D8.5 - Early Warning and Digital Twin Platform release notes and user manual (version 1.0)

DOCUMENT TRACKS DETAILS

Project acronym	SCORE
Project title	Smart Control of the Climate Resilience in European Coastal Cities
Starting date	01.07.2021
Duration	48 months
Call identifier	H2020-LC-CLA-2020-2
Grant Agreement No	101003534

Deliverable Information	
Deliverable number	D8.5
Work package number	WP8
Deliverable title	Early Warning and Digital Twin Platform release notes and user manual
Lead beneficiary	MBI srl
Author(s)	Giovanni Serafino (MBI), Ivan Boscaino (MBI), Andrea Rucci (MBI), Attilio Vaccaro (MBI), José Gómez-Barrón (UCD)
Due date	31/08/2023
Actual submission date	31/08/2023
Type of deliverable	Report
Dissemination level	Public

VERSION MANAGEMENT

Revision table			
Version	Name	Date	Description
V 0.1	Giovanni Serafino (MBI), Ivan Boscaino (MBI), Andrea Rucci (MBI), Attilio Vaccaro (MBI), José Gómez-Barrón (UCD)	10/08/2023	First draft
V 0.2	Cécil Meulenberg (ZRS), Filippo Giannetti (UNIPI)	23/08/2023	Updated draft internally reviewed

V 0.3	Giovanni Serafino (MBI), Ivan Boscaino (MBI), Andrea Rucci (MBI)	25/08/2023	Updated draft after contribution from partners
V1.0	Giovanni Serafino (MBI) Iulia Anton (ATU) Salem Gharbia (ATU)	28/08/2023	Final version

All information in this document only reflects the author's view. The European Commission is not responsible for any use that may be made of the information it contains.

LIST OF ACRONYMS AND ABBREVIATIONS

Acronym / Abbreviation	Meaning / Full text
2D	Two-dimensional
AI	Artificial Intelligence
API	Application Programming Interface
AR5	IPCC fifth Assessment Report
AuthN/AuthZ	Authentication/Authorization
CCLL	Coastal City Living Lab
CSV	Comma-separated Values
DB	Database
DEM	Digital Elevation Model
DMP	Data Management Plan
DSM	Digital Surface Model
DT	Digital Twin
DTM	Digital Terrain Model
EBA	Ecosystem-Based Approach
EWMA	Exponential Weighted Moving Average
EWS	Early-Warning Support
GeoJSON	Geographic Javascript Object Notation
GIS	Geographic Information System
GM	Gaussian Mixture
GUI	Graphical User Interface
IPCC	Intergovernmental Panel on Climate Change

JSON	Javascript Object Notation
MAD	Median Absolute Deviation
ML	Machine Learning
OGC	Open Geospatial Consortium
PCA	Principal Component Analysis
PFA	Principal Feature Analysis
RCP	Representative Concentration Pathway
REST	Representational State Transfer
RPO	Regional Planning Organization
SIP	SCORE Information and communication technologies (ICT) platform
SIR	Servizio Idrogeologico Regione Toscana
SME	Small-Medium Enterprise
UN	United Nations
URL	Uniform Resource Locator
USE	User Scenario Evaluation
WFS	Web Feature Service
WMS	Web Map Service
WMTS	Web Map Tile Service
WP	Work Package

BACKGROUND: ABOUT THE SCORE PROJECT

SCORE is a four-year EU-funded project aiming to increase climate resilience in European coastal cities.

The intensification of extreme weather events, coastal erosion and sea-level rise are major challenges to be urgently addressed by European coastal cities. The science behind these disruptive phenomena is complex, and advancing climate resilience requires progress in data acquisition, forecasting, and understanding of the potential risks and impacts for real-scenario interventions. The Ecosystem-Based Approach (EBA) supported by smart technologies has potential to increase climate resilience of European coastal cities; however, it is not yet adequately understood and coordinated at European level.

SCORE outlines a co-creation strategy, developed via a network of 10 coastal city 'living labs' (CCLLs), to rapidly, equitably and sustainably enhance coastal city climate resilience through EBAs and sophisticated digital technologies.

The 10 coastal city living labs involved in the project are: Sligo and Dublin, Ireland; Barcelona/Vilanova i la Geltrú, Benidorm and Basque Country, Spain; Oeiras, Portugal; Massa, Italy; Piran, Slovenia; Gdansk, Poland; Samsun, Turkey.

SCORE will establish an integrated coastal zone management framework for strengthening EBA and smart coastal city policies, creating European leadership in coastal city climate change adaptation in line with The Paris Agreement. It will provide innovative platforms to empower stakeholders' deployment of EBAs to increase climate resilience, business opportunities and financial sustainability of coastal cities.

The SCORE interdisciplinary team consists of 28 world-leading organisations from academia, local authorities, Regional Planning Organizations (RPOs), and small-medium enterprises (SMEs) encompassing a wide range of skills including environmental science and policy, climate modelling, citizen and social science, data management, coastal management and engineering, security and technological aspects of smart sensing research.

EXECUTIVE SUMMARY

This document is a deliverable of the SCORE project, funded under the European Union's Horizon 2020 research and innovation programme under grant agreement No 101003534.

The D8.5 represents the release notes and user manual of the geographic information system (GIS)-Based Digital Twin – Early-Warning Support (DT-EWS) prototype, a result of the activities carried on in task T8.2 "Development of the Early Warning Support and Digital Twin Platform". It is worth noting that the present report complements the deliverable D8.4 "Early Warning and Digital Twin Platform prototype", submitted at the same time, that describes the first version of the DT-EWS system demonstrator.

The content of the present document exposes in detail the system functional architecture, explaining its structure and main functionalities, as well as its interactions with the users. It is a guide on how to operate and interact with the DT-EWS, describing the functionalities implemented in the Platform and operational procedures to use its services. Since the DT-EWS prototype is a backend system, the guide is meant for technicians.

In the same way as for D8.4, multiple versions of the deliverable will be produced during the project, since the first one will be refined at different stages to correct bugs and add new features. Each release will be tagged with a unique version number.

The system is composed of two main blocks, the User Scenario Evaluation (USE) subsystem and the EWS subsystem. The former is meant to be employed for running simulations enabling what-if and long-term analyses on scenarios built by users as a mixture of real, ground-truth data from the coastal city, and hypothetical data, e.g., regarding weather events, sea state, and solutions based on EBAs implemented within the urban study area. The EWS subsystem, on the other hand, is intended to run continuously without inputs from the users, since it has to monitor the current weather situation. By leveraging on real-time data from sensors distributed over the study area and on weather forecasts, it can make projections of the city's floodings risk. This way, it can raise alerts in case floodings are foreseen to happen with water levels that can put at risk inhabitants, or cause major economic damages. The final version of the system will take into account not only risks related to flooding due to rain and river discharge, but also the effects of coastal erosion. The here-described, first released version of the DT-EWS system does not yet include functionalities related to erosion, since they depend on other work package (WP) activities that have not yet been completed at this stage of the project.

LINKS WITH OTHER PROJECT ACTIVITIES

This document takes inputs that drove the DT-EWS system design, from the SCORE project deliverables D8.1 - GIS Based Digital Twin Platform functional requirements [1], D8.2 - GIS Based Early Warning and Digital Twin Platform system architecture and design [2], and GIS Based Early Warning and Digital Twin Platform Interface Control Document [3]. It is also based on the downscaled models elaborated in SCORE project WP3, the SCORE Information and communication technologies (ICT) platform (SIP) developed in SCORE project WP5, the exposure and vulnerability maps provided by SCORE project WP6, the list of solutions based on EBAs specified in SCORE project WP7.

The outputs of this document are relevant for the SCORE WP8 system further development, integration, deployment, (project tasks T8.2, T8.3, T8.4), and for the subsequent activities of system validation and assessment, whose plan is reported in detail in D8.10 [4] (project task T8.5).

TABLE OF CONTENT

INDEX OF FIGURES

INDEX OF TABLES

1.   INTRODUCTION

The activities carried out in WP8 of the SCORE project are focused on the design, development, and deployment of a DT-EWS system capable of supporting the increase of the resilience of coastal cities against the ongoing climate change. Nowadays, the use of digital models of real systems to simulate the systems behaviour and evolution under a set of starting boundary conditions is increasingly gaining momentum. Such a model is known as Digital Twin [5], a dynamic virtual copy of a physical asset, process, system or environment that behaves identically to its real-world counterpart. Typically, a digital twin ingests data and virtually reproduces processes, enabling the prediction of possible outcomes, level of performance, and issues that the "physical twin" may experience, without the need of observing really happening processes. Initially conceived for system maintenance and product design or testing, this paradigm gained popularity in many other fields, in particular in the design of smart cities in connection with climate change [6] and flooding hazard [7].

In the context of the impact of extreme events related to climate change, the early-warning systems play a crucial role in the possibility of mitigating the damage on population and infrastructures in case of floods, by detecting pre-conditions and predicting the onset of a disaster before it occurs, and by providing real-time information and alerts during an event. Usually, early-warning systems are managed by institutional agencies that leverage on them for decision support, to send alarms to multiple stakeholders including government, private companies, and the general public. Early-warning systems have been clearly defined by United Nations (UN) General Assembly resolution 69/284 (December 2016),

> "*An integrated system of hazard monitoring, forecasting and prediction, disaster risk assessment, communication and preparedness activities systems and processes that enables individuals, communities, governments, businesses and others to take timely action to reduce disaster risks in advance of hazardous events.*"

The importance of this kind of systems is witnessed by the recent initiative of the UN Organizations, that through the World Meteorological Organization launched Early Warnings For All, which is an attempt "to ensure that everyone on Earth is protected from hazardous weather, water, or climate events through life-saving early warning systems by the end of 2027" [8].

In this scenario, the SCORE DT-EWS system aims at simulating the impact of extreme weather events on coastal cities, also considering the adoption of possible solutions based on EBAs, providing relevant stakeholders and decision-makers with an effective tool for urban and territorial planning, and for supporting the early-warning and decision activities in case of oncoming extreme events. More specifically, the USE subsystem natively supports participatory approaches to co-design adaptation solutions, particularly focusing on the ecosystem-based ones, allowing users to run simulations regarding urban weather, what-if scenarios or climate long-term scenarios of global warning. By providing a quantification of the flooding hazard, it can support technical figures, policymakers, and citizens as well, in realistically evaluating, also economically, the impact of adaptation and other possible actions aimed at enhancing the coastal cities resilience against coastal erosion and adverse events caused by the weather. On the other hand, the EWS subsystem processes can run continuously, periodically checking the status of environmental parameters collected by official or citizen-science [9] sensors and models allowing an all-day monitoring of the flooding risk, raising alerts when some relevant parameters (or combination of parameters) exceed a set threshold.

Typically, early-warning systems are supposed to be used by authorized users and are fed by real-time data coming from all the sensors disseminated on the urban area, together with weather forecasts

provided by local services. The issuing of different levels of warnings or alerts is always supervised and is regulated by national or regional laws that imply specific legal responsibilities. The SCORE EWS subsystem does not mean to replace already existing and running early-warning frameworks, but aims at supporting their activities, guaranteeing a largest as possible interoperability with other systems.

Finally, it is important to highlight that, after the first release related to D8.4 and D8.5, the SCORE DT-EWS system design will be fine-tuned and further driven by feedback and inputs from users, especially from those with a technical background that can benefit from the DT-EWS for their evaluation and planning activities. Moreover, the test plan reported D8.10 [4] will be followed for the validation of the system as a whole, allow detecting and fixing eventual bugs, and improve the system design. In addition, the system will be later enriched with other functionalities related to coastal erosion, to enable the users consider risks related to erosive phenomena. This set of functionalities is not included in the present system release, since in it relies on results from other project activities that are expected to be completed in the second half of the project.

1.1 Scope of the document

This document is part of the technical deliverables of WP8 (Development of integrated Early Warning Support System and spatial Digital Twin solution prototypes) of the SCORE project. In particular, this deliverable is aimed at describing the developed GIS-Based DT-EWS system in its architecture and functionalities. Here, we report on the modules that compose the system, as well as the application programming interfaces (APIs) employed to share and exchange information and commands between internal interfaces and external systems.

This document is structured as follows: after the Introduction, in Sect. 2 the set of possible system inputs are listed and classified, whereas Sect. 3 introduces the high-level system architecture and its modules, divided in the two subsystems USE and EWS, showing the block scheme of the system and the involved processes, including the underlying software functionalities, like the employed digital hydraulic model. Then, Sect. 4 explains the user operations and the Sect. 5 describes the system outputs. Finally, Sect. 6 is dedicated to the interactions with other systems, including the system front-end and other external systems. The integration of the GUI and the inter-operation with other systems is part of the activities of the ongoing task T8.4 "Integration and Deployment of GIS based Early Warning and Digital Twin Platform". The last Sect. 7 briefly summarizes the document content and outlines the way ahead in the WP activities.

1.2 Massa, the development pilot case

SCORE established 10 CCLLs bringing together citizens, scientists, policy makers and other stakeholders around specific challenges related to coastal city risks in an open collaborative spirit. Within the project, each CCLL is supported by creating a core team including members of academia, public actors, private actors and civil groups, to ensure that all parties play their part in the different stages of the process. During the project proposal stage, a rapid needs assessment via an online questionnaire survey was conducted by the project consortium to understand the specific needs and expectations of the CCLLs. The questionnaire survey included, among others, basic information (geographical coastal area, population, environmental contextual conditions, natural and climatic related hazards, impacts), objectives for creating and implementing the CCLL, expected outcomes, potential stakeholders, technical and financial capacities, previous experience in relevant projects and level of expertise and experience in different activities associated with each WP. A rubric was developed to rate the CCLLs based on their responses which allowed

for cross-comparison of their responses to a questionnaire probing their experience, capacity and resource. The top-scored CCLLs per WP suggested the frontrunners for that WP. The results from the rapid needs assessment then enabled the initial identification of frontrunner and fellow CCLLs for each WP. This allows all CCLLs to be engaged as both a frontrunner and a follower on different project activities. This mutual exchange of knowledge will return multiple benefits of engagement, empowerment and learning throughout the whole process.

Three out of the ten CCLLs involved in the SCORE project are front-runners of WP8, namely Oarsoaldea, Vilanova i la Geltrú (both located in Spain), and Massa (in Italy). Their geographic positions are shown in Figure 1, showing how they are somehow representative of all the CCLLs in the project: two of them are located on the Mediterranean Sea, one on the Atlantic Ocean.

The demonstrator of the SCORE DT-EWS system has been developed taking the city of Massa as a pilot case; therefore, all the maps, results, and examples shown in this deliverable represent almost exclusively that coastal city. All the activities of tasks T8.4 that are focused on the system deployment will start from this city, and many interactions are expected with the final users and relevant stakeholders in Massa, for a demonstrator fine-tuning driven by users' needs.

The motivation behind the choice of Massa as the "forerunner of the frontrunners" is due to the fact that MBI, which is the partner responsible of the DT-EWS design and development, LaMMA, the leader of WP3 that will provide the down-scaled long-term climate models, and UNIPI, responsible for the development of the SIP, are all part of the Massa CCLL core team. Therefore, information exchange and any kind of relevant interactions with the city stakeholders for the system development and deployment are expected to be particularly direct and smooth.

Figure 1. Geographical position of WP8 front runners.

During the deployment phase on Massa, new versions of the DT-EWS will be gradually implemented, tailored on the other two frontrunners, and later deployed, following a similar process as in Massa, based on frequent interactions with the users an the stakeholders.

2.   SYSTEM INPUTS

The proposed DT-EWS needs a large number of different inputs, provided by a heterogenous set of data sources. As it will be clearer from the following detailed system description , a portion of these inputs are provided by users, whereas some data come from sensors. Among the data provided by users there can be real maps as well as hypothetical pieces of information that the users would like to simulate to understand their impact on the hydraulic context of their coastal city.

2.1 The coastal cities baseline data

The minimum set of data required by the DT-EWS system to properly run consistent and complete simulations, ensuring the minimum functionalities that correspond to the requirements reported in D8.2 [2], is represented by the real maps of the study area identified for each CCLL . More specifically:

1.   The city digital elevation model (DEM), digital terrain model (DTM), or digital surface model (DSM), preferably with high spatial resolution. Maps with grid cells of 2 m x 2 m are needed to get outputs on flooding with a good degree of detail, but 5 m x 5 m can be also acceptable. These maps are provided in the form of .shp files or similar, that can be viewed with tools like Q-GIS. It is also important that buildings are clearly distinguishable from the ground

2.   A map with the land usage and cover of the study area, containing information on the Manning coefficient [10] and terrain porosity

3.   Geometry of the rivers flowing through the CCLL area, i.e., a .txt or .csv file with coordinates of each river course centre, bed width, Manning coefficient, bed elevation.

