probabilities and plots are generated	Must have: - Access to Jupyter Notebook as a service - Access to necessary global data - Access to the ML models Could have: - Interfaces to adapt the visualisation - Possibility to download the results

2.3 DT Application: Flood early warning in coastal and inland regions

Geographical region of interest: Humber, United Kingdom

Summary

The DT for flood early warning in coastal and inland regions will focus on the generation of flood risk maps that trigger early warning alerts when a flood is predicted. The system will be demonstrated for historical flood events of Humber, United Kingdom.

Output flood risk maps are produced from SFINCS, a reduced-complexity model for super-fast dynamic modelling of compound flooding, which receives river discharge data from Wflow, a hydrological model. Both SFINCS and Wflow will be forced by example (historical) weather forecasts. Additionally, the DT will combine the SFINCS flood maps with Sentinel-1 based flood maps generated by the openEO implementation of the Global Flood Monitor. The mentioned software components are described in D7.1 [R6]

The flood early warning system will trigger alerts (e.g. email alerts) if populated areas are at risk of being flooded.

Use case

The goal is to provide Jupyter Notebooks for scientists and decision-makers 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.

Preconditions

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

1. Users can:
    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.

Table 3 – User stories for DT Application: Flood early warning in coastal and inland regions

Ref N	As a Stakeholder	I want to	So that	And it’s considered done when	MoSCoW
4.6-1	… decision maker with little technical expertise	… specify a geographic region and temporal period of interest	… I can set up the automated processing of data for flood inundation and hydrological models and access relevant EO data for flood monitoring and forecasting	… when the system simulates a historic flood event and data is automatically prepared, so an automated early warning system can access it in a standardised form	Must have: - Access to Jupyter Notebook as a service - Access to necessary global data Should have: - Example visualisations of output to support validation - trigger sending alert based on flood extent

					Could have: – option to upload local data – interactive Solara[2]-based front-end Won’t have: – Operational early warning system. This is a demonstrator only.
4.6-2	… expert user with good technical expertise but little domain knowledge	…process and combine modelled and EO-based flood-related data for specific regions of interest	…I can get tailored information on flood monitoring and forecasting	…I can provide decision makers with a thorough overview of the expected flood event	

2.4 DT Application: Alpine droughts early warning

Geographical region of interest: Alps

Summary

This DT aims to develop a prototype of a drought early warning system for the Alps at the river basin scale. The study area is divided into seven Alpine basins to reduce the computational time (Po East, Po West, Danube, Swiss, Drava, Rhone East, Rhone West). A machine learning model (surrogate model) is trained to replicate the WFLOW process-based hydrological model. The parameters of the surrogate model will be optimised based on historical observations and static inputs by minimising a loss function between model output and observations (i.e., soil moisture and possibly evapotranspiration derived from satellite data, and streamflow) in the area of interest. Once the model is trained, seasonal forecasts from ECMWF will be used to forecast discharge and soil moisture and identify regions affected by drought events. A machine learning algorithm to downscale seasonal forecasts is also developed.

Use case

The prototype of a drought early warning system is developed and integrated into openEO as a user interface for researchers and decision makers. Through openEO the user should be able to:

1) Run a trained model for a specific area of interest and temporal extent. The detailed information about openEO is already described and can be available in Deliverable D6.1[R3]
2) Validate results using historical observations and
3) Run the model driven by seasonal forecasts to identify areas affected by hydrological drought.

[2] https://solara.dev/docs/

Preconditions

1) Users have to access DT using OpenEO, available global data, or upload their regional data.
2) The module for downscaling climate data and the trained surrogate model should be implemented and available in openEO.
3) Modules for coregistering and formatting all the required input data have to be available in openEO.
4) The output from DT can be directly examined using openEO and can also be downloaded for later use.

Table 4 – User stories for DT Application: Alpine droughts early warning

Ref N	As a Stakeholder	I want to	So that	And it's considered done when	MoSCoW
4.6-3	Researchers	– Use OpenEO to test a drought early warning system using a trained model and EO data	– I can study potential drought impacts on the Alpine region. – I can contribute to further enhancing the trained model. -I can leverage cloud-based processing capabilities	When simulated results are in good agreement with historical observations and the trained model is able to reproduce past drought events	**Must have:** i) access to openEO Authentication and Access Control, ii) access to modules to ingest and pre-process EO and climate data, iii) access to the trained surrogate model **Should have:** i) estimation of model accuracy, ii) interactive and user-friendly visualisations for simulated results.

4.6-4	Local/Regional public authorities in the field of agriculture, hydrology and river basin management	– Know over which basins there is high probability of water scarcity in the next month	– I can have an overview of the areas where there is a high risk of drought impacts – I can inform the interested water users – I can develop strategies for water allocation and conflict management	– It is possible to visualise and analyse the data and simulated results. – The accuracy of the prediction is high	**Could have:** Access to hydrological data for validation. **Won't have:** Operational drought early warning system
4.6-5	Journalist for environmental topics with little expertises about technical data	Visit OpenEO by visualising fact sheets about drought conditions in the Alpine region.	Publish an article about the impact of extreme events on water availability in the Alpine region.	Information is easily accessible also to a non-expert user	

2.5 DT Application: Extreme rainfall, temperature and wind weather event changes in response to climate change

Geographical region of interest: Europe, but could be applied anywhere in the world

Summary

The DT for impacts of extreme weather event changes in response to climate change will focus on providing to the users information on the changes of the characteristic of those events and impacts compared to a reference period and a specific region. The focus will be on precipitation as well as on temperature and wind extremes. The change of characteristics will be assessed, such as geographical extent, intensity (if relevant), duration, and frequency of occurrence. This DT will use an innovative Artificial Intelligence (AI) method based on a Convolutional Variational Auto-Encoder (CVAE) method to detect anomalies.

Use Case

The goal is to provide Jupyter Notebooks for scientists and decision-makers to:

1. Assess the changes of characteristics of specific weather extreme events and assess their impacts
2. Assess over a selected geographical region, a specific climate reference period, a specific future time period with one or several greenhouse gas scenarios

Preconditions

1. Set up the necessary CVAE model for a user-defined extreme weather event, geographical region of interest, climate reference time period (historical 20-30 years), season, future climate scenarios (RCPs) and climate time period of interest.
2. Train the CVAE model according to end users' choices of step 1, on separate climate models from the CMIP6 archive, using several climate model ensemble members.
3. Run the CVAE model on the end users' choices of extreme weather events, geographical regions, future climate scenarios (RCPs), and climate time periods of interest, for several climate models using specific training information according to each climate model and specific reference period.
4. Generate end users' products related to the changes in characteristics of those events: geographical extent, intensity (if relevant), duration, and frequency of occurrence. Those products will be specific plots and maps. Output data will be also available to end users for further data processing. Multiple climate models and RCPs will be used to provide uncertainties evaluation with a range of characteristic changes.

