



Critical Action Planning over Extreme-Scale Data

D5.1 – Initial Report on Explainable AI, Visual Analytics and Augmented Reality

Version 1.0

Documentation Information

Contract Number	101092749
Project Website	https://crexdata.eu/
Contractual Deadline	M18, June.2024
Dissemination Level	PU - Public
Nature	R - Report
Author	Gennady Andrienko (FR), Natalia Andrienko (FR), Eleonora Cappuccio (CNR), Despoina Dimelli (TUC), Bahavathy Kathirgamanathan (FR), Georgios Lamprinakos (TUC), Katerina Mania (TUC), Salvatore Rinzivillo (CNR), Ioannis Safranoglou (TUC), Alexios Stavroulakis (TUC)
Contributors	
Reviewers	Miguel Ponce de Leon (BSC), Arnau Montagud (BSC)
Keywords	Explainable AI, Visual Analytics, Uncertainty, Augmented Reality



CREXDATA has received funding from the European Union's Horizon Europe programme under grant agreement number 101092749.

Change Log

Version	Author	Date	Description Change
V0.1	S. Rinzivillo	2024-05-08	Initial ToC
V0.2	All partners	2024-05-22	Contributions from the tasks
V0.3	All partners	2024-06-03	Finalizing the contributions
V0.4	All partners	2024-06-25	Improvement of the document after internal review
V0.5	S. Rinzivillo	2024-06-25	Final editing and typesetting
V0.6	S. Rinzivillo	2024-06-26	Pre-final version
v1.0	A. Deligiannakis	2024-06-28	Final pass, minor corrections/formatting

Contents

Executive Summary	7
1 Introduction	8
2 Explainable AI (XAI)	10
2.1 Local Rule-based Explanations - LORE	10
2.2 Instance-based explanation for medical images	12
3 Visual Analytics for Supporting XAI	16
3.1 Visual Analytics for Building Models Recognising Behavioural Patterns in Time Series	16
3.2 TRACE - A visual interface for rule-based explanations	20
4 Visual Analytics for Decision Making under Uncertainty	22
4.1 Comparison of simulated and real developments of the COVID19 pandemics in Spain	22
4.2 Supporting comparative analysis for predictive models dealing with event-based data	26
4.2.1 Predicting event based categories with classification models	29
4.2.2 Explanation of event prediction with custom rules	30
5 Augmented Reality at the Field	37
5.1 Weather emergency use case	37
5.1.1 Contributions	37
5.1.2 Motivation and Previous Work	37
5.1.3 System Overview	38
5.1.4 Urban Management	39
5.1.5 System Architecture	41
5.1.6 Server Function	42
5.1.7 GPS Acquisition	44
5.1.8 AR User Interface	44
5.1.9 Water Visualization Logic and Implementation	48
5.1.10 Weather Emergency Use Case: Plans for the next 18 months	52
5.2 Maritime use case	52
5.2.1 Architecture	52
5.2.2 User Experience and Implementation	53
5.2.3 HoloLens Hardware and Usage	55
5.2.4 AR system's User Interface	57
5.2.5 Water Surface Route	58
5.2.6 Maritime Use Case: Plans for the next 18 months	59
6 Uncertainty Visualization in Augmented Reality	60
6.1 Weather Emergency Uncertainty Use Case	60
6.2 Maritime use case	62
6.2.1 Collision Avoidance Challenges	62

6.2.2	Uncertainty Visualization	63
6.2.3	Google Map's Uncertainty Visualization	63
6.2.4	CesiumJS's Uncertainty Visualization	64
6.2.5	Implementation of Uncertainty Visualization in Maritime Navigation . . .	64
6.2.6	Route Visualization	64
6.2.7	Cone of Uncertainty	65
6.2.8	Color-Coded Probability Indicators	65
6.2.9	User Interaction and Feedback	65
6.3	Uncertainty Visualization: Plans for the next 18 months	65
6.4	Trials: The cities of Dortmund and Innsbruck	66
6.4.1	Data from Mike+	68
6.4.2	Innsbruck Trials	70
6.4.3	Dortmund Trials	71
7	Conceptual Modeling for Explainable AI and Uncertainty Communication . .	74
8	Conclusions	78
9	Acronyms and Abbreviations	79

List of Figures

Figure 1	Overview of the LORE pipeline.	11
Figure 2	Extraction of rules from the local surrogate model.	11
Figure 3	Comparison of LORE architectures	12
Figure 4	Architecture of the ABELE explainer	13
Figure 5	AAE Architecture, Discriminator and Decoder modules	13
Figure 6	A Progressive Growing AAE	14
Figure 7	Synthetic skin lesion samples generated by ABELE	15
Figure 8	Participants confidence towards ABELE explanations.	15
Figure 9	Schematic workflow for acquiring input from domain experts	17
Figure 10	Generation of labelled pattern class examples	19
Figure 11	An example of the TRACE interface for a rule-based explanation	21
Figure 12	Comparing trends of the simulated and actual COVID19 over time	23
Figure 13	Distribution and temporal variation of the simulation model residuals	24
Figure 14	Comparison of simulated COVID19 across time and the set of provinces	24
Figure 15	The provinces with the largest negative and positive forecast errors	25
Figure 16	Spatio-temporal distribution of the residuals by cell colouring	26
Figure 17	Categories of the pandemic and mobility levels for weekly episodes	28
Figure 18	Event sequences generated for Madrid and Barcelona provinces	28
Figure 19	Distributions of the COVID and mobility events over time and provinces	29
Figure 20	Predicted categories of the COVID19 events for one model variant	30
Figure 21	Misclassified events from each of six models	31
Figure 22	Cumulative disagreement from each model with the ground truth	32
Figure 23	Discrepancies between the predictions of 6 models	33
Figure 24	Projection of the predictions of the 6 models on a plane	34
Figure 25	LORE explanation process	35
Figure 26	Bipartite graphs with the probabilities for COVID and Mobility	36
Figure 27	Comparison of the neighborhoods created by two generators	36
Figure 28	Types of urban flood	39
Figure 29	System Architecture	41
Figure 30	Dataflow Diagram	43
Figure 31	Compass	44
Figure 32	Menus	45
Figure 33	AR POIs on Real World	46
Figure 34	Exeamples of icons for POIs	46
Figure 35	Map for Weather Emergency AR Visualization	48
Figure 36	Scanned Area	49
Figure 37	Scanned Area with Water	49
Figure 38	Water Shader Graph	50
Figure 39	Outdoors Flood AR Concept 1	51
Figure 40	Outdoors Flood AR Concept 2	51
Figure 41	Biosignals on the Field Concept	52
Figure 42	AR Elements	54
Figure 43	Information areas and the types of application components	55
Figure 44	Three types of zones of the bridge for the AR application	55
Figure 45	AR Map locked on Sky Band	58

Figure 46	Augmented Reality Visualization Route Concept	59
Figure 47	Uncertainty in predicted depths, extents, and hazard assessment	61
Figure 48	Certainty of Predicted Water Level	61
Figure 49	POI's with Uncertainty	62
Figure 50	Google Maps Uncertainty Radius	63
Figure 51	Uncertainty Cone Visualization Concept	64
Figure 52	a) City of Dortmund, b) City of Innsbruck	66
Figure 53	The selected area for trials	68
Figure 54	MIKE+ software	68
Figure 55	Innsbruck Flood Area Grid	69
Figure 56	MIKE+ FLOOD CSV Sample	69
Figure 57	Innsbruck AR Trial Location	70
Figure 58	Innsbruck AR Trial Location	71
Figure 59	Dortmund Location and Trials	72
Figure 60	Before Changes UI	73
Figure 61	New UI	73
Figure 62	An example of a Conceptual Model for a decision of a ML model	75
Figure 63	A fragment of a conceptual model for a model recognizing vessel movement patterns	76

Executive Summary

This document provides an overview of the research activities of Workpackage 5. These tasks focus on the intersection of Explainable AI, Uncertainty, Visual Analytics, and Augmented Reality. WP5 aims to develop human-centred approaches to creation and explanation of Machine Learning models. The document presents the scientific advances in each of these directions, tailored around the set of use cases of the project. On one side the use cases provides the requirements, the challenges to be addressed, and the testbeds to validate the developed methods. On the other side, the analytical methods developed in WP4 will be used to support the analysis of the data and the development of the models.

The document emphasizes the value of visual analytics and augmented reality in enhancing model interpretability, understanding uncertainties, and aiding decision-making in complex scenarios.

The document is structured as follows:

- Section 2 presents the research activities on Explainable AI, with a focus on the development of the LORE method for generating local rule-based explanations.
- Section 3 gives an overview of the Visual Analytics methods, with a focus on supporting the creation of ML models incorporating expert knowledge and the presentation of explanations to users.
- Section 4 discusses the strategies to exploit Visual Analytics to support the analysis of uncertainty in the models and to extract the knowledge of the expert to be incorporated in the models.
- Section 5 presents the advances on Augmented Reality, with a focus on the development of AR methods to support the use cases of the project.
- Section 6 shows how Augmented Reality could support the analysis of uncertainty in the models.
- Section 7 presents the development of a conceptual model of uncertainty to map the concepts and relationships of the domain and the user toward the data-driven models.

1 Introduction

Understanding of computer models by users has critical importance for their adoption for practical use. The research in eXplainable AI (XAI) develops algorithmic approaches intended to help users understand model decisions. However, these approaches alone are insufficient for two main reason. The first issue is the cognitive mismatch between algorithmically generated explanations and human mental models and logic. A model is truly explained if a domain expert accepts it based on both empirical evidence of satisfactory accuracy and compliance with the domain knowledge and reasoning. The second issue is ignorance of human cognitive capacities. While certain types of models, which are often used as mimic models to represent the work of “black boxes”, are deemed “transparent” or “interpretable” by design, they may be incomprehensible in practice due to their size and complexity.

In CREXDATA WP5 we strive to address these problems by developing human-centred approaches to creation and explanation of ML models. This includes

- supporting interactive exploration of model recommendations in a dialog with an XAI system,
- employing visual analytics techniques for creating human-intelligible models,
- involving easily perceivable and interpretable visual representations in explanations,
- designing interactive visualizations to facilitate understanding of model prediction uncertainties.

Quoting the CREXDATA project proposal, we address the MO3.2 objective: “**Visual analytics coupled with XAI for understanding complexity and reasoning under uncertainty.** Helping users to comprehend the space of possible situations, future worlds and choose a reasonable decision strategy. Giving user access to the internal rationale of AI models, enabling a Human-Machine conversation to improve the quality of the decision process, leveraging a data-driven approach to exploit expert knowledge.”

In this first period of the project, we have focused on four main direction of research:

- Explainable AI (XAI) methods, with a focus on post-hoc explanations, to make accessible the internal processes of Machine Learning models. This activity mainly focused on the study and development of explanation models that can be directly applied to black-box models of the use cases of the project (see Section 2)
- Uncertainty in models, with a focus on the development of a conceptual model of uncertainty to map the concepts and relationships of the domain and the user toward the data-driven models (see Section 7)
- Visual Analytics (VA) methods to support the accessibility of the human to the complexity of models. The VA methods are designed to support the creation of ML models, incorporating expert knowledge, and the presentation of explanations to users (see Section 3). They also include solutions to assist the analysis of uncertainty in the models (see Section 4)
- Augmented Reality (AR) methods, with a focus on supporting the use cases of the project. The AR methods are designed to support the interaction of the user with the

models and the data, and to provide a more immersive experience in the exploration of the models (see Section 5). They are also designed to support the analysis of uncertainty in the models (see Section 6)

2 Explainable AI (XAI)

This section will provide an overview of the explainable AI methods developed within the project. Inspired by [6], we are following two parallel strategies to make the AI models more interpretable and explainable. The first strategy is to use inherently interpretable models, such as decision trees, which are easy to understand and explain. The second strategy is to use post-hoc explainability methods, which can be applied to any black-box model to provide explanations for their predictions. This general approach should also take into account the specific requirements of the application domains of CREXDATA, such as the need for explanations to be consistent with domain-specific knowledge and be compatible with the time-sensitive decision making processes of the CREXDATA users. Besides, the explanation methods should also be able to handle different types of data, ranging from tabular data to unstructured data, such as text, images, and time series.

2.1 Local Rule-based Explanations - LORE

LORE [20] is a local rule-based explanation method that can be applied to any black-box model. LORE generates a set of rules that approximate the behavior of the black-box model in the vicinity of a specific instance. The rules are generated by training a local interpretable model, such as a decision tree, on the neighborhood of the instance to be explained. The rules are then used to explain the prediction of the black-box model for that instance.

LORE has been shown to provide accurate explanations for a variety of black-box models, including deep learning models, and for a variety of data types, like tabular data, images, time series. Within the project we worked to enhance the flexibility and extensibility of the original library. This flexibility is mainly motivated by the requirements arising from the use cases of CREXDATA. In fact, the use cases are characterized by the need of a high level of customization of the explanation methods, to make them compatible with the domain-specific knowledge and the time-sensitive decision making processes of the CREXDATA users. For the use case on health crisis, for example, we have developed a customized neighborhood generation (presented in Section 4.2.2). We already identified other cases where this customization is mandatory, such as the use case on maritime data, using a custom method for predicting the events extracted with the method presented in Section 3.1.

To introduce the improvements implemented in the new version, let us first recall the main steps of the original LORE algorithm. Given a black-box model and an instance to be explained, LORE generates a set of rules that approximate the behavior of the black-box model in the vicinity of the instance. The rules are generated by training a local interpretable model, such as a decision tree, on the neighborhood of the instance to be explained. The rules are then used to explain the prediction of the black-box model for that instance. Figure 1 shows an overview of the process to extract the explanations for a black-box for image classification. The input is first encoded into an n -dimensional space using a pre-trained **encoder** (Figure 1(a)). Then, around the projected point a **neighborhood generator** is used to create synthetic instances that are similar to the instance to be explained (Figure 1(b)). The synthetic instances are decoded and supervised by the black-box that assigns them a label (Figure 1(c-f)). This supervised dataset is used to train a local **surrogate** interpretable model (Figure 1(g)), such as a decision tree, that approximates the behavior of the black-box

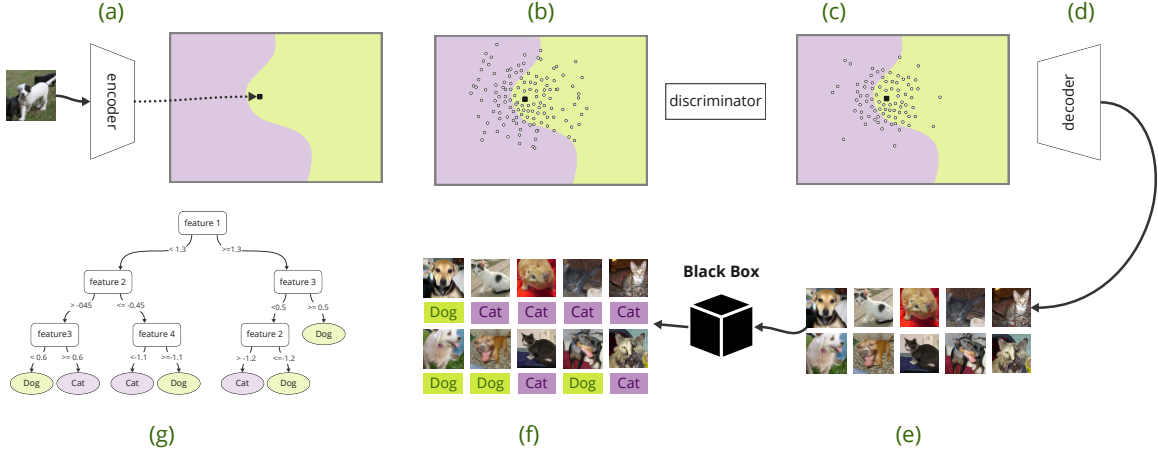


Figure 1: Overview of the LORE pipeline.

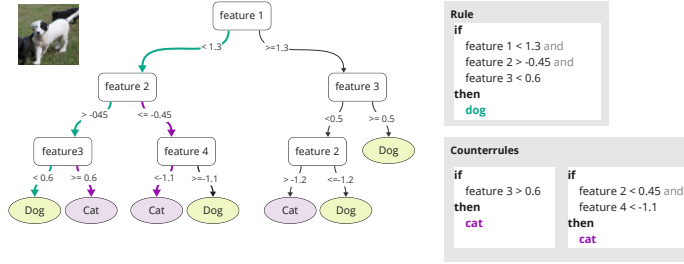


Figure 2: Extraction of rules from the local surrogate model.

model in the vicinity of the instance to be explained.

The rules generated by the local interpretable model are then used to explain the prediction of the black-box model for the instance to be explained.

Formally, a rule is defined as a statement like $p \rightarrow y$, where the *consequence* y is the output of the black box and the *premise* p is a conjunction of split conditions on the observed features. Each condition for a numerical feature can be represented as a predicate of the form $a_i \in [v_{i,l}, v_{i,h}]$, where a_i is one of the features of the data and $v_{i,l}$ and $v_{i,h}$ are respectively the lower and upper bounds (extended with infinity) for the domain of a_i where the predicate is valid. For categorical data types the predicate has the form $a_i \in \{v_l, v_j, \dots, v_k\}$, where each v_i is a value of a_i . An instance x is covered by a rule r if all the predicates of the premise of r are satisfied by x .

Figure 2 shows the extraction of rules from the local surrogate model. The rules are extracted by traversing the decision tree and collecting the conditions along the path from the root to the leaf node that corresponds to the prediction of the black-box model. The conditions are then used to generate a human-readable explanation of the prediction. Counterfactual rules are derived by changing the conditions of the rules to generate instances that would lead to a different prediction by the black-box model.

The general structure of LORE pivots around three main elements: encoder, neighborhood generator, and surrogate model. From the original version of LORE, we focused on a new paradigm to make such elements more flexible. As shown in Figure 3, the new implementation allows the user to easily replace the encoder, neighborhood generator, and surrogate model with custom implementations. We exploited this new flexibility in the use cases of the CREXDATA project, such as the instance-based explanation for medical images described in Section 2.2, the new visual interface for rule exploration in Section 3.2. or the customized neighborhood generation for COVID19 data described in Section 4.1.

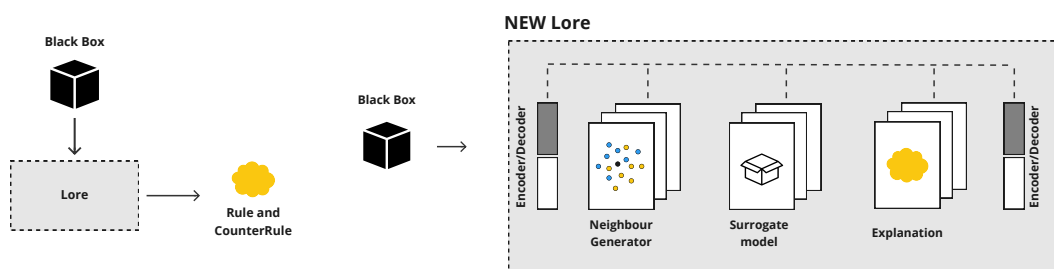


Figure 3: Comparison of the new LORE architecture with the legacy implementation.
(Left) Legacy implementation with monolithic implementation of internal methods.
(Right) New implementation with flexible encoder, neighborhood generator, and surrogate model.

The refactored version of LORE has been made public and is available on the official repository of LORE on GitHub¹.

2.2 Instance-based explanation for medical images

Considering the decision processes investigated in the Health Crisis use case, we have designed an explanation methodology that would be able to deliver explanations to be taken into account by the decision makers. Thus, we consider the problem of defining a decision making pipeline where an image classifier is coupled with an explainer. We proved the proposed methodology with a public accessible dataset on medical images. The approach, of course, is easily extendable to other domains. This section shows a case study where a deep learning model is used to classify medical images and the predictions are explained using the LORE method. This work has been published in several papers: in the Journal on Data Science and Analytics [35] we introduce the main methodology using a legacy implementation of LORE specifically developed for image classification and explanation, and we assess the usefulness of the generated rules with a user study; in [37] we highlight the impact of using LORE paradigm and its interfaces to support clinicians in their decision making processes; in [36] we present a use case on skin lesions image classification. We are actively working to use the new version of LORE to improve the efficiency of rule generation and the quality of the explanations.

The aim [35] is to extend and exploit the methodologies illustrated in [21, 38, 39], and to study the usability of an explanation method into a real medical setting, to assist the practitioner for

¹https://github.com/kdd-lab/LORE_sa

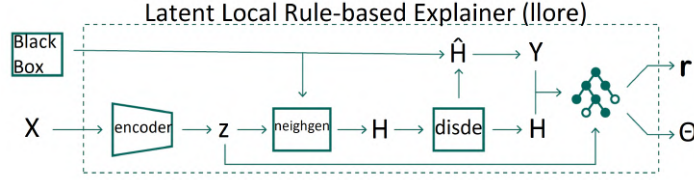


Figure 4: Architecture of the ABELE explainer, [21].

skin lesion diagnosis from images.

We rely on the labeled dataset available from the ISIC 2019 (*International Skin Imaging Collaboration*) image classification challenge. We train a state-of-the-art deep learning classifier using the ResNet [25] architecture on the dataset. After that, we explain the classifier decisions through ABELE [21]. ABELE is a local model agnostic explainer that takes as input an image x and a black-box classifier b , and returns (i) a set of *exemplar* and *counter-exemplar* images, and (ii) a *saliency map*. Exemplars and counter-exemplars are images synthetically generated and classified with the same outcome as x , and with an outcome other than x , respectively. They can be visually analyzed to understand the reasons for the decision. The saliency map highlights the areas of x that contribute to its classification and areas that push it into another class.

Fig. 4 shows the architecture of ABELE, whose general schema is very similar to LORE, and it employs an Adversarial AutoEncoder (AAE) [33] to map each input image to a latent space. We are still working on porting this schema in the new version of LORE, to be used as a new collection of modules, rather than a standalone implementation.

The AAE architecture (Figure 5) includes an encoder: $\mathbb{R}^n \rightarrow \mathbb{R}^k$, a decoder: $\mathbb{R}^k \rightarrow \mathbb{R}^n$ and a discriminator: $\mathbb{R}^k \rightarrow [0, 1]$ where n is the number of pixels in an image and k is the number of latent features. The image $x \in \mathbb{R}^n$ to be explained is passed as input to the autoencoder where the encoder returns the latent representation $z \in \mathbb{R}^k$ using k latent features with $k \ll n$.

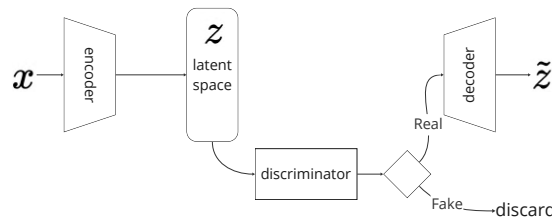


Figure 5: AAE Architecture, Discriminator and Decoder modules, [21].

The quality of the AAE is crucial to the quality of the explanations, in particular for very specific tasks such as medical image classification. We have spent a significant effort to improve the quality of the AAE. Generative Adversarial models are generally not easy to train as they are usually affected by a number of common failures, like, for example, *failures in convergence* or *mode collapse* [57], i.e., the tendency by the generator network to produce a small variety of output types.

In order to overcome such generative failure and dataset limitations, we implemented a

collection of cutting edge techniques that altogether are capable of addressing all the issues we mentioned and successfully training an AAE with adequate performance. In particular, we address mode collapse using ad-hoc tricks like Mini Batch Discrimination [50] and Denoising autoencoders [61].

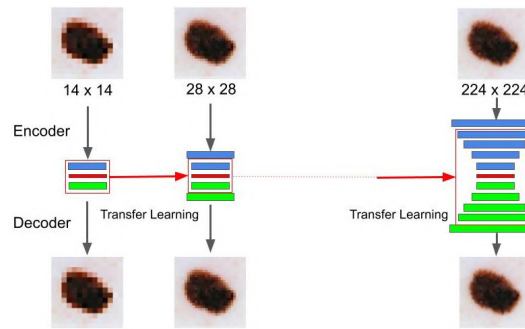


Figure 6: A Progressive Growing AAE. At each step, an autoencoder is trained to generate an image that is twice the size of the previous one, starting from an image of 14x14 pixels and gradually increasing to an image of 224x224 pixels. The learned features from one autoencoder are then transferred to the next. To handle the growing image size, both the encoder and decoder networks are expanded by adding one convolutional block at each step. The transfer learning is confined to the shared network architecture.

We implemented a Progressive Growing Adversarial Autoencoder [27]. Progressive growing helps to achieve a more stable training of generative models for high resolution images like in our case. The main idea is to start with a very low resolution image and step by step adding block of layers that simultaneously increase the output size of the generator model and the input size of the discriminator model until the desired size is achieved. Starting with a single block of convolutional layers for encoder and decoder, we are able to reconstruct low resolution images (14×14 pixels), then step by step we increase the number of blocks until the networks are powerful enough to manage images of the desired size, i.e., 224×224 pixels in our case. The latent space dimension is kept fixed, consequently the discriminator takes as input tensors always of the same size. Although one could fix also the network of the discriminator, we found helpful to progressively increasing also the width of this network so that the discriminator can deal each step with a more structured information. The incremental addition of the layers allows the autoencoder to first learn large scale structure and progressively shift the attention to finer detail. Figure 7 shows some synthetic skin lesion samples generated by our AAE.