The CCLLs must provide these data as maps or other kind of documents and upload them on the SIP as stand alone files, that will be employed by the DT-EWS. Of course, the users must also update the provided maps in case there is any major change in the data previously stored on the SIP.

Referring to D8.2 [2], these minimum baseline data enable the system being compliant with the following functional requirements:

SCORE-REQ-DTEW-FUN-001-01: Early warnings for coastal flooding

SCORE-REQ-DTEW-FUN-003-01: Early warnings for storm surges

SCORE-REQ-DTEW-FUN-006-01: Sand dunes protection

SCORE-REQ-DTEW-FUN-012-01: Spread of overflooding and flooded areas map

SCORE-REQ-DTEW-FUN-013-01: Ground / Surface water level

SCORE-REQ-DTEW-FUN-024-01: Condition of river bed

SCORE-REQ-DTEW-FUN-027-01: Data monitoring at the local level

SCORE-REQ-DTEW-FUN-030-01: City risk assessment

SCORE-REQ-DTEW-FUN-032-01: Coastal erosion

SCORE-REQ-DTEW-FUN-044-01: Land Use

SCORE-REQ-DTEW-FUN-045-01: Land Use Change

SCORE-REQ-DTEW-FUN-046-01: Simulation

SCORE-REQ-DTEW-FUN-047-01: Localisation

Moreover, the above-mentioned data also allow the system implementing the following functional specifications [2]:

SCORE-FSPEC-DTEW-002: GIS-based interface

SCORE-FSPEC-DTEW-051: Map layers selection capability

SCORE-FSPEC-DTEW-052: Types of map layers

SCORE-FSPEC-DTEW-054: Base maps attribute data

SCORE-FSPEC-DTEW-056: Simulation maps attribute data

SCORE-FSPEC-DTEW-070: Thematic layers

SCORE-FSPEC-DTEW-071: Thematic Layers – Coastal flooding

SCORE-FSPEC-DTEW-074: Thematic Layers – Sand Dunes Protection

SCORE-FSPEC-DTEW-079: Thematic Layers – Spread of overflooding and maps overflooding

SCORE-FSPEC-DTEW-087: Thematic Layers – Forest cover

SCORE-FSPEC-DTEW-088: Thematic Layers – Coastal Erosion

SCORE-FSPEC-DTEW-092: Thematic Layers – Land Use

SCORE-FSPEC-DTEW-101: GIS based approach

SCORE-FSPEC-DTEW-110: Map Content

SCORE-FSPEC-DTEW-153: Entity types

SCORE-FSPEC-DTEW-159: Definition of map

SCORE-FSPEC-DTEW-469: Early warnings for Coastal Flooding

SCORE-FSPEC-DTEW-470: Early warnings for Storm Surges

SCORE-FSPEC-DTEW-474: Condition of river bed

2.2 The climate and weather data

In addition to data related to different map layers of the CCLL, other data are relevant to take into account the evolutions of weather, in the short term, and of climate, in the long term.

2.2.1 Long-term models

The SCORE DT-EWS aims at providing users with the possibility of running simulations to understand the impact of eventual policies they would like to adopt, considering a long time horizon, both in the past and in the future. For this reason, the SCORE consortium, through the activities of WP3, is elaborating down-scaled climate models to extract data related to river discharge and sea level for each CCLL. These models are derived either from historical data sets, for the scenarios related to the past years 1956-2005, or from the Representative Concentration Pathways

(RCPs) adopted by the Intergovernmental Panel on Climate Change (IPCC) [11], i.e., on the trajectories of greenhouse gas concentration in the year 2100.

The RCPs are classified depending on the associated radiative forcing, in turn representing a specific global warming future scenario, as reported in the IPCC fifth Assessment Report (AR5) [11], where four possible scenarios are analysed, namely RCP2.6, RCP4.5, RCP6, and RCP8.5. The number after the acronym "RCP" represents the radiative forcing, i.e., the energy flux per surface unit through the Earth atmosphere, expressed in W/m^2. As an example, the RCP2.6 scenario is related to an expected radiative forcing of 2.6 W/m^2. These scenarios can be elaborated to make projections on the evolution of different relevant parameters, like global or regional temperature, or sea level rise. Figure 2 shows the plot reported in [11], representing the global average surface temperature change from 2006 to 2100, obtained by multi-model simulations, with changes relative to 1986–2005. Time series of projections and a measure of uncertainty (shading) are shown for scenarios RCP2.6 (blue line) and RCP8.5 (red line). Furthermore, the mean and associated uncertainties averaged over the years 2081–2100 are given for all RCP scenarios as coloured vertical bars at the right side of the plot. It is worth noticing that this analysis predicts a mean temperature variation of around 1 °C in the most optimistic scenario (RCP2.6), and more than 4 °C in the worst scenario (RCP8.5).
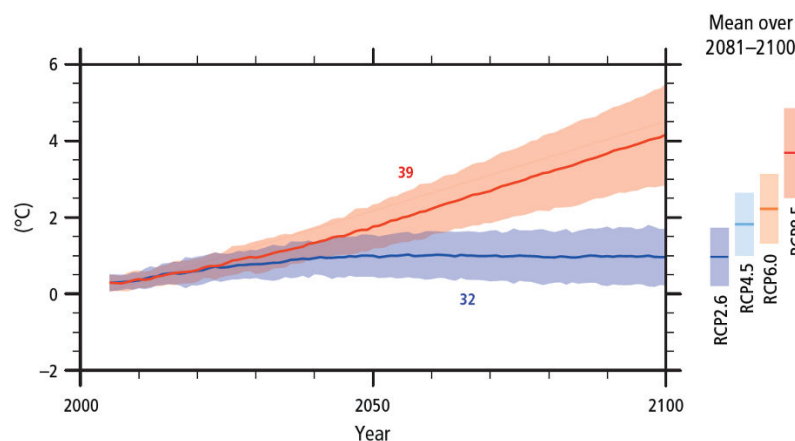
Figure 2. Global average surface temperature change from 2006 to 2100 as determined by multi-model simulations. All changes are relative to 1986–2005 [11].

In the present DT-EWS system release, the data related to the area of Massa that have been integrated in the USE subsystem to let the user create different scenarios, are those extracted from RCP4.5 and RCP8.5, considered as related to medium and high risk, respectively, in two possible 50-year long time spans, namely 2015-2065 and 2045-2095. Finally, among the possible future scenarios with a given level of risk and time span, the user can select events with different return time, i.e. 5, 25, 50, 100, 200, or 500 years.

Referring to D8.2 [2], the data of the long-term past or future scenarios enable the system being compliant with the following functional requirements:

SCORE-REQ-DTEW-FUN-007-01: Easily readable models

SCORE-REQ-DTEW-FUN-012-01: Spread of overflooding and flooded areas map

SCORE-REQ-DTEW-FUN-031-01: Sea level rise

SCORE-REQ-DTEW-FUN-046-01: Simulation

Moreover, the above-mentioned data also allow the system implementing the following functional specifications [2]:

SCORE-FSPEC-DTEW-071: Thematic Layers – Coastal flooding

SCORE-FSPEC-DTEW-079: Thematic Layers – Spread of overflooding and maps overflooding

SCORE-FSPEC-DTEW-158: Simulation scenario

SCORE-FSPEC-DTEW-251: Simulation Events

SCORE-FSPEC-DTEW-351: Simulation

SCORE-FSPEC-DTEW-404: Sea level rise

2.2.2   Weather forecasts

Another important source of data is represented by weather forecasts in a short time horizon. Indeed, the system, in particular the EWS subsystem, to monitor the current hydraulic situation of a city, needs to know the impact of weather in the next hours. Therefore, it needs to be fed with data related to intensity, duration, and distribution of precipitations, to river discharge, to sea state and risk of storm surges.

These data are expected to be provided by local weather services in each CCLL every few hours, possibly directly obtained through dedicated APIs connected to the service data centre, or by acquiring a forecast bulletin in a format that the system can read.

Referring to D8.2 [2], information on weather forecasts enables the system being compliant with the following functional requirements:

SCORE-REQ-DTEW-FUN-001-01: Early warnings for coastal flooding

SCORE-REQ-DTEW-FUN-003-01: Early warnings for storm surges

SCORE-REQ-DTEW-FUN-012-01: Spread of overflooding and flooded areas map

SCORE-REQ-DTEW-FUN-013-01: Ground / Surface water level

SCORE-REQ-DTEW-FUN-019-01: Distribution of precipitation regime

SCORE-REQ-DTEW-FUN-020-01: Humidity level

SCORE-REQ-DTEW-FUN-025-01: Provincial heat islands

SCORE-REQ-DTEW-FUN-027-01: Data monitoring at the local level

SCORE-REQ-DTEW-FUN-028-01: Emergency tools

SCORE-REQ-DTEW-FUN-030-01: City risk assessment

SCORE-REQ-DTEW-FUN-042-01: Risk Maps

Moreover, the weather forecasts data also allow the system implementing the following functional specifications [2]:

SCORE-FSPEC-DTEW-079: Thematic Layers – Spread of overflooding and maps overflooding

SCORE-FSPEC-DTEW-084: Thematic Layers – Temperature of the region

SCORE-FSPEC-DTEW-085: Thematic Layers – Distribution of precipitation regime

SCORE-FSPEC-DTEW-117: Risk maps

SCORE-FSPEC-DTEW-451: Short-Term Forecasting

SCORE-FSPEC-DTEW-452: Warning

SCORE-FSPEC-DTEW-469: Early warnings for Coastal Flooding

SCORE-FSPEC-DTEW-470: Early warnings for Storm Surges

SCORE-FSPEC-DTEW-471: Ground/Surface water level

SCORE-FSPEC-DTEW-473: Humidity Level

SCORE-FSPEC-DTEW-475: Heat intensity

2.2.3   Data from sensors

To realise a system that is capable of monitoring the coastal city current situation, it is crucial to have reliable data coming from a heterogeneous set of sensors distributed on the city area. Both historical and real-time datasets can be important for the system design and operation. Then most important parameters that the system must monitor are:

- Precipitation intensity, especially rain rate (typically measured in mm/h, from rain gauges or disdrometers)

- Rivers flow rate (measured, e.g., in m^3/s) or level (in m)

- Sea state parameters, like:

    o   tide level (from tide gauges, in m)

    o   wave height and period (usually measured in m and s, coming from wave gauges)

- Sewage network state (from dedicated sensors indicating levels or other quantities).

Other data related to values of other weather-related parameters, like humidity, temperature, wind speed and direction, or from other sensors like phreatimeters can also be useful, at least for visualization by the users.

Historical datasets from weather sensors, possibly related to past extreme events, can be very helpful for the system training. Usually, they are .txt or .csv files with values of a physical parameter (e.g., rain intensity, humidity, tide/waves height, etc.) vs. time. In the case of Massa, they are publicly available on the website of the Hydrological Service of the Tuscany Region (Servizio Idrogeologico Regione Toscana, SIR) [12]. On the other hand, real-time or quasi real-time data streams from sensors must be directly integrated into the EWS subsystem, thanks to on-purpose developed APIs that represent the interface between the DT-EWS and the existing sensor network. The faster is the sensors read refresh time, the more accurate is the system monitoring.

It is worth noticing that the developed system will fundamentally relies on information coming from official sensors owned by local/regional agencies or companies. However, consistently with one of the key features of the SCORE projects, the system will also accept as an input data from sensors adopted in the WP4 citizen science activities [9]. Finally, it is also important underlining that, in the absence of datastreams from sensors, there is no use in running the EWS subsystem, since there is no point in attempting making projections if there is not a sufficient awareness of the current situation. In such a case, the system raises a warning to inform the user that the systems stops its operations because it cannot find any monitored parameter. On the other hand, the EWS subsystem is capable of checking if there are missing or unavailable sensors for a specific parameter that is expected to be measured (e.g., rain rate, river flow rate or level, etc.), and informs the users that some needed inputs are missing. This way, the user is always aware that the EWS cannot give a complete picture of the present situation, although it can run anyway.

Details about data usage and usability can be found in the most recent release of the data management plan (DMP) [13].

Referring to D8.2 [2], the possibility of taking as an input historical and real-time data from sensors, enable the system being compliant with the following functional requirements:

SCORE-REQ-DTEW-FUN-001-01: Early warnings for coastal flooding

SCORE-REQ-DTEW-FUN-003-01: Early warnings for storm surges

SCORE-REQ-DTEW-FUN-004-01: Farm safety

SCORE-REQ-DTEW-FUN-008-01: Past flood data and their effects

SCORE-REQ-DTEW-FUN-012-01: Spread of overflooding and flooded areas map

SCORE-REQ-DTEW-FUN-018-01: Temperature of the region

SCORE-REQ-DTEW-FUN-019-01: Distribution of precipitation regime

SCORE-REQ-DTEW-FUN-020-01: Humidity level

SCORE-REQ-DTEW-FUN-022-01: Real-time meteorological data

SCORE-REQ-DTEW-FUN-025-01: Provincial heat islands

SCORE-REQ-DTEW-FUN-027-01: Data monitoring at the local level

SCORE-REQ-DTEW-FUN-030-01: City risk assessment

SCORE-REQ-DTEW-FUN-034-01: Drought-prone regions

SCORE-REQ-DTEW-FUN-042-01: Risk Maps

Moreover, receiving data streams from sensors also allow the system implementing the following functional specifications [2]:

SCORE-FSPEC-DTEW-075: Thematic Layers – Past flood data and their effects

SCORE-FSPEC-DTEW-079: Thematic Layers – Spread of overflooding and maps overflooding

SCORE-FSPEC-DTEW-085: Thematic Layers – Distribution of precipitation regime

SCORE-FSPEC-DTEW-090: Thematic Layers – Drought-prone regions

SCORE-FSPEC-DTEW-110: Map Content

SCORE-FSPEC-DTEW-111: Sensors on map

SCORE-FSPEC-DTEW-112: Data from sensors

SCORE-FSPEC-DTEW-113: Plotting of sensors data

SCORE-FSPEC-DTEW-114: Modify sensors

SCORE-FSPEC-DTEW-115: Status at current time

SCORE-FSPEC-DTEW-119: Real-time meteorological data

SCORE-FSPEC-DTEW-159: Definition of map

SCORE-FSPEC-DTEW-452: Warning

SCORE-FSPEC-DTEW-469: Early warnings for Coastal Flooding

SCORE-FSPEC-DTEW-470: Early warnings for Storm Surges

SCORE-FSPEC-DTEW-471: Ground/Surface water level

SCORE-FSPEC-DTEW-472: Farm Safety

SCORE-FSPEC-DTEW-473: Humidity Level

SCORE-FSPEC-DTEW-475: Heat intensity

2.3 Exposure and hazard maps

Within the SCORE project, hazard maps are produced as a result of WP6 activities, which are fed by data and information produced by WP3. Moreover, WP6 also produced exposure and vulnerability datasets, reported in D6.3 [14], for the three WP8 frontrunner CCLLs (Massa, Vilanova i la Geltrú, and Oarsoaldea). These datasets are represented by maps that show the most important categories that can be impacted by a disastrous event, i.e., i) population, ii) buildings, and iii) roads and railway networks. The developed exposure models distinguish among buildings features and use, and the financial losses that can be caused by extreme events are thoroughly analysed.

The maps elaborated in WP6 and the related vulnerability functions are employed by the DT-EWS to provide the user with information about the urban areas that will exhibit the highest risk of human and/or economical damage, creating hazard maps that are defined considering the system simulation outputs. As it will be clearer in the following, in the case of the USE subsystem, the obtained hazard maps will be hypothetical and will be useful to understand the impact of urban planning on hazard mitigation. The EWS subsystem, on the other hand, generates warnings not simply considering that some city areas are flooded, but taking into account human losses and economical damages, also related to the water level.

Referring to D8.2 [2], the use of hazard maps and exposure and vulnerability data enables the system being compliant with the following functional requirements:

SCORE-REQ-DTEW-FUN-003-01: Early warnings for storm surges

SCORE-REQ-DTEW-FUN-004-01: Farm safety

SCORE-REQ-DTEW-FUN-007-01: Easily readable models

SCORE-REQ-DTEW-FUN-030-01: City risk assessment

SCORE-REQ-DTEW-FUN-042-01: Risk Maps

SCORE-REQ-DTEW-FUN-044-01: Land Use

SCORE-REQ-DTEW-FUN-045-01: Land Use Change

SCORE-REQ-DTEW-FUN-046-01: Simulation.