Table 5 – User stories for DT Application: Extreme rainfall, temperature and wind weather event changes in response to climate change

Ref N	As a Stakeholder	I want to	So that	And it's considered done when	MoSCoW
4.7-1	Policy maker with little technical expertise	Specify an extreme weather event, a geographical region, a reference historical climate time period, a future time period of interest, and the CMIP6 SSP scenarios to be used for extreme event analysis	The CVAE model will be automatically trained and applied on the input climate data	The resulting datasets with extreme events change of characteristics will be calculated and plots are generated	Must have: – Access to Jupyter Notebook as a service – Access to necessary global data – Access to the CVAE model Could have: – Possibility to adjust the visualisation interactively

					– Possibility to download the results
4.7-1	Scientists with some technical expertise	Specify an extreme weather event, a geographical region, a reference historical climate time period, a future time period of interest, and the CMIP6 RCP scenarios as well as CMIP6 specific climate models to be used for extreme event analysis	Can run complex analysis with specific CMIP6 climate models to better assess uncertainties	The resulting datasets with extreme events characteristics and plots are generated	Must have: – Access to Jupyter Notebook as a service – Access to necessary global data – Access to the VAE model Could have: – Interfaces to adapt the visualisation – Possibility to download the results

2.6 DT Application: Flood climate impact in coastal and inland regions

Geographical region of interest: Humber, United Kingdom, and Beira, Mozambique

Summary

The DT for flood climate impact in coastal and inland regions will focus on the generation of flood maps and quantifying impacts on buildings, utilities, roads and accessibility under future climate conditions. Additionally, end-users can select flood mitigation and adaptation measures and test their effectiveness. The system will be demonstrated for flood scenarios under climate change for Humber, United Kingdom, and Beira, Mozambique.

Flood maps under future climate scenarios are produced from SFINCS, a reduced-complexity model for super-fast dynamic modelling of compound flooding, which receives river discharge data from Wflow, a hydrological model. The flood maps are then used by Delft-FIAT, a flood impact assessment tool, and RA2CE, a Resilience Assessment and Adaptation for Critical infrastructurE – model, to quantify impacts and damages to buildings, utilities, roads and accessibility.

Additionally, end-users will be able to select flood mitigation and adaptation measures and re-run flood scenarios to test their effectiveness in reducing flood-related impacts.

Use case

The goal is to provide Jupyter Notebooks for scientists and decision-makers 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.

Preconditions

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

1. Users can:
    a. specify a region of interest
    b. specify a temporal period to simulate
    c. select local data for the models, if available
    d. select and specify mitigation and adaptation measures
2. The user runs the DT workflows for the specified region and scenario 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 and GeoPackage data.

Table 6 – User stories for DT Application: Flood climate impact in coastal and inland regions

Ref N	As a Stakeholder	I want to	So that	And it's considered done when	MoSCoW
4.6-6	… decision maker or planner with little technical expertise	… specify a geographic region, a climate change scenario and select mitigation / adaptation measures of interest.	… I can set up the flood inundation and hydrological models and run flood scenarios under future climate conditions and test the impact	… when the system simulates a flood scenario and quantifies the impacts and damages to buildings, utilities, roads	Must have: - Access to Jupyter Notebook as a service - Access to necessary climate projection data Should have:

			of the selected mitigation / adaptation measures	and accessibility.	- Flood and related impact visualisations and data to support decision-making - Options to select future climate change scenarios - Options to select mitigation and adaptation measures Could have: - Option to upload local data - Interactive Solara-based front-end Won't have: - An extensive list of adaptation and mitigation measures - An operational system, this is a demonstrator only
4.6-7	… expert user with good technical expertise but little domain knowledge	…process and combine different models and tools needed for flood related adaptation planning for specific regions of interest	…I can get tailored information on flood scenarios under future climate conditions and make decisions on what adaptation and mitigation measures to invest in	…I can provide decision makers with a thorough overview on the expected flood scenarios and their impact under future climate conditions	

3 DT Applications Design

3.1 DT Application: Tropical storms change in response to climate change

3.1.1 ML Model Requirements

Since the application input consists of a set of 2-dimensional data, each variable can be easily considered as a 2D image, where each pixel corresponds to a cell of the lat-lon grid. Based on this consideration the ML model identified for the DT is a Convolutional Neural Network (CNN). In order to feed the CNN with the images, input climatic variables are stacked together and tiled into non-overlapping patches of fixed size, generating an input of dimension $H \times W \times C$. Besides traditional CNN, Convolutional Graph Neural Network (CGNN) approaches will be considered too. Data needs to be preliminarily divided into patches in order to improve the model efficiency and, more importantly, to ensure that each patch is more likely to contain at most a single TC. Such patches are mapped with the records provided by IBTrACS so that each patch containing a cyclone is associated with the corresponding (lat,lon) record, while each patch without a cyclone is associated with a negative value, e.g., (-1,-1). Therefore, the CNN will be trained to find a non-linear mapping between input 3D patches and the corresponding lat-lon output coordinates. In terms of software infrastructure, solutions like Keras/Tensorflow and/or PyTorch will be exploited.

A preliminary list of identified data variables is available in Table 7.

Table 7 – Preliminary list of data variables for the DT Application

Variable name	Temporal Resolution	Spatial Resolution	Unit	ERA5 Name	CMIP6 Name
10 m wind gust since previous post-processing	6-hourly	0.25°x0.25°	m/s	fg10	TBD
10 m instantaneous wind gust	6-hourly	0.25°x0.25°	m/s	i10fg	wsgmax10m
temperature at 500 mb	6-hourly	0.25°x0.25°	K	t_500	ta
temperature at 300 mb	6-hourly	0.25°x0.25°	K	t_300	ta
relative vorticity at 850 mb	6-hourly	0.25°x0.25°	1/s	vo_850	rv850
mean sea level pressure	6-hourly	0.25°x0.25°	Pa	msl	psl

3.1.2 Workflow Description

This section describes the logical flow of operations the DT needs to perform to carry out user requests. The links to WP6 core modules and WP7 thematic modules are highlighted.

In terms of computing workflow, the DT on Tropical Cyclones comprises these steps:

1. Users can select the proper data for running their analysis; i.e., specifying the temporal and spatial extensions to be considered or the models from CMIP experiments. *Gathering & Filtering of climate data* thematic modules (WP7) will be used for retrieving the data from the different sources.
2. Selected data will be pre-processed so that it can be used as input to the ML models. Two thematic modules will be exploited for this step: *Subsetting & patches generation on gridded data* and *Data augmentation and pre-processing of gridded data (WP7).* In particular, patches from the input data will be generated, and:
    a. For the training stage: data will be labelled and normalised based on the training set. In addition data augmentation techniques will be applied;

   b. For the inference stage: data will be only normalised according to the training scaler;
3. Only for the training workflow: training/re-training of a ML model can be triggered in order to use different hyper-parameters if results from testing are not satisfactory in terms of evaluation metrics (e.g., error and classification metrics); the *Software Quality Validation core module* (from T6.2) will be used to this end. At this stage, patch generation and data augmentation procedures of ERA5 data are also required to increase the number of training examples. The trained models will be stored in a ML model repository for future usage. The *ML workflow core module* (from T6.5) will be exploited for this step.
4. Only for the inference stage: a ML model from the repository of pre-trained models will be executed on the data from step 2. The *ML workflow core module* (from T6.5) will be exploited for this step.
5. Results from the inference stage on projection data (e.g., CMIP6) will then be post-processed to build the final results (e.g., to build the spatial grid) using the *Geo-referencing data thematic module* (from WP7);
6. Results on different CMIP experiments and/or different ML models can be combined together to reduce the uncertainties of the results from the single ML models and/or CMIP output, according to the user's input. The *Multi-model climate inference thematic module* (from WP7) can be used for this step;
7. Final results can be stored as output files or visualised in the notebook interface using different Python visualisation modules.

The overall workflow of the application will be orchestrated by the *Workflow engine solutions from T6.1*, which will also take care of tracking the DT provenance, and deployed by the *Big Data processing infrastructure T6.4* over the WP5 infrastructure resources. Figure 1 shows a diagram with the DT workflow main steps.