We designed a survey to assess the impact of our visual explanations on skin lesion diagnosis. The primary goal is to determine the usefulness of explanations in supporting doctors and medical experts in diagnosing and treating skin cancers, and to evaluate their confidence in black-box based diagnosis models and the explanations provided by the explainer. The survey is composed of ten questions, each presenting a different medical image case. Each question is organized into four different points, following the same structure for the ten cases. Within each question, the participants are asked to estimate their trust into the suggestions coming from the system, by starting with the classification of the black box and progressively adding exemplars and counter exemplars.

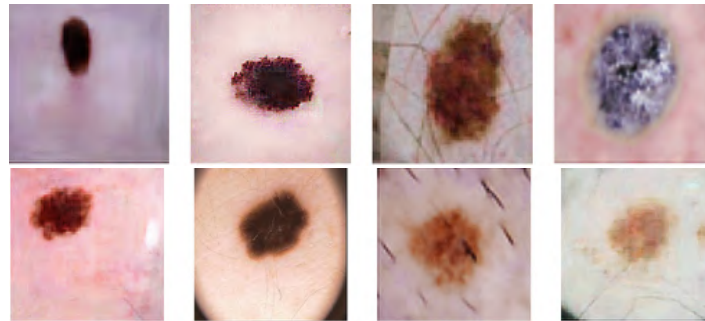


Figure 7: Synthetic skin lesion samples generated by ABELE and classified as Melanocytic Nevus by the ResNet black box, except for the upper right image classified as Actinic Keratosis.

During the survey, participants were not informed of the correctness of their prediction, nor they received further suggestions by looking back at their previous answers or explanations. To investigate the recipient's reaction to incorrect advice, in Q6 we intentionally entered a wrong classification followed by further wrong advice concerning exemplars and counter-exemplars. All the other nine instances were selected among correctly classified cases. The results show a slight mistrust toward the sixth black-box classification, although there is no statistically significant drop in confidence after receiving wrong advice by an AI model (68.75% for Q1 to Q5, 60.03% for Q6 and 66.71% for Q7 to Q10). On the contrary, if we restrict our study to the sub-sample of medical experts, Figure 8 shows a 14% drop of confidence after receiving the wrong advice (78.04% for Q1 to Q5, 56.19% for Q6 and 63.95% for Q7 to Q10), supporting that, after receiving wrong advice from an AI model, domain experts show a decline in confidence and trust toward that model in subsequent instances.

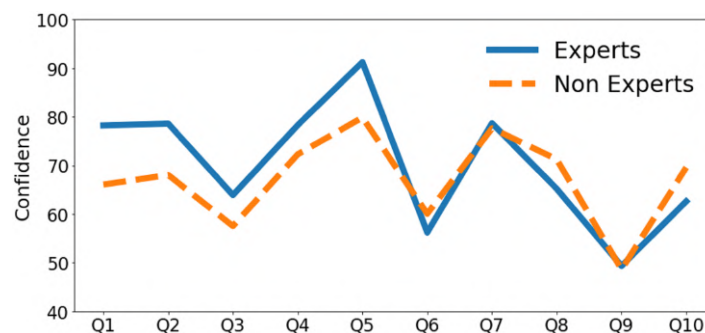


Figure 8: Participants confidence towards ABELE explanations.

This study allowed us to understand the crucial points for medical image classification and explanation. First, the quality of the generated synthetic images is crucial for the success of the explanation and to improve the trust of the expert towards the model. These objectives require an improved quality of the explanations and the quality of the synthetic images generated by the AAE. Second, we experimented an user study to assess the usefulness of the explanations in supporting doctors and medical experts in diagnosing and treating skin cancers. This human-based evaluation is a step that is starting to be more and more important in the field of XAI, and we are planning to extend this study to other use cases.

3 Visual Analytics for Supporting XAI

There are two main uses of visual analytics (VA) for supporting XAI:

- Employ VA in the process of model building to create models incorporating expert knowledge, which makes them better understandable to users. The knowledge extracted in the course of model building can, potentially, be used in generating user-oriented explanations of the model logic and decisions.
- Employ VA techniques in presenting explanations to users and enabling them to explore and thereby better understand the model work.

In CREXDATA WP5, we develop VA approaches aiming to attain both objectives: help domain experts and data scientists create well performing and appropriately explainable models and generate easily understandable explanations of model work for end users.

3.1 Visual Analytics for Building Models Recognising Behavioural Patterns in Time Series

This piece of work is mostly concerned with using VA for model development. It was conducted in relation to the CREXDATA Maritime Use Case (task T2.3). We considered the problem of recognizing different movement patterns in trajectories of vessels. The results of the work have been published in paper [3].

The motivation for developing a VA approach was the difficulty of formulating precise definitions of possible movement patterns and lack of understanding how these patterns may be manifested in data. Due to these problems, rule-based pattern recognition does not attain desired performance. An alternative could be the use of ML methods, but an obstacle is the lack or insufficient amount of labelled training data.

To address this challenge, we propose a novel visual analytics approach (see Figure 9), in which a domain expert detects different movement patterns in visual representations of trajectories and uses domain knowledge for constructing features capable of effectively distinguishing these patterns from other types of behaviour. It is essential that these features need to characterise the *behaviour* of relevant variables *on time intervals*, whereas raw data consist of elementary values referring to individual time steps. Hence, feature construction requires knowledge of (a) what aspects of the behaviour are important, e.g., the range of the values or the development trend, and (b) what kinds of computationally derivable aggregate characteristics can represent these aspects.

The capability of the features to characterise and differentiate patterns is explored using interactive visualisations, which allow an expert to check whether groups of data items that are similar in terms of the features instantiate the same behaviour patterns and whether groups instantiating different patterns are well separated by the features. The visual aids also allow the expert to select and label groups of representative examples of different pattern types and check the suitability and sufficiency of these examples for generating an automated classifier by means of a machine learning algorithm. The interactive visual interface enables simultaneous consideration and labelling of multiple data items, which saves the precious

time of the human while allowing creation of a sufficiently large set of data examples for model training.

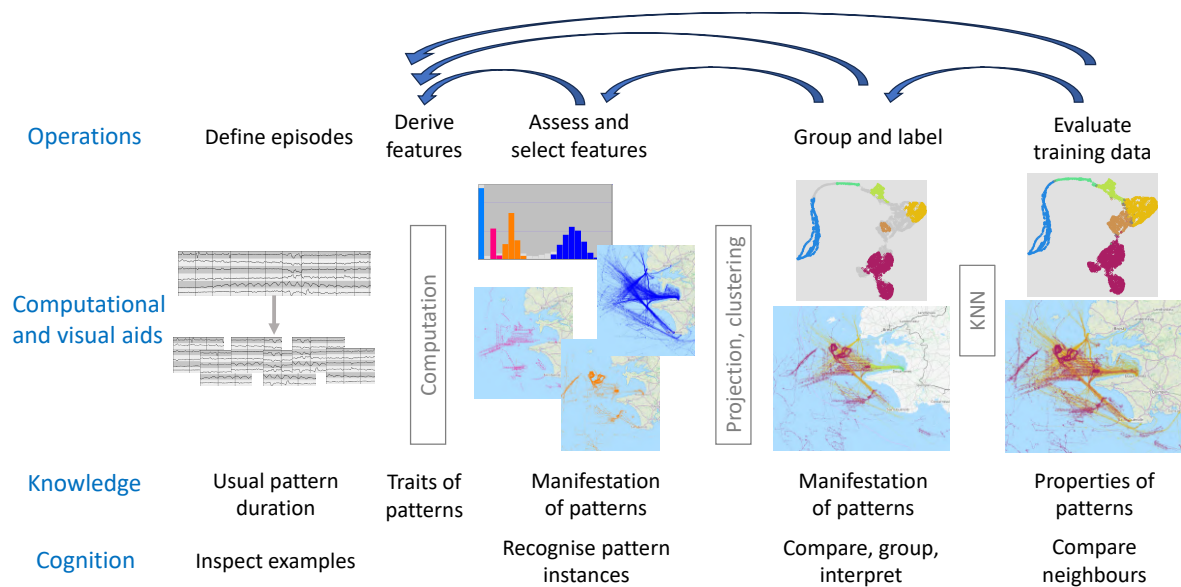


Figure 9: A schematic representation of the workflow of the visual analytics approach to involving domain expertise and human cognition in acquisition of training data for creation of a pattern classification model.

The approach encompasses the following key components (see Figure 9):

1. **Representation of time series:** Convert time series into sequences of *episodes*, each of an appropriate duration to encapsulate occurrences of patterns of interest.
2. **Temporal abstraction and feature generation:** Abstract elementary data to interval-based features, capturing pertinent aspects of behaviour or development during the episodes.
3. **Feature assessment and selection:** Explore and assess the utility of the individual features in discriminating patterns. Choose a combination of features that comprehensively represents different aspects of the patterns. If necessary, return to step 2 to generate additional features.
4. **Grouping, examining, and labelling:**
 - Identify groups of similar episodes in terms of the chosen features.
 - Examine whether the episodes within the groups encapsulate the same pattern types. If not, return to the step of feature selection (step 3) or feature generation (step 2).
 - Label the group members as occurrences of known patterns, thereby producing labelled examples.
5. **Evaluation of the training data:** Check the suitability of the produced training data for derivation of a classification model, i.e., whether unlabelled data can be correctly

classified based on their similarity to the labelled examples.

The efficacy of the entire process depends on the appropriate representation of pattern-characteristic dynamics in attribute values through interval-based features. Key aspects, including *value level*, *variability*, *general trend*, *curvature*, and *fluctuations*, can be effectively captured through various computationally derivable interval-based attributes. These attributes encompass:

- **Value level:** minimum, maximum, mean of the attribute values, quantiles, histograms of relative value frequencies (proportions within predefined intervals).
- **Variability:** summary statistics of the positive and negative changes between consecutive time steps, such as the mean of the changes, their variance, amplitude, etc.
- **General trend:** parameters A and B of the trend line $y = Ax + B$; alternatively, the trend can be described by the angle of the trend line inclination.
- **Curvature:** ratio of the sum of consecutive value changes to the difference between the maximum and minimum, sums of positive and negative deviations from the trend line.
- **Fluctuations:** numbers of positive and negative deviations from the trend line, number of intersections of the trend line.

The effectiveness of different features in distinguishing patterns can be assessed by examining the distributions of feature values and visually exploring groups of episodes with low, high, and medium values.

The approach was evaluated in application to trajectories of fishing vessels. To distinguish their movement patterns, the analysts generated and tested several potentially expressive features and finally chose the following features: the minimum and third quartile of the speed, the logarithm of the curvature, and the logarithm of the minimal distance to the nearest port. The approach employs dimensionality reduction to construct a 2D spatial embedding (projection) based on these features (Figure 10, left). The purpose of the embedding is to expose groups of episodes with similar feature values and enable interactive selection of groups for inspection and eventual labelling. We employ the UMAP method oriented to preserving local neighbourhoods. This means that the method prioritises placing close neighbours in proximity in the embedding, usually at the expense of potentially distorting distances between non-neighbouring objects. Hence, groups of similar episodes manifest as compact clusters of points in the projection, enabling easy visual detection and interactive selection for inspection.

A sufficiently large set of labelled examples can be generated by selecting groups of points directly in the projection plot by means of brushing, view the visual representations of the respective episodes, including a map showing their shapes (Figure 10, right), and assign class labels, if the episodes are deemed suitable to serve as class examples. This process is supported by a visualisation that enables seeing the whole group of selected episodes rather than considering them one by one. In this way, a domain expert can create many examples simultaneously. An alternative or complementary way is to employ clustering. The 2D embedding can be used for informed setting of clustering parameters and assessing the clustering results. Representative examples of different patterns can be selected from cluster members that are close to cluster centres.

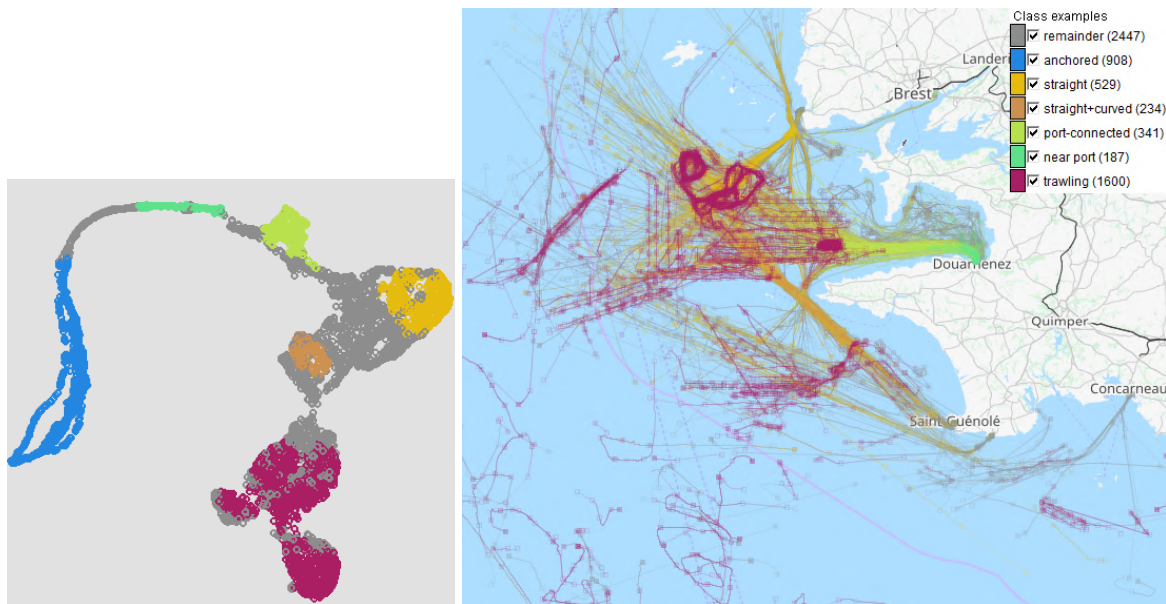


Figure 10: Generation of labelled pattern class examples using 2D embedding (left) and visual representation of movement episodes on a map (right).

As a tool for evaluation, our approach employs the kNN (k-Nearest Neighbours) algorithm. Being applied to unlabelled data, it shows whether their similarity to the examples in terms of the features implies the presence of the same patterns as in the examples.

The outcomes of the workflow can not only contribute to the development of a well performing classifier but also provide materials that can be used in the future for explaining the model's work to users. The first kind of material is the understandable features, for which it is known what types of behaviour they can distinguish and what feature values are pertinent to different behaviours. The second kind of material is labelled pattern examples, which can be used for providing similarity-based justifications of model decisions.

The approach proves particularly valuable in scenarios where patterns of interest lack precise definitions, which is a common challenge in many real-world applications. Furthermore, it addresses the complexity of unknown possible manifestations of these patterns in data. For example, maritime traffic managers know how fishing vessels typically move while trawling but have quite little understanding of the corresponding properties of the vessel tracks.

A crucial step involves temporal abstraction, where elementary attribute values referring to individual time steps are transformed into interval-based summary features expressing the way in which the elementary values vary. This adaptation is essential since behaviours inherently occur within time intervals, whereas the raw data pertains to discrete time points. We propose a catalogue of summary attributes meant for expressing different aspects of distribution and variation of numeric values within intervals. An expert or data analyst can choose potentially suitable attributes corresponding to the relevant facets by which distinct behaviours may differ from each another.

Designed to enable involvement of domain experts, the approach aids in structuring and refining their prior knowledge and empowers them to explore how expected patterns manifest

in the data by assessing different characteristics of variation within time intervals. The expert knowledge and evolving understanding of the data semantics are captured in the form of data-derived relevant features and labelled examples, which become vital inputs for training a machine learning model. This integration of human expertise and machine learning not only enhances model interpretability but also ensures alignment with human domain knowledge.

3.2 TRACE - A visual interface for rule-based explanations

In this section we describe TRACE (a Tool for Rule And Counter-rules Exploration). TRACE is an interactive interface that leverages visual analytics to provide a user-friendly representation of explanatory rules. An early version of this work was published under the name FIPER in [9, 10]. The primary goal of TRACE is to aid human interpretation by allowing users to explore and interact with rule explanations. Through various interactive features, users can enhance their understanding of the underlying prediction algorithm.

This study is inspired by the explanation process outlined by DARPA [22], which involves coupling an Explainable AI algorithm with an Explanation User Interface that serves as an intermediary between users and the algorithm-generated explanations. In our implementation, TRACE exploits on explanations generated by the new version of LORE 2.1, though it can also be adapted for other rule-based Explainable AI techniques.

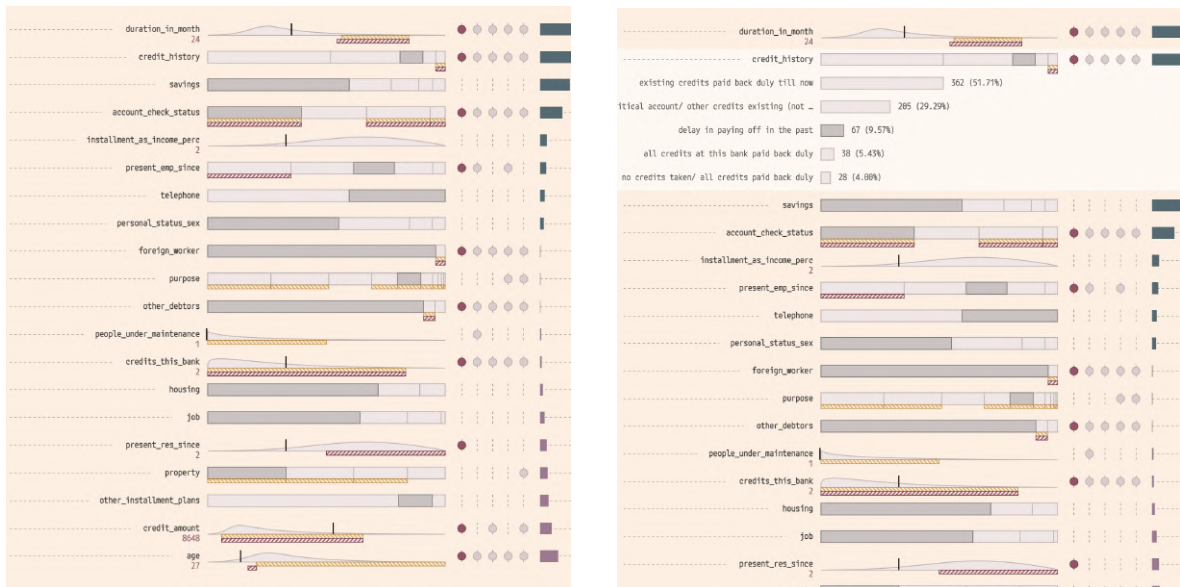
As detailed in Section 2.1, LORE generates a rule along with multiple counter-rules based on an instance. The formal rule definition provided above, which differentiates between numerical and categorical features, led to a distinct visual representation of the instance's features.

The visual interface is under development to finalize the functionalities and improve the user experience. We are currently try many different types of benchmarks and synthetic datasets to test the effectiveness of the interface. As an example, Figure 11a illustrates the organization of the visual space for a rule derived from an instance of the UCI German Credit Risk dataset [16]. The dataset's features are represented using two different types of charts, depending on the feature type. For categorical data, a stacked bar chart is used to display the part-to-whole relationship of each possible value, allowing users to understand the internal distribution of values. A white bar highlights the observed value for the attribute on the x-axis. For numerical data, a density plot provides a compact visualization of the data distribution, with the observed value for x represented by a vertical line within the plot's scale. The specific instance value for that feature is listed below the feature label. The user can delve into the category distribution of each feature by clicking on the row. Also, min, max, first quartile, third quartile, and median values can be visualized with a box plot by clicking on each numerical feature.

For attributes where a predicate p exists in rule r , a second layer is added to highlight the rule's intervals. This interval visualization adapts to the data type: for categorical data, the intervals in the rule premise are underlined with a yellow bar, while for numerical data, the yellow bar represents the range of predicate values.

A summary view of the generated counterfactuals is presented next to the feature representation. Features affected by the counterfactuals are marked with a red dot. Users can increase the granularity of the counterfactuals by clicking on each one to view the involved intervals and categories.

Finally, features are ordered based on their importance values. The Feature Importance (FI) panel uses colour coding to represent the weights, with blue indicating a positive contribution and magenta indicating a negative contribution from the corresponding feature.



(a) An Instance of the German Credit Risk Dataset visualized with TRACE

(b) Bar charts showing the count on the different categories of a feature

Figure 11: An example of the TRACE interface for a rule-based explanation

TRACE incorporates interactivity through various modalities to provide users with information tailored to their cognitive needs. For instance, users can choose to view only the features involved as predicates in the rule or access more detailed feature information by expanding each distribution plot. Additionally, users can explore different counterfactuals by comparing them with the rule. Tooltips offer textual support to the visualization. These actions are guided by the concept of progressive disclosure [55], enabling users to manage the complexity and amount of information displayed.

4 Visual Analytics for Decision Making under Uncertainty

The objective of this task is to help users of simulation and ML models to gain awareness and understanding of uncertainties involved in model forecasts. For this purpose, visual representations and interactive operations can enable viewing and comparing different forecasts. One anticipated use case has been exploration of multiple alternative forecasts for understanding the space of possibilities and selecting the most critical ones to be taken into account in decision making. However, as a large set of alternative forecasts has not yet been produced within the project, we have been focusing on the problem of pairwise comparison, which may refer to comparing two forecasts or to comparing model predictions with real data.

By the current date, we have addressed two examples of comparative exploration tasks, both in the context of the CREXDATA Health Crisis Use Case (task T2.2): (1) comparison of a simulated scenario of COVID19 pandemic development (performed by our project partners in task T2.2) with real data for the provinces of Spain [45]; (2) comparison of predicted impacts of different mobility behaviours on COVID19 dynamics with historical data.

4.1 Comparison of simulated and real developments of the COVID19 pandemics in Spain

For the provinces of Spain, we have obtained historical data [45] about the numbers of new COVID19 cases in each day during the period from 01/01/2020 till 12/08/2021. For the same period, there are results of simulation of the COVID19 development also including the number of new cases in each day and each province. In both datasets, the values have been transformed to the number of cases per 100K of the resident population of each province to ensure comparability.

We have generated several interactive visual displays for comparing the forecast of the COVID19 development with the real dynamic of the pandemic and in this way gaining understanding of the uncertainties involved in the simulation outcomes.

Figure 12 demonstrates a visualization supporting comparison at a high level of abstraction. Two juxtaposed time series plots are meant to represent the overall temporal development trends in real data and simulation outcomes (or, potentially, in results of two simulations). Each line in a plot represents the dynamics of the values of the target variable (here, the number of COVID19 new cases) in one geographic location (here, a province of Spain). The thick black lines represent the dynamics of the mean values over all locations. While the overall shapes of the plots look similar, the comparison is complicated due to heavy overplotting. Therefore, the line plots are accompanied by segmented bar charts representing the data in an aggregated form. The value range of the target variable is divided into intervals, and each interval is represented by a colour using a diverging colour scale (common in both plots). In our example, shades of green represent lower values (dark green encodes the lowest value interval) and shades of red encode higher value intervals (the darkest red represents the highest values). Each bar corresponds to one day, and the segment lengths are proportional to the numbers of values belonging to the corresponding intervals. With this representation, one can notice some differences in the distributions of the daily values across the intervals,

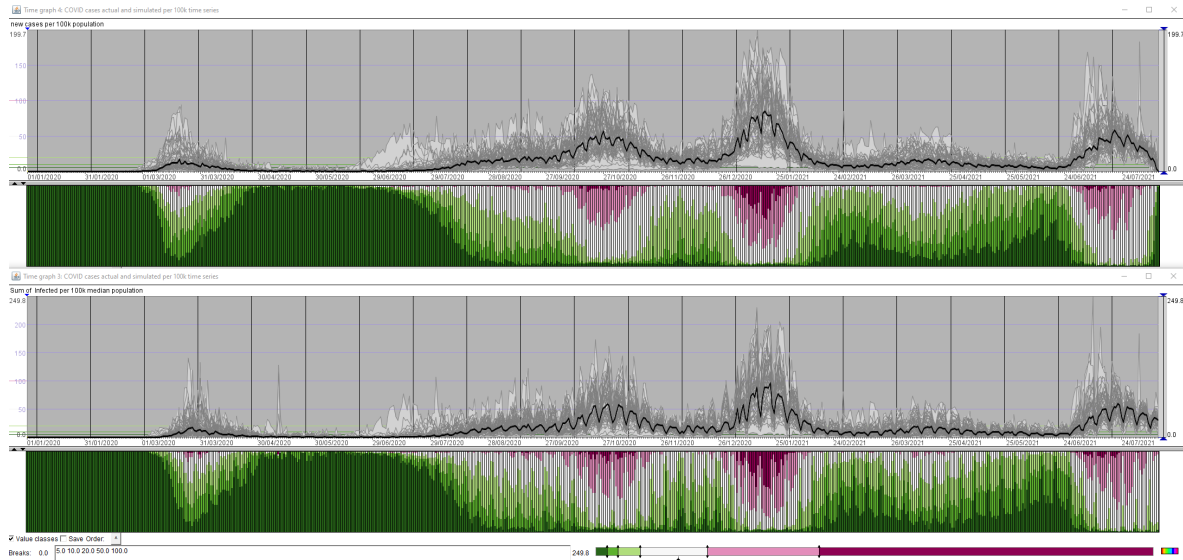


Figure 12: For comparing the overall trends of the simulated and actual COVID19 developments over time, the real [45] and simulated (by task T2.2) data are represented by lines in two juxtaposed time series plots. Below the line plots, the data are represented in an aggregated form by segmented bar charts where segment colours correspond to different intervals of the numbers of new cases per 100K population, from very low (dark green) to very high (dark red).

while the overall trends still appear similar. However, it should be taken into account that identifying corresponding time steps in two line plots or bar charts is quite difficult. Therefore, after comparing the overall trends, it is necessary to take a closer look at the differences.