Moreover, the hazard maps and exposure and vulnerability data also allow the system implementing the following functional specifications [2]:

SCORE-FSPEC-DTEW-052: Types of map layers

SCORE-FSPEC-DTEW-070: Thematic layers

SCORE-FSPEC-DTEW-092: Thematic Layers – Land Use

SCORE-FSPEC-DTEW-110: Map Content

SCORE-FSPEC-DTEW-115: Status at current time

SCORE-FSPEC-DTEW-117: Risk maps

SCORE-FSPEC-DTEW-153: Entity types

SCORE-FSPEC-DTEW-157: Land Use Change

SCORE-FSPEC-DTEW-159: Definition of map

SCORE-FSPEC-DTEW-351: Simulation

SCORE-FSPEC-DTEW-354: Simulation Evaluation

SCORE-FSPEC-DTEW-452: Warning

SCORE-FSPEC-DTEW-469: Early warnings for Coastal Flooding

SCORE-FSPEC-DTEW-470: Early warnings for Storm Surges

SCORE-FSPEC-DTEW-472: Farm Safety.

2.4 Other data

In the deployment phase, the DT-EWS system framework will be adapted to each CCLL, at first by ingesting the above-listed minimum required data that must be made available by the city. Then, there will be room for negotiation for adding other data specific to each particular CCLL, i.e., other map layers that can be integrated in the system, that users wish at least to visualise, when it is not possible to employ them in simulations like, e.g., biodiversity conservation, wildlife conservation, past fires and their effects, and so on.

Referring to D8.2 [2], these additional data enable the system being compliant with the following functional requirements:

SCORE-REQ-DTEW-FUN-002-01: Biodiversity conservation

SCORE-REQ-DTEW-FUN-005-01: Wildlife conservation

SCORE-REQ-DTEW-FUN-009-01: Past fires and their effects

SCORE-REQ-DTEW-FUN-010-01: Previous earthquakes and their effects

SCORE-REQ-DTEW-FUN-011-01: Previous landslides and their effect

SCORE-REQ-DTEW-FUN-014-01: Maps of air polluted areas

SCORE-REQ-DTEW-FUN-015-01: Maps of contaminated water

SCORE-REQ-DTEW-FUN-016-01: Water resource variation (surface and ground)

SCORE-REQ-DTEW-FUN-017-01: Quality of soil and water

SCORE-REQ-DTEW-FUN-021-01: Land ownership

SCORE-REQ-DTEW-FUN-026-01: Daily water consumption by local citizens

SCORE-REQ-DTEW-FUN-029-01: Data coming from people on infrastructure conditions

SCORE-REQ-DTEW-FUN-033-01: Information about the transport and mobility

SCORE-REQ-DTEW-FUN-034-01: Drought-prone regions

SCORE-REQ-DTEW-FUN-035-01: Waste water management

Moreover, the following functional specifications can be implemented [2]:

SCORE-FSPEC-DTEW-051: Map layers selection capability

SCORE-FSPEC-DTEW-052: Types of map layers

SCORE-FSPEC-DTEW-070: Thematic layers

SCORE-FSPEC-DTEW-072: Thematic Layers – Biodiversity

SCORE-FSPEC-DTEW-073: Thematic Layers – Wildlife conservation

SCORE-FSPEC-DTEW-076: Thematic Layers – Past fires and their effects

SCORE-FSPEC-DTEW-077: Thematic Layers – Previous Earthquakes and their effects

SCORE-FSPEC-DTEW-078: Thematic Layers – Previous landslides and their effects

SCORE-FSPEC-DTEW-080: Thematic Layers – Maps of air-polluted areas

SCORE-FSPEC-DTEW-081: Thematic Layers – Maps of contaminated water

SCORE-FSPEC-DTEW-082: Thematic Layers – Water resource variation

SCORE-FSPEC-DTEW-083: Thematic Layers – Quality of soil and water

SCORE-FSPEC-DTEW-086: Thematic Layers – Land distribution and ownership

SCORE-FSPEC-DTEW-091: Thematic Layers – Waste water management

SCORE-FSPEC-DTEW-110: Map Content

SCORE-FSPEC-DTEW-159: Definition of map

2.5 User-defined scenarios and current scenarios

The DT-EWS system is intended to run simulations taking as inputs a collection of data coming from different sources, that we term as *scenarios*. Once all the data needed by the user and by the system for a simulation have been collected in a scenario, the simulation can run. We can distinguish between two different kind of scenarios that are employed in the DT-EWS system:

- *User-defined scenarios* are heterogeneous collections of data representing the input for a simulation of the USE subsystem, containing the city base-line data, as geographical maps, DEM/DTM/DSM, infrastructure or human hazard maps, that represent the "ground truth" on which any scenario is built; moreover, it includes other parameters related to hypothetical EBAs solutions and to the weather and hydrological events that can be either decided by the user, or be the product of real climate and sea-state model projections

- *Current-state scenarios*, on the other hand, are not created by users, but set up by the system itself. In addition to the city base-line data, they take into account the real weather and hydrological conditions by including *i)* real-time data coming from the official and eventual citizen-science [9] sensors available on the considered study area, and *ii)* eventually the weather forecasts and nowcasts. These scenarios are employed by the EWS subsystem, as it will be clearer in the following.

From this point on, for simplicity, in the text we will adopt the acronym "EBA" to indicate in general solutions inspired by ecosystem-based approaches. In the creation of a scenario, the user can add some EBAs that can be placed on the city map, by tracing one or multiple polygons in the urban area where the EBAs are intended to be implemented. Moreover, the user can include weather events, by considering either the long-term historical or future models, or by creating hypothetical events setting the related parameters (timeline, intensity, area of occurrence). The system is intended to take into account also the sea state whose parameters, as explained in the following, represent the boundary condition for the hydraulic model. To achieve this, in this release of the system, the user can choose among a set of prepared sea-state scenarios, related to different tide levels and wave height.

This way, the simulation can take into account the EBAs to show the impact of these measures on the hydraulic situation of the city in case of occurrence of the simulated weather events.

Referring to D8.2 [2], the definition and creation of scenarios enables the system being compliant in particular with the following functional requirements:

SCORE-REQ-DTEW-FUN-001-01: Early warnings for coastal flooding

SCORE-REQ-DTEW-FUN-003-01: Early warnings for storm surges

SCORE-REQ-DTEW-FUN-012-01: Spread of overflooding and flooded areas map

SCORE-REQ-DTEW-FUN-013-01: Ground / Surface water level

SCORE-REQ-DTEW-FUN-019-01: Distribution of precipitation regime

SCORE-REQ-DTEW-FUN-022-01: Real-time meteorological data

SCORE-REQ-DTEW-FUN-045-01: Land Use Change

SCORE-REQ-DTEW-FUN-046-01: Simulation.

Moreover, it allows implementing the following functional specifications [2]:

SCORE-FSPEC-DTEW-101: GIS based approach

SCORE-FSPEC-DTEW-103: Entity attributes update

SCORE-FSPEC-DTEW-115: Status at current time

SCORE-FSPEC-DTEW-154: Ecosystem-Based Adaptation

SCORE-FSPEC-DTEW-155: Edited maps

SCORE-FSPEC-DTEW-157: Land Use Change

SCORE-FSPEC-DTEW-158: Simulation scenario

SCORE-FSPEC-DTEW-159: Definition of map

SCORE-FSPEC-DTEW-160: Definition of events timeline

SCORE-FSPEC-DTEW-251: Simulation Events

SCORE-FSPEC-DTEW-252: Simulation Event Properties

SCORE-FSPEC-DTEW-301: Simulation timeline definition

SCORE-FSPEC-DTEW-351: Simulation

SCORE-FSPEC-DTEW-352: Simulation Execution

SCORE-FSPEC-DTEW-451: Short-Term Forecasting

SCORE-FSPEC-DTEW-452: Warning

SCORE-FSPEC-DTEW-469: Early warnings for Coastal Flooding

SCORE-FSPEC-DTEW-470: Early warnings for Storm Surges

SCORE-FSPEC-DTEW-471: Ground/Surface water level.

3.   SYSTEM ARCHITECTURE

In this section, we briefly describe the system architecture. Beyond exposing the system block scheme, to give a clearer idea of the system functionalities, we also outline the architecture in terms of process and interactions with the users.

As mentioned above, the SCORE GIS-based Early-Warning and Digital Twin Platform, that in this document in indicated by the acronym DT-EWS, is composed of two main subsystems: USE, and the EWS. They can run mostly independently, since they respond to different user's needs:

- The USE subsystem will allow the users to perform simulations regarding urban weather, what-if scenarios or long-term climate scenarios. By providing a quantification of the flooding hazard, it can support technical figures and policymakers in evaluating the impact of adaptation measures inspired by EBAs and other possible actions aimed at enhancing the resilience of coastal cities against extreme weather events and coastal erosion.

- The EWS subsystem processes can run continuously, periodically checking the coastal city situation, by monitoring the actual rivers streamflow, sea state and weather depending on available data, allowing an all-day monitoring of the flooding risk, and raising alerts when some flooding distribution exceeds a set threshold. It will be activated by authorized users, and fed by real-time data coming from all the sensors disseminated in the urban area, together with weather forecasts provided by local services.

3.1   The User Scenario Evaluation subsystem

3.1.1   Subsystem architecture

Different categories of users will have the possibility to use the DT-EWS system, as described in more detail in [2]. Only fully authorized users, provided with their own personal credentials, will be allowed to access the complete functionalities of the system, i.e., viewing sensible data and launching simulations.

As illustrated in Figure 3, users will interact with the DT-EWS system thanks to a Graphical User Interface (GUI) [2], [3], that will allow them to view the base-line data of the city represented as layers of interest, e.g., the city map, or its DEM/DTM/DSM, the map of the sensors in the urban area, and all the other layers and maps listed in Section 2. The majority of all these data are stored in the SIP or are part of data streams coming from active sensors.

When logged on to the DT-EWS, the users do not interact directly with the SIP; instead, thanks to a Data Service function, they can search files and data on the SIP, since all the files therein are associated to metatags for this purpose. Indeed, the representational state transfer (REST) APIs are set to let the users query data on the SIP, exploiting communications based on the XML-RPC protocol [16]. The APIs on the SIP can return a list of uniform resource locators (URLs), to documents (PDF, XML, JSON, GeoTiff, etc.), map layers, or Open Geospatial Consortium (OGC) services, like Web Map Service (WMS), Web Map Tile Service (WMTS), Web Feature Service (WFS), etc., depending on the user's request [2].

All the retrieved data are stored locally on the DT-EWS database (DB) and, once all the data needed by the user or by the system for the simulation are available and have been collected in a scenario, as it has been defined in Subsection 2.5, the simulation can be launched.
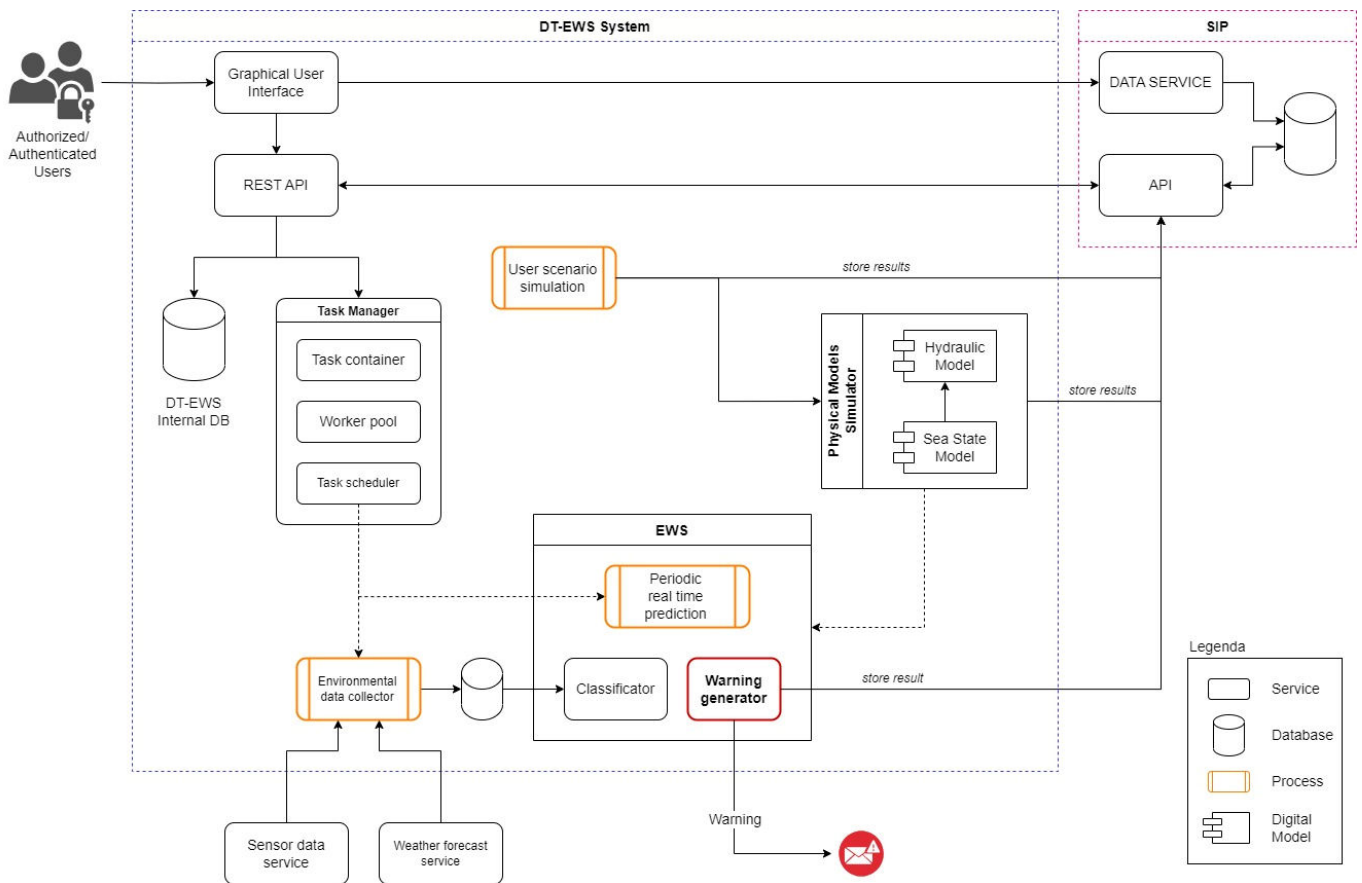
Figure 3. DT-EWS overall architecture. SIP: SCORE ICT platform; API: application programming interface; REST: representational state transfer; DB: database.

A user-defined scenario can be employed in the case of what-if or long-term analyses, performed to test the impact of the conditions of a given scenario on the urban area. In the first case, except the base-line data present on the SIP, these conditions can be completely hypothetical, e.g., intended to simulate extreme weather occurrences and their impact in case of simulated EBAs. In the second case, the simulations focus on the climate and sea state conditions evolution over a long-time horizon, considering a mixture of real data, like city base-line maps and the real climate models mentioned in Subsection 2.2.1, and other elements decided by the user, such as the hypothetical implementation of EBAs. Finally, an *event* is an output of the simulation, also represented by a collection of different data and pieces of information, as better detailed in the following.

In Figure 3, the system task manager is in charge of managing the simulation tasks, that are queued by the task container service and passed on to the worker pool. The latter block performs the tasks following an order set by the task scheduler. The simulation process is performed by software tools that we refer to as "digital models", i.e., a numerical representation of physical models related to the coastal sea state and to the studied area hydrology, by numerically solving the mathematical equations dictated by the underlying physics. Some of the information and data contained in the user-defined scenario represent the boundary conditions for these models, for a consistent solution of the equations. In particular, the scenario takes at first into account the data set by the users about sea state, like tide level, waves height and period, etc., or from downscaled models describing the evolution of sea state over time, that have been processed by a sea-state model software. The output of this processing, together with the remaining data provided in the user-defined scenario, e.g., represented by data about rain rate, wind speed and direction, river level and flow rate, and so on, represent the boundary conditions for the hydrological model simulator. The latter generates maps of flooding events, i.e., raster maps with the same spatial resolution of the employed DEM, containing the water level in each resolution cell. Finally, the system can relate the obtained flooding

map with the human and financial exposure maps of the same coastal city present on the SIP and the associated tables, to precisely evaluate the number of people impacted by the events and to calculate the effective damage in €, considering the water depth [14]. At the end of the simulation, all these maps and data, along with the created scenario are stored together on the SIP, so that they can be viewed and further analysed by users.

Inside the system depicted in Figure 3, the EWS subsystem does not perform what-if and long-term analyses, but it enables continuous monitoring of the city hydrological situation, since it employs current-state scenarios, with data coming from available sensors, like the ones mentioned in Subsection 2.2.3, and all the available sources of relevant data. Moreover, a process of environmental data collection also gathers the weather forecasts for a few following hours, i.e., the nowcasting. All these data and pieces of information are employed to give support in the knowledge of the possible impact and risks associated with the oncoming weather phenomena on the urban area as considered in its actual conditions, thanks to the exposure maps.