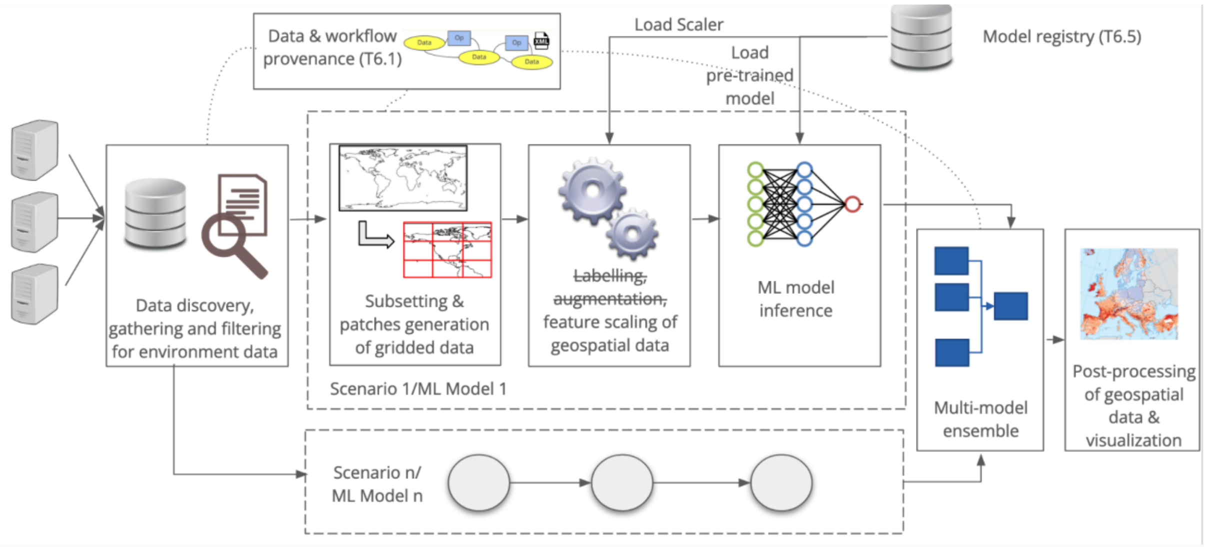
Data & workflow provenance (T6.1)
Data
Op
Data
Op
XML
Data
Load Scaler
Model registry (T6.5)
Load pre-trained model
Data discovery, gathering and filtering for environment data
Subsetting & patches generation of gridded data
~~Labelling, augmentation,~~ feature scaling of geospatial data
ML model inference
Scenario 1/ML Model 1
Multi-model ensemble
Post-processing of geospatial data & visualization
Scenario n/ ML Model n

Figure 1 – Overview of the workflow for DT on extreme events

3.2 DT Application: Wildfire risk assessment in response to climate change

3.2.1 ML Model Requirements

The main goal of DT Wildfire case study is to develop several Deep and Convolutional Neural Network-like architectures capable of learning complex relationships between chosen input variables and extract global fire risk maps containing wildfire burned areas probabilities. Several architectures could be investigated, such as UNet and UNet++. The proposed ML architectures will take as input a stack of climatic variables (similarly to Section 2.1.2) of dimension $H \times W \times C$ (where H and W are height and width, respectively, and C is the number of drivers) and provide as output a map of dimension $H \times W \times C$ with wildfires risk probabilities inside the patch. The input data will be split into non-overlapping patches of fixed size as mentioned in Section 2.1.2.

Preliminary list of identified data variables is available in Table 8.

Table 8 – SeasFire Cube and corresponding to CMIP6 data variable chosen to carry out the Wildfire case study

Full name	SeasFire Cube name	Unit	CMIP6 name
ERA5 Meteo Reanalysis Data			
Total precipitation	tp	m	pr
Relative humidity	rel_hum	%	hur
Vapor Pressure Deficit	vpd	hPa	N.A.
Sea Surface Temperature	sst	K	tos
Temperature at 2 meters – Min	t2m_min	K	tasmin
Surface net solar radiation	ssr	MJ m-2	rss
Land-Sea mask	lsm	0-1	sftlf

Nasa MODIS MOD11C1, MOD13C1, MCD15A2			
Land Surface temperature at day	lst_day	K	ts
Normalised Difference Vegetation Index	ndvi	unitless	N.A.
Global Wildfire Information System (GWIS)			
Burned Areas from GWIS	gwis_ba	ha	Used only for training
Valid mask of GWIS burned areas	gwis_ba_valid_mask	0–1	Used only for training

3.2.2 Workflow Description

This section describes the logical flow of operations the DT on WildFires needs to perform to carry out user requests. The links towards WP6 core modules and WP7 thematic modules are highlighted also in this case.

In terms in computing workflow, the DT on Wildfires risk management comprises these steps:

1. Users can select the proper data for running their analysis; i.e., specifying the temporal and spatial extensions to be considered or the models from CMIP experiments. *Gathering & Filtering of climate data* thematic modules (WP7) will be used for retrieving the data from the different sources.
2. Selected data will be pre-processed so that it can be used as input to the ML models. Two thematic modules will be exploited for this step: *Subsetting & patches generation on gridded data* and *Data augmentation and pre-processing of gridded data (WP7).* In particular, patches from the input data will be generated and:
    a. For the training stage: data will be normalised based on the training set from SeasFire. In addition data augmentation techniques will be applied;
    b. For the inference stage: CMIP6 data will be normalised according to the training scaler and pre-processed to be compatible with SeasFire cube data structure;
3. Only for the training workflow: training/re-training of a ML model can be triggered in order to use different hyper-parameters, if results from testing are not satisfactory in terms of evaluation metrics (e.g., error and classification metrics); the *Software Quality Validation core module* (from T6.2) will be used to this end. At this stage, patch generation and data augmentation procedures of SeasFire data are also required to increase the number of training examples. The trained models

will be stored into a ML model repository for future usage. The *ML workflow core module* (from T6.5) will be exploited for this step.

4. Only for the inference stage: a ML model from the repository of pre-trained models will be executed on the data from step 2. The *ML workflow core module* (from T6.5) will be exploited for this step.
5. Results from the inference stage on projection data (e.g., CMIP6) will then be post-processed to build the final results (e.g., to build the spatial grid) using the *Geo-referencing data thematic module* (from WP7);
6. Final results can be stored as output files or visualised in the notebook interface using different Python visualisation modules.

The overall workflow of the application will be orchestrated by the *Workflow engine solutions from T6.1*, which will also take care of tracking the DT provenance, and deployed by the *Big Data processing infrastructure (T6.4)* over the WP5 infrastructure resources. The overall DT workflow is similar to the one shown in Figure 1.

3.3 DT Application: Flood early warning in coastal and inland regions

3.3.1 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. openEO satellite-based flood monitoring: An existing workflow for flood monitoring [R4] will be re-developed in the openEO syntax[R5] for being usable on several platform backends. The process graph will be gradually enhanced to create a fully automatic processing chain, based on Sentinel-1 σ_0 and PLIA image collections.

Wflow simulates the volume and timing of water flow (runoff) from a catchment into a river system, based on meteorological inputs and the catchment characteristics. To do so a user has to define catchment boundaries, set up hydrological response units based on combining e.g. soil, land cover and topography data. Simulations are forced using precipitation data, and results are affected by land use / land cover, soil types, topography and evapotranspiration rates.