Figure 13 uses the same visualization technique as in Figure 12 to represent the calculated differences between the corresponding values of the target variable in two datasets (here, the actual values are subtracted from the predicted ones for the same day and province). In the segmented bar chart, another colour scale is used, from dark blue representing large negative differences (i.e., underestimations of the simulation model) to dark red representing large positive differences (i.e., overestimations). The light yellow encodes differences close to zero. With this representation, it becomes obvious that the forecast tends to lag behind the actual development of the phenomenon (COVID19 pandemic). Along the time axis, one can see a repeated pattern of a period of significant underestimations (prevalence of blue bar segments) followed by a period of significant overestimations (prevalence of red bar segments). This means that the model predicts increases of the values to occur later than in reality. When the actual values have already increased, the model still predicts lower values, and it starts predicting higher values when the actual values are already going down.

For yet more detailed investigation of the differences we propose a matrix display, as in Figure 14, where the columns correspond to time intervals (weekly in this example) and rows to geographic locations (provinces). We apply a row arrangement algorithm that strives to put geographic neighbours close on the vertical axis. Like in the segmented bar chart, the colours from dark blue to dark red represent negative and positive differences. This visualization clearly shows that (1) the lag of the prediction with respect to real development is mostly

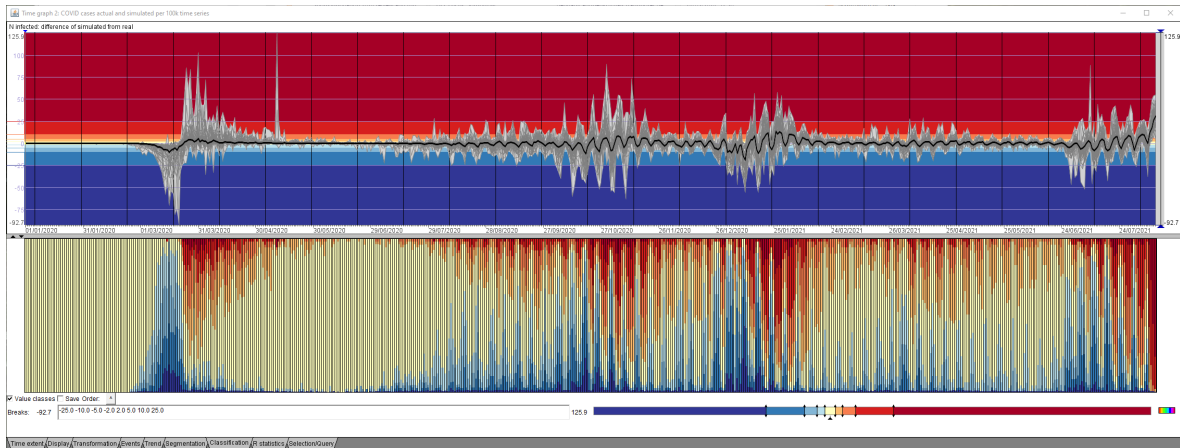


Figure 13: The time graph and segmented bar chart represent the distribution and temporal variation of the simulation model residuals with respect to the actual values. Blue shades represent underestimations and red - overestimations.

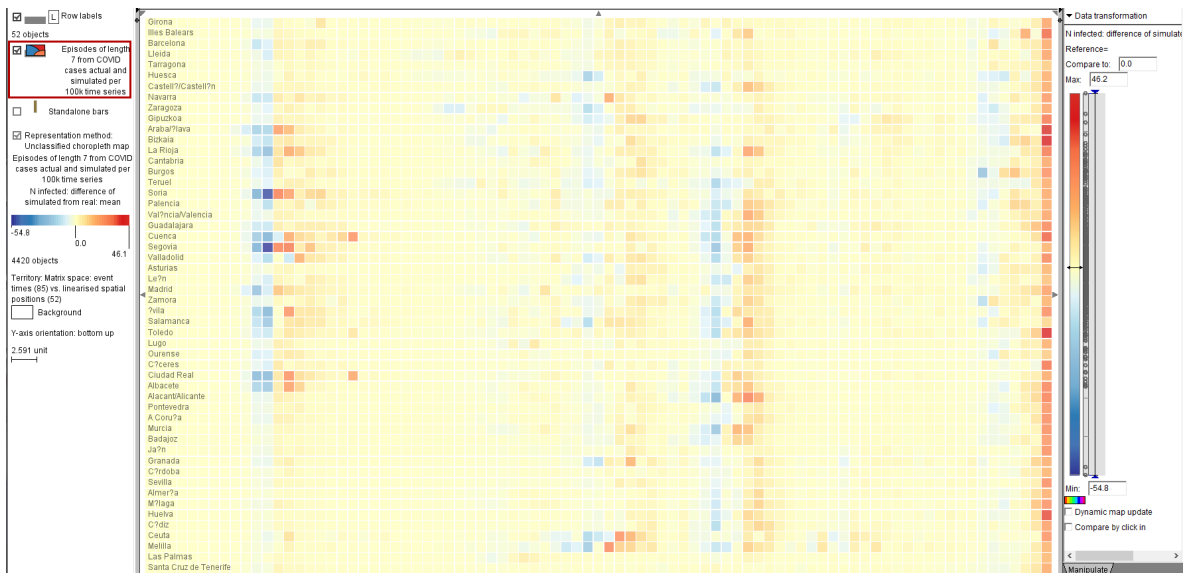


Figure 14: Comparison of simulated COVID19 development with the real evolution across time (horizontal axis) and the set of provinces (vertical axis). Each cell of the matrix correspond to an interval of 7 days and represents the mean simulation residual for the respective week and province. As before, underestimation is represented by shades of blue and overestimation by shades of red.

about 3 weeks; (2) the deviations for the first and third waves of the pandemic are higher than for the second wave; (3) the deviations in the same time period are not uniform throughout the geographic space, but there are groups of provinces where the under- and overestimations are much higher than in the others. To identify these provinces, one can interactively select the lines with extremely low and/or high values of the residuals in the period of interest, and the corresponding provinces will be highlighted on a map, as shown in Figure 15. In this illustration, it is visible that the provinces with large deviations of the predictions from the real values in the first wave of the pandemic form a geographically contiguous region. A domain expert can try to relate the model errors to the geography, including the road network.

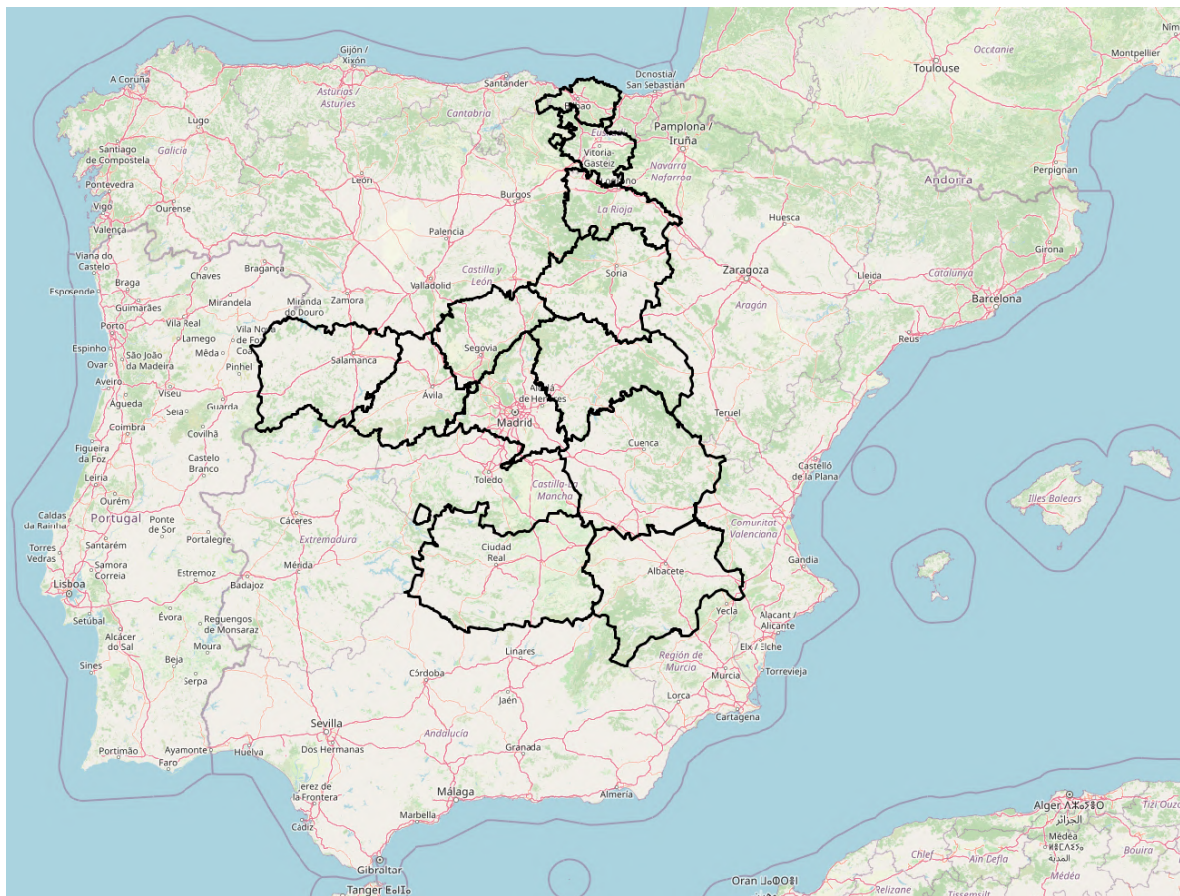


Figure 15: The provinces with the largest negative and positive forecast errors for the first wave of the COVID19 pandemic are highlighted on a map of Spain.

It is also useful to investigate whether the model prediction lags and residuals are related to the amount of increase of the real values. To support this, we complement the matrix visualization with bars put on top of the cells (Figure 16). The bar lengths are proportional to the weekly mean real values attained in the weeks and provinces represented by the matrix cells. It is visible that the temporal lag in the prediction is mostly constant (about 2-3 weeks) irrespective of the real value levels. The amounts of under- and overestimation are the largest in the first pandemic wave, when the actual values were not very high. The model errors are significantly lower in the third wave, when the actual values were the highest. This is a positive property of the model.

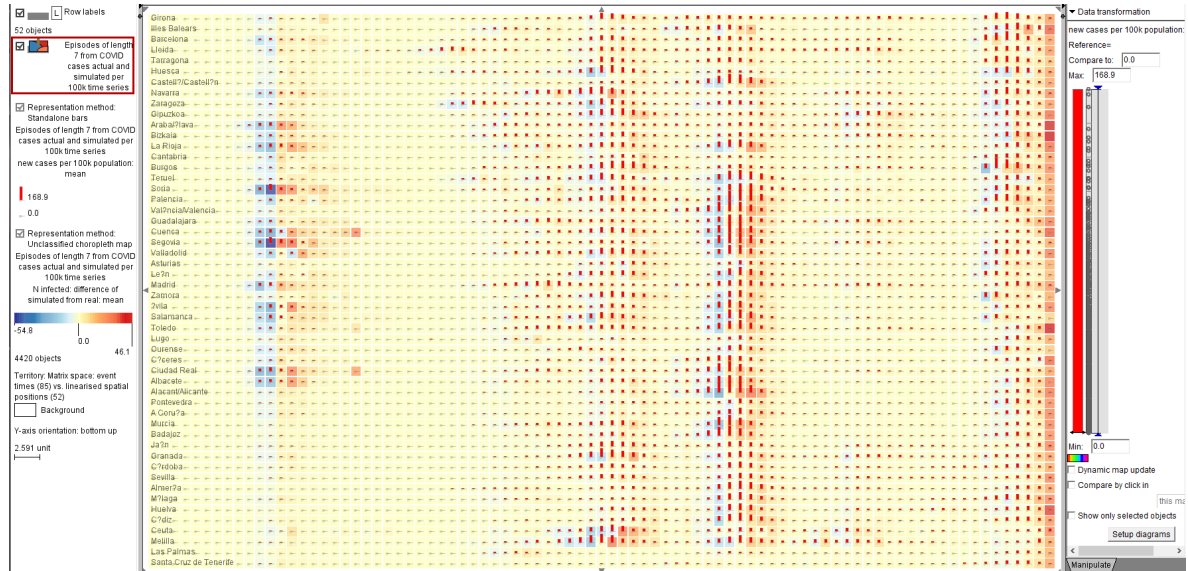


Figure 16: On top of the matrix representing the spatio-temporal distribution of the residuals by cell colouring, the bright red bars represent the actual numbers of the COVID19 new cases per 100K of the population.

Hence, it has to be concluded that the model is not very reliable regarding the timeliness of predicting pandemic increases, as we observed time lags between the actual and predicted values. If no better model is available, decision makers need to take the temporal lags into account when developing policies and strategies. It means that an increase of the pandemic should be expected to come 2-3 weeks earlier than predicted by the model.

4.2 Supporting comparative analysis for predictive models dealing with event-based data

In this section we focus on classification models dealing with data representing sequences of events. Unlike simulation models, which predict the development of some dynamic phenomenon over an extended time interval, here we consider models predicting the kind of a future event based on the kinds of events that occurred earlier. The possible kinds of events are specified as discrete *classes*. Hence, while the previous section was focusing on numeric continuous time series, here we deal with discrete categorical data. The motivation is to develop approaches for different types of possible model outputs.

An event can be defined as a discrete entity or happening that exists or takes place within a certain time interval $[t_1, t_2]$ and is associated with a particular type or category. Examples of events related to pandemics are rises and drops of morbidity, periods of restricted mobility, lockdowns, and recovery of normal mobility. An event sequence can be defined as a series of observed occurrences of various types of events linked to the same dynamic phenomenon unfolding over time. The events occurring in a particular temporal order reflect the evolution or progression of the phenomenon through identifiable stages or manifestations. For example, the progression of a pandemic can be represented as a sequence of pandemic events. The evolution of the population mobility behaviour during a pandemic can be represented as a

sequence of mobility events.

The modelling task we consider is to represent the interplay of two or more phenomena as manifested through respective event sequences, such that each phenomenon may influence and be influenced by others. An example is parallel sequences of pandemic and mobility events, where mobility events stand for changes in population behaviour in response to pandemic events, and these changes affect the types of the following pandemic events.

The research is done in the context of the CREXDATA Health Crisis Use Case (task T2.2). The modelling task is to predict the class of the next COVID19 event (i.e., the level of morbidity) based on preceding disease events and levels of population mobility. The event data consist of two parallel event sequences: one for disease events and another for mobility events. These parallel sequences are available for different geographic areas; in our case study, these are 52 provinces of Spain.

The event data were derived from original data having the form of continuous numeric time series. We used two openly available datasets prepared by Ponce-de-Leon et al. [44]: daily counts of new COVID19 cases for the 52 provinces of Spain covering the period from 01/01/2020 to 12/08/2021 [45] and daily counts of trips within and between the provinces covering the period from 14/02/2020 to 09/05/2021 [46]. The trip counts were estimated based on anonymised data containing locations of mobile phones. For our analysis and modeling, we used data from 17/02/2020 (Monday) to 09/05/2021 (Sunday), encompassing 64 full weeks. We transformed the disease case counts into counts per 100,000 province inhabitants using openly available population data [47]. The counts of trips within the provinces were normalised to the pre-pandemic levels still attained in the period from 14/02/2020 to 10/03/2020. To minimise the influence of the irrelevant weekly cycle in the data variation, we smoothed both sets of time series using a forward moving average with a sliding window of 7 days.

To obtain event sequences for using in the research, we then divided the continuous time series into segments called *episodes*. Each segment has the length of one full week from Monday till Sunday. We categorised the episodes into four levels of pandemic severity and four levels of mobility, using an interactive approach with feature-based spatial embedding, clustering, and kNN classification [4]. For this purpose, the value variation in the episodes was represented by summary features including the median, maximum, and trend line angle. The COVID19 levels were denoted as 'c1', 'c2', 'c3', and 'c4' and the mobility levels were labeled as 'm1', 'm2', 'm3', and 'm4', where 'm4' represents normal or near-normal mobility, and 'm1' indicates predominant staying at home. The definition of the event classes is illustrated in Figure 17.

After assigning the categories to the weekly episodes, each subsequence of consecutive episodes of the same category was combined in a single event. The event duration is determined by the number of concatenated episodes. In the result, we obtain for each province a sequence of events differing in duration, each event having a particular category. This transformation is done separately for the COVID19 data and for the mobility data. Hence, for each province, we get two distinct event sequences: COVID19 events and mobility events. This is illustrated in Figure 18. The time intervals of the events from the two sequences are not the same.

To visualise the resulting events, we use a kind of Gantt chart display shown in Figure 19. Fragments from this display were used in Figure 18. The horizontal dimension represents

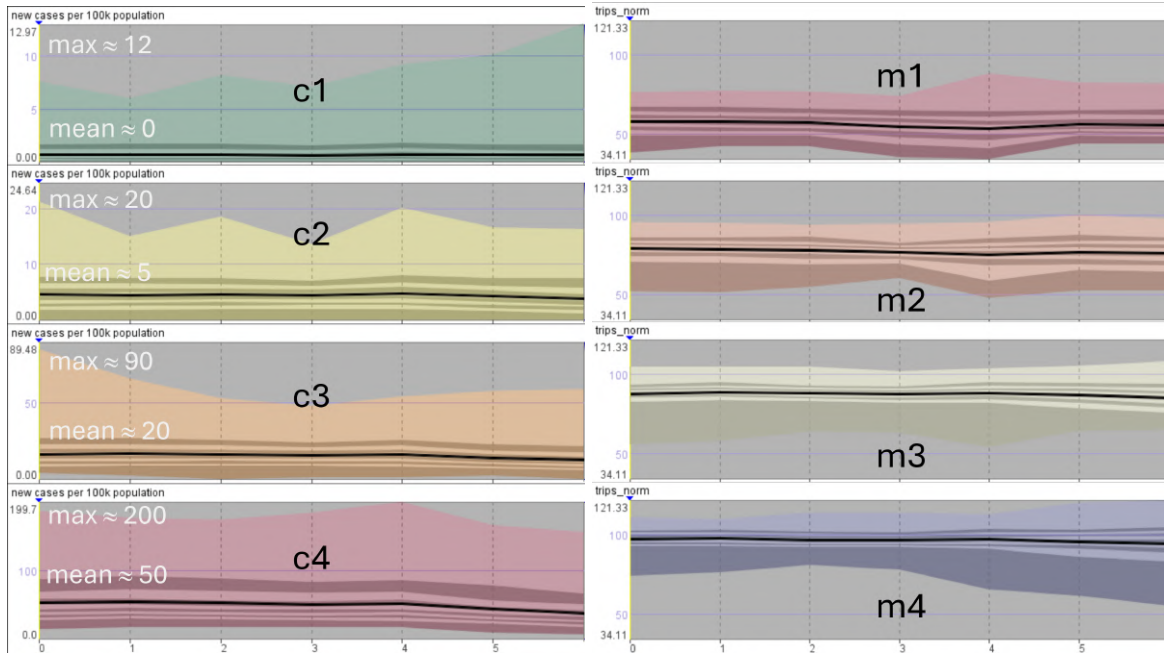


Figure 17: Definition of the categories of the pandemic and mobility levels for weekly episodes. The time graphs represent in an aggregated form the weekly variations of the numbers of COVID19 cases per 100K of population (left) and the levels of population mobility compared to the baseline (right). Each time graph corresponds to one category, denoted c1 (green) to c4 (red) for the COVID19 and m1 (blue) to m4 (red) for the mobility. Please note the different scales of the vertical axes in the graphs representing the COVID19 levels. For the mobility, the scale of the y-axes is the same in all graphs.

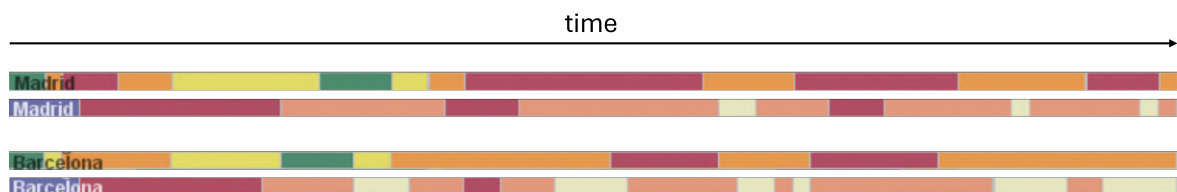


Figure 18: Examples of event sequences generated for the provinces of Madrid (top) and Barcelona (bottom). The horizontal axis represents time. The events are represented by consecutive horizontal bars coloured according to the event classes. The bar positions along the time axis correspond to the times of their occurrences, and the bar lengths are proportional to the event durations. The upper and lower bar sequences for Madrid and Barcelona represent, respectively, the sequences of the pandemic and mobility events.

time, and the rows correspond to the provinces. The display contains horizontal bars differing in the lengths. Each bar represents one event. Its horizontal position corresponds to the time when the event was occurring, and the length is proportional to the event duration. The bar colour encodes the event class.

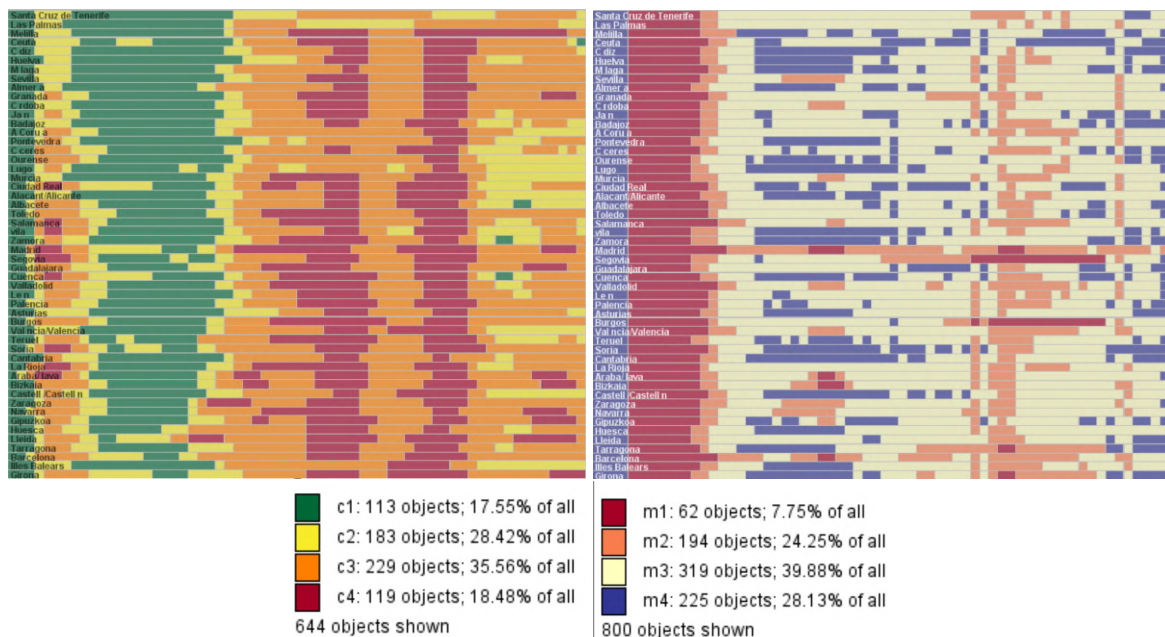


Figure 19: Two Gantt charts represent the distributions of the COVID and mobility events (left and right, respectively) over time and the Spanish provinces. Each bar represents an event, and its colour represents the event category. The horizontal axis of each display represents time, and the vertical dimension represents the set of provinces. The legends below the charts show the counts of the events of each class and their percentages in the whole sets of the pandemic events (left) and mobility events (right).

4.2.1 Predicting event based categories with classification models

We use the transformed data for developing a model predicting the category of the next COVID19 event based on the categories of both the COVID19 and mobility events that have taken place within several weeks preceding the given time. The motivation for the model development was to obtain a different type of model output than numeric time series in order to develop an approach to model explanation suitable for models with the categorical type of output.

By experimenting with different time intervals, we found that the events that happened in weeks from -5 to -2 with respect to the given week have the highest importance for predicting the category of the COVID19 event starting in the given week.

To assess the model performance, we provide a Gantt chart of the model predictions, which can be juxtaposed with the display of the real event categories, and a Gantt chart of the model errors, i.e., differences of the predicted categories from the actual ones. Figure 20 demonstrates the visualizations for one model variant.