A fundamental difference between the EWS module and the rest of the system lies in the fact that this subsystem leverages on Artificial Intelligence (AI) and Machine Learning (ML) algorithms for classification of the current scenario and to decide if the flooding hazard is so high that a warning must be raised. The EWS operates on past scenarios and events simulated by the physical model and operates a classification of the current state scenario based on their similarities. In other words, it associates flooding events to the current observed situation based on a spectrum of previously computed scenarios guaranteeing fast and frequent monitoring of the city status, as better detailed in the following section.

Referring to D8.2 [2], the described architecture has been conceived to be compliant with all the functional and development requirements and functional specifications mentioned in the previous subsections. In addition, it makes the system also compliant with the security requirements:

SCORE-REQ-DTEW-SEC-039-01: User Authentication

SCORE-REQ-DTEW-SEC-040-01: User Authentication – Credentials

SCORE-REQ-DTEW-SEC-041-01: Categories of users,

and with the associated functional specifications:

SCORE-FSPEC-DTEW-003: User authentication

SCORE-FSPEC-DTEW-004: Categories of users

SCORE-FSPEC-DTEW-005: User management

SCORE-FSPEC-DTEW-020: User capabilities – Viewer

SCORE-FSPEC-DTEW-030: User capabilities – Standard user

SCORE-FSPEC-DTEW-040: User capabilities – Power User

SCORE-FSPEC-DTEW-045: User capabilities – Administrator.

3.1.2   USE subsystem services

Here are described with some more detail the services of the USE subsystem. As depicted in Figure 4, which is a detail of Figure 3, the USE subsystem consists of the following services:

1. The REST APIs, which are implemented using the Python Django high-level web framework, together with the Django Ninja API builder. The service is based on the open standards for APIs OpenAPI and JSON Schema.
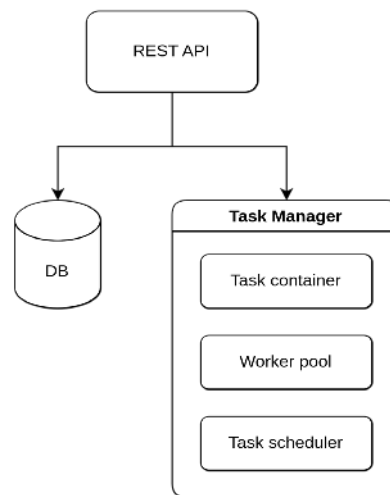
Figure 4. Structure of the USE subsystem services. REST API: Representational state transfer application programming interface; DB: Database.

2. The database, built using the PostgreSQL object-relational database system. In this framework, the API allows a CCLL user to create scenarios of possible EBAs and weather conditions (sea states, river flows, precipitation events) which can be simulated, obtaining at the output URLs for the calculated results, i.e., to objects like map layers, reports, etc. The entity relationship diagram reported in Figure 5 summarizes the conceptual schema of the database content:
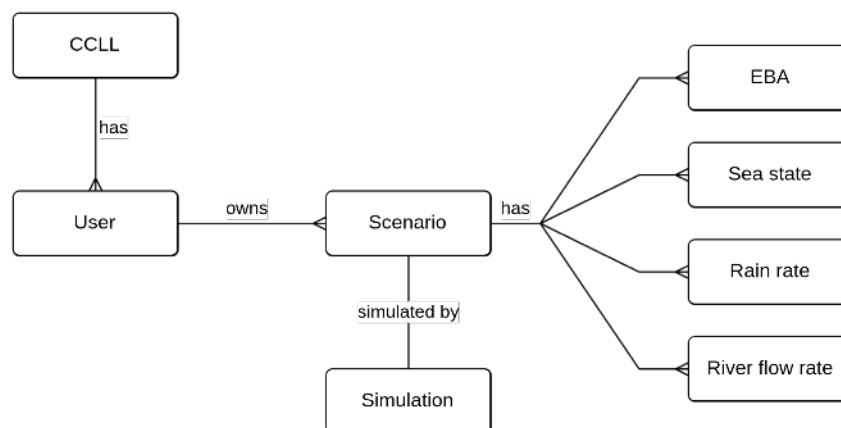
Figure 5. Conceptual scheme of the DB.

3. The task Manager, based on the Python Celery distributed system, in which tasks are treated as messages sent to application queues managed by the RabbitMQ message broker. Whenever a scenario needs to be simulated, a task is queued and assigned to an available worker that will run the simulation and collect the results in reports, images, etc.

3.1.3 The adopted hydraulic model: LISFLOOD-FP

LISFLOOD-FP [17] is the hydraulic digital model employed by the SCORE DT-EWS system to perform hydraulic simulations. It is an open-source raster-based two-dimensional (2D) hydrodynamic model, designed to simulate floodplain inundation in a computationally efficient manner over complex topography. It is capable of simulating grids with high spatial resolution for dynamic flood events, allowing the integration with maps obtained thanks to new sources of terrain information from remote sensing techniques, such as airborne laser altimetry and satellite interferometric radar. LISFLOOD-FP can accurately take into account the behaviour of water flow in rivers, channels,

and floodplains by solving the Saint-Venant equations, allowing to include crucial factors like topography, boundary conditions, and hydraulic processes [18]. Therefore, to run simulations, LISFLOOD-FP requires input data such as DEMs or DSMs, river channel networks, land cover information, and river flow and sea-state boundary conditions. Parameterization is a critical step in configuring the model, involving the setting of values for parameters that control flow resistance, channel roughness, and infiltration rates. Moreover, precipitations can be included in simulations as *.NetCDF* files, in which the intensity is specified on a timeline, for each cell of the DTM/DEM/DSM that is employed. Calibration of these parameters is commonly performed by comparing model outputs with observed data to ensure accuracy. In case the considered modelled area includes a large water body, like the sea, LISFLOOD-FP needs to set boundary conditions on it. Indeed, the sea state in particular, is crucial not only for coastal flooding due to storm surges, but also for analysing river discharge dynamics which, at the end, play a key role in flooding events. The related boundary conditions must be set by the user. In the present version of the system, the user can choose a particular scenario among a set of possible sea states, that are tailored to each considered coastal city.

The outputs of LISFLOOD-FP provide insights into flood patterns and impacts, including flood extent maps, water depth distributions, flow velocities, aiding in visualizing flood dynamics and identifying flood-prone areas. In this way, it enables risk assessment, real-time flood forecasting, and emergency response planning. An example of the output of a hydraulic simulation carried out with LISFLOOD-FP is depicted in Figure 6, where a flood event on a part of the urban area of Massa is represented. In the USE subsystem, LISFLOOD-FP represents the digital hydraulic model at the core of the simulation engine; on the other hand, its specific role in the EWS subsystem is different and will clarified in the following.
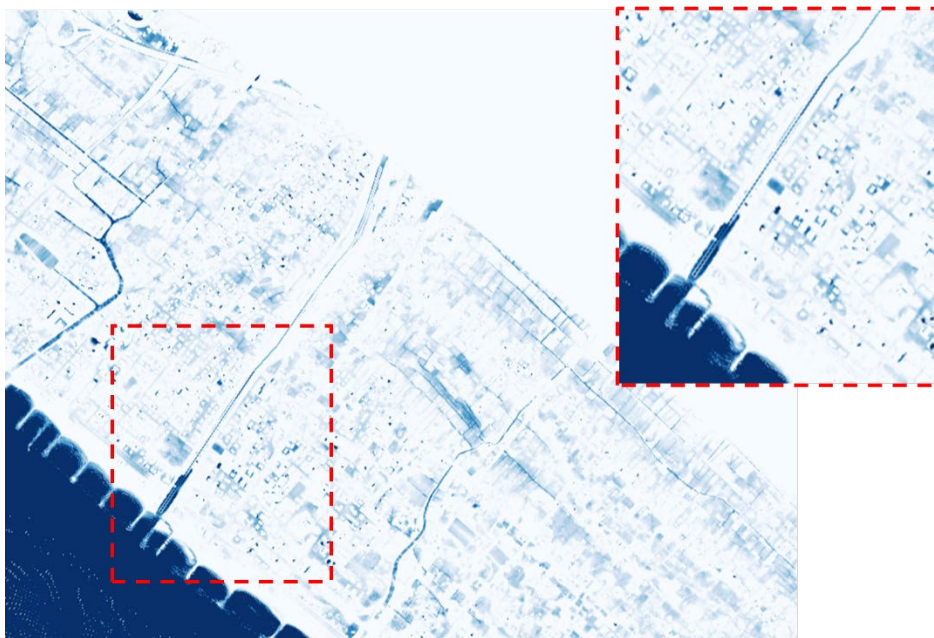
Figure 6. Example of the output of a simulation with Lisflood-FP. The obtained flooding map is related to a part of the study area of the coastal city of Massa. Here, only the presence of water is reported, in blue. In the top-right inset, a zoom on the area near the mouth of the Frigido river, where the shapes of the buildings are clearly distinguishable due to the presence of water.

LISFLOOD-FP has been selected among other similar available off-the-shelf products (like, e.g., HEC-RAS) since it is a non-commercial, research tool. Being an open-source model, LISFLOOD-FP encourages collaboration, customization, and ongoing development. Researchers and practitioners can freely access its source code, enabling them to adapt and improve it according to specific project needs and local conditions. This open approach fosters innovation and supports the advancement of flood risk management practices.

Referring to D8.2 [2], the adoption of LISFLOOD-FP makes the system compliant with the following functional requirements:

SCORE-REQ-DTEW-FUN-001-01: Early warnings for coastal flooding

SCORE-REQ-DTEW-FUN-003-01: Early warnings for storm surges

SCORE-REQ-DTEW-FUN-007-01: Easily readable models

SCORE-REQ-DTEW-FUN-008-01: Past flood data and their effects

SCORE-REQ-DTEW-FUN-012-01: Spread of overflooding and flooded areas map

SCORE-REQ-DTEW-FUN-013-01: Ground / Surface water level

SCORE-REQ-DTEW-FUN-030-01: City risk assessment

SCORE-REQ-DTEW-FUN-031-01: Sea level rise

SCORE-REQ-DTEW-FUN-032-01: Coastal erosion

SCORE-REQ-DTEW-FUN-046-01: Simulation.

Moreover, LISFLOOD-FP allows implementing the following functional specifications [2]:

SCORE-FSPEC-DTEW-051: Map layers selection capability

SCORE-FSPEC-DTEW-052: Types of map layers

SCORE-FSPEC-DTEW-053: Map layers properties

SCORE-FSPEC-DTEW-070: Thematic layers

SCORE-FSPEC-DTEW-071: Thematic Layers – Coastal flooding

SCORE-FSPEC-DTEW-075: Thematic Layers – Past flood data and their effects

SCORE-FSPEC-DTEW-079: Thematic Layers – Spread of overflooding and maps overflooding

SCORE-FSPEC-DTEW-088: Thematic Layers – Coastal Erosion

SCORE-FSPEC-DTEW-101: GIS based approach

SCORE-FSPEC-DTEW-110: Map Content

SCORE-FSPEC-DTEW-116: Time snapshots

SCORE-FSPEC-DTEW-154: Ecosystem-Based Adaptation

SCORE-FSPEC-DTEW-155: Edited maps

SCORE-FSPEC-DTEW-159: Definition of map

SCORE-FSPEC-DTEW-251: Simulation Events

SCORE-FSPEC-DTEW-252: Simulation Event Properties

SCORE-FSPEC-DTEW-301: Simulation timeline definition

SCORE-FSPEC-DTEW-351: Simulation

SCORE-FSPEC-DTEW-352: Simulation Execution

SCORE-FSPEC-DTEW-353: Simulation Duration

SCORE-FSPEC-DTEW-354: Simulation Evaluation

SCORE-FSPEC-DTEW-401: Simulation Results presentation

SCORE-FSPEC-DTEW-404: Sea level rise

SCORE-FSPEC-DTEW-471: Ground/Surface water level.

3.2 The Early-Warning Support Subsystem

In this chapter, the EWS sub-system is described. In the first two sections, the functional architecture and an overview of the series of processes involved in the software is described at a high level. The following sections are, instead, devoted to a more detailed and technical description of the single parts, starting from the configuration of the EWS and then describing the chain of processes implemented in the system, depicting the role and the use of the parameters on which the EWS subsystem operations are based.
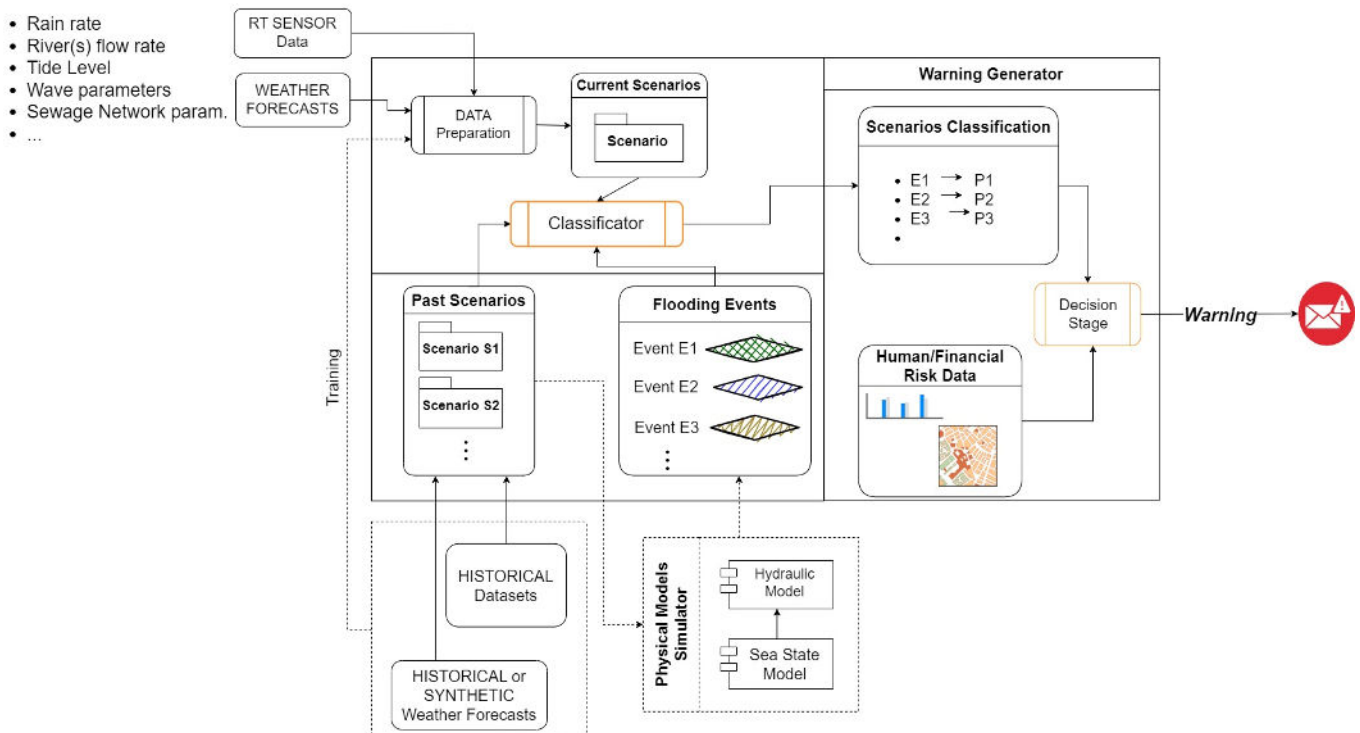
Figure 7. A schematic representation of the early-warning support subsystem working principle.

3.2.1 Functional architecture

The EWS subsystem is in charge of a continuous monitoring of the current situation of a coastal city area of interest, making projections on its hydraulic state, and generating alarms in case of elevated risk of flooding or inundation. Figure 7 reports a schema illustrating its working principle at a higher level. The subsystem involves the use of AI/ML algorithms and relies on the use of data from available sensors, as listed in subsection 2.2.3, by forecast data and by a library of historical and synthetic scenarios.

In the library, each given scenario S_k, is created from a set of historical or synthetic series, related to the relevant element of the city (rain rate, sea level, rivers flows...). It is fed at the input of the simulations of the physical models, in this case represented by LISFLOOD-FP, in the same way as it is done in the USE subsystem.