For inland flooding, the hydrological model (Wflow) simulates how water flows through river systems and floodplains, and the flood inundation model (SFINCS) simulates how this water might spread during flood events. The flow and spread of water is affected by the geometry of the river channel and floodplain (usually derived from Digital Elevation Models), the upstream inflow, downstream water levels, location of levees, dams, and channels and how these are operated.

For coastal flooding additional factors and data need to be considered, including tides, storm surges, sea level rise projections, waves, beach morphology and dynamics, coastal infrastructure and defenses, wind speed and direction and atmospheric pressure.

An overview of data requirements to run Wflow and SFINCS can be found in Tables 4 and 6 of interTwin D7.1 [R6]
To produce deterministic flood maps for a given scenario or event, a specific (extreme) rainfall or surge event is selected and the hydrological model (Wflow) simulates river inflows. These together with the aforementioned data representing relevant processes associated with coastal flooding are fed into the flood inundation model (SFINCS) which simulates the flood extent and depth. The flood extent and depth is then used to trigger an early warning.

To produce probabilistic flood maps, an ensemble of scenarios that represent a range of possible rainfall and surge events is simulated. This is done in order to take into consideration potential variations in e.g. intensity, duration and spatial distribution. Each scenario is associated with a return period (e.g. 1 in 100 years) or a probability of occurrence (e.g. 1% chance each year). Then based on the ensemble simulations the likelihood of flooding to occur for each point on the map is determined. The resultant flood map shows the probability of inundation for each location.

To ensure accurate simulations, models are usually calibrated and validated against historical data from in situ measurements and / or satellites. Those value-added satellite data will be processed, using a global flood monitoring workflow that is being adapted to project specific requirements and that is being automated. In a first iteration the flood monitoring workflow is based on Bayesian decision making, exploiting data cubes of Sentinel-1 data with its orbit repetition and a-priori generated probability parameters for flood and non-flood conditions. Therefore, local seasonal non-flood conditions for each day-of-year are defined by pre-processing harmonic parameters of each pixel's full time series. As a stretched goal, processing this firstly static information can be embedded as a dynamic workflow into the operational process chain, only analysing e.g. the recent two years and thus considering possible changes in the backscatter by e.g. changes of land use / land cover. Another stretched goal is the usage of ML-based training instead of the lightweight Bayesian approach, making use of the available big data processing capabilities within the project. A data cube based masking of no-sensitivity resulting from ill-posted satellite geometries or impeding land cover further enhances the process' robustness. Implementing the described stretched goals will depend on 1) analyses on their potential to increase the product's accuracy, and 2) on decisions throughout the project regarding implementing redundant processing libraries, usable for this and for other workflows.

To set up Wflow for the purposes described above, a Digital Elevation Model is used to delineate the watershed boundaries. The catchment can then be broken down into smaller sub-basins or hydrological response units based on land use, soil type and topography. For each sub-basin processes such as runoff coefficients and infiltration rates are parameterised.

To set up SFINCS, the same Digital Elevation Model is used to define the river network, i.e. the geometry of the rivers and floodplains. Boundary condition points are determined for the upstream inflows from Wflow and downstream water levels or tidal conditions. The influence of bridges, dams, levees, and other infrastructure on flow dynamics should also be included.

An overview of data requirements to set up Wflow and SFINCS can be found in Tables 3 and 5 of interTwin D7.1 [R6]

In this digital twin application functionality is being developed that enables an end user

1. Easily set up Wflow and SFINCS for a user-defined region of interest
2. Easily run the models and produce deterministic and probabilistic flood maps

3.3.2 Workflow Description

Three workflows are described in this section:

1. Setting up the necessary Wflow and SFINCS models
2. Producing a deterministic flood map
3. Producing a probabilistic flood map

Setting up Wflow and SFINCS models for a specific region of interest comprises the following steps:

1. Set up SFINCS
    a. A user defines a geographic region of interest
    b. A user defines the SFINCS model resolution
    c. A user selects available global datasets from which to build the model. Optionally a user can upload and use local data instead of the global data.
2. Set up Wflow
    a. The area of interest defined for SFINCS is used to define the upstream catchment area for Wflow
    b. A user defines the Wflow model resolution
    c. A user selects available global datasets from which to build the model. Optionally a user can upload and use local data instead of the global data

Once the SFINCS and Wflow models have been set up they can be run to produce deterministic and probabilistic flood maps as follows.

For deterministic flood maps:

1. A user selects a time period to simulate, e.g. a specific historical event
2. A user selects the preferred forcing data for SFINCS and Wflow, e.g.
    a. Wflow – temperature, precipitation, and potential evapotranspiration
    b. SFINCS – in addition to precipitation, for a coastal flood, select data for tides, surges, waves, and sea level
3. A user then runs Wflow
4. A user then runs SFINCS with river discharges from Wflow
5. The water levels produced by SFINCS are interpolated to the digital elevation model selected by the user in the Setup SFINCS step 1c, to estimate flood depths and extent

For probabilistic flood maps

1. A user selects a time period to simulate, e.g. a specific historical event
2. A user selects the preferred *ensemble of* forcing data for SFINCS and Wflow, e.g.
    a. Wflow – temperature, precipitation, and potential evapotranspiration
    b. SFINCS – in addition to precipitation, select data for tides, surges, waves, sea level
3. A user then runs many Wflow simulations forced with the ensemble forcing data
4. A user then runs many SFINCS simulations with river discharges from Wflow
5. The many water levels produced by SFINCS are interpolated to the digital elevation model selected by the user in the Setup SFINCS step 1c, to estimate flood depths and extent and e.g. the likelihood of a flood occurring at a specific location is computed.

In the above workflows, this DT application will leverage the following capabilities from other WPs of interTwin (summarised in Fig.2)

- The DT will leverage capabilities developed in WP7, T7.6: Hydrological model data processing thematic module.
- Additionally for preprocess forcing and boundary condition data, this DT aims to leverage developments from WP6, T6.3 Data fusion.
- To run the models using containers, this DT will rely on functionality from WP6, T6.4 Container workload management and batch queue system.
- For post-processing and visualisation, this DT will rely on WP6, T6.2 FAIR data quality evaluation.
- For workflow composition and execution, this DT will leverage developments from WP6, T6.1 Workflow composition and WP6, T6.4 Workflow backend.
- Finally, all data and compute resources are leveraged from WP5, the DTE infrastructure including Notebooks as a Service.