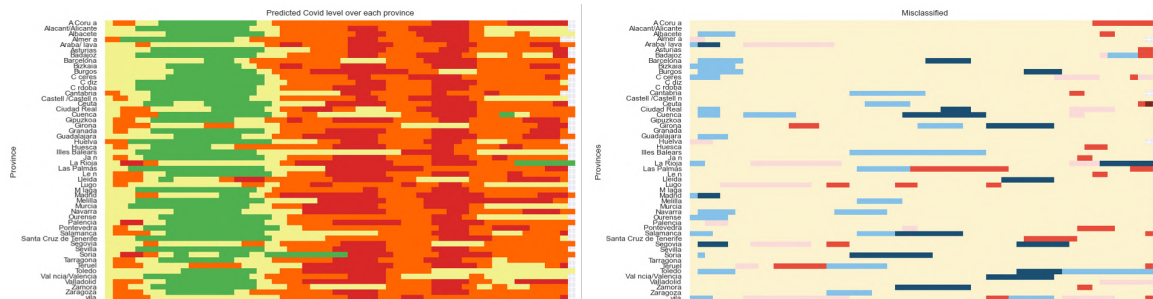


Figure 20: The Gantt chart on the left represents the predicted categories of the COVID19 events obtained from one model variant. On the right, the model errors are represented by shades of blue for underestimation and shades of red for overestimation.

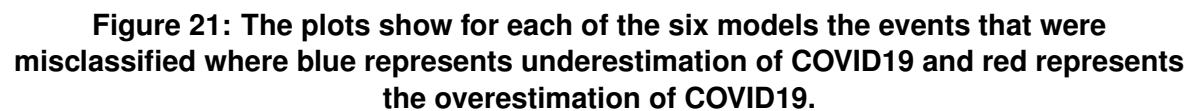
Following this, variants of the initial model were developed by using human identified patterns in the data to modify the data and features used. The base model (M1) uses a random forest model that takes in weeks -5 to -2 of Mobility data and -5 to -3 of COVID19 data as input. Weeks -2 and -1 of COVID19 were not used to account for the realistic lag of COVID19 data that may be available when making a prediction. Individual changes are made to this base model and re-evaluated. The changes include: the removal of the first 30 days which has unpredictable behaviour (M2), the addition of days elapsed since the start (M3), the addition of the duration of events (M4), the removal of all islands as these were seen as outliers (M5), and finally a combination of all these modifications (M6). Figure 21 shows how each change improves the number of misclassifications. It is evident that the combination of all these modifications aids to improve the overall classification (From an accuracy of 0.7124 for M1 to 0.8633 for M6).

Figure 22 then goes on to combine these misclassifications across all of the models which helps to visualise the times and provinces for which the models are more likely to be misclassified. Further possibilities for interactive comparative analysis of the predictions of the 6 models are illustrated in Figure 23 and Figure 24. The captions of these figures provide detailed explanation of their contents.

4.2.2 Explanation of event prediction with custom rules

Starting from the Random Forest classifier implemented in subsection 4.2.1, we focused on the task of providing an accessible explanation for the prediction of the class of a new instance. In particular, which part of the input was relevant for the ML model? We exploit LORE 2.1 to build the explanation in form of rule. We also focused on the custom implementation of our own neighbourhood generator using the new version of LORE presented in Section. 2.1. Our goal was to create a neighbourhood that was as close as possible to the original data to generate more meaningful rules, taking into account the internal dependencies of data.

LORE generates a set of rules that approximate the behavior of the black-box model in the vicinity of a specific instance. Figure 25 shows an overview of the process to extract the explanations for a black-box. The input is first encoded into an n -dimensional space using a pre-trained **encoder** (Figure 25). Then, around the projected point a **neighborhood**



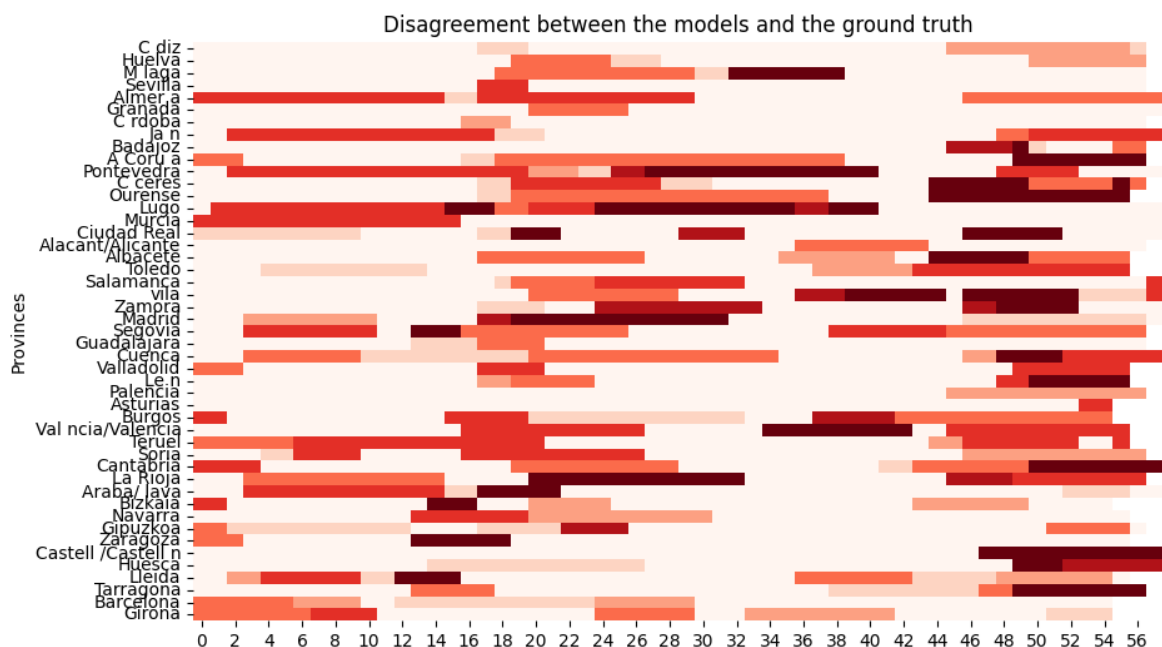


Figure 22: The plot shows the cumulative amount of disagreement from each model with the ground truth. The darker regions indicate the events that had a greater misclassification rate across the six models investigated.

generator is used to create synthetic instances that are similar to the instance to be explained. The synthetic instances are decoded and supervised by the black-box that assign them a label. This supervised dataset is used to train a local **surrogate** interpretable model, such as a decision tree. This transparent model is used to classify the local instance and the predicates selected along the descending of the tree are used to generate the rules. The counter-rules are generated by selecting variation of the input features to follow different paths in the tree that lead to different outcomes.

We implemented a customized Neighborhood generator to handle the internal dependencies and constraints of the data. Each event record has three sections: 1) attributes about temporal position; 2) classes of epidemic diffusion of covid; 3) classes of mobility activities. The neighborhood generation is based on a aperturbation of the given instance. When generating this perturbation, the values for COVID19 or Mobility levels should observe some constraints. For example, we should avoid abrupt changes from a low value of epidemics to the maximum. We should avoid spikes or holes in the sequence of classes, such as an empty cell in the middle of a sequence of high-incidence epidemics.

In the LORE pipeline, neighborhood generation is the step where we can inject the constraints defined by the specific application domain. As a baseline, we use one of the generators coming with the library: `Random Generator`, which creates neighbor instances by perturbing each value independently with an offset extracted from a Gaussian distribution. However, this procedure does not ensure that the constraints described above will be held.

Thus, we modeled our generator, exploiting transition probabilities customized for COVID and mobility features. The COVID (mobility) level at week -4 is more likely to influence

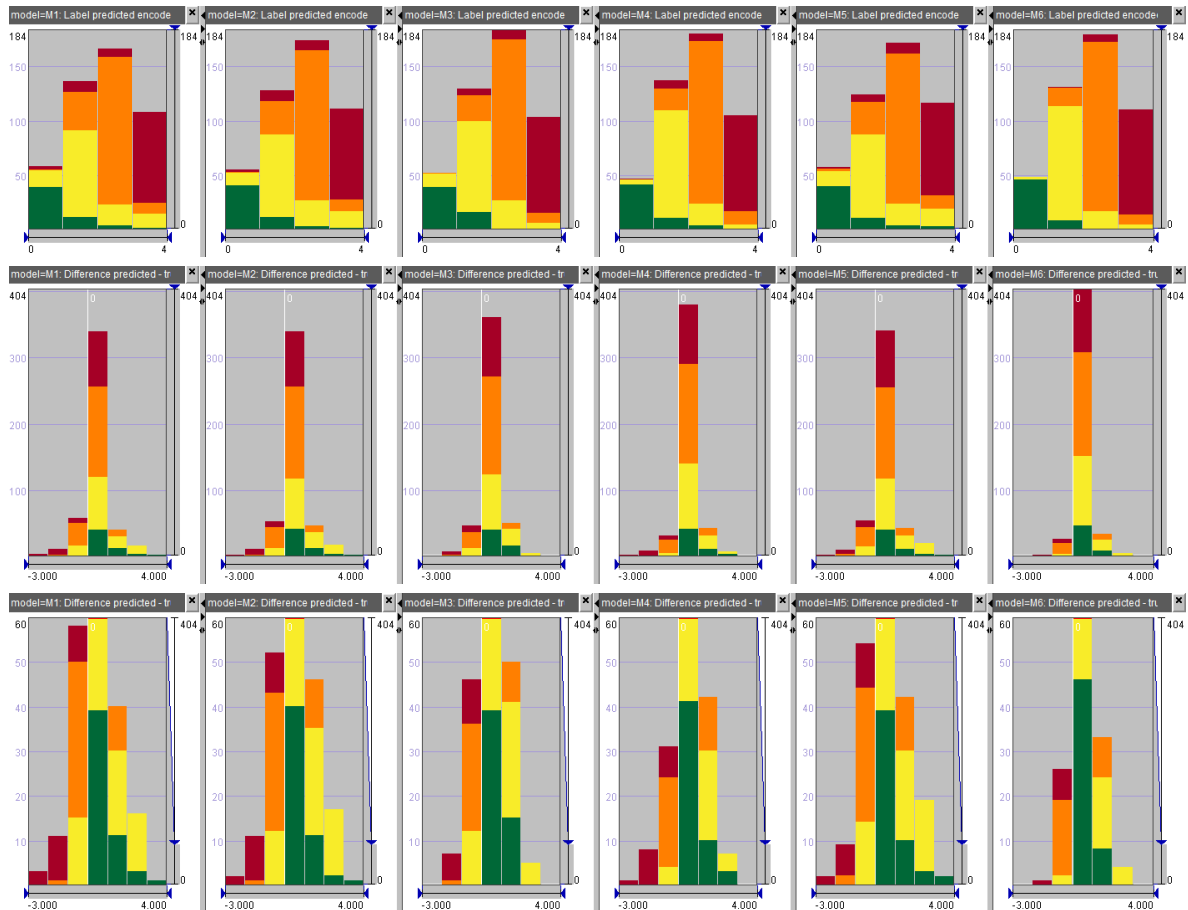


Figure 23: Exploration of the discrepancies between the predictions of 6 models by means of interactive histogram displays. In each row, the histograms correspond to the models from 1 to 6. In the top row, the bars of the histograms correspond to the COVID19 classes predicted by the respective models. The bars are divided into segments coloured according to the true COVID19 classes. In the second and third row, the bars correspond to the deviations of the predictions from the true classes ranging from -3 to 3. The bar in the centre corresponds to zero deviations, i.e., correct predictions. In the bottom row, the vertical axis of the histograms is trimmed to count = 60, and the bar length are extended to make the non-zero deviations better visible.



Figure 24: The projection plot presents the result of applying the UMAP dimensionality reduction method (distance metric: Hamming distance, N neighbours = 20, min distance = 0.1, random state = 1) to the predictions of the 6 models. The dot colours correspond to the true classes of the instances the dots represent. The clusters correspond to groups of instances for which all or almost all models give the same predictions. Separated dots correspond to instances that have been inconsistently classified by different models. In the clusters, the dots differing in colours from the prevalent colour of the cluster signify consistent misclassifications, i.e., multiple models tend to assign these instances to the same but wrong class. Clusters in the projection plot can be selected for detailed exploration.

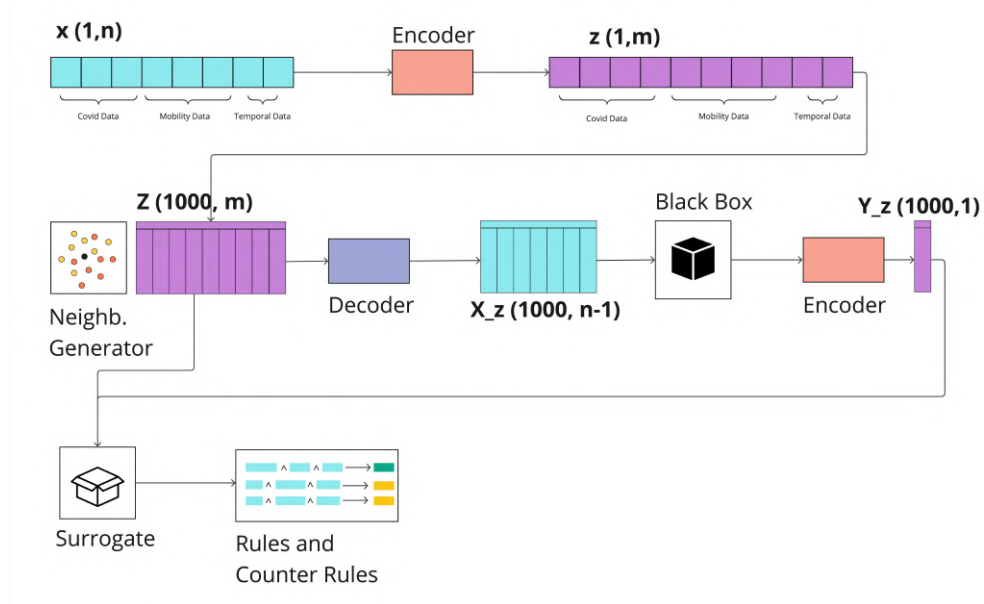


Figure 25: LORE explanation process. Starting with an instance x from the training dataset, the instance is first encoded to generate a neighborhood dataset consisting of m elements. This dataset is then decoded and classified by the black box model. Finally, the surrogate model takes the encoded neighborhood dataset Z and the prediction Y_Z as input to generate rules and counter-rules.

week -3. This type of observation is implemented as a transition function that decides the perturbed value of a feature by looking at its current value. The probability of transition changes according to the actual level of the attribute. For example, a high value of epidemics has a higher probability of yielding a higher value. And these transitions will be different for COVID or mobility values. Figure 26 shows two bipartite graphs for COVID (Left) and mobility (Right), representing the weighted probabilities for each starting state to the destination one. It can be noted that the generated data does not map directly to class 0.

For brevity, we present the formal description of the transition function for COVID, but a similar definition is straightforward for mobility. The transition function for COVID states is given by a matrix A with size 5×5 , where the element $a_{i,j}$ is the probability of perturbing the value c_i into the value c_j .

As a result, we compared analytically the two generators, in comparison with the training data we used to learn the model. Figure 27 shows the projection of the training data and the neighborhood generated by the two generators. It is evident how the custom neighborhood is much more compact and dense than the random one. This is a good sign that the custom generator is able to generate instances that are more similar to the training data.

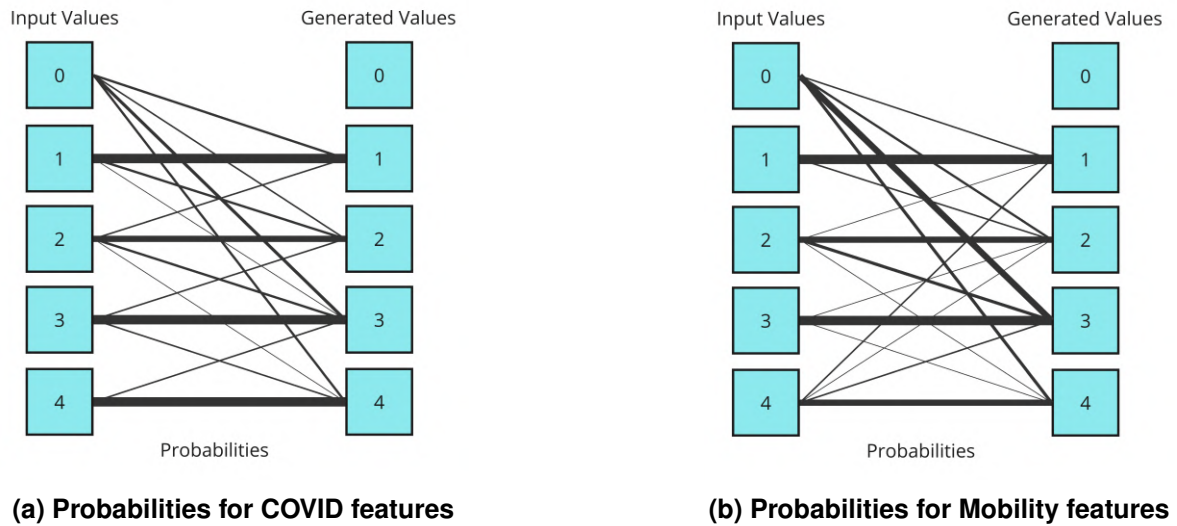


Figure 26: Bipartite graphs with the probabilities for COVID and Mobility, the thickness of the arc depends on the probability of a value to be perturbed into another value

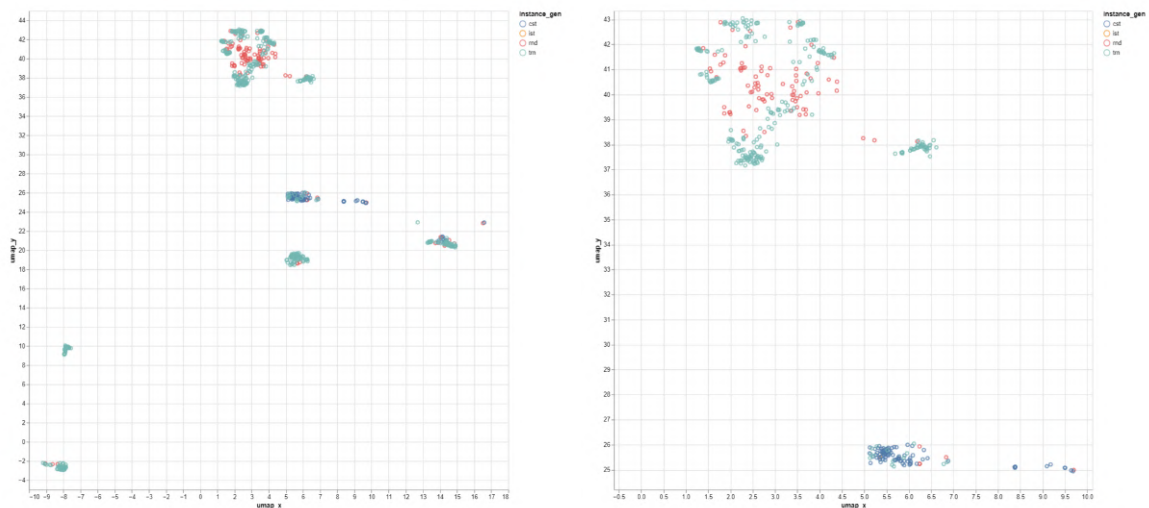


Figure 27: Comparison of the neighborhoods created by two generators. The multidimensional datasets are projected on a two-dimensional space for the visualization using UMAP algorithm. On the left the whole dataset. On the right a zoom around a selected instance used to generate the neighborhood. It is evident how the custom neighborhood is much compact and dense than the random one.

5 Augmented Reality at the Field

5.1 Weather emergency use case

5.1.1 Contributions

T5.4 introduces the architecture of a novel system for real-time urban flood management utilizing the capabilities of head-worn AR, integrating extreme scale data. Previous work has been limited to utilizing low-scale static data for visualization [51]. Our system's architecture visualizes multiple urban flood scenarios by consuming dynamic data, enhancing users' situational awareness through cutting-edge AR. According to the system architecture presented, the user wears a Hololens 2 AR headset, allowing for hands-free operation. Our system dynamically generates predicted flood level (water level, velocity and direction of water) on-site, in real-time, contributing to the planning of an efficient rescue. This represents a notable advancement over past work, where routing to avoid pluvial floods was primarily conducted within VR environments, focusing on simulation rather than operation on the field.

Our system components include:

- A novel AR-based system architecture for flood visualization that utilizes extreme scale and complex data analytics regarding critical urban infrastructure and weather forecasts, based on open data sources. Our system's architecture offers a depiction of urban flood inundation as well as water level, velocity and direction of floods among other user interface elements. Unlike VR that confines rescuers in immersive simulations in laboratories, our system encompasses on-site rescuers' intervention.
- An AR system architecture that integrates a first-person perspective for situational awareness improvement allowing users to experience seamless integration of the real-world enriched with digital elements based on spatial data such as water topography, urban networks such as manholes, flood risk of points in the horizon as well as visualization of scientific data.
- Enhanced decision-making of flood risks based on AR visualization but also awareness of real-world surroundings, integrating emergency response protocols and ensuring safe evacuation.

5.1.2 Motivation and Previous Work

Amidst the increasing frequency and severity of flooding incidents in recent years, emergency response and rescue operations face significant challenges. Navigation and risk assessment methods, in dynamically changing flood environments, are dependent on flood visualization methods, which should efficiently communicate risk, without amplifying hazards for first responders [11]. Modern digital visualization tools are increasingly vital in enhancing the management of flood risks in urban areas [30]. Rather than commonly used 2D visualization [59], prominent technologies, including virtual reality (VR), augmented reality (AR), and digital twin simulations, are extensively utilized to depict urban flood scenarios [43]. Among these, VR has gained extensive usage [8], [53] for informing on simulated scenarios, though without awareness of the real world. While research efforts focusing on urban flood visualization are dedicated to the phases of readiness and prevention, there is a noticeable gap in the

capabilities for real-time observation and handling of urban flooding events in AR [5]. The reliance on mobile phones or tablets in previous AR urban flood visualization has constrained on-site operations because the interaction is not hands-free [40].

In addressing the increasing frequency of global flooding events and the need for heightened public awareness, past work showcases AR as an effective medium for educating the public on local flood risks [58], [48]. Users are able to engage with potential flood levels in their local flood zones, by leveraging an AR app that offers in-situ modeling of basic 3D building prototypes (cuboids) along a riverside, facilitating the visualization of an augmented flood plane [24]. Notably, users can actively adjust the flood plane height, enhancing the interactive experience. A real-time prototype for mobile augmented reality (MAR) is put forward merging real-time building model updates, interactive flood visualization, and seamless integration with live sensor readings, including water level, humidity, and soil moisture, accessible through a network [23]. These sensor readings contribute to detailed real-time annotations. Feedback indicates the need for increased geometric model complexity to enhance operation on-site. A MAR coastal erosion 3D visualization system is put forward leveraging geographical data visualizing future shoreline changes due to coastal erosion [28]. The study shows that MAR remains operational under challenging situations, including outdoors when bright lights may interfere with the user's vision. Accurate registration of 3D segments with the real-world coastline is achievable and can be viewed seamlessly through a smartphone. However, the application's efficiency is affected during bad weather and sea waves, causing the 3D content to drift in the scene. MAR systems require the user to hold a mobile device which, while flooding is taking place, is restrictive. To address this issue, our system architecture provides 3D visualization utilizing head-worn AR, enabling users to operate hands-free, while forecasting of events is streamed to the AR user, in real-time.

Another approach for effective flood visualization, is combining AR with a 3D-printed terrain model [62]. The study explores adaptive flood data processing and hybridizing virtual flood and terrain models. The researchers simulate a barrier lake dam-break scenario, comparing a flood visualization placed on a 3D printed terrain model and one on a 3D digital terrain model. Results show improved flood hazard understanding when a 3D-printed model is involved. In our work, we offer a system architecture that includes real-time urban flood management, through head-worn AR on-site rather than in the control room, without the need for 3D printed models.

5.1.3 System Overview

T5.4 proposes the system architecture for AR-based flood visualization. An innovative approach is put forward for designing a system managing urban floods and conducting on-site rescue operations in real-time through the use of head-word AR, combined with large-scale data integration. Targeting to enhance real-time decision-making in flood scenarios, our system's architecture facilitates effective coordination and execution of rescue operations. Utilizing AR headsets, first responders and emergency management personnel will access and interact with real-time visual representations of urban floods, including water levels, affected areas, and safe routes. The system's integration with large-scale data analytics will allow the dynamic combination of weather forecasts, sensor networks, historical flood data and urban topography. The result of data processing is stored in Kafka topic(s). Subsequently, the Hololens 2 device connects to the corresponding topic through a client as a consumer

and receives the appropriate messages. Upon receipt of the message from the AR device, it undergoes processing to isolate the valuable information necessary for visualization. The AR environment is designed in a way that ensures vital information like flood level forecasting will be accessible without overwhelming the user or blocking users' field of view. The head-mounted AR device offers a hands-free experience, enabling users to remain fully engaged with their surroundings while receiving crucial data updates and navigational assistance.