To each scenario S_k, the simulator will output an event E_k, associated to a series of maps generated on the city area DEM/DTM/DSM, representing the evolution of the flooding and where a value for the water depth level is associated to each cell of the maps, that will be greater than zero in case of flooding.
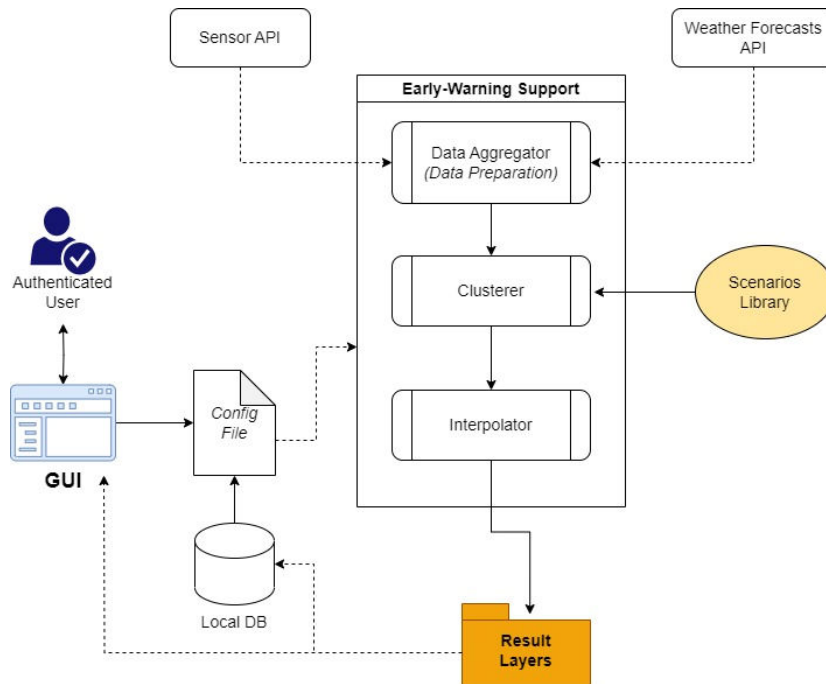
Figure 8. EWS block scheme. API: Application programming interface; GUI: Graphic user interface; DB: Database.

On the other hand, the subsystem also takes as input real-time data from sensors, together with real weather forecasts and, after a data preparation process, the current-state scenario is constructed. This process is devoted to pre-process and aggregate input data by using ML techniques such as feature selection strategies or dimensionality reduction algorithms, possibly using also measured or validated historical data, to handle and manage correctly the input data based on their reliability or informative content [19], [20]. Afterwards, this real-time scenario, along with the obtained set of pre-computed scenarios from the library and the set of associated events, are the input for the Classificator, as depicted in Figure 7. This block represents the AI/ML-based process extracting the similarity between the observed scenario and the past scenarios, thus establishing an occurrence probability P_k associated to each possible k-th event. The set of events and probabilities is then ingested to the Warning Generator stage, where a decision process is launched considering the probability of flooding events and the correlated risks from a human and financial point of view, retrieving the exposure and hazard maps described in subsection 2.3. The final output of this process is an alert that will be notified to a pool of relevant users, together with the expected flooding layer and projected risks and damages whose computation is based on the WP6 work described in D6.3 [14]. These results will be stored in a warning repository.

As it will be clearer in the next sections, a peculiarity of the EWS subsystem consists in running simulations avoiding the use of LISFLOOD-FP, the digital hydraulic model, thanks to AI/ML algorithms. This implies very fast operations and a hugely reduced need for computational resources, compared to the USE subsystem. For this reason, in this stage of the development, where few realizations of the EWS are required, there is no need for a particular software architecture to guarantee parallel runs of the EWS on the same machine. However, in the perspective of a better scalability of the system, future EWS versions may include, as in the USE subsystem, the Celery Python library for task management, with RabbitMQ as message-broker software, that will allow transparently parallel runs on different machines.

3.2.2   Subsystem run overview

Each single run of the EWS involves a cascade of processes that is depicted in Figure 8. During a run, in each stage, the software provides all the necessary information about the relevant inner operations or problems by logging at screen. Different levels of logging are implemented depending on the use by a user or for validation-test purposes. The logs are also exported as a file to allow user and developer to keep track of the past runs. An example of internal logs is in Figure 9.

```
2023-07-28 10:16:30,813 EWS          INFO     Running EWS
2023-07-28 10:16:30,813 EWS          INFO     Setting now time to 2023-07-28 08:16:31+00:00
2023-07-28 10:16:30,814 EWS          INFO     Reading configuration
2023-07-28 10:16:30,815 EWS          INFO     Setting log file ../data/massa/warnings/2023-07-28 08:15:00+00:00_log
2023-07-28 10:16:30,817 EWS          INFO     Validating configuration
2023-07-28 10:16:30,836 EWS          INFO     Setting up sensor datastreams
2023-07-28 10:16:30,837 EWS          DEBUG    Activated local test mode
2023-07-28 10:16:30,837 EWS          INFO     Reading scenarios from library
2023-07-28 10:16:30,881 EWS          DEBUG    Loaded scenario scenario_1
2023-07-28 10:16:30,919 EWS          DEBUG    Loaded scenario scenario_2
2023-07-28 10:16:30,919 EWS          DEBUG    Processing element sea
2023-07-28 10:16:30,919 EWS          INFO     Processing datastream sea@test_sensor_sea_1
2023-07-28 10:16:31,226 EWS          WARNING  Found missing values
2023-07-28 10:16:31,226 EWS          WARNING  Skipping for missing values over threshold
2023-07-28 10:16:31,226 EWS          INFO     Processing datastream sea@test_sensor_sea_2
2023-07-28 10:16:31,633 EWS          DEBUG    Processing element rain
2023-07-28 10:16:31,634 EWS          INFO     Processing datastream rain@test_sensor_rain_1
2023-07-28 10:16:31,967 EWS          DEBUG    Processing element river_1
2023-07-28 10:16:31,967 EWS          INFO     Processing datastream river_1@test_sensor_river_1_1
2023-07-28 10:16:32,275 EWS          WARNING  Found negative values
2023-07-28 10:16:32,278 EWS          WARNING  Found 0.42% of possible outliers
2023-07-28 10:16:32,279 EWS          INFO     Processing datastream river_1@test_sensor_river_1_2
2023-07-28 10:16:32,606 EWS          INFO     Creating current scenario
2023-07-28 10:16:32,711 EWS          DEBUG    Predicting layer at 15min
2023-07-28 10:16:32,712 EWS          INFO     Selecting result layer
2023-07-28 10:16:32,713 EWS          DEBUG    Selecting mode scenario scenario_1 with probability 0.93
2023-07-28 10:16:33,175 EWS          INFO     Exporting result layer over threshold
2023-07-28 10:16:33,175 EWS          INFO     Exporting result layer
2023-07-28 10:16:33,631 EWS          INFO     Computing risk
2023-07-28 10:16:33,631 EWS          DEBUG    Processing raster to polygons
2023-07-28 10:17:00,625 EWS          INFO     Working on exposure roads
2023-07-28 10:17:03,751 EWS          DEBUG    Saved risk map on ../data/massa/warnings/2023-07-28 08:15:00+00:00_roads.geojson
2023-07-28 10:17:03,751 EWS          INFO     Working on exposure population
2023-07-28 10:17:14,679 EWS          DEBUG    Saved risk map on ../data/massa/warnings/2023-07-28 08:15:00+00:00_population.geojson
2023-07-28 10:17:14,679 EWS          INFO     Working on exposure buildings_potential
2023-07-28 10:17:32,569 EWS          DEBUG    Saved risk map on ../data/massa/warnings/2023-07-28 08:15:00+00:00_buildings_potential.geojson
```

Figure 9. Example of internal logs of the EWS in a test case for the CCLL of Massa, with a few synthethic scenarios and test sensors

The system is initialized by a set of configurable parameters, to ensure the system flexibility and adaptability to the users' and CCLLs' needs. These parameters can be contained in a configuration file present on the DT-EWS local DB or on the SIP. After login, the configuration can be also modified by the user, who can change the initial parameters, related to the possible inputs, such as:

- CCLL baseline data

- Elements defining the scenario, i.e., entities like rain intensity, sea state, rivers flow rate, etc.

- Information about the considered sensors and forecasts

- The set of past or synthetic scenarios

- Optional technical parameters related to the EWS subprocesses,

where the latter are intended to be used especially during the validation or test purposes. During each run, the system takes care of properly parsing the configuration, alerting the user about eventual anomalies and/or errors about the inputs.

The Data Aggregator block is fed with data streams coming from the sensors and from the weather forecasts and, for each physical element, a list of data sources from which data will be read through the APIs is provided. Therefore, the Data Aggregator takes care of processing the data and of constructing a set of time series defining the current state of the city system. Clearly, this process strongly depends on the actual availability of the data at run time. Therefore, a set of integrity checks is performed on the acquired data and the software takes care of logging possible relevant notices for the users, e.g., missing data, presence of possible outliers or any information that could be relevant about the sensors and the forecasts feeds.

Once defined, the current scenario is passed through the classification process. As depicted in Figure 8Figure 7, the procedure is pursued by a Clusterer block, which is deputed to compare and evaluate similarities of the actual state with the scenarios in the library. More precisely, the scenarios are modelled as a statistical mixture allowing to formally define and associate an occurrence probability to each scenario. It is important noticing that the possible actual occurrence frequency of past historical scenarios – or, equivalently, an *a priori* occurrence associated to synthetic scenarios – must not be considered in this process, to avoid naively associating small probabilities to rare extreme events. In other words, if current information from sensors and weather forecasts outline a scenario that can be correlated to a past (or synthetic) disastrous event which, usually, is related to a long return time (i.e., a small occurrence probability), the system must not make the mistake of attributing to the emerging, current scenario a low probability due to the fact that it is similar to a rare event. Indeed, the EWS is not intended to predict how likely an extreme event is, but to promptly recognize it from the observed current situation.

Finally, the Interpolator block in Figure 8 is deputed to the management of the events corresponding to the different scenarios, each one now associated to its own occurrence probability. Each of these scenarios is equipped with a time series of output layers, computed using the hydraulic physical model LISFLOOD-FP, i.e., maps in the form of raster files of the CCLL indicating the flooding level in each cell of the map. The selected layer, which therefore represents the immediate future forecast of the flooding distribution over the map, can be also filtered by a specified mask allowing to restrict the relevant city areas to be monitored. In addition, an alerts generation control is guaranteed by the possibility of setting up warning thresholds for both the probability of occurrence and the water depth level. Finally, the selected layer is used to compute the expected damage on the exposure elements of the city, i.e., population, roads and railways, or buildings in compliance with the procedures and the vulnerability functions described in [14]. The expected calculated damages are quantified in number of affected people and in € for infrastructures and buildings.

The obtained output raster with flooding water depth (in GeoTIFF format), together with the damage layers corresponding to the different exposures (in GeoJSON format), are exported in the warning repository and therefore are ready to be visualized from the GUI. In addition, each run of the EWS is equipped with a copy of the configuration file for validation and replication purposes and with a log file containing all the information and possible warnings and errors generated during the run. All outputs are also uniquely associated by their generation runtime to keep the historical set of records.

It is worth underlining that, compared to the USE subsystem, the interactions of the users with the EWS are limited. Indeed, the simulations must not be launched each time by an operator, but they run continuously for monitoring purposes. Moreover, the considered scenarios are synthetic or past ones, employed by the EWS subsystem to create its library, or the current one agreeing to the sensors outputs and weather nowcasts, with no room for the user to alter real parameters. However, the user can visualize :

- past and present results

- map of the available official and/or citizen science sensors and their status

- plots of sensors data streams vs. time.

Referring to D8.2 [2], the described architecture of the EWS subsystem is compliant with the following functional and development requirements:

SCORE-REQ-DTEW-FUN-001-01: Early warnings for coastal flooding

SCORE-REQ-DTEW-FUN-003-01: Early warnings for storm surges

SCORE-REQ-DTEW-FUN-004-01: Farm safety

SCORE-REQ-DTEW-FUN-007-01: Easily readable models

SCORE-REQ-DTEW-FUN-012-01: Spread of overflooding and flooded areas map

SCORE-REQ-DTEW-FUN-013-01: Ground / Surface water level

SCORE-REQ-DTEW-FUN-019-01: Distribution of precipitation regime

SCORE-REQ-DTEW-FUN-027-01: Data monitoring at the local level

SCORE-REQ-DTEW-FUN-028-01: Emergency tools

SCORE-REQ-DTEW-FUN-030-01: City risk assessment

SCORE-REQ-DTEW-FUN-031-01: Sea level rise

SCORE-REQ-DTEW-FUN-042-01: Risk Maps

SCORE-REQ-DTEW-FUN-046-01: Simulation

SCORE-REQ-DTEW-FUN-047-01: Localisation

SCORE-REQ-DTEW-SEC-037-01: Ways to access the system

SCORE-REQ-DTEW-SEC-039-01: User Authentication

SCORE-REQ-DTEW-SEC-040-01: User Authentication – Credentials

SCORE-REQ-DTEW-SEC-041-01: Categories of users.

Moreover, the here-illustrated subsystem is also compliant with the following functional specifications [2]:

SCORE-FSPEC-DTEW-001: Ways to access the system

SCORE-FSPEC-DTEW-003: User authentication

SCORE-FSPEC-DTEW-004: Categories of users

SCORE-FSPEC-DTEW-006: Localization

SCORE-FSPEC-DTEW-030: User capabilities – Standard user

SCORE-FSPEC-DTEW-040: User capabilities – Power User

SCORE-FSPEC-DTEW-045: User capabilities – Administrator

SCORE-FSPEC-DTEW-053: Map layers properties

SCORE-FSPEC-DTEW-054: Base maps attribute data

SCORE-FSPEC-DTEW-055: Baseline maps attribute data

SCORE-FSPEC-DTEW-056: Simulation maps attribute data

SCORE-FSPEC-DTEW-070: Thematic layers

SCORE-FSPEC-DTEW-071: Thematic Layers – Coastal flooding

SCORE-FSPEC-DTEW-079: Thematic Layers – Spread of overflooding and maps overflooding

SCORE-FSPEC-DTEW-085: Thematic Layers – Distribution of precipitation regime

SCORE-FSPEC-DTEW-101: GIS based approach

SCORE-FSPEC-DTEW-110: Map Content

SCORE-FSPEC-DTEW-111: Sensors on map

SCORE-FSPEC-DTEW-112: Data from sensors

SCORE-FSPEC-DTEW-113: Plotting of sensors data

SCORE-FSPEC-DTEW-114: Modify sensors

SCORE-FSPEC-DTEW-115: Status at current time

SCORE-FSPEC-DTEW-116: Time snapshots

SCORE-FSPEC-DTEW-117: Risk maps

SCORE-FSPEC-DTEW-119: Real-time meteorological data

SCORE-FSPEC-DTEW-251: Simulation Events

SCORE-FSPEC-DTEW-354: Simulation Evaluation

SCORE-FSPEC-DTEW-401: Simulation Results presentation

SCORE-FSPEC-DTEW-402: Time navigation of simulation results

SCORE-FSPEC-DTEW-451: Short-Term Forecasting

SCORE-FSPEC-DTEW-452: Warning

SCORE-FSPEC-DTEW-460: Warning Accessibility

SCORE-FSPEC-DTEW-469: Early warnings for Coastal Flooding

SCORE-FSPEC-DTEW-470: Early warnings for Storm Surges

SCORE-FSPEC-DTEW-471: Ground/Surface water level

SCORE-FSPEC-DTEW-472: Farm Safety.

3.2.3   Configuration and directory structure

As a first step of the run, the software proceeds in reading and validating a configuration file containing all the necessary parameters needed by the module to perform a run. Part of these parameters are mandatory and are related to the general setup of the specific CCLL. Instead, other parameters are optional, i.e., the EWS subsystem will use default values if these are not specified: these are mostly technical or specific parameters that are intended to be used mostly or almost exclusively for validation purposes or corrections. During the

validation process, the program will check the presence and the correctness of all the inserted parameters. In addition, it will also perform some controls on the directories and user files structure. In case of wrong, misspelled or incompatible values, the EWS will stop the run and print a list of all the errors with a brief description, to help the user understand and identify the eventual mistakes. Conversely, if the configuration file is valid, the program will log the success of the validation step and proceed with the run.

Key	Description
name	Name or identifier of a single CCLL
elements	List of city elements identifiers used to construct the current scenario
exposures	List of exposures on which damages are calculated
workdir	Working directory name
datastream_sensors	URL for sensors datastreams
datastream_forecasts	URL for forecasts datastreams (optional)

Table 1. CCLL Configuration parameters

The description of all the configuration parameters is reported in Table 1 and Table 2. In particular, the former lists the set of mandatory user parameters needed to run, while the latter describes the set of optional technical parameters.