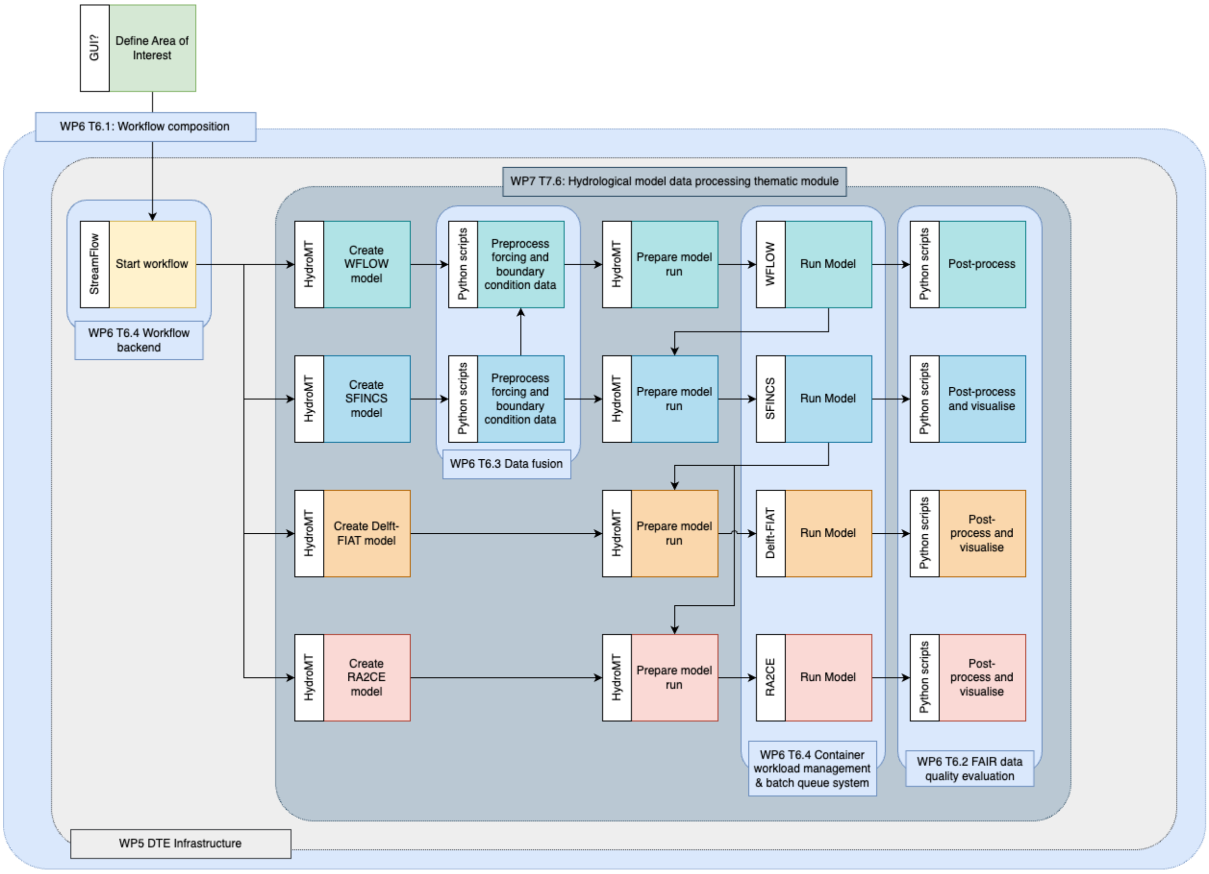

Figure 2 – Overview of the workflow components mapped against developments in other work packages and tasks of the project in C4-Model Level 3 format.

3.4 DT Application: Alpine droughts early warning

3.4.1 Model Requirements

The drought early warning DT Application aims to develop a deep learning model capable of reproducing the physical representation of hydrological processes of the model Wflow, and learning complex physical aspects, not yet described in the physical model, from spatially distributed data derived from EO. The model will be used to simulate hydrological fluxes (streamflow, soil moisture and evapotranspiration) and predictively identify drought prone areas. In this study, both dynamic and static types of data are used as input to run hydrological as well as its surrogate model. The dynamic input data include meteorological variables (temperature, precipitation and potential evapotranspiration) and vegetation-related information, such as the leaf area index. Meteorological inputs from 2000 to 2020 derive from daily ERA5 reanalysis. Both the dynamic and static data are resampled to the resolution of the hydrological model grid (~ 1 km). Once the model is set, it will be run with meteorological forcing inputs derived from a downscaled version of ECMWF SEAS5 seasonal forecasts in place of ERA5 reanalysis to make predictions for future drought conditions. The downscaling of forecast fields will be performed through machine learning schemes optimised for each specific variable. Continuously update the model with new forecast data as it becomes available to generate near-real-time drought predictions. Table 9 and 10 presents the type of dynamic and seasonal forecast data used in this study. The description of static data is already available in D7.1[R6].

Table 9 – Details of dynamic data used in this study and their sources.

Dynamic Data	Data Source	Time	Temporal Resolution	Spatial Resolution	Product Description
2m_temperature	Copernicus Climate Data Store (CDS)	2000–2020	Daily	0.25º x 0.25º 0.0495º x0.0495º	ERA5 & CERRA
Surface pressure					
Total precipitation					
Surface net solar radiation	ECMWF				
Surface solar radiation downwards					
Surface Pressure	CDS				
Potential evapotranspiration	CDS				de Bruin et al. (2016).
Leaf Area Index	Earth Data		Monthly	500 m	MODIS

Table 10 – Details of seasonal forecast data used in the study and their sources.

Variable	Spatial Resolution	Temporal Resolution	Product Description
2m_dewpoint_temperature	0.25° x 0.25°	Hourly	Seasonal Forecast ECMWF
2m_temperature			
Geopotential (at 500 and 850 hpa)			
Mean_sea_level_pressure			
Specific_humidity (at 500 and 850 hpa)			
Surface_pressure			

Temperature (at 500 and 850 hpa)			
Total precipitation			
u_component_of_wind (at 500 and 850 hpa)			
v_component_of_wind (at 500 and 850 hpa)			

3.4.2 Workflow Description

The purpose of this section is to describe the logical flow of operations that the DT drought early warning system needs to carry out both from development side as well as from user perspective.

1) Computing workflow from the development side of the DT drought early warning system. Three primary steps are involved:

i) Setting Up physical based model (Wflow_sbm) and develop its surrogate model by keeping physical aspects of Wflow_sbm

ii) Calibrate parameters of the hydrological (surrogate) model.

iii) Predict drought anomalies using trained model and seasonal forecast data from

ECMWF

This activity will use the spatially distributed hydrologic model Wflow_sbm [R1, R2] to estimate hydrological fluxes including snow accumulation and melt, actual evapotranspiration, soil moisture, streamflow. The study area is divided into seven Alpine basins to decrease computational time (Po east, Po west, Danube, Swiss, Drava, Rhone East, Rhone West). Throughout the study domain, all models are set up in the same manner with 0.008333-degree cell sizes, which correspond to approximately one kilometer. The Wflow_sbm performance will examine using observed discharge, *MODIS/Terra Net Evapotranspiration* and Sentinel-1 based soil moisture both at temporal and spatial scale. A surrogate model for differentiable hydrological models, like Wflow_sbm, offers computational efficiency, improves convergence while reducing noise, making it suitable for distributed computing. In this study, a surrogate model based on dPL framework is developed in two phases as shown in Fig.3. The first phase involves training the LSTM to reproduce the performance of the process-based model Wflow_sbm by minimising the loss function (RMSE). It is necessary to include this step in order to support a differentiable workflow as well as to save time during computation. The LSTM surrogate model is trained using dynamic forcings and static attributes, using soil moisture and evapotranspiration simulated by Wflow_sbm as a target of emulations. By using the Inter-Quartile Range (IQR) approach, the data is preprocessed to remove outliers. Next, the data is normalised using the min–max normalisation technique. Finally, the preprocessed data is split into a 70% training set and a 30% testing set to evaluate the model's performance. This process ensures that the LSTM model receives clean and standardised data for effective training and testing. Once the loss function is minimised, parameters of surrogate hydrological models are estimated based on historical observations and static inputs by minimising the loss function between model outputs and observations, namely soil moisture, which plays an integral role in drought prediction. As soon as the model is trained and the parameters are calibrated, seasonal forecast data from ECMWF at daily scale will be used in order to forecast drought conditions over the following two weeks to two months.