5.1.4 Urban Management

Flooding is one of the serious natural climate concerns, that is intensified by climate change and can cause major economic, social, and environmental consequences. Urban flooding, which refers to the inundation of a densely populated area due to excess rainfall on a continuous and impervious stretch of land that mostly arises due to an overwhelming capacity of the drainage system and reduced infiltration rate, [17] is a major problem in many parts of the world. Urban flooding, which can originate from coastal, pluvial, or fluvial flooding (Figure 1), is the primary source of flood losses worldwide. Among the categories of flood hazards, pluvial flooding, which is brought on by excessive precipitation combined with insufficient stormwater infrastructure or restricted infiltration capacity, has traditionally received less attention as it is thought to be controlled and generally causes less harm. Nonetheless, data indicate that pluvial flooding is a major contributor to cumulative damage over time, and that hazard exposure changes, aging infrastructure, urbanization, and global warming are all increasing the likelihood of these catastrophes, while the necessary infrastructure to mitigate floods is lacking in most urban areas, rendering them acutely vulnerable [7]. The exposure of an urban area to flooding comprises the population, its uses and infrastructure, environmental and cultural assets, economic activities, and all the city's elements and facilities that cause changes in physical processes, socioeconomic growth, migration, and economic changes. The growing trend of urban flooding is a global phenomenon that has become an important field of study and provides a significant challenge, especially for modeling communities and urban planners.

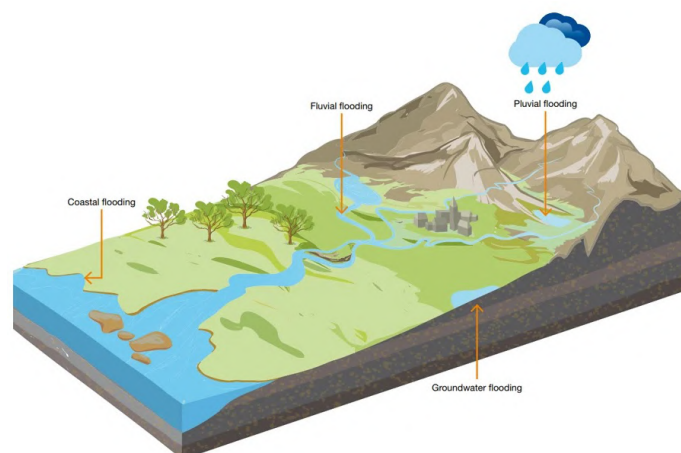


Figure 28: Types of urban flood

Flood risk reduction measures are categorized into structural and nonstructural measures. The structural measures are mainly major public projects that require moderate-to-major

planning and design efforts, while on the other hand, the nonstructural measures aim to improve urban planning and management. The nonstructural measures can be categorized into emergency planning and management, including warning, evacuation, preparedness, and flood insurance; speeding up recovery to increase resilience by enhancing building design and construction; and flood avoidance and reduction. The last category is related to land use planning as land use, public spaces, relocation, and forestation plans, and architectural planning as dry and wet floodproofing techniques, structural retrofitting or reinforcement, and facility maintenance and repair plans [1].

Effective flood risk management is related to urban planning factors, such as the adjustment of urban infrastructure, and land-use practices that can integrate sustainable drainage systems, create impermeable surfaces, preserve and restore natural floodplains, incorporate green infrastructure, propose limits in the construction of flood-prone areas, and formulate emergency response plans [12], [2]. Our proposed system architecture addresses the management of urban floods with the use of hydro-meteorological, topography, and urban data that ensure real-time rescue operation and coordination.

A 3D flood model is a dynamic visual rendering that considers multiple variables, along with collected and existing data points from various sources, to show the movement and behavior of floodwaters in a three-dimensional space — and how those floodwaters relate to the physical world around them. Geographic Information Systems (GIS)-based and remote sensor-based data models can integrate a broader range of information, combine different data types and support spatial visualization — providing actionable insights at critical junctures.

In T5.4 we have proceeded with the evaluation of vulnerable flood-prone areas in the cities of Innsbruck and Dortmund. The basic aim is to analyze areas that present high flooding risks because of their geo-morphological characteristics and concentrate public uses such as schools, museums, hospitals, etc. The hydrodynamic modeling process, automatically incorporates a range of different data and data sources as High-resolution terrain model, River depth data, e.g., from echosounder measurements, land use data, building models, vector data on hydrological catchments, river courses, lakes, culverts, soil maps to derive infiltration parameters, sewer network data (manholes, reaches) rainfall data, and water level data (water level and flow). The question that arose was what to present in an emergency case in a simple way which would make the rescuers procedures precise and fast. The realistic 3D visualization helps practitioners, who are not domain experts, to better estimate the scale as well as the impact of a flood. The main features that will be presented are:

- water level (grid points and nodes)
- flow (grid points, pumps, orifices, weirs, valves)
- velocity (grid points)

These elements are necessary to better describe the existing and future flood status. The basic idea is to create a system that informs users every 10 minutes about the changes in water height, velocity, and direction.

5.1.5 System Architecture

The system (Figure 29) that we developed has three main components: a Dockerized Kafka Server, an Intermediate Node Server, and an Augmented Reality App.

The Dockerized Kafka Server which was set up locally on our computer with specific ip, simulates the CREXDATA Kafka Server and is used to configure our application according to the Big Data logic. Furthermore, by following this approach, our application will be easily integrated with the Crexadata system with only minor adjustments when the time comes. We initialized the Kafka server container using the Zookeeper "wurstmeister/zookeeper" and Kafka "wurstmeister/kafka" images, assigning them to the corresponding ports. Inside the container, the Kafka cluster is accessible via the container port 9092. Accessing the Kafka Cluster inside the container, we created a custom topic that serves as the Complex Event Forecasting topic. From this custom topic, we receive the synthetic forecasting data and visualize it in an Augmented Reality Environment.

Initially, we used the Confluent Kafka client in order to consume messages from the topic and send them directly to the Unity3D app (not AR app). Despite our success, we soon discovered that the Confluent Kafka client is not compatible with UWP (Universal Windows Platform), the platform utilized by the Hololens 2. Thus, we were forced to change our architecture for two main reasons:

- UWP Incompatibility with Confluent Kafka Client.
- Limit the number of processes running on the Hololens 2 to prevent overheating and reduce the likelihood of frame drops. The processing power of the Hololens 2 device is limited to 4-GB LPDDR4x system DRAM, Qualcomm Snapdragon 850 Compute Platform.

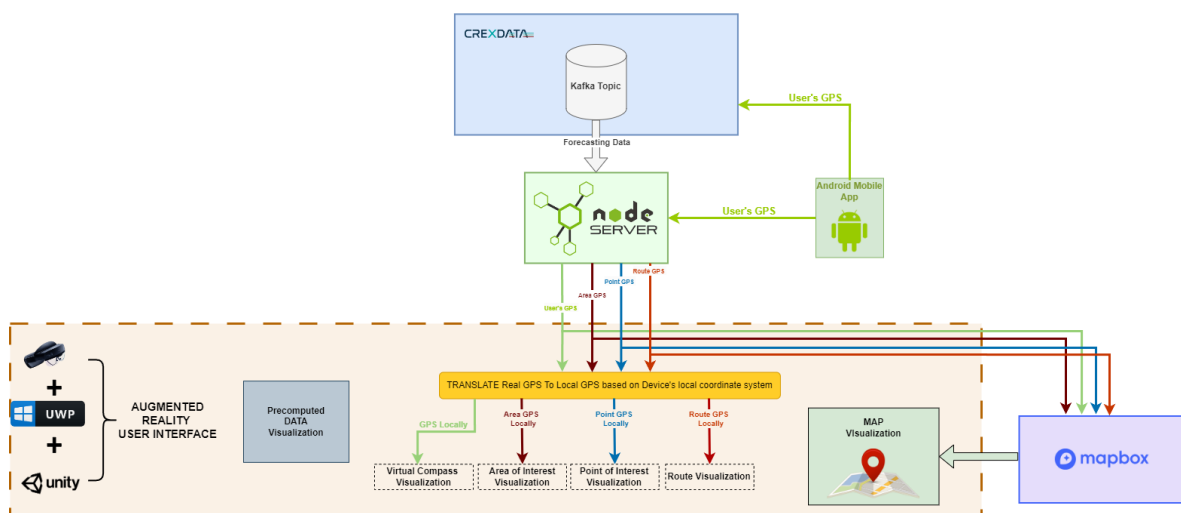


Figure 29: System Architecture

Therefore, we revised our initial architecture, which involved sending messages directly from the Kafka topic to the Augmented Reality app. Instead, we implemented an Intermediate Server to receive messages from the topic, process the data, and subsequently transmit the processed data to the corresponding Augmented Reality device(s).

For that purpose, we decided to deploy Node.js, which is an asynchronous event-driven JavaScript runtime that was designed to build scalable network applications. Thus the reasons that drove us to use Node.js are:

- **Asynchronous Event-Driven Architecture:** Node.js, with its proficiency in managing asynchronous operations, is well-suited for real-time applications, offering efficient handling of multiple connections, which proves crucial for managing data flow between Unity and Kafka in an Asynchronous Event-Driven Architecture.
- **Vast Ecosystem:** Utilizing Node.js's rich ecosystem, which includes a broad array of libraries and modules, such as KafkaJS for modern Apache Kafka client support and the ws library for WebSocket functionality, enhances real-time communication capabilities and simplifies interaction with Kafka topics.
- **Community Support and Documentation:** Node.js benefits from robust community support and comprehensive documentation, providing developers with valuable resources and assistance for building and maintaining projects effectively.

To meet project requirements, we need to transmit data from the Kafka topic to the Hololens 2 device in real time, precisely when new data becomes available. For that reason, we have to transmit the data a) from the Kafka topic to the Node server for internal processing and b) transmit the processed data to the Hololens 2 device(s)

To transmit data from a Kafka topic to the Node server, as the a step implies, we utilized the Kafka.js client. Kafka.js uses a custom binary TCP-based protocol specific to Apache Kafka, designed for efficiency and high performance in large-scale, real-time message-passing environments.

To transmit data between the Node server and Hololens 2 devices as the b step implies, we utilized WebSockets, a duplex protocol primarily used for client-server communication which is ideal for real-time and fast data transmission. On the Node server side, we used the ws library, and on the client side, we employed the websocket-sharp library, which is compatible with Hololens 2 and the UWP environment. The AR system architecture is analyzed in: *Stavroulakis, A., Dimelli, D., Roumeliotis, M. and Mania, A. (2024). An Augmented Reality System Architecture for Flood Management. Angers, Proceedings of International Conference on Geographical Information Systems Theory, Applications and Management Conference (GISTAM 2024) [56]*

5.1.6 Server Function

The server is attempting to establish connections between the Kafka Cluster and the HoloLens 2 client. Once connections are successfully established on both sides, the Node server informs the HoloLens 2 client via a message that the system is ready to process messages. The client then sends its unique ID back to the Node server and waits for messages. The Node server creates a new consumer and a corresponding consumer group named after the HoloLens 2 device's ID. Thus, each consumer group will have only one consumer, corresponding to the specific HoloLens 2 client ID.

Message consumption from the topic is configured to start from the oldest message, without assigning an offset. This means that if the consumer (or the members of that consuming

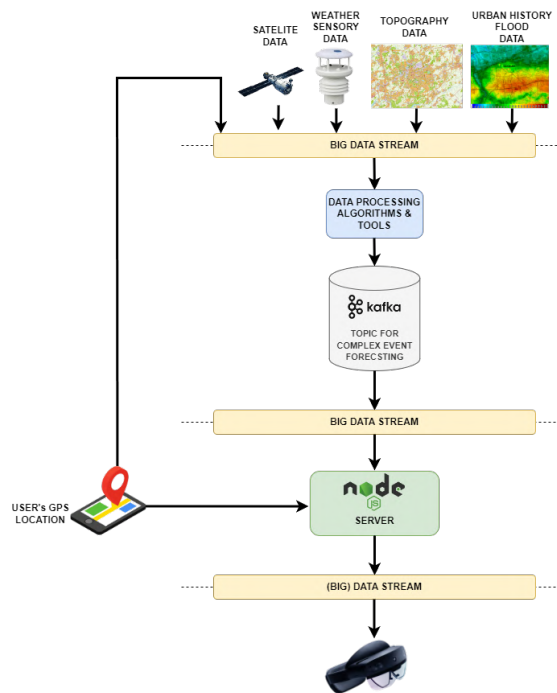


Figure 30: Dataflow Diagram

group) disconnects and reconnects, it will not resume from the last consumed message but again from the start.

On the other hand, the Node server sends the actual message along with the offset number from the topic to the HoloLens 2 client. We did this because there are two cases:

1. The connection between Node Server and HoloLens 2-Client has been lost
2. The HoloLens 2 shut's down

In both cases, the consumer group to which the client belonged continues to exist. It only gets erased if the topic is deleted. Thus, if the HoloLens 2 shuts down and reconnects to the server, it will be reassigned as a consumer in the existing group named after its ID. If the offset was in autocommit mode, consumption would start from the last committed offset before the HoloLens 2 shut down, meaning that all previously received information would be lost.

To address this, we configured the system so that after a shutdown or connection loss, the client tries to reestablish the WebSocket connection with the Node server. When it does, it sends a message informing the server of the last consumed offset. If the connection is merely lost, message consumption must continue from the last message, meaning the HoloLens 2 sends the last offset number back to the Node server. If there was a complete shutdown, consumption starts from the beginning, meaning the client sends back offset number 0.

On the HoloLens 2 side, we store the last consumed offset number to send back in case of reconnection. For a new connection, the client sends back offset number 0.

5.1.7 GPS Acquisition

The HoloLens 2 device does not have a built-in GPS system. To address this, we developed an Android app that provides the Node server via websocket with the GPS location of the HoloLens 2 user. This solution is intended for experimental purposes only and substitutes for the lack of an actual GPS system on the HoloLens 2.

5.1.8 AR User Interface

When the HoloLens 2 device starts, it initializes the virtual world in the Unity3D app, oriented to the user's initial position. Thus, the virtual and real worlds must be aligned. To achieve this alignment, the user must undergo a calibration process. The calibration process is as follows: The user must be oriented to true north when the application starts, using a compass. For convenience, a virtual compass is provided by our Android application. The calibration process can be initiated at any time the user wants through the main menu.

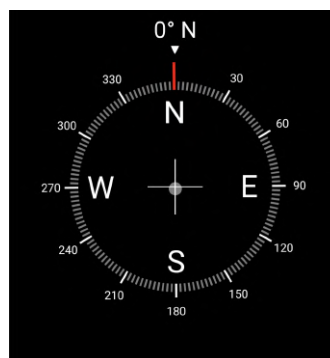


Figure 31: Compass

In the context of rescue operations, the design and development of user interfaces play a pivotal role in ensuring the effectiveness and efficiency of response efforts. For rescuers, having interfaces that are intuitively understandable, easy to use, and unobtrusive to their field of view is imperative. Firstly, in emergency situations where every second is critical, intuitive interfaces enable quick access to vital information without the need for extensive training or complex navigation. This streamlined access facilitates efficient decision-making processes, potentially saving lives in the process. Moreover, clear and unobtrusive interfaces help reduce the cognitive load on rescuers, allowing them to focus more on the task at hand rather than struggling with the interface. Maintaining situational awareness is paramount during rescue operations, and interfaces that do not obstruct the field of view enable rescuers to remain vigilant of their surroundings and adapt to changing conditions effectively. Additionally, seamless communication and coordination among team members are essential for successful rescue missions. Intuitive interfaces facilitate collaborative efforts by ensuring easy access to shared information, enhancing overall coordination and response effectiveness. Lastly, in high-stress environments, where adrenaline can impair cognitive functions, interfaces that are easy to use even under pressure become invaluable tools for rescuers, enabling them to remain effective and responsive in the face of adversity.

Menus: The main menu has two layers:

1. The first layer appears when one of the palms is facing the user, utilizing the hand tracking feature of the HoloLens 2. It has six toggles and one calibration button. Each toggle, by turning it on or off, enables or disables the corresponding virtual objects as implied by their names from the user's field of view.
2. The calibration button, when pressed, opens the second menu layer. This layer is a panel that contains a calibration button in case the user wants to recalibrate the HoloLens 2 and also provides exact instructions that the user should follow for the calibration process

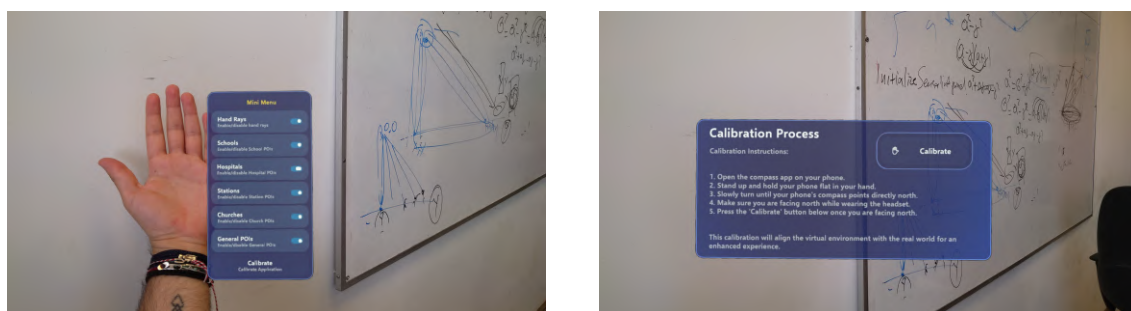


Figure 32: Menus

Points Of Interest (POI): The Point of Interest (POI) visualization feature in our AR application enhances situational awareness by providing real-time 3D representations of key locations such as schools, hospitals, churches, stations, and general points of interest. This feature leverages GPS coordinates to accurately place POIs within the Unity environment, ensuring that emergency responders can easily identify critical locations during a crisis. Every category of POIs has a unique design that allows the user to intuitively understand what a POI is by its icon. Below, we present the visual concept for each of the POIs that are significant for the weather emergency use case.

There are two major categories of POIs:

1. **Static POIs**
2. **Dynamic POIs**

Static POIs are locations that remain unchanged during a weather emergency and are known beforehand, meaning their GPS locations have been preloaded into the app. In our case, the static POIs are crowded buildings or areas such as schools, hospitals, churches, and stations, which are prioritized by the rescuers.

Dynamic POIs are not known at the start and their locations are determined during the event. The acquisition of the data follows the Kafka Cluster - Node Server - HoloLens 2 pipeline. One case of dynamic POI is the GPS location of the other rescuers of the team. Our app is designed to handle multiple users simultaneously through the Node server. Each user sends their GPS data from the mobile app to the Node server via WebSocket. The Node server then broadcasts every user's GPS location to all other end-users. It is crucial for each member of the rescue team to know each other's location in the field. We have managed



Figure 33: AR POIs on Real World



Figure 34: (a) General POI (b) Church POI (c) Hospital POI (d) School POI (e) Station POI

to broadcast every user's GPS location to other team members, but the visualization is still under development

For every GPS location to be visualized in an Augmented Reality environment in the correct location and distance from the user, it must undergo a conversion process. More specifically, the real Latitude and Longitude must be convert into Augmented Reality Δ Latitude and Δ Longitude as follows:

The Δ Latitude (Δ lat) is calculated every frame by taking the difference between the latitude of the target location and the latitude of the reference location which in our case is the GPS location of the user, multiplying this difference by conversion factor 111,000. This conversion factor is based on the fact that each degree of latitude corresponds to approximately 111 kilometers, or 111000 meters. This approximation holds true because the distance represented by a degree of latitude is nearly constant everywhere on Earth. Therefore, the formula Δ lat=(latitude - usersLatitude) \times 111000 converts the latitude difference into meters.

The Δ Longitude (Δ lon) is calculated by taking the difference between the longitude of the target location and the longitude of the reference location, which in our case is the GPS location of the user, then multiplying this difference by a factor that accounts for the Earth's curvature at the given latitude. Unlike latitude, the distance covered by a degree of longitude

varies depending on the latitude, decreasing as you move away from the equator. This is because the Earth is a sphere, so the lines of longitude converge at the poles. To convert longitude degrees into meters, the formula includes the cosine of the reference latitude (in radians). Specifically, $\Delta\text{Lon} = (\text{longitude} - \text{myLongitude}) \times (111000 \times (\cos(\text{myLatitude} \times \pi/180)))$. Here, 111000 is the approximate length of a degree of longitude at the equator in meters, and $\cos(\text{myLatitude} \times \pi/180)$ adjusts this length based on the reference latitude, converting the longitude difference into meters.

We enhanced the previously developed algorithm by implementing it as a script in C#, which executes at every frame to accurately position the AR object relative to the user's location.

Weather Emergency Map: Our Hololens 2 application utilizes the MapBox Unity SDK to create an interactive and real-time 2D map experience. This integration is crucial for providing spatial context and navigation assistance to the user, enhancing the overall effectiveness of the application.

Key Components and Functionality: MapBox Unity SDK:

The MapBox Unity SDK is employed to render a high-quality 2D map within the Hololens environment. The SDK offers comprehensive tools for managing map tiles, layers, and geographical calculations, ensuring accurate and responsive map displays. Real-Time GPS Integration:

The center and content of the map are dynamically updated based on real-time GPS coordinates received from an Android application. The Android app sends the user's GPS location to a central server, which then transmits this data to the Hololens application. By constantly updating the map center with the latest GPS coordinates, the map remains aligned with the user's current location. User Representation on the Map:

A graphical representation of the user, such as a small figure or icon, is displayed on the map. The position of this user icon is continually updated based on the GPS data from the Android app. Utilizing MapBox's latitude and longitude capabilities, the icon's placement on the map is both precise and dynamic, reflecting the user's real-world movements. Visualization of Points of Interest (POIs):

The map is being enhanced to include the visualization of Points of Interest (POIs). These POIs will be displayed on both the 2D map and the 3D world view within the Hololens. Each POI is positioned using its geographical coordinates, allowing for accurate representation on the map. This dual visualization strategy helps users navigate more effectively by providing relevant spatial context in both augmented reality and 2D map formats. Technical Implementation Details:

Map Centering: The map's center point is updated using the latest GPS coordinates to ensure it accurately reflects the user's current location. User Icon Placement: The user icon's position on the map is calculated based on real-time GPS data, ensuring it moves in sync with the user's physical movements. POI Visualization: POIs are integrated into the map as markers, with their positions determined by their specific latitude and longitude coordinates. Future Enhancements:

Future developments will include real-time updates and interactions with the POIs, providing

users with detailed information about each point of interest directly within the HoloLens interface. Additional features such as route planning and navigation guidance will be incorporated to further assist users in spatial orientation and decision-making processes. By leveraging the capabilities of the MapBox Unity SDK and integrating real-time GPS data, our application provides a comprehensive and interactive mapping solution within the HoloLens environment. This setup not only enhances user navigation but also improves the overall user experience by seamlessly blending digital map information with the physical world.

1. Through the app by utilizing the MapBox toolkit. Directly on the site by checking or unchecking the POIs you want to visualize
2. The map is centralized around the user.



Figure 35: Map for Weather Emergency AR Visualization

5.1.9 Water Visualization Logic and Implementation

The water visualization component of our AR application is designed to aid emergency responders by providing an intuitive, real-time 3D representation of predicted flood levels. The system leverages the capabilities of Microsoft HoloLens 2 and is developed using Unity and the Mixed Reality Toolkit (MRTK).

Core Components and Workflow

Ground Height Calculation

The first step in the water visualization process is to accurately determine the ground height relative to the user's position. This is achieved by casting a downward ray from the user's head (the main camera in HoloLens 2). When this ray intersects with the ground or any surface tagged as "ground," the point of intersection is recorded. This point's Y-coordinate provides the ground height, serving as a reference for positioning the water plane accurately.

User Interaction and Input

The user interacts with the system through a slider and a toggle button. The slider allows the user to select different forecast hours, which correspond to different predicted water levels. The toggle button enables or disables the water visualization. When the button is pressed, the system either initiates or halts the visualization process.

Environmental Scanning

Upon enabling the water visualization, the ARMeshManager component starts scanning the environment. This scanning process generates a mesh of the surroundings, which is essential for understanding the spatial layout and ensuring that the water plane interacts correctly with the environment. The system displays an instruction panel to guide the user to look at the ground during this scanning period, ensuring a reliable ground height calculation.

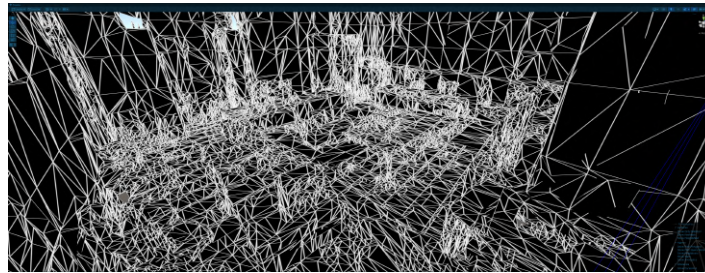


Figure 36: Scanned Area

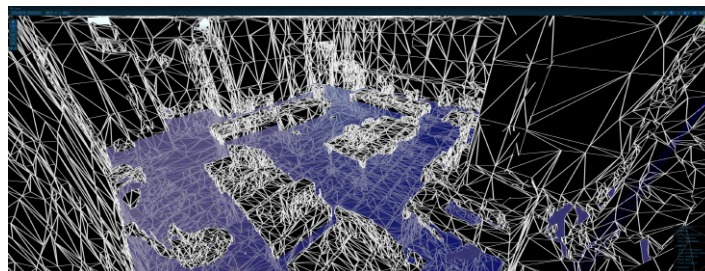


Figure 37: Scanned Area with Water

Water Plane Placement

After the scanning period, the water plane is instantiated at the correct position. The height of the water plane is determined by adding the calculated ground height to a predefined water level. This predefined level is selected based on the current slider value, which represents the forecasted flood height for a specific time. The water plane is then positioned at this adjusted height, ensuring it accurately reflects the predicted flood level.