Key	Description
forecast_length	Forecast layer target interval (in minutes) from run time. Default: 60 minutes
scenario_length	Time length (in minutes) of the current scenario to be created. Default: 180 minutes
scenario_forecast_length	Time length (in minutes) of the part of the scenario time length using data from forecasts datastreams (if provided). Default: 90 minutes
scenario_validation_length	Time length (in minutes) of the validation sample used to process data from sensors. Default: 10080 minutes
missing_interpolation_mode	Mode used for time series data inputation. Possible values are "linear" or "median". Default: "linear".
missing_threshold	Maximum allowed percentage of missing values of a single datastream in the current scenario interval to be valid and usable. Default: .5
outlier_detection_mode	Mode used for time series outlier detection. Possible values are 'rolling_mad' or 'isolation_forest'. Default: 'rolling_mad'
outlier_detection_halflife	Exponential smoothing half-life (in minutes) used in the time series outlier detection process. Default: 1440 minutes
outlier_detection_threshold	Modified z-score threshold for outlier detection in 'rolling_mad' mode (unused in other modes). Default: 2
outlier_detection_contamination	Contamination percentage level for outlier detection in 'isolation_forest' mode, unused in other modes. If empty, uses automatic mode as described in [21]. Default: None

feature_selection_mode	Mode used for feature selection/aggregation. Possible values are 'PFA', 'mean' and 'mode'. Default: 'PFA'
feature_selection_pfa_quantile	Minimum amount of variance explained in the Principal Component Analysis step [20] of the PFA [22]. Unused in other modes. Default: 0.9
feature_selection_pfa_n_features	Number of principal features selected by the PFA and then aggregated. Unused in other modes. Default: 2
scenario_metric_sensitivity	Sensitivity or rounded digit for finding best alignment between scenarios. Default: 2
scenario_mixture_cluster_number	Number of multivariate gaussian samples used to define the scenario mixture. Default: 5000
scenario_mixture_cluster_scale	Scale or relative standard deviation of the multivariate gaussian. Default: 1.8
warning_mode	Mode for warning layer selection. Possible values are 'mode' or 'worst'. Default: 'mode'
warning_layer_mask	Raster dataset defining the sub-region(s) of the full layer to be used for the warning processes. Default: the system uses the whole map
warning_layer_max_dist	Maximum time distance (in minutes) to match the forecast layer with the flooding layers of the predefined scenarios. If greater, the next is selected. Default: 10
warning_layer_probability_threshold	Minimum probability for selecting the flooding layer in the 'worst' mode (unused in other cases). Default: 0.1
warning_layer_flooding_threshold	Minimum average flooding level (in meters) to start the export process. If 'warning_layer_mask' is given, the value is computed only on the map sub-regions. Default: 0.1
exposures_bins	List of floats defining a binning for the damage of the exposure categories. Each list must refer to a key corresponding to an exposure defined in the parameter 'exposures' and must contain the ordered right-endpoints of each bin. Default: no bins
random_state	Random seed used shared by the internal algorithms. Default: 42
test_mode	Boolean value to activate test mode. Default: False

Table 2. Technical optional configuration parameters.

The software retrieves all necessary inputs in a common data repository where a specific working directory is defined for each CCLL. In particular, the EWS fetches the needed files from several folders, each one related to a different object. Details about this structure will be described along with the processes characterization in the following sections.

3.2.4   Datastream processing

During this step, the EWS subsystem collects and processes the data from the datastreams from the sensors and, if provided, of the forecasts. For each CCLL element specified in the configuration, the software asks through purposely prepared APIs the corresponding datastreams list, i.e., those containing data related to the element itself. Hence, each datastream is read and passed through a chain of data-preparation processes:

1. *Data normalization and check*. Time series are regularized and made compliant with a common system format, i.e., pre-processed to be compliant with a system time parameters and units. Missing or negative values in the time-series are detected, the latter replaced by empty entries. In both cases the information is logged to make the user aware of possible problems with the datastream. If the percentage of missing data in the current scenario time interval turns out to be higher than the specified threshold (see the configuration parameter missing_threshold in Table 2): the datastream is skipped.

2. *Outliers detection*. On the full validation data sample, an outlier detection procedure aimed at detecting possible wrong or suspect values is applied. These kinds of techniques represent a common step in the pre-processing of data for analysis or model-building purposes. More in-depth information and overviews about this topic can be found in [23] - [25], whereas more specific discussions about the applications of these methods on time series can be found in [26], [27]. In our case, the EWS implements two different algorithms which can be selected in the configuration of the run. The first one is based on the so-called Median Absolute Deviation (MAD), a robust statistic method allowing to detect possible outliers based on the value of a modified z-score [28], [29]. The other, instead, implements an Isolation Forest [21], [30], a well-known ML algorithm that aims to identify outliers relying on their isolation in the feature space without using any distance or density measure. In both cases, the algorithms are applied on top of an exponential weighted moving average (EWMA), in order to keep into account possible trends or regular components of the time series. At the end of the procedure, if some points are identified as possible outliers, they are removed. Similarly as in the previous step, the information is logged to let users be aware of possible problems with the sensor data.

3. *Data imputation*. Once cleaned up, if the time series are valid, i.e., the number of missing elements does not exceed the fixed threshold in the scenario time interval, empty entries are imputed in order to get a complete time series. Two possible modes are implemented to achieve this task: interpolation or median imputation, both selectable in the configuration. In the first case, the time series values are linearly interpolated, keeping into account the actual frequency, whereas in the second case the median of the values in the series interval is used.

At the end of these steps, collected data from different datastreams belonging to a specific element are processed together, in order to obtain a single time series associated to it in the current scenario. Clearly, at this stage, it is necessary to have previously collected at least one valid time series from a datastream for the specific element. In case this is not possible, the software will notify on the log that the element itself cannot be used for the creation of the current scenario. This feature aggregation/selection process implements a Principal Feature Analysis (PFA) [22]. More in details, this procedure stacks the well-known Principal Component Analysis (PCA) and a hard clustering of the components, a *k*-Means in the original paper where details about these methods can be found by the interested reader [20]. Essentially, the dimensionality reduction brought up by the PCA, which keeps the most relevant components in the sense of how much they explain the observed variance of the data, is followed by a clusterization step that is aimed at selecting the features that represent mostly all the data. In our case, a sample is represented by time series in the time-domain, and the process allows obtaining a single series that captures the relevant behaviour associated to the element, keeping into account information from all the sensors, at the same time being robust to single, local behaviours. Together with the PCA, also a median and mean aggregation procedure are implemented and can be selected from the configuration settings. In these cases, all the series are averaged in the time-domain.

Finally, if at least one element possesses a valid time series resulting from the processes above, the run proceeds in the creation of the current scenario. If forecast datastreams are provided, data from forecast corresponding to the valid elements are merged to those coming from the feature aggregation of the sensors. It is important to notice

that also forecast datastreams will undergo the same regularization and control procedures already described in the first process, logging any possible error or warning to the user.

3.2.5   Scenario classification

The above-defined set of time series, each one related to an element of the city, formally defines the input conditions of the current scenario. As regards the EWS subsystem, it is convenient to consider a scenario as an abstract object that incapsulates both input time series related to the city elements and output data as collection of time-ordered layers describing the related events. In this sense, a current scenario defined from the actual situation does not take into account the immediate future events, which are instead associated by the classification procedure performed by the EWS subsystem.

3.2.5.1 Scenarios library

The described procedure relies on a set of predefined scenarios related to the specific CCLL and describing a suitably large spectrum of flooding events, i.e., covering as much as possible the space of events that may occur in the city. It is important to notice that this scenarios library could contain actual known past events, but also synthetic scenarios describing hypothetical extreme situations, especially with the aim of giving the subsystem the ability to detect unexpected or very rare extreme events. The events related to these scenarios are the outcome of the simulations performed using the hydraulic physical model LISFLOOD-FP, described in more detail in the first chapters of this document.

The EWS fetches the scenario from a scenarios folder under the city specific working directory. Each scenario is equipped with a set of comma-separated values (CSV) files containing the time series related to the city elements used as input conditions by the physical model LISFLOOD-FP [17], whose format is compliant with the software. Output layers are also stored in and are uniquely identified by the recorded occurrence time during the run.
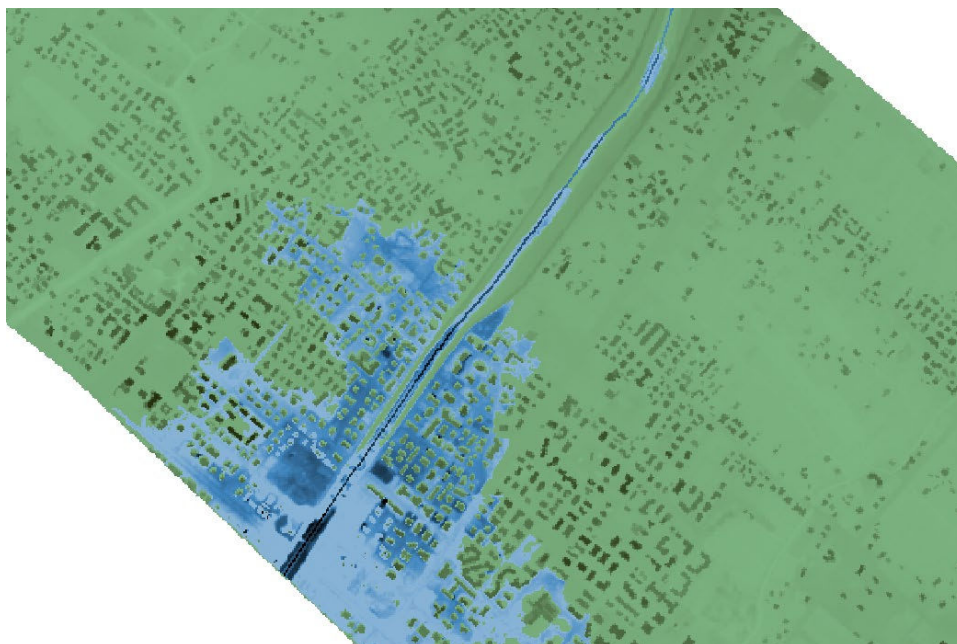

Figure 10: Example of results of flooding in the case of the CCLL of Massa, obtained using hydraulic model LISFLOOD-FP. Here, the detail of a zone of the city is shown using QGIS software [41].

3.2.5.2 Computation

In these terms, the EWS subsystem classification is a procedure that associates the observed current scenario with an output event based on a concept of similarity with the scenarios in the library. Thanks to this procedure, we can obtain the state of the city system in the immediate future in short times and with suitable frequency, integrating the most recent measures from sensors and predictions from short-term forecasts.

More in detail, the idea is to consider a space of events in which both the scenarios in the library and the current one exist and determine to which extent the latter overlaps with the ones contained in the library. In the EWS subsystem, this is achieved by defining a Gaussian Mixture (GM), a probabilistic clustering model allowing to formalize statistically the partial membership of points in the space to a set of gaussian multivariate distributions [20]. In our case, it is applied to the input time series in time-delay embedding. In [31] - [34], the interested reader can find reviews and applications of clustering algorithms, in particular GMs on time series. For the definition of the subpopulations, the subsystem relies on the use of well-known data augmentation techniques [35] - [37] applied to the case of time series (see reviews in [38], [39]). Notice that each scenario in the library, that could be in principle synthetic or based on historical records, defines a subpopulation that is equally represented in the event space. This means that, as explained previously, even if a scenario is related to an extreme, rare past event based on historical data, the system prevents it to be erroneously associated to a small cluster due to the lower frequency of occurrence.

Scenarios are read from a common library and are processed to make them compliant with the mixture model. During this procedure, the software performs some integrity checks on the scenarios, logging possible errors, e.g., scenarios in the library must contain the set of elements used to define the current scenario. In addition, in principle each scenario could have a specific length that is independent from that of the current one and the only assumption is that they must cover both the time span of the current one and of the forecast step (otherwise they cannot provide an output layer compatible with the target forecast time). A procedure of alignment between scenarios, based on the minimization of a metric, the L1 distance between the input time series, is also pursued to make them compliant and well defined in the same event space.

Finally, the classification procedure produces a set of probabilities, one for each scenario in the library, of representing the current scenario.

3.2.6   Warning results

The EWS subsystem implements two different configurable modes for the selection of the forecasted flooding event. The first one is 'mode', which associates the scenario with the greatest probability. The other is 'worst', allowing to select a scenario with probability greater than a certain threshold specified if the average value of the flooding exceeds a fixed value. In both modes, if set up during configuration, raster layers with the water depth can be filtered by a mask specifying the sub-regions of the map over which the computation must be restricted, e.g., cutting off less relevant regions.

The flooding threshold is also used as trigger for the actual alerting process. If the expected level is lower than this value, the computation stops, and the flooding layer is not saved. In other words, this value is intended to be used to avoid storing non-relevant results of the EWS subsystem in the database. Conversely, in case of relevant flooding level, raster of the flooding layer is exported and stored in the warnings sub-directory in GeoTIFF format. An example of flooding layer is given in Figure 10.
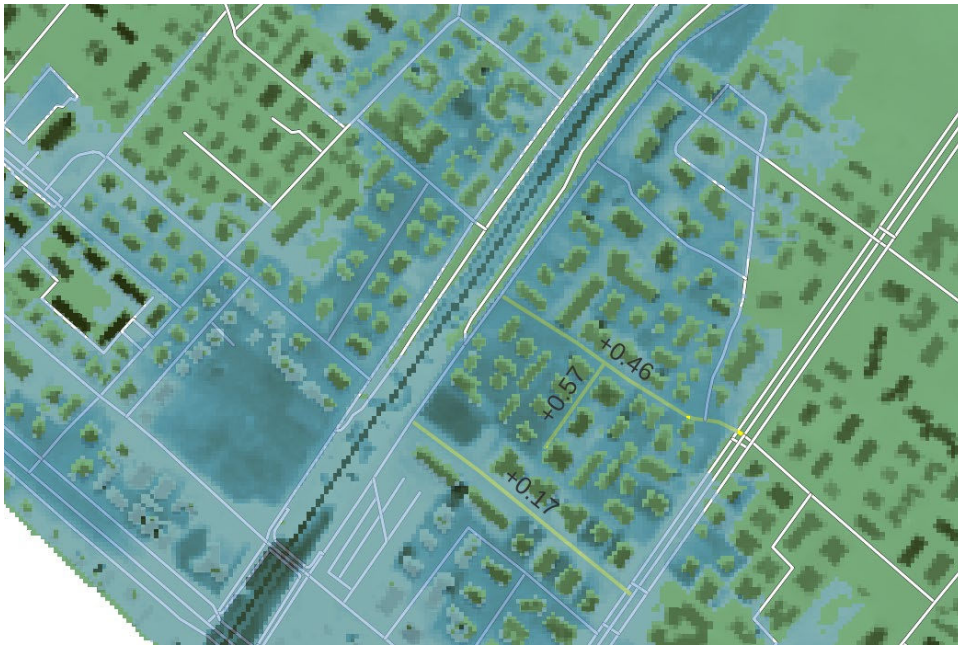
Figure 11: Example of flooding layer and its effect on the road network in the case of Massa, obtained using exposure vector layers extracted from D6.3 [14]. In the picture, the average depth of water (in meters) on some selected roads is shown using QGIS software [41].

3.2.7   Damage computation

The DT-EWS system includes a specific sub-package devoted to the computation and management of risk. The implementation is based on the work of WP6, comprehensively described in document D6.3 [14]. In this context, the vulnerability functions are modelled as abstract functions mapping the value of a hazard, which in our case is the flooding water depth, in a damage depending not only on the hazard itself but also on the attributes of the specific vector features defining the exposure, e.g., a road described by a line shape belonging to a certain vulnerability category or a polygonal zone including many buildings belonging to different vulnerability classes. The specific vulnerability functions defined for each exposure type, e.g., population, roads, and so on, are implemented in the package and are applied to the exposure maps specified in the configuration, keeping into account any possible spatial distribution of the flooding water level resulting from the processes above.

Exposure maps provided by WP6 are stored in the exposure subdirectory as vector layers in geoJSON format, while the flooding raster layer is translated in a vector format through a polygonization process, whose sensitivity can be set in the configuration. For each feature in the exposure map, damage is computed by integrating the partial damage over each sub-feature determined by the intersections between the features itself and the flooding layer. As an example, if a railway intersects two regions with different water depths, the damage must be computed by summing up the partial damages obtained in each of the two segments belonging to each region, and potentially belonging to different vulnerability classes. Notice that this procedure does not depend on the specific shape of the considered features and no approximations are introduced, meaning that the precision of the computation reflects those given by the resolution of the layers.

Finally, the resulting risk maps are stored in the warnings subdirectory in GeoJSON vector format. An example of layer is given in Figure 11 and Figure 12.

Figure 12: Example of flooding layer and its effect on the buildings in the case of Massa, obtained using exposure vector layers by D6.3 [14]. In this picture, the computed damage (in €) on three selected areas is reported in the related yellow-marked cells, using QGIS software [41].