2) Based on the user's perspective within openEO, the drought early warning system consists of the following steps**:**

1. Defining Geographical Extent

To begin the analysis, the user must specify the geographical area of interest. The user can select the watershed of interest among the ones included in the study area.

2. Data Selection and Preprocessing

Data selection is one of the key components in simulating DT results. The drought DT Application provides flexibility to their users to choose globally available data on the portal or bring their own local data both for historical and future simulation. The user must preprocess and harmonise the data according to model resolution. This process ensures that the ML model receives clean and standardised data for simulation.

3. Model Selection/ Training

Following the preprocessing of the data, the user can select the trained model to simulate historical drought conditions as well as forecast drought anomalies in the future. The current model is trained and tested for the Alps. If the user wishes to run the model in a different region or use a different type of model, it will be necessary to retrain the model.

4. Post-Processing

Users can check the performance of trained models and simulations by comparing them with historical observations of streamflow, evapotranspiration, and soil moisture. Upon satisfactory model performance, users can also run the model using seasonal forecasts for predicting drought anomalies.

5. Visualisations

Final results can be visualised in the openEO interface by using different Python visualisation modules or they can be downloaded for further analysis and shared with the appropriate authorities.

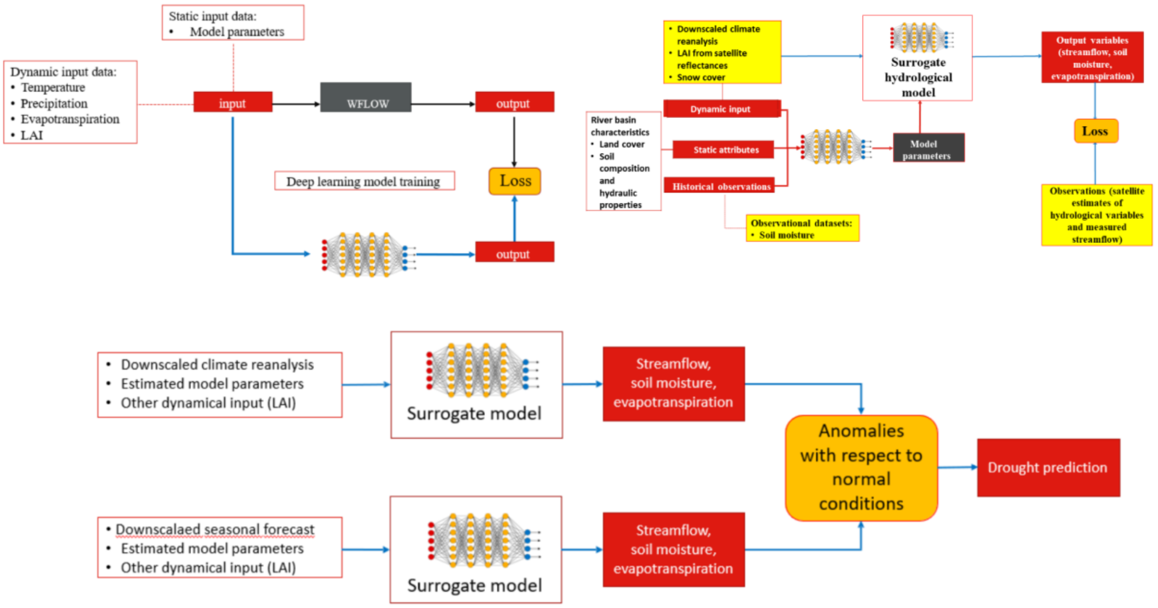
Static input data:
• Model parameters
Dynamic input data:
• Temperature
• Precipitation
• Evapotranspiration
• LAI
input
WFLOW
output
Deep learning model training
Loss
output
• Downscaled climate reanalysis
• LAI from satellite reflectances
• Snow cover
Surrogate hydrological model
Output variables (streamflow, soil moisture, evapotranspiration)
Dynamic input
River basin characteristics
• Land cover
• Soil composition and hydraulic properties
Static attributes
Historical observations
Model parameters
Loss
Observational datasets:
• Soil moisture
Observations (satellite estimates of hydrological variables and measured streamflow)
• Downscaled climate reanalysis
• Estimated model parameters
• Other dynamical input (LAI)
Surrogate model
Streamflow, soil moisture, evapotranspiration
Anomalies with respect to normal conditions
Drought prediction
• Downscalaed seasonal forecast
• Estimated model parameters
• Other dynamical input (LAI)
Surrogate model
Streamflow, soil moisture, evapotranspiration

Figure 3 – (i) A deep learning model is trained to emulate a physical based hydrological model (Wflow_sbm) (ii) work plan of differential parameter learning (dPL) for parameter estimation using trained model in first phase, past observations, reanalysis inputs and static attributes (iii) the trained model is finally run using downscaled dynamic forcing inputs derived from seasonal forecasts of ECMWF to predict drought conditions.

3.5 DT Application: Extreme rainfall, temperature, and wind weather event changes in response to climate change

3.5.1 ML Model Requirements

The application inputs are sets of 2-dimensional data for each climate variable. Each of them can therefore be considered as a 2D image, where each pixel coincides with a cell of the lat-lon grid. Considering three climate variables (temperature, precipitation, and wind), the images take the form of “RGB” (Red Green Blue) images, carrying three components for each pixel. Based on this consideration, the deep learning model identified for the DT is based on convolutional layers. Given the unsupervised learning conditions, the model used for anomaly detection is a Convolutional Variational Auto-Encoder – an image-compressing/rebuilding neural network (CVAE). Its inputs are daily squares over Western Europe, with the three climate variables values for each pixel - dimension n_{lat} x n_{lon} x 3. The tool can be extended to any geographical zone. The CVAE compresses (encodes) the input with convolutional layers to a smaller latent space. Distribution parameters are sampled from the encoded space, and the image is rebuilt with symmetrically transposed convolutional layers. A backward loop assesses the loss between the original and reconstruction image. The model is trained on historical data (about 30 years), assuming that history is the “normal” situation. The trained network is then applied to projection data: when the reconstruction loss is unusually high, the

situation is considered an anomaly. Post-processing these findings helps characterise the events with their duration, frequency, and intensity.
The software infrastructure is PyTorch.

A preliminary list of used data variables is available in Table 10.

Table 10 – Preliminary list of data variables for the DT Application

Variable name	Temporal Resolution	Spatial Resolution	Unit
Daily Maximum Near-Surface Air Temperature	daily	100km x 100km	K
Daily Precipitation	daily	100km x 100km	kg m^{-2} s^{-1}
Daily-Mean Near-Surface Wind Speed	daily	100km x 100km	m s^{-1}

3.5.2 Workflow Description

This section describes the logical flow of operations the DT needs to perform to carry out user requests.
In terms of computing workflow, the DT on extreme events follows those steps:

1. **Data Selection** Users select the relevant data to run their analysis; i.e., specify the geographical region of interest, climate reference time period (~30 years), time period of interest, season, and future climate scenarios (RCPs).
2. **Data Preprocessing** Selected data will be pre-processed so that it can be used as input to the ML model. In particular, data will be normalised with respect to the full dataset, and split into four season datasets.
3. **Model Training** The network will be pre-trained for Western Europe on a specific climate model, but for any other location or model it will have to be trained again. The training is based on 30 years of historical data. The hyperparameters are tuned to reach a trade-off between various applications.
4. **Projection Data Inference** The weights of the trained network are saved and applied to projection data of the same climate model – to avoid detecting biases between climate models as anomalies.
5. **Post-processing** The daily reconstruction errors will be analysed with respect to history or other IPCC scenarios, for each season. Percentiles and other statistical methods are used to determine the behaviour changes of historically rare events: their frequency of occurrence, duration, intensity (when relevant), and geographical extent.