Data Acquisition from Kafka

The forecast data necessary for determining the water levels are acquired from a Kafka topic. This data is processed through a custom Node.js server, which acts as an intermediary.

The server subscribes to the Kafka topic, retrieves the forecast data, and decrypts it before sending it to the AR application. This ensures that the water levels displayed in the AR environment are based on the latest and most accurate forecast information available.

Water Shader for Realistic Visualization

To enhance the realism of the flood visualization, a custom water shader was created using Unity's Universal Render Pipeline (URP). This shader simulates water properties such as transparency, reflection, and movement, providing a visually convincing representation of water. The shader's parameters can be adjusted to reflect different water conditions, such as calm or turbulent water, adding to the immersive experience.

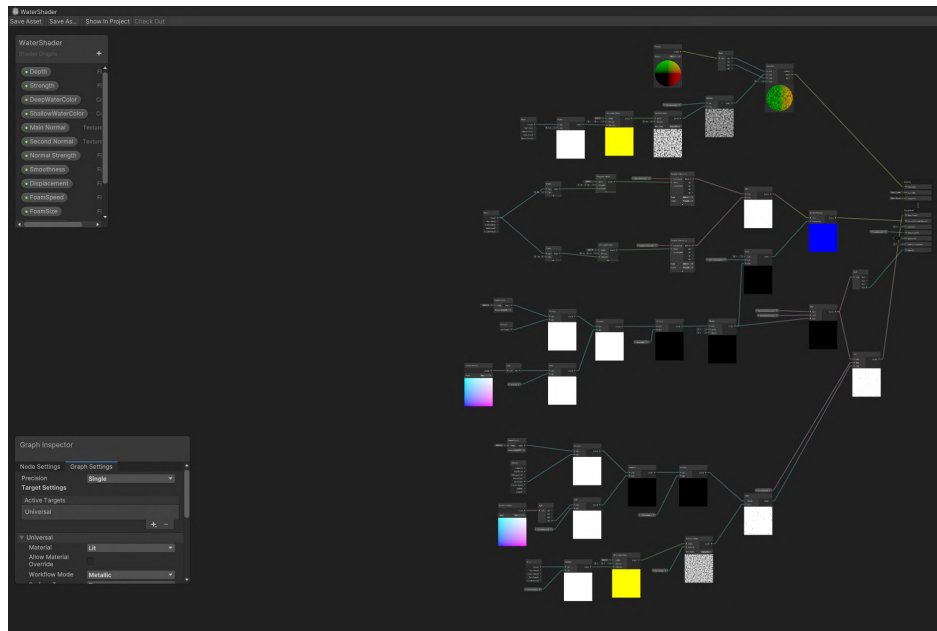


Figure 38: Water Shader Graph

Dynamic Water Level Adjustment

The water level can be dynamically adjusted based on user input. As the user moves the slider to different values, the system recalculates the appropriate height for the water plane and updates its position in real-time. This feature allows users to visualize different flood scenarios seamlessly, enhancing their situational awareness and decision-making capabilities.

Performance and Cleanup

To ensure optimal performance, the system includes mechanisms to manage and clear the scanned meshes when they are no longer needed. When the water visualization is disabled, all previously generated meshes are destroyed, and the ARMeshManager is turned off. This cleanup process helps maintain the application's performance and prevents unnecessary resource usage.



Figure 39: Outdoors Flood AR Concept 1



Figure 40: Outdoors Flood AR Concept 2

Physiological Monitoring of Rescuers A prototype device mounted on the Hololens 2 device will be able to capture physiological data of rescuers. An AR user interface is now placed on the rescuer's arm, showcasing vital signs. The goal of this AR user interface is to detect the physical state of the rescuer and communicate distress to control.

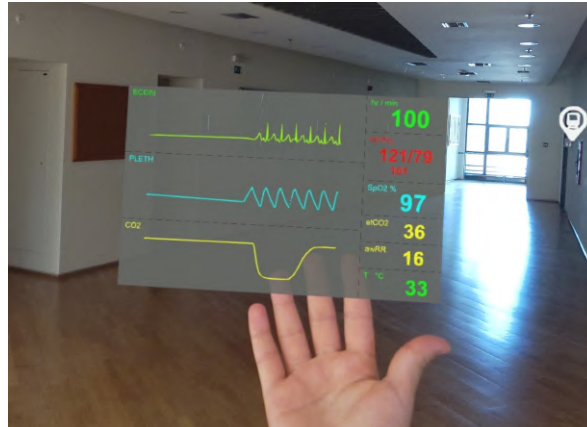


Figure 41: Biosignals on the Field Concept

5.1.10 Weather Emergency Use Case: Plans for the next 18 months

For the next 18 months of CREX DATA, these are the technical plans for AR system development for the Weather Emergency Use Case:

- Incorporate data and optimize functionality based on the evaluation testing of AR prototype for flood management, while in Dortmund and Innsbruck, in June 2024
- Implement additional functionality for gaze-based interaction
- Experiment with real-time streaming of data, connecting to the forecasting architecture
- Implement real-time physiological data monitoring of rescuers while on the field.

5.2 Maritime use case

In this work, we explore the development of an augmented reality (AR) application for the HoloLens 2, designed to assist ship captains in avoiding collisions and navigating hazardous weather conditions. The application leverages smart user interfaces (UIs) to provide real-time navigational guidance using predictive data generated by a collaborating team. Our role focuses on the representation of this data within the AR environment to enhance situational awareness and decision-making for maritime navigation.

5.2.1 Architecture

Overview The architecture of our augmented reality (AR) application for the HoloLens 2 is meticulously designed to provide ship captains with real-time navigational guidance, enhancing their ability to avoid collisions and navigate through hazardous weather conditions. This section delves into the data flow from our project partner, Marine Traffic, through various processing stages, and finally to the display on the HoloLens 2 device.

Data Flow and Integration The process begins with data acquisition from Marine Traffic, which supplies critical information for ship collision and weather avoidance. This data is encapsulated in a `.json` file and includes navigational parameters such as the MMSI (Maritime

Mobile Service Identity) of vessels, target speeds, courses, coordinates, collision points, and more. Marine Traffic provided a test file that allowed us to understand the exact data format and content.

At the moment, for our tests we use this test file but we have already made the infrastructure to receive data from a Kafka server. Kafka acts as a reliable data pipeline, facilitating the seamless transfer of large volumes of data with minimal latency. The data from the Kafka server is then retrieved by a Node server, which we designed and implemented specifically for this purpose. The Node server processes the incoming data, preparing it for transmission to the end devices.

Since we do not have direct access to the ship's instruments to obtain actual location data, we developed a mobile application to capture the ship's current GPS location. This mobile app sends the GPS location through a WebSocket connection back to the Node server. The Node server then integrates this real-time location data with the collision and weather data received from Marine Traffic.

The combined data is then transmitted to the HoloLens 2 device. The HoloLens 2 leverages smart user interfaces (UIs) to display real-time navigational information, which aids ship captains in making informed decisions to avoid collisions and navigate safely. This AR presentation enhances situational awareness and decision-making processes.

Data Format The `.json` file provided by Marine Traffic includes the following information:

- **MMSI of vessel 1 and 2:** Unique identifiers for each vessel.
- **Target speed for vessels 1 and 2:** Defined by the collision avoidance algorithm.
- **Course for vessels 1 and 2:** Directional path based on the collision avoidance algorithm.
- **Current coordinates (longitude and latitude) for vessels 1 and 2:** Real-time location data.
- **Linestring for collision avoidance path:** Detailed path solution including longitude, latitude, and time.
- **Collision point coordinates:** Longitude and latitude of potential collision point.
- **Current time and collision event ID:** Timestamp and unique identifier for the collision event.

Data Processing Script A script has been developed to extract and format the relevant data from the `.json` file, ensuring it is in a usable format for the HoloLens 2. This preprocessing step is crucial for maintaining data integrity and ensuring accurate AR representation.

5.2.2 User Experience and Implementation

Overview: Navigational safety and efficiency are paramount in maritime operations, especially under adverse conditions. Our AR application leverages augmented reality technology to present critical information within the ship's bridge, assisting the crew with navigation and

collision avoidance. Drawing upon principles from [42] the system enhances the Open Bridge UI architecture to meet the unique demands of maritime navigation.

Methodology: The design of the AR elements in our application is categorized into five types as per the Augmented Open Bridge framework: App Display, Widget Display, AR Map, Annotations, and Ocean Overlay. These elements are strategically placed within defined information zones on the bridge to optimize visibility and utility, minimizing obstruction of the crew’s view of critical navigational areas.



	 App display	 Widget display	 AR map	 Annotation	 Ocean overlay
Arctic specific 	Convoy app Voyage planning tool AR map	Ice pressure indicator	Satellite/drone images Ice forecast charts	Ice sheets Ice bergs Snowmobile	Ice status overlay Ice 3D mesh overlay
General 	ECDIS Radar Conning Tasks / checklist Echosounder DP	Compass Heading & course Thrusters Speed Machine power Wind	Land Depth/safety contours Planned track Cross track distance Waypoints Vessels	Vessels Waypoints People Animals Shallow water Helicopter / drone	Drawing/annotation Safety contour Depth Planned track Cross track distance Other vessel tracks

Figure 42: AR Elements

AR Element Types and Information Zones: AR elements are crafted to provide varying levels of information, from comprehensive application displays to simple annotations linked to physical objects in the real world. The placement of these elements follows a structured approach within four information zones:

- **Sky Band:** Displays information such as weather data and navigational warnings.
- **Horizon Band:** Provides a panoramic view of navigational paths and nearby vessels.
- **Water Surface:** Highlights critical data like collision avoidance paths and current vessel positions.
- **Masked Areas:** Ensures essential controls and displays are not obstructed by AR elements.

This zoning approach ensures that information is presented clearly without overwhelming the crew, enhancing situational awareness and decision-making.

AR Zones and User Interaction: The bridge is divided into zones that dictate the behavior and accessibility of AR components, catering to the needs of users in different areas. This zoning approach, combined with advancements in AR technology, allows for dynamic interaction with virtual elements, significantly extending the capabilities of the original Augmenting OpenBridge framework. Users can interact with these elements through gestures, voice commands, and other input methods supported by the HoloLens 2, providing an intuitive and efficient user experience.

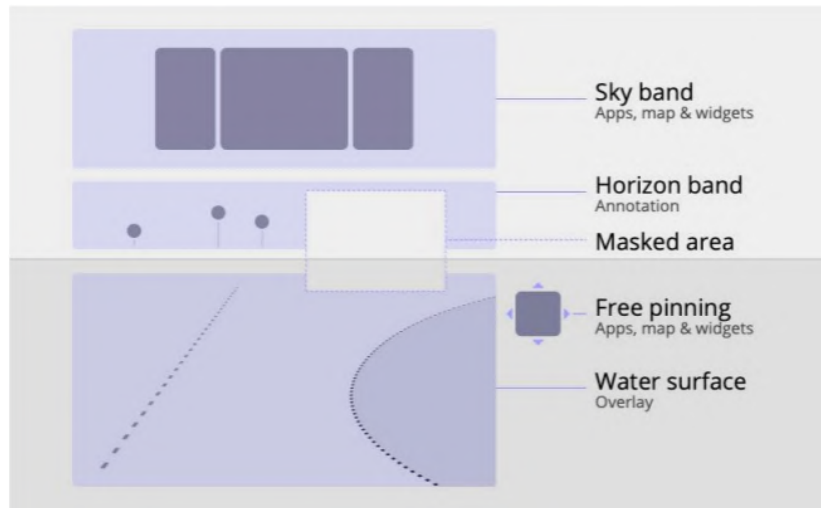


Figure 43: Suggested information areas and the types of application components they may contain

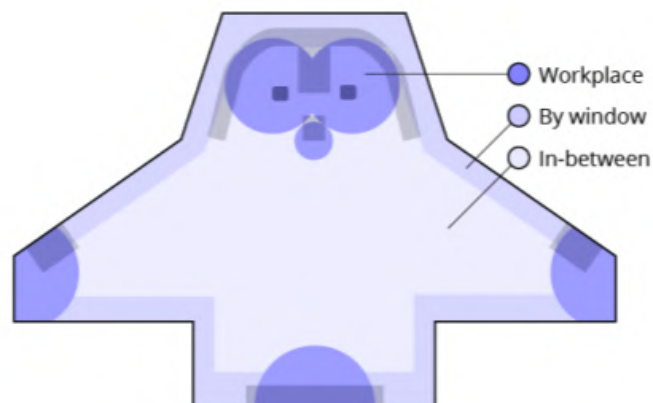


Figure 44: Dividing the bridge into three types of zones that affect the behaviour of the AR application components.

5.2.3 HoloLens Hardware and Usage

HoloLens Orientation Awareness: We need to know where the horizon is to divide the water surface zone from the horizon band.

Flat World Assumption

Initialization of Orientation: Upon launching the application, the HoloLens initializes its starting point (0,0,0) using its gyroscope and other orientation sensors. This initialization ensures that the forward direction of the HoloLens is perpendicular to the ground, irrespective of the device's initial orientation. This is crucial for establishing a consistent frame of reference.

Setting the Forward Direction: In a flat world, where the horizon is at eye level, the forward direction after initialization directly corresponds to the horizon. The HoloLens' sensors ensure that the forward vector is parallel to the ground plane, meaning that looking straight ahead naturally aligns with the horizon.

Locating the Horizon in the Real World

Initialization of Orientation: As in the flat world scenario, the HoloLens initializes its orientation upon app launch. Using the gyroscope and other orientation sensors, the device establishes a (0,0,0) starting point with the forward direction perpendicular to the ground.

Calculating the Horizon Line: In the real world, the Earth's curvature must be taken into account. The application needs to calculate the dip of the horizon based on the user's height above sea level. This involves a few steps:

Height Measurement: The application may prompt the user to input their height or use additional sensors to estimate it.

Horizon Distance Calculation: First, let's determine the distance to the horizon using the formula:

$$d = \sqrt{2Rh}$$

where:

- d is the distance to the horizon.
- R is the radius of the Earth (approximately 6,371 kilometers).
- h is the height of the observer above sea level.

Calculate the Angle: The angle α can be calculated using the following relationship derived from the geometry of the circle:

$$\cos(\alpha) = \frac{R}{R+h}$$

Solving for α :

$$\alpha = \arccos\left(\frac{R}{R+h}\right)$$

HoloLens Scan Area

HoloLens 2 uses a combination of sophisticated sensors and cameras to scan and understand a room. Equipped with a depth sensor, environment understanding cameras, and an Inertial Measurement Unit (IMU), HoloLens captures detailed spatial data. The depth sensor emits infrared light to measure distances to various points, while the cameras capture visual features of the environment. The IMU tracks the device's orientation and movement, ensuring accurate mapping even when the user is mobile. A custom Holographic Processing Unit

(HPU) processes this data in real-time, constructing a 3D model of the room that updates continuously as the environment changes.

Dividing Sky Band from Horizon Band: The HoloLens Orientation Awareness helps to locate the horizon line within the real-world environment, as described in the previous section. By understanding the layout and dimensions of the bridge, the HoloLens can accurately distinguish between the Sky Band and the Horizon Band. This distinction is vital for placing relevant AR elements in appropriate areas.

Mapping the Bridge Environment: The detailed spatial map created by the HoloLens allows for precise placement and interaction with digital elements within the physical space. With these mapping capabilities, we can divide the bridge into the distinct zones described previously. This capability ensures that the system can identify the user's location within the bridge at any given time, allowing the user interface to adapt dynamically based on the current zone.

5.2.4 AR system's User Interface

2D AR Map: This map features a dynamic overlay that highlights areas of concern—such as hazardous weather zones enabling the crew to visually assess and anticipate navigational challenges ahead. A continuous line on the map plots the optimal alternative route, carefully calculated to circumnavigate these hazards, ensuring safe passage. Additionally, the ship's current position is prominently marked, providing a real-time reference point that integrates the vessel's location with weather updates and route alternatives to assist in decision-making and strategic planning for safer voyages. For collision avoidance, the map will also display the collision point and the route that the captain should follow to avoid collisions, further enhancing navigational safety and situational awareness.

Upon MarineTraffic's advice and [42], we have decided to secure the 2D map's location within the Sky band, which provides spatial context without concealing crucial views. This fixed position aligns with the ship's Skyband, ensuring consistent access to navigational data within the natural field of view and potentially reducing physical strain. Also because this area doesn't occlude with the reality this map will be viable from all 4 workplaces.

For the map, we will use Mapbox, a powerful mapping and location platform that enables developers to integrate custom maps and geospatial data into their applications. Mapbox provides highly customizable maps, real-time data visualization, and efficient handling of large datasets, making it ideal for creating detailed and dynamic navigation aids in our AR application.

Annotations: In building upon our collision avoidance system's foundational features, we are introducing advanced functionalities that will significantly enhance the interactive aspect of the system, particularly through expandable annotations and a sophisticated pinning mechanism. These features are designed to provide the captain and crew with a more dynamic and informative interface, without overwhelming them with constant information.

The distinct markers used for highlighting the vessel at risk of collision and the projected collision point on the horizon band will come equipped with annotations that can be expanded.



Figure 45: AR Map locked on Sky Band

This means that, upon interaction, these markers will unveil a deeper layer of information about the potential threat, including but not limited to the vessel's name, its current speed and trajectory, and the estimated time until the potential collision occurs. This expandability allows for a tailored informational approach, where the depth of data displayed can be adjusted based on the immediacy and severity of the navigational situation.

Furthermore, the system will feature a pinning function, crucial for maintaining ongoing awareness of Points Of Interest (POIs) such as nearby vessels. This function allows for critical information about these POIs to be 'pinned' onto the AR display, ensuring that this information remains readily accessible and continuously visible, regardless of any changes in the user's focus or location within the AR zones of the bridge. The pinned information, illustratively shown in our system's design, will default to a symbolic representation encircled, with an arrow indicating the relative heading of the other vessel. This symbol's color will transition from green to orange, and eventually to red, as the proximity risk increases, thereby offering an intuitive visual cue of the escalating risk.

Upon focusing on a vessel's symbol, additional details will become visible, such as the distance to the vessel. The user can then decide to pin this information pane, keeping it constantly open on their display or integrated into a widget for easy reference. This pinning can be activated through various user-friendly inputs such as voice commands, specific gestures, or control buttons, as validated in our usability tests.

5.2.5 Water Surface Route

To enhance our collision avoidance strategy, the system will display an alternate route marked on the water's surface. This route will be delineated by a semitransparent line, with arrows interspersed to indicate direction, providing a clear path for navigation while minimizing visual disruption. This design choice ensures that the captain retains full situational awareness, focusing on safe navigation without losing sight of the real-world surroundings.

The dashed line with arrows will be semi-transparent and visible only within the designated

work zones on the bridge, catering specifically to the navigating personnel. This approach keeps the navigation guidance discreet, avoiding unnecessary distractions for other crew members and ensuring it does not interfere with other bridge operations. The line's visibility will be restricted to ensure it assists only those directly involved in the vessel's navigation, thus maintaining an efficient and focused navigational environment.



Figure 46: Augmented Reality Visualization Route Concept

5.2.6 Maritime Use Case: Plans for the next 18 months

For the next 18 months of CREX DATA, these are the technical plans for AR system development for the Maritime Use Case:

- Plan for initial testing of AR functionality in real world conditions outdoors
- Incorporate data and optimize functionality based on the evaluation testing of AR prototype
- Experiment with real-time streaming of data, connecting to the forecasting architecture.

6 Uncertainty Visualization in Augmented Reality

Uncertainty visualization is a significant research challenge for visual analytics and data visualization, especially if such uncertainty is required to be communicated in real-time and plays a central role in decision making while a user is in an extreme situation and wears or holds AR-capable devices (head-mounted or mobile). As uncertainty visualization is considered one of the top research problems in visualization, we will focus on data extended from regular non-uncertainty processing to an uncertainty-aware dataset in the weather emergency use case and maritime use case, focusing on visualizing uncertainty of data in each use case. In this document, uncertainty mappings for AR uncertainty visualization, on-site is explained, being a feature of the AR system development which is visible to the AR users deploying a head-worn AR system. The AR system deploys interactive visualization specifying visual elements connected to uncertainty of data.

6.1 Weather Emergency Uncertainty Use Case

Flood modeling and forecasting are crucial for managing and preparing for extreme flood events. Data gathered at different stages of disaster risk management may exhibit varying degrees of variability and uncertainty, often due to incomplete knowledge about each phase and its associated timelines. The impact of floods can vary significantly across different timelines in disaster management. Uncertainty arises from a complex interplay of errors related to source data (land use changes, population growth, and urbanization), sampling, and model representation. These uncertainties cascade through the risk assessment system (Figure 47), diminishing confidence in individual predictions and causing variability in predicted depths and extents across multiple forecasts. This variability, shown as differences in predicted depths and flows, further impacts the analysis of flood effects. Uncertainties in predicted depths and flows combine with errors from data on buildings and infrastructure, as well as with the statistical models used to estimate damage. Consequently, there is uncertainty in the quantified damage for a flood scenario, complicating decision-making processes, such as whether to invest in enhanced mitigation measures.

In relation to the AR prototype, these are the main visualization paradigms showcasing uncertainty of data, while a rescuer is located in the centre of the city of Dortmund and Innsbruck. The visualization paradigms for uncertainty visualization are going to be tested during the initial trials of the AR prototype, in June 2024.

- The current water height is visualized alongside a percentage value that represents the prediction certainty. This value indicates the confidence level that the predicted water height will match the visualized height. As predictions extend further into the future, the certainty value decreases, reflecting the increased uncertainty. This real-time feedback mechanism allows users to make more informed decisions based on the reliability of the forecast data.
- While visualizing the water level based on water rendering, there is a visual grid and/or number (scientific data) which communicates the 'best case scenario', therefore, the lowest elevation of water predicted as well as the 'worst case scenario' which communicates the highest level of water predicted. The distance between the two water levels signifies the range of uncertainty for these predictions.

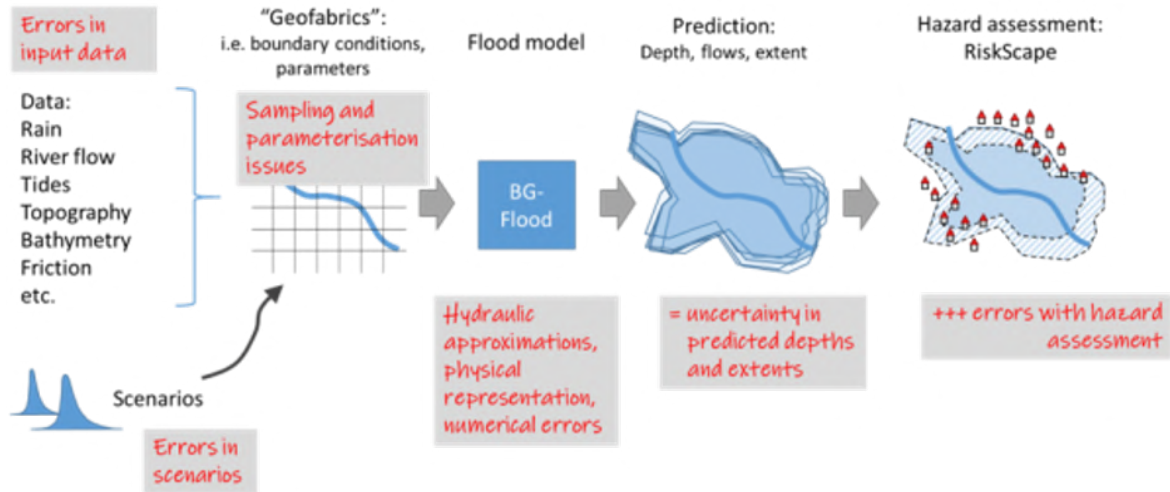


Figure 47: Errors from multiple sources “cascade” through the analysis system, resulting in uncertainty in predicted depths and extents and the derived hazard assessment.

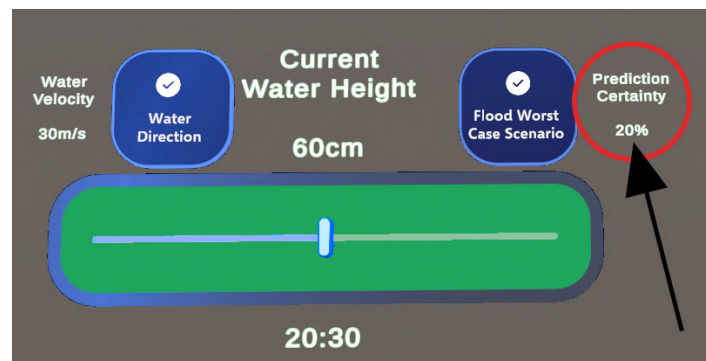


Figure 48: Certainty of Predicted Water Level

- While rescuers are able to visualize the water level as well as its velocity and its direction at specific points in the centre of Innsbruck and Dortmund, they are also able to see significant points in the horizon (schools, public buildings etc). These points are visualized in different colours to showcase their respective flood risk. Attached to these points, there is an info window which may contain scientific data communication uncertainty of flood risk.