The structure of each map is compliant with that of the corresponding exposure map and includes all the initial attributes in order to keep all the specific information that could be needed by the user. The result of the computation is exported appending two new attributes, respectively indicating the average water depth (in meters) and the computed damage of the feature. In addition, if needed for visualization or classification purposes, it is also possible to define, for each exposure type, classes of damage at custom levels. In this case, a new attribute with integer values will denote the category of the damage with respect to a specified list of values.

3.2.8   Test mode

The EWS includes a test mode, that can be activated using the configuration file, allowing to perform possible more detailed tests for validation and parameter tuning purposes. In this mode, it is possible to use custom sensor time series, specified on CSV files, to be read in place of the default sensor datastreams. Similarly, also the use of the default weather forecast is disabled. The EWS subsystem will therefore read and process them exactly as if they were a sensor read from the APIs, therefore passing through all the operations described in the sections before.

4.   USER OPERATIONS

As explained in Section 2, scenarios are crucial elements for the DT-EWS system operations. These collections of inputs, either user-created or outlined by real situations, are the baseline for the system to run simulations.

4.1 User-created scenarios

Users must create scenarios to let the USE subsystem run the simulations for what-if or long-term analyses, depending on the kind of inputs that are gathered in the scenario. In the following, two examples of the possible steps that the user can carry out to create the two kind of scenarios. It is worth noticing that the order of the steps can be chosen by the end-user, and that they will be performed through the GUI that is described in the deliverables D8.6 [42] and D8.7 [43], where related screenshots of the GUI are also reported.

4.1.1   Structure of a scenario for what-if analysis

What-if analyses develop on simulations performed on scenarios including both real data, like the coastal city baseline data, and data prepared or hypothesised by the user, including weather events, EBAs, and so on. In the following, the steps that a user can follow to create such a scenario:

1. Login onto the system georeferenced on the user's city, thanks to user ID and Password

2. Once redirected on the city maps, the user can select a language among English and the local language

3. Visualize baseline layers

4. Visualize sensor distribution maps

5. Zoom in/out on a city portion (already associated to a risk map), or consider the whole area (also in 3-D), selecting available map layers

6. Open EBAs palette, where the EBAs available for that particular city are listed. Each CCLL will have a different list of EBAs, since they are related to the specific geomorphological scenario

    6.1. Select an EBA and apply it to a selected area on the city map, tracing polygons

    6.2. Repeat step 6.1 or go to the next one

7. Set the weather events parameters (Timeline, intensity, etc.)

    7.1. Set intensity on the timeline divided into slots or by compiling a table

8. Select sea state settings: in the first version of the system, the sea state scenarios are preset and the user will choose one among the available ones

9. Select hydrography settings

    9.1. Set water bodies levels

    9.2. Set flows (in m^3/s) for all the rivers and canals

    9.3. Set water bodies inputs/outputs (water sinks/sources) OR use the already set defaults

At this point, the user can launch a simulation, whose duration must be consistent with the timeline set in the scenario. At the end of the simulation, the outputs can be visualised as new layers and new risk maps and saved and uploaded on the SIP with associated metadata. Later, the simulation results stored on the SIP can be retrieved and downloaded by the user.

4.1.2   Structure of a scenario for long-term analysis

Long-term analyses are based on scenarios that largely use "real" data, meaning that, beyond the city baseline data as in the previous case, the user must consider also the long-term climate scenarios as defined in the RCPs reported in Subsection 2.2.1. Then, hypothetical information can also be included, like possible EBAs, to test their impact on the city resilience against the evolutions of weather along this century, following these steps:

1. Login onto the system georeferenced on the user's city, thanks to user ID and Password

2. Once redirected on the city maps, the user can select a language among English and the local language

3. Visualize baseline layers

4. Visualize sensor distribution maps

5. Zoom in/out on a city portion (already associated to a risk map), or consider the whole area (also in 3-D), selecting available map layers

6. Open EBAs palette, where the EBAs available for that particular city are listed. Each CCLL will have a different list of EBAs, since they are related to the specific geomorphological scenario

   6.1. Select an EBA and apply it to a selected area on the city map, tracing polygons

   6.2. Repeat step 6.1 or go to the next one

7. Select a simulation timeline, in the order of months or years, not exceeding the available RCP scenarios durations

8. Upload available climate models, in the form of raster files that will be at the input of LISFLOOD-FP

9. Upload available sea state models, that will be employed as boundary conditions for LISFLOOD-FP

Exactly as in the previous case, after the last step, the user can launch a simulation, whose duration must be consistent with the timeline set in the scenario. At the end of the simulation, the outputs can be visualised as new layers and new risk maps and saved and uploaded on the SIP with associated metadata. Later, the simulation results stored on the SIP can be retrieved and downloaded by the user.

4.1.3   User authentication

To use the API, a user must first authenticate by sending a username and password to the "/tokens" endpoint to get a <token>. Each subsequent call must use the Barer HTTP authentication scheme:

```
Authorization: Bearer <token>
```

The token expires after 60 minutes of inactivity.

4.1.4   API endpoints

In this subsection, we introduce the REST APIs to make the users operate with the USE subsystem that have been prepared up to now to start testing the system and GUI functionalities, in compliance with [3]. The complete development of all the APIs is still ongoing as a part of the activities of task T8.4.

4.1.4.1 Tokens (login and logout)

POST /tokens

Checks the username and password provided and creates an authentication token (login).

DELETE /tokens

Deletes the current authentication token (logout).

4.1.4.2 Users information

GET /users/me

Gets the data of the currently logged in user.

4.1.4.3 Operations on scenarios

POST /scenarios

Creates a scenario.

GET /scenarios

Lists the available scenarios.

GET /scenarios/{scenario_id}

Gets a scenario.

PUT /scenarios/{scenario_id}

Updates a scenario.

DELETE /scenarios/{scenario_id}

Deletes a scenario.

4.1.4.4 Operations on the EBAs of a scenario

POST /scenarios/{scenario_id}/ebas

Creates an EBA in a scenario.

GET /scenarios/{scenario_id}/ebas

Lists the defined EBAs in a scenario.

GET /scenarios/{scenario_id}/ebas/{eba_id}

Gets an EBA in a scenario.

PUT /scenarios/{scenario_id}/ebas/{eba_id}

Updates an EBA in a scenario.

DELETE /scenarios/{scenario_id}/ebas/{eba_id}

Deletes an EBA from a scenario.

4.1.4.5 Operations on the sea state of a scenario

POST /scenarios/{scenario_id}/seastates

Creates sea states in a scenario.

GET /scenarios/{scenario_id}/seastates

Gets the sea states in a scenario.

PUT /scenarios/{scenario_id}/seastates

Updates the sea states in a scenario.

DELETE /scenarios/{scenario_id}/seastates

Deletes the sea states from a scenario.

4.1.4.6 Operations on the rain rates of a scenario

POST /scenarios/{scenario_id}/rainrates

Creates rain rates in a scenario.

GET /scenarios/{scenario_id}/rainrates

Gets the rain rates in a scenario.

PUT /scenarios/{scenario_id}/rainrates

Updates the rain rates in a scenario.

DELETE /scenarios/{scenario_id}/rainrates

Deletes the rain rates from a scenario.

4.1.4.7 Operations on the river flows of a scenario

POST /scenarios/{scenario_id}/riverflows

Creates river flows in a scenario.

GET /scenarios/{scenario_id}/riverflows

Gets the river flows in a scenario.

PUT /scenarios/{scenario_id}/riverflows

Updates the river flows in a scenario.

DELETE /scenarios/{scenario_id}/riverflows

Deletes the river flows from a scenario.

4.1.4.8 Simulations

POST /scenarios/{scenario_id}/simulations

Creates (queues) a simulation task for a scenario.

GET /scenarios/{scenario_id}/simulations

Gets the simulation state of a scenario.

DELETE /scenarios/{scenario_id}/simulations

Deletes (stops) the simulation task for a scenario.

4.1.4.9 Reports

POST /scenarios/{scenario_id}/reports

Publishes the documents resulting from the scenario simulation on the SIP server.

4.2 EWS subsystem operation

The EWS subsystem is planned to run automatically in background with a dedicated instance for each CCLL and, therefore, the interaction with the user are more limited with respect to the USE subsystem of the prototype. As regards the configuration parameters, these are intended to be used especially or solely for validation/test purposes, and only skilled, technical users are expected to be able to access and modify them through the GUI, and hence to retrieve the modifications in the back-end. Also in this case, then, the user can interact with the subsystem, although in a more limited way. In particular, he or she can:

1. Login onto the system georeferenced on the user's city, thanks to user ID and Password

2. Once redirected on the city maps, the user can select a language among English and the local language

3. Visualize baseline layers like, e.g., DEM/DTM/DSM, hydrography, exposure maps, etc.

4. Visualize sensor distribution maps,

   4.1. Read the status of each sensor, e.g., whether it is active or inactive, and eventually exclude the related data stream from the simulation, in case of sensor inactivity, or if the EWS subsystem detects anomalies in the provided data

   4.2. Plot sensor data

5. Zoom in/out on a city portion (already associated to a risk map), or consider the whole area

6. Run iterative simulations with selected time projection depending on the needed time for monitoring, (e.g., 3, 6, or 12 hours)

7. Create temporary risk maps updating the baseline ones

8. Visualize results (short-term projections, warning list or table)

   9. Download Simulation results

Alerts and results of the EWS runs will be stored and historicized, so that it will possible from the GUI to get an alert when needed, and visualize both current and past situations.

5. SYSTEM OUTPUTS

Since the two subsystems composing the DT platform have different scopes, they produce different outputs. This section describes the outcomes produced by the simulations and made available by the DT-EWS system. Therefore, we divide them into USE subsystem outputs and EWS subsystem outputs. Moreover, each subsystem can produce "raw" or "elaborated" outputs, as explained in the following.
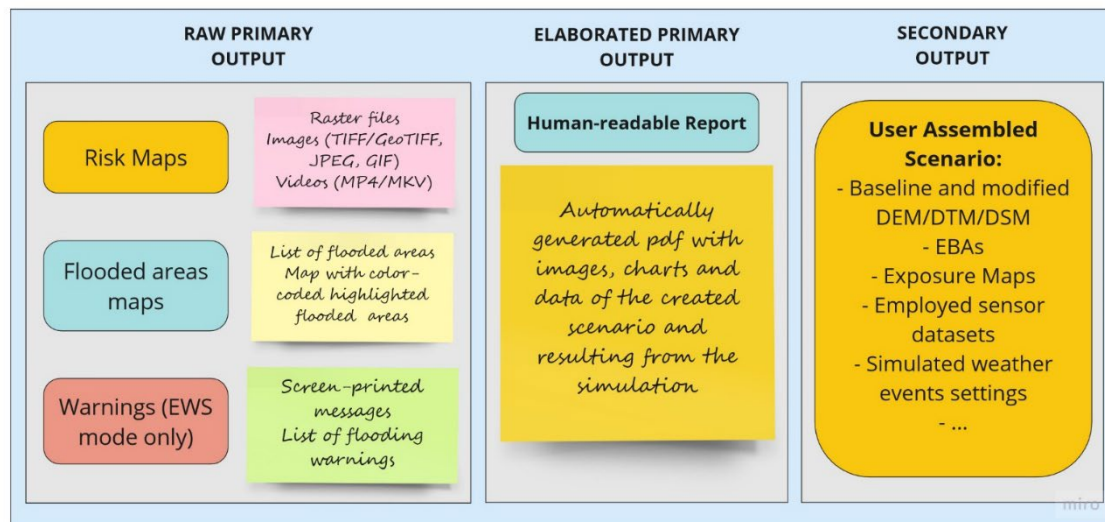

Figure 13. Diagram describing the DT-EWS outputs.

5.1 USE outputs

When operating on the USE subsystem, the user can launch simulations, whose results can be immediately visualized and saved by the user and, once the results are saved, they are automatically stored on the SIP. The system produces a heterogeneous set of outputs, composed by different files. They are grouped in primary and secondary outputs; on their turn, the primary outputs can be raw or elaborated.

5.1.1 USE primary outputs

The scheme structure of the output of the USE subsystem is depicted in Figure 13 and mainly consists of three blocks.

The raw main output contains all the simulation results, in the form of:

1. maps of flooded areas, where the presence of water with a level higher than zero is reported on the city maps as a blue layer

2. human/financial risk maps, produced as a colour-coded map obtained by superposition of the available exposure maps and the simulated flooding maps

The flooding maps can be represented as raster files or images (TIFF/GeoTIFF, JPEG, GIF), or even in video format (MP4/MKV) to visualise the time evolution of the outputs in time. The risk maps, on the other hand, are a grid in which each cell is filled with a different colour, depending on the degree of damage consequent to the registered water level in the cell.

Among the primary outputs, the USE subsystem automatically generates an elaborated output, which is a human-readable report in the form of a PDF file containing images, charts, and data of the scenario assembled by the user and from the results from the simulation; this can help the user having a glance to the simulation outcomes for a first evaluation, before taking the decision about saving the outputs on the SIP or not.

5.1.2   USE secondary outputs

The USE subsystem also generates a secondary output, which assembles all the inputs of the simulation and its outcomes. The former is constituted by the whole scenario assembled by the user, regardless of being a completely hypothetical scenario or including projections from real climate models. Therefore, it must contain:

- The real or modified by the user city DTM/DEM/DSM

- The exposure maps

- The maps of the realised EBAs

- The raster and NetCDF files related to the simulated events, including the sea state boundary conditions.

It is necessary to store these data together with the simulations outcomes, since they report the scenario that generated the outputs, and their knowledge by the user is crucial for a consistent analysis.

Referring to D8.2 [2], the described primary and secondary outputs of the USE subsystem is compliant with the following requirements:

SCORE-REQ-DTEW-FUN-030-01: City risk assessment

SCORE-REQ-DTEW-FUN-042-01: Risk Maps

SCORE-REQ-DTEW-FUN-046-01: Simulation

SCORE-REQ-DTEW-DEV-038-01: GIS-based interface

Moreover, the here-illustrated subsystem is also compliant with the following functional specifications [2]:

SCORE-FSPEC-DTEW-002: GIS-based interface

SCORE-FSPEC-DTEW-051: Map layers selection capability

SCORE-FSPEC-DTEW-052: Types of map layers

SCORE-FSPEC-DTEW-053: Map layers properties

SCORE-FSPEC-DTEW-060: Transition to Map Visualization Mode

SCORE-FSPEC-DTEW-070 – 92: Thematic layers

SCORE-FSPEC-DTEW-101: GIS based approach

SCORE-FSPEC-DTEW-103: Entity attributes update

SCORE-FSPEC-DTEW-110: Map Content

SCORE-FSPEC-DTEW-117: Risk maps

SCORE-FSPEC-DTEW-151: Modified maps updates

SCORE-FSPEC-DTEW-154: Ecosystem-Based Adaptation (EBA)

SCORE-FSPEC-DTEW-155: Edited maps

SCORE-FSPEC-DTEW-156: Entities Comments

SCORE-FSPEC-DTEW-159: Definition of map

SCORE-FSPEC-DTEW-351: Simulation

SCORE-FSPEC-DTEW-354: Simulation Evaluation

SCORE-FSPEC-DTEW-401: Simulation Results presentation

SCORE-FSPEC-DTEW-402: Time navigation of simulation results.

5.2 EWS outputs

The EWS sub-system, during each run produces a log with all the relevant information about each process involved. In particular, it will notify about possible problems, missing data or outliers points coming from sensor data streams (see Subsect. 3.2.4 for details).

5.2.1 Warnings

If the current state represents a dangerous situation, depending on the thresholds that have been set in the configuration, it will export a raster layer in GeoTIFF format representing flooding water depth in each cell of the map, and will process the exposure computing the damage associated to them. The results are then exported in GeoJSON vector format, with same structure of the original exposure file. In addition, a copy of the configuration file and the log is stored. Produced files are stored separately for each city and are associated to the runtime, in order to keep track of the runs history. Layers format, being in well-known and used format, can therefore be easily loaded and visualized by the users using the GUI.

5.2.2 The Common Alerting Protocol

The Common Alerting Protocol (CAP) [40] is a standard open non-proprietary format used for public exchange of public warnings and adopted by many applications and government agencies. Based on a XML structure, it allows to define specific alerting messages equipped with a large number of information about any particular kind of event: meteorological, fire, public health, security, and so on. In the context of SCORE, despite not directly present in the software requirements, it could be possible to include, along with the outputs already described before, a message in CAP format which can be useful for the integration of the DT-EWS system with possible other external system. At current stage, our system does not implement the CAP, however we plan to further investigate and possibly develop the software in order integrate this standard in future versions.