6. **Improving results** Only one climate model can be studied at a time, but variants of the same model (ensemble members) can be used simultaneously and therefore augment available data. A trained network on more data will generate more accurate results and reduce uncertainty – it can be evaluated depending on climate models and RCPs. Results from different models (and therefore different networks) can also be aggregated.
7. **Visualising** Final results can be stored as output files or visualised in the notebook interface using different Python visualisation modules, e.g. plots, maps.

The script of the core model is available on GitHub (in progress):
https://github.com/cerfacs-globc/xtclim.

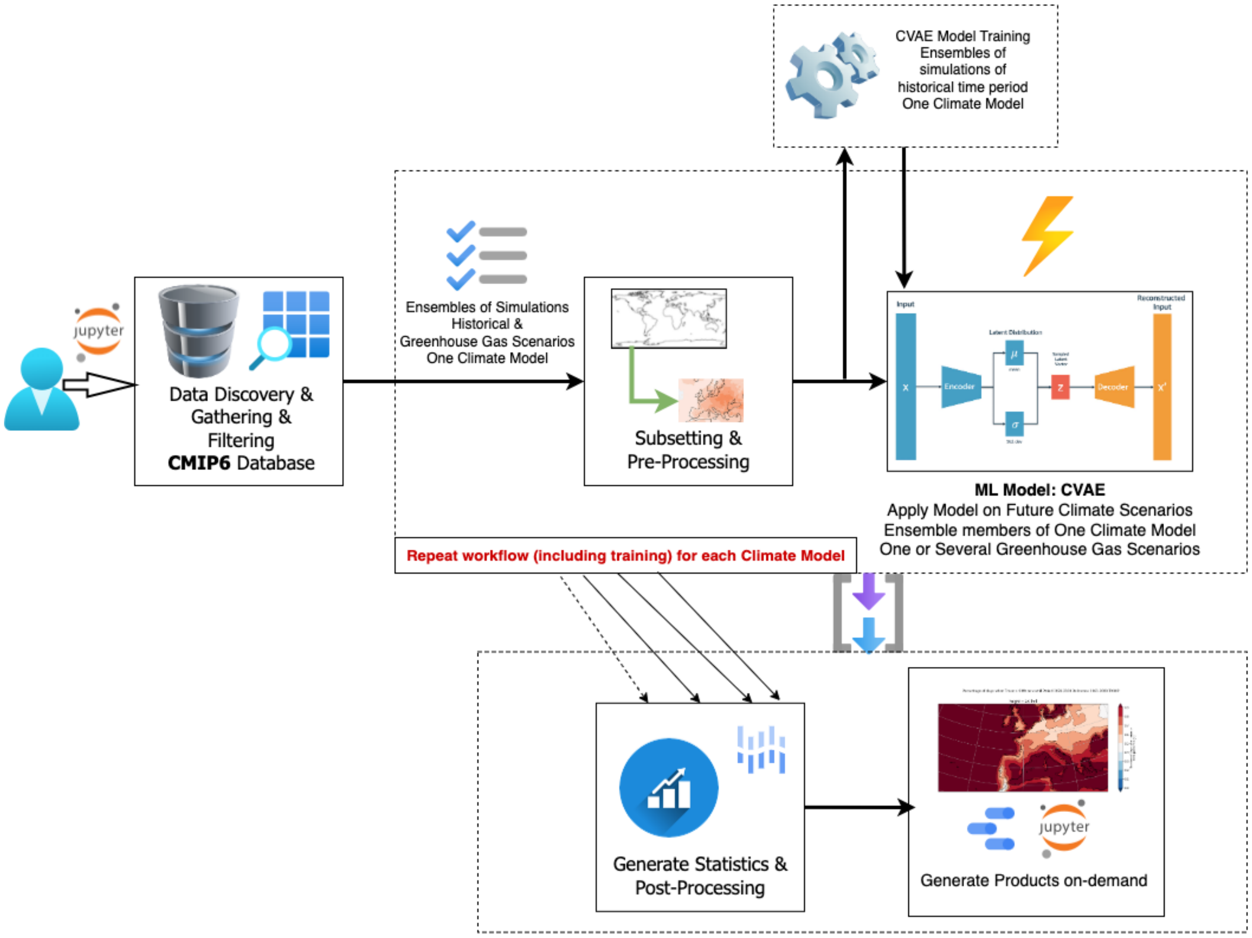
CVAE Model Training
Ensembles of simulations of historical time period
One Climate Model
jupyter
Data Discovery & Gathering & Filtering
CMIP6 Database
Ensembles of Simulations
Historical &
Greenhouse Gas Scenarios
One Climate Model
Subsetting & Pre-Processing
Input
Reconstructed Input
Latent Distribution
X
Encoder
μ
σ
Sampled Latent Vector
Z
Decoder
X'
ML Model: CVAE
Apply Model on Future Climate Scenarios
Ensemble members of One Climate Model
One or Several Greenhouse Gas Scenarios
Repeat workflow (including training) for each Climate Model
Generate Statistics & Post-Processing
jupyter
Generate Products on-demand

Figure 4 – Schematic overview of the main components of the DT Application: Extreme rainfall, temperature, and wind weather event changes in response to climate change. It uses a Convolutional Variational Auto-Encoder method to detect anomalies.

3.6 DT Application: Flood climate impact in coastal and inland regions

3.6.1 Model Requirements

In addition to the hydrological and flood inundation models and workflow described in Section 3.3, the flood *climate impact* DT for coastal and inland regions uses models and tools to quantify impacts on buildings, utilities, roads and accessibility under different future climate scenarios applying different mitigation and adaptation strategies.

Instead of using the meteorological data referred to in Section 3.3 to run the hydrological models (Wflow) and the flood inundation model (SFINCS), here climate projection data will be used to simulate flood depth and extent under future climate conditions. The requirements and workflow to arrive at the deterministic and probabilistic flood maps are the same as in Section 3.3, but using climate projection data instead of weather forecasts.

Once the deterministic and probabilistic flood maps have been generated, these are fed into the flood impact assessment tool (Delft-FIAT), and the Resilience Assessment and Action perspective for Critical infrastructurE (RA2CE) model.

Using the flood maps, and additional inputs such as depth-damage functions, asset locations and their maximum potential damages, Delft-FIAT derives asset-level and aggregated damages and risk. For each asset specified in the exposure dataset, the water depth or elevation is subtracted from the flood map at the location of the assets; water elevations are converted to water depths using the ground elevation of each asset. When calculating partial flooding, Delft-FIAT will extract either the average or maximum water depth and the fraction of the building that is flooded. The inundation depth within buildings is obtained by subtracting from the water depth the ground floor height. Delft-FIAT derives the damage fraction for each asset using its inundation depth and interpolating over its depth-damage curve. The damage to the asset is then calculated as the product of the maximum potential damage and the damage fraction. When calculating partial flooding, the damages will be reduced by the fraction of the building that is dry. When the user inputs return-period flood maps, Delft-FIAT will calculate the associated return-period damages, and then integrate these to derive the expected annual damages.

Similarly to Delft-FIAT, RA2CE uses the flood maps combined with road network data, road damage functions, road depth damage curves, population data, and important locations to calculate damages to road networks including the cascading effects on society due to disruptions of the infrastructure network.