Figure 49: POI's with Uncertainty

6.2 Maritime use case

In maritime navigation, accurately predicting routes and weather conditions is vital for the safety and efficiency of ship operations. However, these predictions often come with inherent uncertainties. To address this, we included an uncertainty visualization component within the application being developed for the HoloLens 2 augmented reality (AR) device. This feature will help ship captains understand the variability and confidence levels of the predicted routes and weather conditions. By visually representing the range of possible outcomes, we aim to enhance situational awareness and support more informed decision-making in navigation.

6.2.1 Collision Avoidance Challenges

One of the primary challenges in maritime navigation that our approach aims to address is collision avoidance. This problem is particularly complex due to the limitations in how ships' positions are communicated and updated in real-time. Our project partner, MarineTraffic, plays a crucial role in this process by supplying data on potential collisions and providing alternative routes and other relevant information.

MarineTraffic relies on the Automatic Identification System (AIS), a critical maritime communication system designed to enhance maritime safety by automatically transmitting vessel information to other ships and coastal authorities. AIS data includes essential details such as a ship's identity, position, course, and speed, which are vital for navigation and collision avoidance. However, a significant issue arises from the fact that AIS does not always update a ship's location in real-time.

The frequency of AIS updates can vary widely. In some instances, ships update their positions every few seconds, while in others, updates occur every few minutes. There are even cases where ships only update their location every few hours. This inconsistency leads to a significant challenge: predictions and decisions regarding collision avoidance are often based on outdated positional data. Consequently, there is a risk that navigational decisions are made using information that does not accurately reflect the current positions of vessels,

potentially leading to unsafe situations.

This lack of real-time updates in AIS data underscores the importance of integrating uncertainty visualization into our navigation aid. By understanding and visualizing the inherent uncertainties in ship positions, we can provide a more comprehensive and realistic picture of the maritime environment. This approach enhances situational awareness and supports more informed decision-making, ultimately improving the safety and efficiency of maritime operations.

6.2.2 Uncertainty Visualization

Uncertainty visualization is a critical aspect of modern navigation systems, enabling users to comprehend and make informed decisions based on the inherent uncertainties in positional data. A prime example of this is the approach taken by Google Maps.

6.2.3 Google Map's Uncertainty Visualization

Google Maps employs a simple yet effective method to convey the uncertainty associated with GPS data. Recognizing that GPS devices often have an offset of several meters, Google Maps places a semi-transparent circle around the most likely position of the user (Figure 50). This circle indicates the area within which the user might be located, with the highest probability concentrated at the center, where a dot marks the estimated position [34][49]. This visualization helps users understand that their exact location may vary within the circle, thus providing a clear representation of the uncertainty inherent in GPS data.

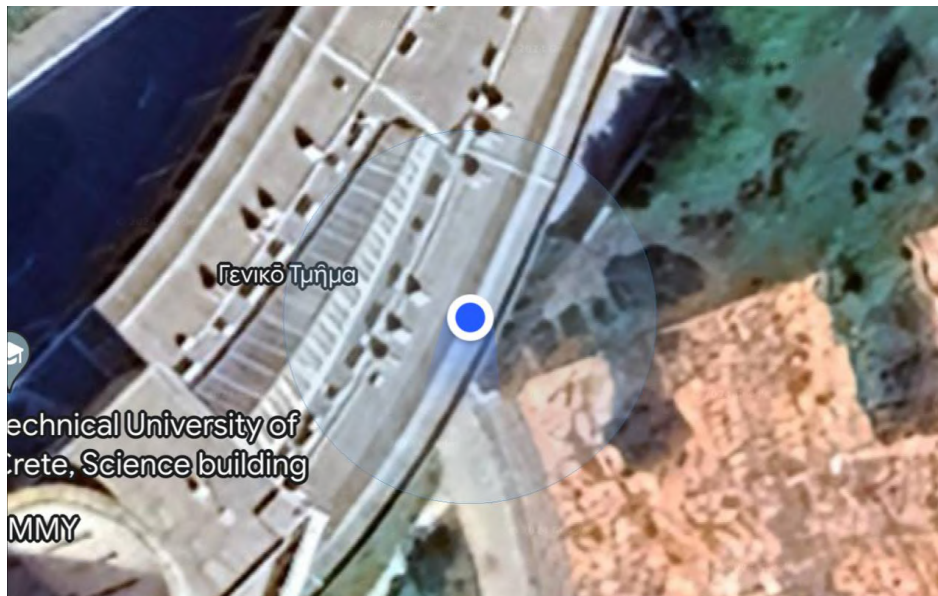


Figure 50: Google Maps Uncertainty Radius

6.2.4 CesiumJS's Uncertainty Visualization

CesiumJS is an open-source JavaScript library for creating 3D globes and maps. It supports uncertainty visualization by allowing developers to create semi-transparent cones or spheres around predicted positions, which represent the range of possible locations an object might occupy over time. This is particularly useful in applications like satellite tracking and environmental monitoring.

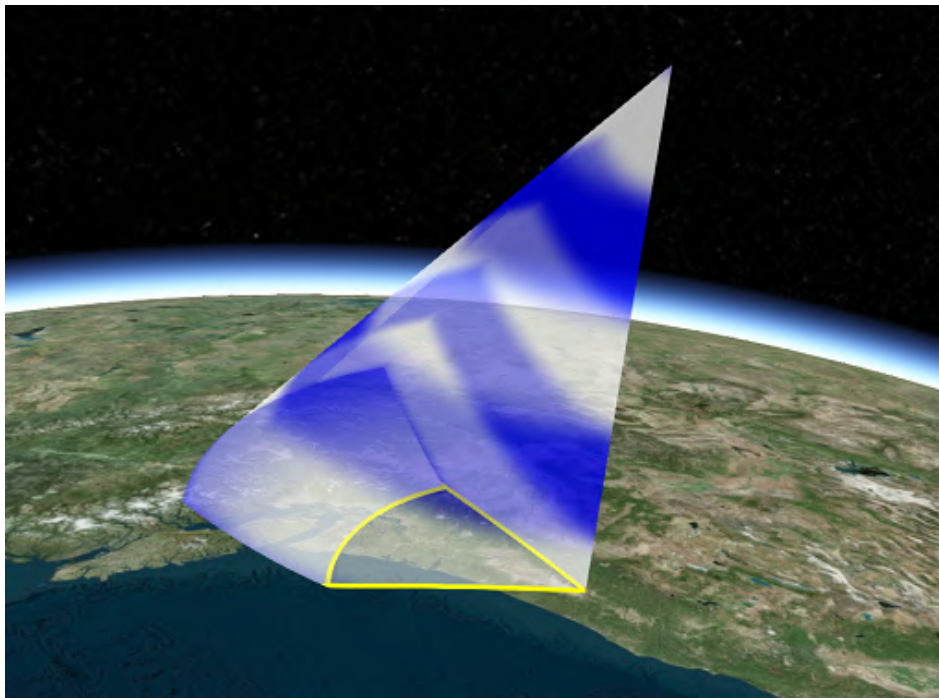


Figure 51: Uncertainty Cone Visualization Concept

6.2.5 Implementation of Uncertainty Visualization in Maritime Navigation

In our project, we have developed an advanced uncertainty visualization component tailored specifically for maritime navigation. Our approach builds on the concepts discussed in the previous sections, applying them to create a robust and intuitive navigation aid for ship captains.

6.2.6 Route Visualization

To enhance navigational safety and efficiency, we plot the predicted route of the ship using real-time data provided by our project partner, MarineTraffic. This route is represented as a semi-transparent line, which is designed to minimize occlusion of the real-world view through the AR device. The transparency ensures that captains can maintain situational awareness of their immediate surroundings while still clearly seeing the predicted path[29].

6.2.7 Cone of Uncertainty

Alongside the primary route line, we overlay a cone-like shape that indicates the general direction in which the ship should move (Figure 51). This cone represents the range of possible paths the ship could take, given the inherent uncertainties in the positional data. The widest part of the cone expands over time, reflecting the increasing uncertainty as predictions extend further into the future.

The cone is also semi-transparent to ensure it occludes the real world as little as possible. By using transparency, we maintain the captain's ability to see critical visual cues from the environment while providing them with crucial predictive information [26][31][41][60].

6.2.8 Color-Coded Probability Indicators

To differentiate between the most likely path and the range of possible routes, we use distinct colors for the line and the cone. The central line, representing the most probable path, is displayed in a different color from the surrounding cone. This visual distinction helps captains quickly and easily identify the primary route while understanding the variability and potential deviations.

The color-coding of the cone itself reflects the probability of different routes. Gradients or varying shades within the cone indicate higher or lower likelihoods of the ship following particular paths. This nuanced visualization allows captains to gauge the level of certainty associated with the predicted routes and make more informed navigational decisions. The central line represents the most likely path, while the surrounding cone indicates the range of possible routes.

6.2.9 User Interaction and Feedback

Our implementation also allows for interactive elements, enabling captains to receive real-time feedback and additional details about the uncertainty visualization. For instance, tapping on different parts of the cone can provide more information about the probability associated with that segment. This interactive functionality enhances the utility of the visualization, making it a powerful tool for real-time decision-making in complex maritime environments.

By integrating these advanced visualization techniques, we aim to provide ship captains with a comprehensive and realistic understanding of their navigation environment. This approach not only improves situational awareness but also supports safer and more efficient maritime operations by effectively communicating the inherent uncertainties in predicted routes.

6.3 Uncertainty Visualization: Plans for the next 18 months

For the next 18 months of CREX DATA, these are the technical plans for AR system development for the Maritime Use Case:

- Based on visibility testing of AR elements outdoors, optimize uncertainty visualization paradigms to be suitable for outdoors viewing
- Manipulate data, especially in the weather emergency use case, so that scientific data signifying uncertainty are evident

- Experiment with real-time streaming of data, when connecting to the forecasting architecture, so that uncertainty visualization is calculated on-the-fly.

6.4 Trials: The cities of Dortmund and Innsbruck

The proposed AR-based flood visualization system is going to be evaluated in Dortmund and Innsbruck, while a rescuer is deploying a head-word AR system as analyzed in T5.4.



Figure 52: a) City of Dortmund, b) City of Innsbruck

The city of Dortmund, situated in the metropolitan area of Ruhr and the heart of Westphalia, is in the catchment area of three river systems, and has a stable population of 590.000 according to the 2024 census. It is a postindustrial city that, in recent years, has been developing the sectors of IT, logistics, and biotechnology. In recent decades, many plans have attempted to manage the profound structural change in the economy and create the future of the municipality and region. The authorities, inhabitants, and experts of the city aim to develop fewer industrial sites, more residential areas, a lot more landscape, and more area for nature. Urban planning promotes equality and balance between the economic, environmental, and social sectors, sustainability, and participation in all planning strategies, and aims to establish a reliable framework for public and private investment in public space, retail, office buildings, housing, and large-scale projects [54]. Dortmund has suffered from floods in its current history, so it is important to manage floods in a way that will secure resilient urban environments. In this direction, images that create a sense of 3D space and depth create a more realistic and interactive experience for the viewer, so they can provide important information for flood management [5]. With a population exceeding 160,000, including 35,000 students Innsbruck is one of the preeminent Alpine towns. The metropolitan area of Innsbruck is projected to experience population growth, which underscores the burgeoning demand for employment, educational facilities, and a spectrum of municipal services. Furthermore, mobility remains a critical concern due to the city's unique topographical challenges. The city's location within a narrow valley basin and its proximity to a sensitive, high alpine environment impose considerable restrictions on spatial development opportunities. Innsbruck's mountain landscape results in a significant altitude variation of 535 meters between its lowest and highest inhabited points coping with natural hazards.. This altitude disparity presents substantial obstacles in optimizing intra-city traffic connectivity. Innsbruck is facing threats such as mudflows, avalanches, rockfalls, and floods. These hazards are projected to intensify due to the impacts of climate change. Consequently, the city must continuously adapt and enhance its strategies for risk management and sustainable urban development to mitigate these increasing threats.

AR can be applied to indicate the progress and expansion of the flood occurrence and to assess the flood damage and vulnerability of urban structures. AR models of the built environment enhances realism and enables more accurate assessments of flood damage. Hence, combining AR technology with flood data can provide decision-makers with valuable insights in urban flood management [52].

The urban elements that need to be visualized are related to the city's exposure hazards and vulnerability. These are topography data as Global digital terrain models (DTMs)—Shuttle Radar Topography Mission (SRTM) and Multi-Error-Removed Improved-Terrain (MERIT) as well as locally available data such as laser imaging, detection, and ranging (LiDAR); high-resolution satellite or orthophoto imagery; drone survey data; and bathymetric surveys of water bodies (Ferguson, et al., 2023). Data to be integrated may be the Hydrometeorological data global datasets such as Multi-Source Weighted-Ensemble Precipitation (MSWEP) or European Centre for Medium-Range Weather Forecasts (ECMWF) Reanalysis v5 (ERA5) or the Copernicus Emergency Management system for flood monitoring and forecasting and local datasets with time series of rainfall and winds, water levels, or river discharges. Another important urban element is the existing flood protection infrastructure and numerical models for flood hazard modeling, as well as the layout and dimensions of primary and secondary drainage infrastructure such as road drainage, canals, culverts, pump stations, and tidal gates, as well as the dimensions and characteristics of coastal or fluvial embankments, dunes, and existing hydrological, hydraulic, and risk models [19]. Existing flood, exposure, or vulnerability data such as global or local flood hazard maps, (mapped flood hot spots), OpenStreetMap data, cadastre system data, building types, population distribution, and characteristics are also elements that may be visualized as having decisive role in flood management. Finally, the city's infrastructure, such as roads, drinking water, sanitation, drainage, and flood protection infrastructure; health care and school facilities; and environmental and cultural assets; are urban elements that have a decisive role in city's resilience and urban flood management. In the proposed system architecture, the available data are the available data sets from the EU sites as Copernicus browser [13], the European floodviewer [18], fire data [14] and urban data as mobility networks, public infrastructures, available 3d models, e.t.c

The team has so far analyzed both cities on order to find areas for the trials. The criteria for these areas-routes selection is:

1. **Safety:** This is a pedestrian area, making it ideal for conducting Augmented Reality experiments safely and away from vehicular traffic.
2. **Points of Interest (POIs):** The area contains several vulnerable infrastructures, such as public schools and kindergartens, which are often prioritized by rescue teams during emergencies.
3. **Flood Risk:** The proximity to the river and areas likely to experience pluvial floods under emergency weather conditions makes it a pertinent location for our trials. After the evaluation of these three factors in both cities, the team has proceeded to the selection of two 100-meter length routes.

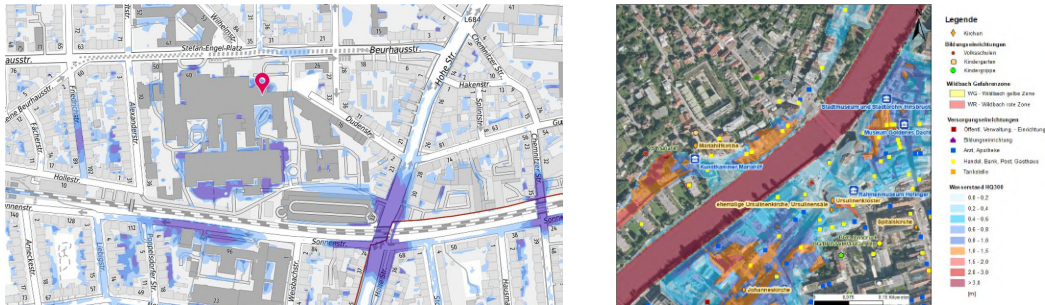


Figure 53: The selected area for trials

6.4.1 Data from Mike+

Mike+ system is mainly composed of specific procedures for multiple data collection, processing and storing and a wide range of numerical models incorporating machine learning. It is a decision support system based on complex algorithms capable of forecasting high water levels events and continuously updating the data, and it provides a wide range of tools for data as warning dissemination, including a specific web interface through a graphical user interface for the configuration of advance management of the system.

The MIKE+ software can lead to a scenario analysis, with a list of different related alternatives for flood mitigation. The simulation results can be maps of flood inundation depth and area for different mitigation scenarios. The combination of mitigation scenarios can gear toward the effectiveness of optimal solutions for the reduction in frequency and intensity of flood events.

The basic idea is the visualization of the water distribution with the use of data such as bathymetry, geographical information, and other infrastructures as pipes and canals in a simulation that can be saved every 60 seconds (Fig MIKE+).

Urban flooding simulator: MIKE+

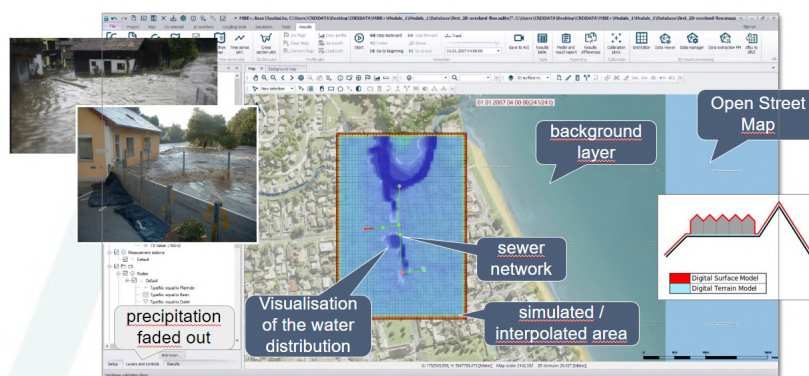


Figure 54: MIKE+ software

For flood visualization, we utilize data exported by MIKE+ in CSV format. MIKE+ divides the area into a grid of rectangles, as seen in the picture. Each rectangle corresponds to a forecasted water level for a 1-minute time interval.

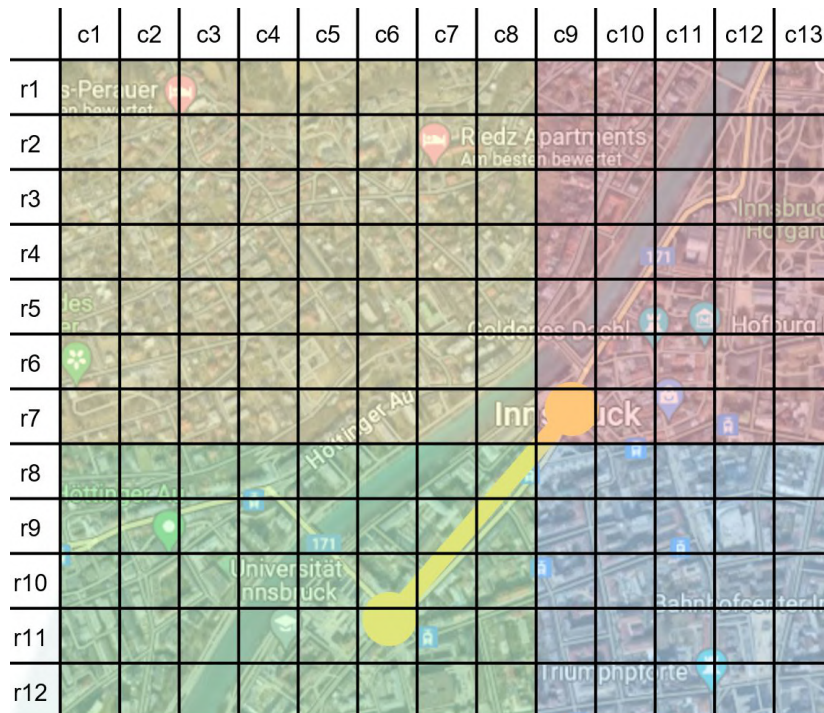


Figure 55: Innsbruck Flood Area Grid

To simplify data processing, we assume that each colored area corresponds to a single grid area. Therefore, we select one rectangle from each grid area to extract the water level forecast. For example, for the yellow area, we utilize the c8r7 rectangle, for the pink area, we use c9r7, for the green area, we choose r8c8, and for the blue area, we select c9r9.

2	c10r8	c11r8	c12r8	c13r8	c1r9	c2r9	c3r9	c4r9	c5r9	c6r9	c7r9	c8r9	c9r9	c10r9	c11r9	c12r9	c13r9	c1r10	c2r10	c3r10	c4r10	c5r10	c6r10
3																							
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 56: MIKE+ FLOOD CSV Sample

We process the CSV data on our Node server and generate JSON files containing relevant information for visualization. Each JSON file corresponds to one of the four colored areas in Innsbruck mentioned earlier.

Our AR device deserializes these JSON files and, by utilizing its GPS location, it identifies one of the four areas and visualizes the corresponding flood level data from the appropriate JSON file.

Utilizing the sliding window concept, the app filters out values from the JSON files that precede the current time of the user. Additionally, it updates the visualization every minute, discarding timestamps that match the current time

6.4.2 Innsbruck Trials

Introduction

In a recent series of trials held in Innsbruck, our AR team conducted extensive testing of our prototype application. These trials aimed to gather feedback and insights from firefighting experts to enhance the application's functionality and user interface.

Location and Participants

The trials in Innsbruck took place at <https://maps.app.goo.gl/5tZHdyXiwGzJmBZz7> on Tuesday, 11th June, between 9:00 am and 12:00 pm. Firefighters-Experts from the fire brigade (ÖBFV) participated, each equipped with the Microsoft HoloLens 2 device to evaluate the AR application. Among those experts were the President of the fire brigade (ÖBFV) and the Senior Researcher/Department Head of the fire brigade (ÖBFV)

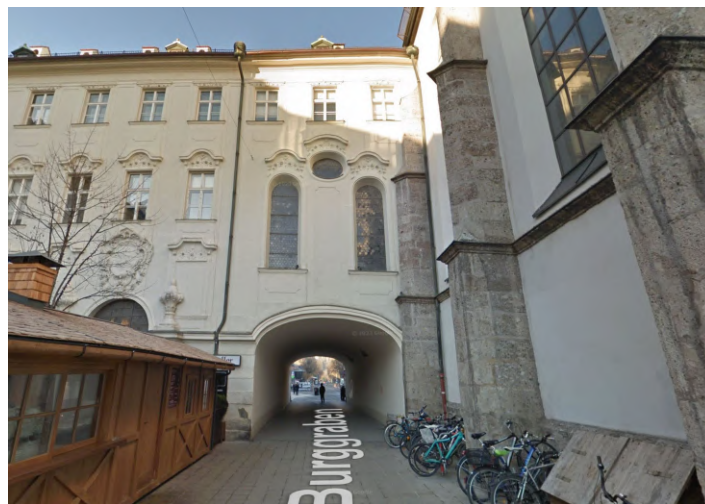


Figure 57: Innsbruck AR Trial Location

Testing Methodology and Feedback

During the trials, the firefighters engaged with the application in its entire scope. The experts provided real-time feedback on usability, interface intuitiveness, and overall functionality. Based on their insights, we promptly made several adjustments to the user interface to enhance ease of interaction and overall usability for the end users prior to the trials in Dortmund.



Figure 58: Innsbruck AR Trial Location

6.4.3 Dortmund Trials

Introduction

Following the initial trials in Innsbruck, our AR team conducted further testing in Dortmund with the primary aim of gathering additional feedback from experts in the field. These trials were part of our ongoing efforts to ensure that the AR application is on the right track for the needs of emergency responders.

Location and Participants

The trials took place at the facilities of Deutsches Rettungsrobotik Zentrum (DRZ) in Dortmund. Four experts from the Dortmund firefighting department participated, each equipped with the Microsoft HoloLens 2 device to evaluate the AR application. Among the participants were the Director of the Fire Department of Dortmund, the Chief of the Fire Department and the press spokesman of the city and the UBA Diplom chemist for the fire department.



Figure 59: Dortmund Location and Trials

Testing Methodology and Feedback

The Dortmund trials followed initial testing in Innsbruck, where we received valuable feedback. Based on this feedback, we made prompt adjustments, as mentioned earlier, to the user interface to facilitate easier interaction. The revised version tested in Dortmund incorporated these improvements. During the trials, the firefighters engaged with the application and tested the current capabilities. They provided real-time feedback on usability, interface intuitiveness, and overall functionality. Their interactions with the application were closely monitored, and their feedback was gathered to identify any further adjustments needed. This feedback is crucial for refining the application to better meet the needs of emergency responders.

The changes that were made involves removing the main menu with the three basic buttons that were permanently placed in the upper zone of the user's field of view (FOV). Instead, we have relocated the main menu to the redesigned pop-up menu accessible with the left hand.



Figure 60: Before Changes UI



Figure 61: New UI

7 Conceptual Modeling for Explainable AI and Uncertainty Communication

In CREXDATA, conceptual models are meant to serve two purposes: (1) explaining models and their predictions using domain-specific concepts at appropriate level of abstraction and (2) defining the types and semantics of possible uncertainties and effective ways to communicate this information to analysts. A conceptual model (CM) represents concepts and relationships of a given domain, i.e., the domain semantics. The fundamental reason for using conceptual models is the semantic poorness of ML models derived from data. Matching components of a ML model, its results, or explanations to concepts and relationships from a CM can make them more meaningful and understandable to users. Obviously, to achieve this goal, two components are necessary:

1. An appropriate CM including concepts and relationships that are relevant to a given ML model. Most commonly, conceptual models are formally represented by entity-relationship (ER) diagram, i.e., a graph with vertices representing concepts and edges representing relationships.
2. A “semantic mapper”, that is, a component or schema specifying the links between the CM and the ML model. For example, data-specific features used in a ML model need to be matched to attributes defined in a CM, and decision rules used for prediction and/or explanation need to be linked to domain relationships.