6.   INTERACTIONS WITH OTHER SYSTEMS

The presented DT-EWS system will interact with other systems implemented in the SCORE project, like the GUI, developed in task T8.3 of WP8, to allow the user easily operate with the system, and the SIP, developed by the WP5, as a project knowledge marketplace where the city baseline data and DT-EWS outputs will be stored and organised. Finally, as mentioned in subsect. 5.2, the DT-EWS will eventually have the possibility of interfacing with other existing systems developed outside SCORE.

Here, we briefly outline some elements regarding the interfacing and interactions of the DT-EWS with other systems, since the integration of these systems are the object of ongoing activities of the Task T8.4 "Integration and Deployment of GIS based Early Warning and Digital Twin Platform", focused also on the DT-EWS deployment.
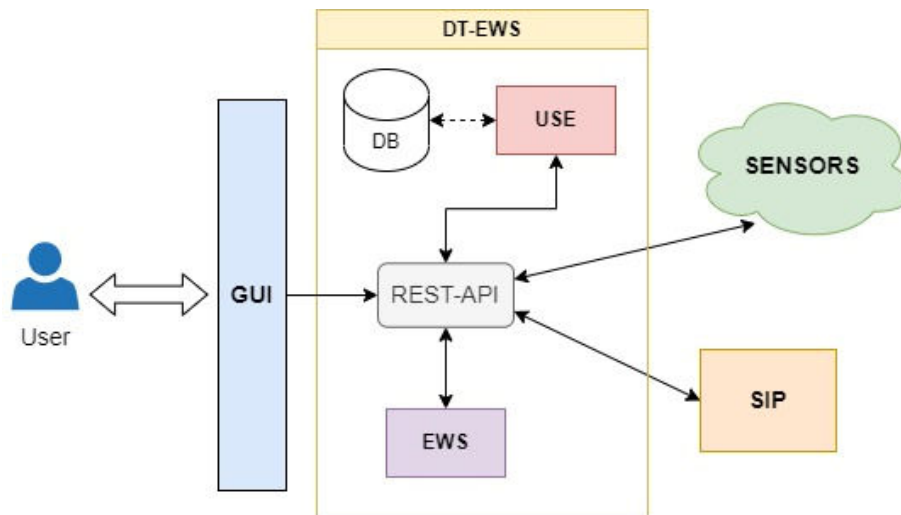

Figure 14. Diagram of the interactions between the DT-EWS subsystems with the other systems. GUI: Graphical users interface; USE: User scenario evaluation; EWS: Early-warning support. REST-API: Representational state transfer-application programming interface; DB: Database; SIP: SCORE ICT platform.

6.1 Interactions with the GUI

The USE and EWS subsystems that has been described in the previous sections represent the back-end of the presented DT-EWS system. Indeed, the final users will employ the GUI to operate with the two subsystem, as more extensively described in [43]. Thanks to the GUI, depending also on the particular subsystem that is considered, users can visualize baseline data as map layers, create scenarios, launch simulations, visualise and save simulations results.

Figure 14 schematically summarizes the interactions among all the involved systems and subsystems. As depicted in figure, the REST-API service contains a set of APIs, whose development is still ongoing in task T8.4, that ensures the communication between back-end and front-end components. The REST-API also contains allows the USE and EWS subsystem obtain the datastreams coming from the sensors.

6.1.1   APIs for interfacing back-end and front-end components

REST APIs that the front-end part of the system will call to send instructions to the back-end part on the user's request are illustrated in subsect. 4.1.4, along with the functionalities they implement.

6.2 Interactions with the SIP

6.2.1   DT-EWS outputs uploading

As already mentioned in Subsect. 3.1, the users can indirectly interact with the SIP once logged on the DT-EWS, by retrieving the baseline data of the coastal city that will be included in the scenario for the simulation. As shown in Figure 14, the communication among the two systems is always guaranteed by the REST-API service.

Given the amount and the size of data that will be transferred from the SIP to the DT-EWS DB, the following solution is at the moment considered: once an input dataset has been downloaded from the SIP, its creation date will be checked on subsequent requests to avoid further downloads if the resource has not changed since the first download, in practice making the DT-EWS implement a basic caching mechanism. The outputs will be sent in GeoTIFF format, with internal compression adopting, e.g., Lempel-Ziv-Welch (LZW) or Deflate lossless algorithms. Further optimizations can also be taken into account, like tiling and the creation of internal overviews to increase the rendering performance on the side of SIP.

The inputs and outputs of the DT-EWS system that have to be uploaded on the SIP must be somehow classified and organised, because the users have to easily retrieve, select, and incorporate them in a scenario. Therefore, the SIP administrator and the data providers (i.e., CCLLs, sensors owners, local weather services, and so on) will agree on a set of keywords to be assigned to inputs, so that the system will be able to query and retrieve the datasets that are useful to its functionality. One or more keywords will be assigned to outputs, once they have been uploaded to the SIP, in order to support content organization and classification. The DT-EWS system will leverage on the resources' extra_metadata field for its internal operation, like querying. An on-purpose solution will be designed for the classification of contents, reflected in the creation of dedicated pages where filtered content (based on one or multiple keywords) is collected and displayed.

6.2.2   AuthN/AuthZ and users management

An important matter for partners involved in WP8, that are responsible for the development of the DT-EWS system, and in WP5, responsible for the creation and administration of the SIP, regards the authentication, authorization and the management of DT-EWS users in relation to SIP. In principle, if this is feasible, the two systems should use the same user base. The SIP can be configured as an Oauth2 provider for DT, in which case DT users must be pre-registered in the SIP.

For users to be able to use the SIP they should be assigned to a specific SIP group. The DT-EWS system, at login time, will query the SIP API to verify if the user is a member of the designated group. In that case, the user will be allowed to operate on the DT-EWS, otherwise access will be forbidden. The DT frontend will query and retrieve the input data for its operations on behalf of the current user, based on an authentication token (e.g., Oauth2 token). On the other hand, the DT-EWS backend will operate with fixed, elevated privileges for uploading the outputs, tagging and assigning permissions on the new datasets. It will take care of assigning the current user as the owner of the new SIP resources.

Permissions on resources will be assigned on the base of users groups to be pre-configured on the SIP. SIP user groups will be the mechanism by which access to the resources will be partitioned for types of users (e.g. the technical personnel of a city town hall, the administrators of the CCLL, etc.).

As regards the provisioning of users, and their assignment to groups at this stage, the SIP offers a global users management system (within its back-office) which is only available to platform administrators. SIP Group managers

can manage group members, but they cannot create users if they are not platform administrators, and there is not a solution to restrict the creation of users only to the groups they manage. The possibility of employing a third-party Authentication service like, e.g., Keycloak, to serve both the SIP and the DT-EWS is under investigation.

7.   CONCLUSIONS

The first version of the DT-EWS back-end system has been implemented in the activities of WP8 of the SCORE project. In this document, the complete description of the system is reported, illustrating its architecture and functionalities, its inputs and outputs, and its interactions with other systems like the front-end and the SIP.

At the moment, users can employ the DT-EWS to run simulations on user-created scenarios, thanks to the USE subsystem. The scenarios include baseline data of the user's coastal city, as well as events related to river discharge and precipitations that the user can arbitrarily set or extract from long-term models based on official RCP scenarios. Moreover, sea-state settings can be chosen as boundary conditions for the digital hydraulic model, among a pool of possible events made available to the users, that can be completely hypothetical or again extracted from RCP scenarios. However, the possibility of letting the users to arbitrarily tune sea-state settings, like tide level and wave height and period, integrating a digital model for sea state, is under investigation. In particular, the aim is to provide the users with this additional degree of freedom in building their scenarios, without making the operations too complex for people that are not expert in wave propagation models.

Moreover, the EWS is also available to run continuous simulations, independently from the USE, for monitoring the city hydraulic situation, fed by the real-time datastreams from sensors, including short-term weather forecast, to predict and assess the impact of oncoming weather events in terms of affected population economical damage.

As a next step, the currently ongoing WP8 activities are aimed at continuing developing the DT-EWS. In particular, the SCORE team is now jointly working at:

i. Completing integration of the back-end and front-end parts of the system, to make it attractive and friendly to the user, thanks to the on-purpose developed GUI

ii. Completing integration of the system with the SIP, where many necessary system inputs and outputs are stored, with particular attention to users management

iii. The deployment of the system in the WP8 front-runners CCLLs, starting from the city of Massa, the pilot case. Meetings with the end users are foreseen to train them on using the system and to fine tune it, based on the users' feedback. It will be important helping the end users to learn a correct use of the DT-EWS system, avoiding misunderstandings. Therefore, with the support of partners active in WP2 and WP9, and with the CCLL core teams, training sessions will be needed and a correct, accurate, and clear communication with the end-users will be crucial, among the activities within the SCORE project.

iv. Test and validate the complete back-end and front-end system functionalities and performance, following the procedures extensively described in the assessment plan D8.10 [4].

8. REFERENCES

[1] SCORE D8.1 – GIS Based Early Warning and Digital Twin Platform functional requirements

[2] SCORE D8.2 – GIS Based Early Warning and Digital Twin Platform system architecture and design

[3] SCORE D8.3 – GIS Based Early Warning and Digital Twin Platform Interface Control Document

[4] SCORE D8.10 – Early-Warning and Spatial Digital Twin Assessment Plan

[5] M. Liu, Sh. Fang, H. Dong, C. Xu, "Review of digital twin about concepts, technologies, and industrial applications", Journal of Manufacturing Systems, Vol. 58, Part B, 2021, pp. 346-361, https://doi.org/10.1016/j.jmsy.2020.06.017.

[6] P. Bauer, B. Stevens, W. Hazeleger, "A digital twin of Earth for the green transition", *Nat. Clim. Chang.* **11**, 80–83 (2021). https://doi.org/10.1038/s41558-021-00986-y.

[7] Oliveira, A., B. Fortunato, A., de Jesus, G., Rodrigues, M., & David, L. (2023). Coastal Digital Twins: building knowledge through numerical models and IT tools. Copernicus GmbH. https://doi.org/10.5194/egusphere-egu23-8041

[8] https://public.wmo.int/en/earlywarningsforall (last accessed 17/08/23)

[9] SCORE D4.2 - Catalogue of low-cost sensors viable for citizen science activities

[10] https://en.wikipedia.org/wiki/Manning_formula (last accessed 17/08/23)

[11] IPCC, *Climate Change 2014: Synthesis Report*. Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change [Core Writing Team, R.K. Pachauri and L.A. Meyer (eds.)]. IPCC, Geneva, Switzerland, 2014, 151 pp.

[12] http://www.sir.toscana.it/consistenza-rete (last accessed 17/08/23)

[13] SCORE D5.5 – Data Management Document, Updated release

[14] SCORE D6.3 – Exposure database and vulnerability curves for the frontrunners CCLLs

[15] Wright, L., & Davidson, S. (2020). How to tell the difference between a model and a digital twin. In Advanced Modeling and Simulation in Engineering Sciences (Vol. 7, Issue 1). Springer Science and Business Media LLC. https://doi.org/10.1186/s40323-020-00147-4

[16] http://xmlrpc.com/ (last accessed 17/08/2023)

[17] https://www.seamlesswave.com/LISFLOOD8.0.html (last accessed 17/08/2023)

[18] T.J. Coulthard, J.C. Neal, P.D. Bates, J. Ramirez, G.A.M. de Almeida, G.R. Hancock, Integrating the LISFLOOD-FP 2D hydrodynamic model with the CAESAR model: implications for modelling landscape evolution, BSG Earth Surface Processes and Landforms, Wiley Online, vol. 38 (15), Dec. 2013, pp. 1897-1906.

[19] T. Hastie, R. Tibshirani, J.H. Friedman, *The Elements of Statistical Learning*, in Springer series in statistics, Springer Science+Business Media (2001). https://doi.org/10.1007/978-0-387-21606-5

[20] G. James, D. Witten, T. Hastie, R. Tibshirani, *An Introduction to Statistical Learning*, New York, Springer, 2013.

[21] Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2012). Isolation-based anomaly detection. ACM Transactions on Knowledge Discovery from Data, 6(1), 1–39. doi:10.1145/2133360.2133363

[22] Lu, Y., Cohen, I., Zhou, X. S., & Tian, Q. (2007, September 29). Feature selection using principal feature analysis. Proceedings of the 15th ACM International Conference on Multimedia. Presented at the MM07: The 15th ACM International Conference on Multimedia 2007, Augsburg Germany. doi:10.1145/1291233.1291297

[23] Iglewicz, Boris. (1993). How to detect and handle outliers. Milwaukee, Wis. :ASQC Quality Press

[24] Pimentel, M. A. F., Clifton, D. A., Clifton, L., & Tarassenko, L. (2014). A review of novelty detection. Signal Processing, 99, 215–249. doi:10.1016/j.sigpro.2013.12.026

[25] Samara, M. A., Bennis, I., Abouaissa, A., & Lorenz, P. (2022). A survey of outlier detection techniques in IoT: Review and classification. Journal of Sensor and Actuator Networks, 11(1), 4. doi:10.3390/jsan11010004

[26] Blázquez-García, A., Conde, A., Mori, U., & Lozano, J. A. (2022). A review on outlier/anomaly detection in time series data. ACM Computing Surveys, 54(3), 1–33. doi:10.1145/3444690

[27] Ma, L., Gu, X., & Wang, B. (2017). Correction of outliers in temperature time series based on sliding window prediction in meteorological sensor network. Information (Basel), 8(2), 60. doi:10.3390/info8020060

[28] Leys, C., Ley, C., Klein, O., Bernard, P., & Licata, L. (2013). Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. Journal of Experimental Social Psychology, 49(4), 764–766. doi:10.1016/j.jesp.2013.03.013

[29] Hochenbaum, J., Vallis, O. S., & Kejariwal, A. (2017). Automatic anomaly detection in the cloud via statistical learning. doi:10.48550/ARXIV.1704.07706

[30] van de Wiel, L., van Es, D. M., & Feelders, A. J. (2020). Real-time outlier detection in time series data of water sensors. In Lecture Notes in Computer Science. Advanced Analytics and Learning on Temporal Data (pp. 155–170). doi:10.1007/978-3-030-65742-0_11

[31] Rani, S., & Sikka, G. (2012). Recent Techniques of Clustering of Time Series Data: A Survey. In International Journal of Computer Applications (Vol. 52, Issue 15, pp. 1–9). Foundation of Computer Science. https://doi.org/10.5120/8282-1278

[32] Eirola, E., & Lendasse, A. (2013). Gaussian Mixture Models for Time Series Modelling, Forecasting, and Interpolation. In Advances in Intelligent Data Analysis XII (pp. 162–173). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-41398-8_15

[33] Maharaj, E. A., D'Urso, P., & Caiado, J. (2019). Time Series Clustering and Classification. Chapman and Hall/CRC. https://doi.org/10.1201/9780429058264

[34] Povinelli, R. J., Johnson, M. T., Lindgren, A. C., & Jinjin Ye. (2004). Time series classification using Gaussian mixture models of reconstructed phase spaces. In IEEE Transactions on Knowledge and Data Engineering (Vol. 16, Issue 6, pp. 779–783). Institute of Electrical and Electronics Engineers (IEEE). https://doi.org/10.1109/tkde.2004.17

[35] M.C. Bishop (2006). Pattern Recognition and Machine Learning (1st ed.). New York, NY: Springer.

[36] Zhang, H., & Wang, Z. (2011). A Normal Distribution-Based Over-Sampling Approach to Imbalanced Data Classification. In Advanced Data Mining and Applications (pp. 83–96). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-25853-4_7

[37] Guyon, I. (2004). Madelon [Data set]. UCI Machine Learning Repository. https://doi.org/10.24432/C5602H

[38]   Iglesias, G., Talavera, E., González-Prieto, Á., Mozo, A., & Gómez-Canaval, S. (2023). Data Augmentation techniques in time series domain: a survey and taxonomy. In Neural Computing and Applications (Vol. 35, Issue 14, pp. 10123–10145). Springer Science and Business Media LLC. https://doi.org/10.1007/s00521-023-08459-3

[39]   Iwana, B. K., & Uchida, S. (2021). An empirical survey of data augmentation for time series classification with neural networks. In F. Schwenker (Ed.), PLOS ONE (Vol. 16, Issue 7, p. e0254841). Public Library of Science (PLoS). https://doi.org/10.1371/journal.pone.0254841

[40]   http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.html (last accessed 17/08/2023)

[41]   QGIS Development Team (2023). QGIS Geographic Information System. Open Source Geospatial Foundation Project. http://qgis.osgeo.org (last accessed 17/08/2023)

[42]   SCORE D8.6 – Tools to access and visualise SCORE data and outcomes

[43]   SCORE D8.7 – Tools to access and visualise SCORE data and outcomes release notes