Additionally for this DT, users can select and input flood mitigation and adaptation strategies such as flood walls, levees, pumps and culverts, raising properties and flood proofing properties and run scenarios to estimate the impact said strategies have on mitigating damages related to floods.

Setting up Delft-FIAT for the purposes described above requires exposure data including building footprints, roads, and asset classification data, as well as vulnerability data including depth damage curves and functions.

To set up RA2CE, additionally requires road infrastructure data as well as data on population and important locations.

An overview of data requirements to set up Delft-FIAT and RA2CE can be found in Tables 7-10 of interTwin D7.1 [R6]

3.6.2 Workflow Description

The workflow to set up SFINCS and Wflow is described in Section 3.3.2. For the flood climate impact DT, there are 3 additional workflows, namely:

1. Setting up the Delft-FIAT and RA2CE models
2. Producing a baseline damage and impact assessment, based on flood depth and extent from running Wflow and SFINCS with future climate projection data
3. Selecting flood mitigation and adaptation strategies and rerunning the models

Setting up the Delft-FIAT and RA2CE models for a specific region of interest comprises the following steps:

1. Set up Delft-FIAT
    a. A user defines a geographic region of interest.
    b. A user selects available global datasets from which to build the model. Optionally a user can upload and use local data instead of global data.
    c. A user checks and/or links the classification of the assets to the asset damage functions.
2. Set up RA2CE
    a. A user defines a geographic region of interest.
    b. A user selects available global datasets from which to build the model. Optionally a user can upload and use local data instead of global data.
    c. A user selects the road types to include.
    d. A user checks and/or links the road types to the road damage functions.

Once the SFINCS, Wflow, Delft-FIAT and RA2CE models have been set up, they can be run to produce baseline damage assessments under future climate scenarios as follows:

1. A user selects a time period to simulate, e.g. a specific historical event
2. A user selects the preferred forcing data for SFINCS and Wflow, e.g.
    a. Wflow – temperature, precipitation, and potential evapotranspiration
    b. SFINCS – in addition to precipitation, select data for tides, surges, waves, and sea level
3. A user then runs Wflow
4. A user then runs SFINCS with river discharges from Wflow
5. The water levels produced by SFINCS are interpolated to the digital elevation model selected by the user in the Setup SFINCS step 1c (Section 3.3.2), to estimate flood depths and extent
6. A user then runs Delft-FIAT with the flood depth maps from SFINCS
7. A user then runs RA2CE with the flood depth maps from SFINCS
8. A user can visualise the resulting damages and impacts spatially in the Jupyter Notebook or in any GIS software

Once the baseline damage assessments under a future climate scenario has been run, the workflow to assess the impact of selecting mitigation and adaptation strategies can be run as follows:

1. A user selects the measures and specifies the properties of the measures that they want to test
2. A user runs a FloodAdapt module that implements the changes in the models corresponding to the measures. It is currently possible to implement measures that make changes to the SFINCS and Delft-FIAT models.
3. Depending on the choice of measures, the user must rerun the following models with the updated model data:
    a. If the user selected a flood wall, pump, levee, and/or culvert, the user must rerun SFINCS, Delft-FIAT and RA2CE with the updated models.
    b. If the user selected buyouts, flood proofing, and/or raising properties, the user can just rerun Delft-FIAT.
4. A user can visualise the resulting changes in damages and impacts spatially in the Jupyter Notebook or in any GIS software

In the above workflows, this DT application will leverage the following capabilities from other WPs of interTwin (summarised in Figure 2, Section 3.3)

- The DT will leverage capabilities developed in WP7, T7.6: Hydrological model data processing thematic module.
- Additionally for preprocess forcing and boundary condition data, this DT aims to leverage developments from WP6, T6.3 Data fusion.
- To run the models using containers, this DT will rely on functionality from WP6, T6.4 Big Data Alalysys.
- For post-processing and visualisation, this DT will rely on WP6, T6.2 FAIR data quality evaluation.
- For workflow composition and execution, this DT will leverage developments from WP6, T6.1 Workflow composition and WP6, T6.4 Workflow backend.
- Finally, all data and compute resources are leveraged from WP5, the DTE infrastructure including Notebooks as a Service.

4 Conclusions

The first version of the architecture design of interTwin DTs applications for WP4 concerned with the environmental domain was developed during the first year of the project. In the current deliverable D4.1, the main focus is on defining a preliminary workflow and identifying possible beneficiaries for each DTs application. As part of each task (4.5, 4.6, and 4.7), we described user stories, key requirements, expected outcomes, and the steps they planned to accomplish their specific goals. Additionally, we outlined in section 3 the layout and necessary steps that should be considered when designing an individual DT application.

The next step will be to complete the development of the first version of each DT Application, followed by their release and integration with the DTE corresponding to the deliverable D4.3 planned for April 2024. That deliverable will also serve as feedback for the first DTE components to be released in Dec 2023 and integrated by the DTs.

5 References

Reference	
No	**Description / Link**
R1	Imhoff, R., et al., Scaling point-scale (pedo) transfer functions to seamless large-domain parameter estimates for high-resolution distributed hydrologic

	modelling: An example for the Rhine River. Water Resources Research, 2020. **56**(4): p. e2019WR026807. DOI: **10.1029/2019WR026807**
R2	Schellekens, J., Van Verseveld, W., Visser, M., Winsemius, H., Euser, T., Bouaziz, L., Boisgontier, H, *Openstreams/wflow:*. 2021. DOI: **10.5281/zenodo.593510**
R3	Isabel Campos, Donatello Elia, Germán Moltó, Ignacio Blanquer, Alexander Zoechbauer, Eric Wulff, Matteo Bunino, Andreas Lintermann, Rakesh Sarma, Pablo Orviz, Alexander Jacob, Sandro Fiore, Miguel Caballer, Bjorn Backeberg, Mariapina Castelli, Levente Farkas, & Andrea Manzi. (2023). interTwin D6.1 Report on requirements and core modules definition (V1 Under EC review). Zenodo DOI: **https://doi.org/10.5281/zenodo.8036987**
R4	Bauer-Marschallinger, B., Cao, S., Tupas, M. E., Roth, F., Navacchi, C., Melzer, T., Freeman, V., & Wagner, W. (2022). Satellite-Based Flood Mapping through Bayesian Inference from a Sentinel-1 SAR Datacube. Remote Sensing, 14(15), Article 3673. DOI: **https://doi.org/10.3390/rs14153673**
R5	Schramm, M., Pebesma, E., Milenković, M., Foresta, L., Dries, J., Jacob, A., Wagner, W., Mohr, M., Neteler, M., Kadunc, M., Miksa, T., Kempeneers, P., Verbesselt, J., Gößwein, B., Navacchi, C., Lippens, S., & Reiche, J. (2021). The openEO API - Harmonising the Use of Earth Observation Cloud Services Using Virtual Data Cube Functionalities. Remote Sensing, 13(6), 1125. DOI **https://doi.org/10.3390/rs13061125**
R6	Michele Claus, Alexander Jacob, Björn Backeberg, Frederique de Groen, Joost Buitink, Roel de Goede, Donatello Elia, Gabriele Accarino, Sandro Fiore, Christian Pagé, Matthias Schramm, Bernhard Raml, & Christoph Reimer. (2023). interTwin D7.1 Report on requirements and thematic modules definition for the environment domain (V1 Under EC review). Zenodo DOI. **https://doi.org/10.5281/zenodo.8036991**