A simple example of using a CM in combination with a ML model is described in paper [32] and illustrated in Figure 62. The authors propose a Model Embedding Method (MEM) for embedding machine learning models into conceptual models. The method employs bidirectional mapping of concept attributes contained in a CM to all features available in a dataset. The contribution of each feature to a ML model decision is calculated using the SHAP values method. The feature contributions are transformed into contributions of the concepts associated with these features. Using the links specified in the CM, concept contributions are represented as scores that measure directed local contributions of one concept to another derived from feature contributions; see Figure 62.

It has to be admitted, however, that a formalized representation like in Figure 62 may not be well suited for presenting to end users or even to domain experts, who may find such a diagram difficult to interpret. It seems to be more beneficial for users if relevant concepts and relationships are presented using texts in a natural language and, possibly, simple schematic drawings that can be easily understood by humans.

While it may be possible to employ generative AI, particularly, large language models, to transform formal representations into human-understandable descriptions, there is still a need to obtain a domain CM from an expert or literature; hence, human involvement is required anyway. Then, there is no real need in constructing a fully formal representation of a CM. Instead, there should be a collection of human-understandable definitions of domain concepts and relationships provided by a domain expert or extracted from literature. These definitions need to be matched to the features available in data the ML model operates on.

Generally, acquiring concept definitions entails a problem known as “knowledge acquisition bottleneck” [15]. However, an interactive visually supported workflow like represented in

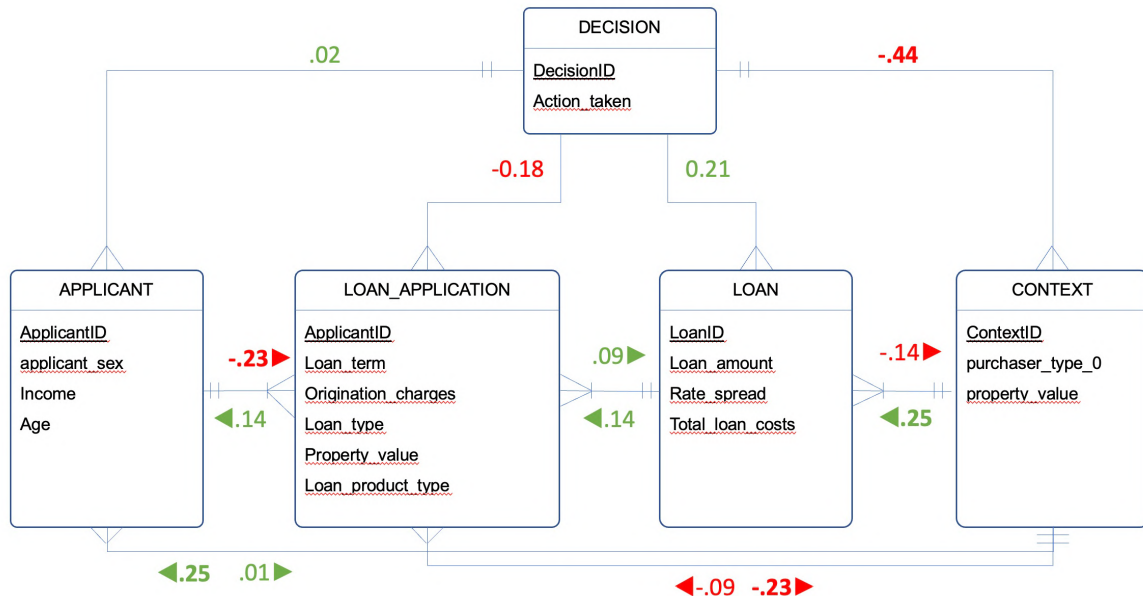


Figure 62: An example of using a conceptual model to represent features contributions to a decision of a ML model. Source: [32].

Figure 9 suggests a way to facilitate the acquisition of domain concepts and relationships. In this workflow, a human expert creates a set of labelled examples for model training and testing. It consists of groups of similar data instances representing different patterns that need to be distinguished based on features selected or constructed by the expert. It is possible to extend the workflow by asking the expert to give additionally to class labels or explanatory annotations for the patterns and the features used to distinguish them. These annotations will form a human-readable conceptual model, which can later be used for explaining model results and possible uncertainties to model users.

Figure 63 presents an illustrative example of a fragment of a conceptual model corresponding to a ML model for recognizing different types of vessel movement behaviours. It includes human-oriented definitions of three behaviour types (movement patterns), examples of these patterns, textual descriptions of their distinctive characteristics, relevant attributes by which the patterns are distinguished, and names of data-derived features that are used in the model for pattern recognition. The attributes are connected to the descriptions of the pattern characteristics referring to these attributes, on the one side, and to the related features contained in data, on the other side. The connections between the attributes and the features represent the semantic mapping between the conceptual model and the data.

For explainable AI, a domain conceptual model can be used for translating explanations provided by an XAI component to descriptions or statements using relevant domain concepts and terminology instead of feature names and human-understandable approximate characteristics instead of meaningless numeric thresholds. For example, using the CM illustrated in Figure 63, an explanation for recognizing some movement behaviour of a vessel as trawling can include the statement “slow and steady speed” instead of “*min_speed* > 0.1 and *Q3_speed* < 7.2”.

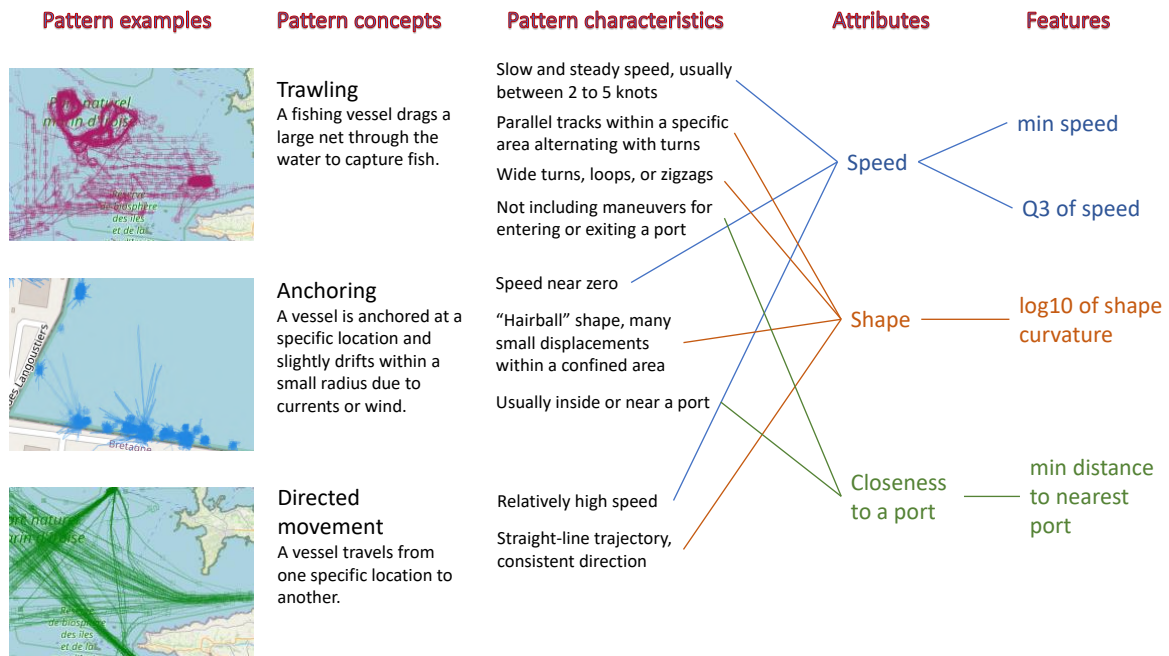


Figure 63: An illustrative example of a fragment of a conceptual model corresponding to a model recognizing vessel movement patterns.

Regarding the communication of uncertainties, a most obvious use is in presenting predictions of probabilistic models. For example, if a model for recognizing vessel movement patterns identifies a specific part of a vessel trajectory as “trawling with 60% probability”, an explanatory module can further elucidate this prediction by investigating other patterns that received relatively high scores. This module can analyze the features responsible for these patterns, map these features to the pattern characteristics specified in the CM, and present these characteristics to the user. For instance, it might explain, “It may be a directed movement because of a relatively high speed”, or “It may be a port-related maneuver due to closeness to a port”.

Further possible uses of a domain CM for informing users about uncertainties include:

- It can specify the scope of a ML model and the underlying assumptions, clarifying what the model is designed to predict and under what conditions it is expected to perform well. This helps users understand the limitations and the specific context in which the predictions should be trusted.
- It can represent the distribution of training data, highlighting any biases or gaps. If the training data lacks representation for certain conditions or populations, users can be alerted to the potential for increased uncertainty in those areas.
- It can map out which areas of the feature space are well-covered by the training data and which are not. This helps users understand where the model might extrapolate and, consequently, where predictions might be less reliable.
- It can describe external factors (e.g., economic conditions, environmental factors) that

might impact the accuracy of predictions. Understanding these influences helps users contextualize the predictions and their uncertainties.

In the further research, we plan to develop an approach to acquiring a conceptual model from a domain expert in the process of model building or examination. We also plan to design an interactive UI that can facilitate the acquisition process. It may not be feasible to develop a universal approach suitable for any kind of ML model. Therefore, we shall focus on one model type within one of the CREXDATA use cases.

8 Conclusions

In this deliverable, we have presented the results of the activities of Workpackage 5 in the first period of the project. The tasks of the WP implement the research directions that lies at the intersection points of a mesh where the Explainable AI and Uncertainty access are crossed with the Visual Analytics and Augmented Reality approaches. The main goal of the WP is to develop a set of methodologies to enhance the explainability of AI models and to communicate the uncertainties in the data and models to the end-users. This initial phase allowed us to align the research directions with the needs of the use cases and to define the main research directions for the remaining period.

This initial phase is the foundation upon which we will build and solidify the next steps of the project. One research direction will be to strengthen the link with the Machine Learning models developed in Workpackage 4. We have explored single models and their explainability. We will continuing the continuous interaction with the use cases to refine the conceptual models and the ML models, aiming to provide a more human-understandable representation of the data and models. We will also continue the development of the AR prototypes and Visual Analytics for the use cases, focusing on the visualization of uncertainty. This interaction with the use cases will drive the continuous development of the conceptual model.

9 Acronyms and Abbreviations

- AAE - Adversarial Autoencoder
- ABELE - Adversarial Black-box Explanation Learning Engine
- AI - Artificial Intelligence
- AR - Augmented Reality
- CM - Conceptual Model
- CNN - Convolutional Neural Network
- COVID-19 - Coronavirus Disease 2019
- EU - European Union
- ISIC - International Skin Imaging Collaboration
- ML - Machine Learning
- PCA - Principal Component Analysis
- PGAAE - Progressive Growing Adversarial Autoencoder
- ResNet - Residual Neural Network
- SHAP - SHapley Additive exPlanations
- t-SNE - t-distributed Stochastic Neighbor Embedding
- VA - Visual Analytics
- XAI - Explainable Artificial Intelligence

References

- [1] Karim I. Abdrabo, Sameh A. Kantosh, Mohamed Saber, Tetsuya Sumi, Dina Elleithy, Omar M. Habiba, and Bahaa Alboshy. *The Role of Urban Planning and Landscape Tools Concerning Flash Flood Risk Reduction Within Arid and Semiarid Regions.*, pages 283–316. Springer Singapore, 2022.
- [2] Iulia Ajtai, Horațiu Ștefănie, Cristian Maloș, Camelia Botezan, Andrei Radovici, Maria Bizău-Cârstea, and Călin Baciuc. Mapping social vulnerability to floods. a comprehensive framework using a vulnerability index approach and pca analysis. *Ecological Indicators*, 154, 2023.
- [3] Natalia Andrienko, Gennady Andrienko, Alexander Artikis, Periklis Mantenoglou, and Salvatore Rinzivillo. Human-in-the-loop: Visual analytics for building models recognising behavioural patterns in time series. *IEEE Computer Graphics and Applications*, pages 1–15, 2024.
- [4] Natalia Andrienko, Gennady Andrienko, Alexander Artikis, Periklis Mantenoglou, and Salvatore Rinzivillo. Human-in-the-loop: Visual analytics for building models recognising behavioural patterns in time series. *IEEE Computer Graphics and Applications*, pages 1–15, 2024.
- [5] Vahid Bakhtiari, Farzad Piadeh, Kourosh Behzadian, and Zoran Kapelan. A critical review for the application of cutting-edge digital visualization technologies for effective urban flood risk management. In *Sustainable Cities and Society*, volume 99, 2023.
- [6] Francesco Bodria, Fosca Giannotti, Riccardo Guidotti, Francesca Naretto, Dino Pedreschi, and Salvatore Rinzivillo. Benchmarking and survey of explanation methods for black box models. *Data Min. Knowl. Discov.*, 37(5):1719–1778, 2023.
- [7] Samuel Brody, Yoonjeong Lee, and Baukje Bee Kothuis. Chapter 5 - urban flood modeling: Perspectives, challenges, and opportunities. In *Coastal Flood Risk Reduction*, pages 47–60. Elsevier, 2022.
- [8] Juliano Calil, Geraldine Fauville, Anna Carolina Muller Queiroz, Kelly L. Leo, Alyssa G. Newton Mann, Tiffany Wise-West, Paulo Salvatore, and Jeremy N. Bailenson. Using virtual reality in sea level rise planning and community engagement—an overview. *Water*, 13(9), 2021.
- [9] Eleonora Cappuccio, Daniele Fadda, Rosa Lanzilotti, and Salvatore Rinzivillo. an interactive visualization combining rule-based and feature importance explanations. In *Proceedings of the 15th Biannual Conference of the Italian SIGCHI Chapter*, pages 1–4, 2023.
- [10] Eleonora Cappuccio, Daniele Fadda, Rosa Lanzilotti, and Salvatore Rinzivillo. Fiper: a visual-based explanation combining rules and feature importance. *arXiv preprint arXiv:2404.16903*, 2024.
- [11] Steve Carver. Developing web-based gis/mce: Improving access to data and spatial decision support tools. In *Spatial multicriteria decision making and analysis*, pages 49–76. Routledge, 2019.

- [12] Luís Cea and Pierfranco Costabile. Flood risk in urban areas: Modelling, management and adaptation to climate change. a review. *Hydrology*, 9(3), 2022.
- [13] Copernicus browser, 2024. <https://dataspace.copernicus.eu/browser/>.
- [14] Copernicus fire data, 2024. https://effis.jrc.ec.europa.eu/apps/effis_current_situation/.
- [15] J. Cullen and A. Bryman. The knowledge acquisition bottleneck: Time for reassessment? *Expert Systems*, 5(3):216–225, 1988.
- [16] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.
- [17] T. Eldho, P. Zope, and A. Kulkarni. Urban flood management in coastal regions using numerical simulation and geographic information system. In *Integrating Disaster Science and Management*, pages 205–219. Elsevier, 2018.
- [18] European floodviewer, 2024. <https://discomap.eea.europa.eu/floodviewer/>.
- [19] Scott Ferguson, Mathijs Van Ledden, Steven Rubinyi, Ana Campos, and Tess Doeffinger. *Urban Flood Risk Handbook: Assessing Risk and Identifying Interventions*. GFDRL, 2023.
- [20] Riccardo Guidotti, Anna Monreale, Fosca Giannotti, Dino Pedreschi, Salvatore Ruggieri, and Franco Turini. Factual and counterfactual explanations for black box decision making. *IEEE Intell. Syst.*, 34(6):14–23, 2019.
- [21] Riccardo Guidotti, Anna Monreale, Stan Matwin, and Dino Pedreschi. Black box explanation by learning image exemplars in the latent feature space. In *ECML/PKDD (1)*, volume 11906 of *Lecture Notes in Computer Science*, pages 189–205. Springer, 2019.
- [22] David Gunning and David W. Aha. Darpa’s explainable artificial intelligence (XAI) program. *AI Mag.*, 40(2):44–58, 2019.
- [23] Paul Haynes, Sigrid Hehl-Lange, and Eckart Lange. Mobile augmented reality for flood visualisation. In *Environmental Modelling & Software*, volume 109, pages 380–389. Elsevier, 2018.
- [24] Paul S. Haynes and Eckart Lange. Mobile augmented reality for flood visualisation in urban riverside landscapes. In *Journal of Digital Landscape Architecture*, pages 254–262. Herbert Wichmann Verlag, 2016.
- [25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE Computer Society, 2016.
- [26] Hurricane cone uncertainty. <https://www.nhc.noaa.gov/aboutcone.shtml>.
- [27] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018.
- [28] Minas Katsiokalis, Lemonia Ragia, and Katerina Mania. Outdoors mobile augmented reality for coastal erosion visualization based on geographical data. In *Proceedings of the International Workshop on Cross-Reality (XR) Interaction co-located with 14th ACM International Conference on Interactive Surfaces and Spaces (ACM ISS 2020)*, volume 2779, pages 380–389. CEUR-WS.org, 2020.

- [29] Francesco Laera, Michele Fiorentino, Alessandro Evangelista, Antonio Boccaccio, Vito M. Manghisi, Joseph Gabbard, Michele Gattullo, Antonio E. Uva, and Mario M. Foglia. Augmented reality for maritime navigation data visualisation: a systematic review, issues and perspectives. *Journal of Navigation*, 74(5):1073–1090, 2021.
- [30] Johannes G. Leskens, Christian Kehl, Tim Tutenel, Timothy Kol, Gerwin de Haan, Guus Stelling, and Elmar Eisemann. An interactive simulation and visualization tool for flood analysis usable for practitioners. In *Mitigation and Adaptation Strategies for Global Change*, page 307–324, 2017.
- [31] Le Liu, Alexander P. Boone, Ian T. Ruginski, Lace Padilla, Mary Hegarty, Sarah H. Creem-Regehr, William B. Thompson, Cem Yuksel, and Donald H. House. Uncertainty visualization by representative sampling from prediction ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 23(9):2165–2178, 2017.
- [32] Wolfgang Maass, Arturo Castellanos, Monica C. Tremblay, Roman Lukyanenko, and Veda C. Storey. Ai explainability: Embedding conceptual models. In Niels Bjørn-Andersen, Roman Beck, Stacie Petter, Tina Blegind Jensen, Tilo Böhmann, Kai Lung Hui, and Viswanath Venkatesh, editors, *Proceedings of the 43rd International Conference on Information Systems, ICIS 2022, Digitization for the Next Generation, Copenhagen, Denmark, December 9-14, 2022*. Association for Information Systems, 2022.
- [33] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian J. Goodfellow. Adversarial autoencoders. *CoRR*, abs/1511.05644, 2015.
- [34] Grant McKenzie, Mary Hegarty, Trevor Barrett, and Michael Goodchild. Assessing the effectiveness of different visualizations for judgments of positional uncertainty. *International Journal of Geographical Information Science*, 30(2):221–239, 2016.
- [35] Carlo Metta, Andrea Beretta, Riccardo Guidotti, Yuan Yin, Patrick Gallinari, Salvatore Rinzivillo, and Fosca Giannotti. Improving trust and confidence in medical skin lesion diagnosis through explainable deep learning. *Int. J. Data Sci. Anal.*, 2023.
- [36] Carlo Metta, Andrea Beretta, Riccardo Guidotti, Yuan Yin, Patrick Gallinari, Salvatore Rinzivillo, and Fosca Giannotti. Advancing dermatological diagnostics: Interpretable ai for enhanced skin lesion classification. *Diagnostics*, 14(7), 2024.
- [37] Carlo Metta, Andrea Beretta, Roberto Pellungrini, Salvatore Rinzivillo, and Fosca Giannotti. Towards transparent healthcare: Advancing local explanation methods in explainable artificial intelligence. *Bioengineering*, 11(4), 2024.
- [38] Carlo Metta, Riccardo Guidotti, Yuan Yin, Patrick Gallinari, and Salvatore Rinzivillo. Exemplars and counterexemplars explanations for image classifiers, targeting skin lesion labeling. In *IEEE ISCC*, 2021.
- [39] Carlo Metta, Riccardo Guidotti, Yuan Yin, Patrick Gallinari, and Salvatore Rinzivillo. Exemplars and counterexemplars explanations for skin lesion classifiers. In *Frontiers in Artificial Intelligence and Applications*, volume 354, 2022.
- [40] Domenica Mirauda, Ugo Erra, Roberto Agatiello, and Marco Cerverizzo. Mobile augmented reality for flood events. *International Journal of Sustainable Development and Planning*, 13(3):418 – 424, 2018.

- [41] Timothy J Naegeli and Jason Laura. Back-projecting secondary craters using a cone of uncertainty. *Computers & geosciences*, 123:1–9, 2019.
- [42] Kjetil Nordby, Etienne Gernez, Synne Frydenberg, and Jon Olav Eikenes. Augmenting openbridge: An open user interface architecture for augmented reality applications on ship bridges. In *19th conference on computer applications and information technology in the maritime industries (COMPIT'20)*, 02 2020.
- [43] Marzan Tasnim Oyshi, Verena Maleska, Jochen Schanze, Franziskus Bormann, Raimund Dachzelt, and Stefan Gumhold. Floodvis: Visualization of climate ensemble flood projections in virtual reality. In *Workshop on Visualisation in Environmental Sciences (EnvirVis)*. The Eurographics Association, 2022.
- [44] Miguel Ponce-de Leon, Javier del Valle, José María Fernandez, Marc Bernardo, Davide Cirillo, Jon Sanchez-Valle, Matthew Smith, Salvador Capella-Gutierrez, Tania Gullón, and Alfonso Valencia. COVID-19 Flow-Maps an open geographic information system on COVID-19 and human mobility for Spain. *Scientific Data*, 8(1):310, Nov 2021.
- [45] Miguel Ponce-de Leon, Javier del Valle, José María Fernández, Marc Bernardo, Davide Cirillo, Jon Sanchez-Valle, Matthew Smith, Salvador Capella-Gutierrez, Tania Gullón, and Alfonso Valencia. COVID19 Flow-Maps daily cases reports, 2021.
- [46] Miguel Ponce-de Leon, Javier del Valle, José María Fernández, Marc Bernardo, Davide Cirillo, Jon Sanchez-Valle, Matthew Smith, Salvador Capella-Gutierrez, Tania Gullón, and Alfonso Valencia. COVID19 Flow-Maps daily-mobility for Spain, 2021.
- [47] Miguel Ponce-de Leon, Javier del Valle, José María Fernández, Marc Bernardo, Davide Cirillo, Jon Sanchez-Valle, Matthew Smith, Salvador Capella-Gutierrez, Tania Gullón, and Alfonso Valencia. COVID19 Flow-Maps population data, 2021.
- [48] Jerónimo Puertas, Luis Hernández-Ibáñez, Luis Cea, Manuel Regueiro-Picallo, Viviana Barneche-Naya, and Francisco-Alberto Varela-García. An augmented reality facility to run hybrid physical-numerical flood models. *Water*, 12, 2020.
- [49] Champika Ranasinghe, Nicholas Schiestel, and Christian Kray. Visualising location uncertainty to support navigation under degraded gps signals: a comparison study. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '19, New York, NY, USA, 2019. Association for Computing Machinery.
- [50] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen and. Improved techniques for training gans. In *NIPS*, 2016.
- [51] Froso Sarri, Lemonia Ragia, Antigoni Panagiotopoulou, and Katerina Mania. Location-aware augmented-reality for predicting sea level rise in situ. In *2022 International Conference on Interactive Media, Smart Systems and Emerging Technologies (IMET)*, pages 1–8, 2022.
- [52] Kai Schröter, Stefan Lüdtkke, Richard Redweik, Jessica Meier, Mathias Bochow, Lutz Ross, Claus Nagel, and Heidi Kreibich. Flood loss estimation using 3d city models and remote sensing data. In *Environmental Modelling & Software*, volume 105, pages 118–131, 2018.

- [53] Yusuf Sermet and Ibrahim Demir. Flood action vr: a virtual reality framework for disaster awareness and emergency response training. In *ACM SIGGRAPH 2019 Posters*. Association for Computing Machinery, 2019.
- [54] U. Sierau. *Dortmund's Urban Development Strategies for the Future*. Dortmund: Dortmund Municipality, 2005.
- [55] Aaron Springer and Steve Whittaker. Progressive disclosure: When, why, and how do users want algorithmic transparency information? *ACM Trans. Interact. Intell. Syst.*, 10(4):29:1–29:32, 2020.
- [56] Alexios Stavroulakis, Despina Dimelli, Michail Roumeliotis, and Aikaterini Mania. An augmented reality system architecture for flood management. In *In Proceedings of the 10th International Conference on Geographical Information Systems Theory, Applications and Management (GISTAM 2024)*, pages 211–218, 2024.
- [57] Hoang Thanh-Tung and Truyen Tran. Catastrophic forgetting and mode collapse in gans. In *IJCNN*, 2020.
- [58] Adam Tomkins and Eckart Lange. Interactive landscape design and flood visualisation in augmented reality. In *Multimodal Technologies and Interaction*, volume 3, 2019.
- [59] Ross Towe, Graham Dean, Liz Edwards, Vatsala Nundloll, Gordon Blair, Rob Lamb, Barry Hankin, and Susan Manson. Rethinking data-driven decision support in flood risk management for a big data age. *Journal of Flood Risk Management*, 13(4):e12652, 2020.
- [60] Uncertainty visualization.
- [61] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.
- [62] Guoyong Zhang, Jianhua Gong, Yi Li, Jun Sun, Bingli Xu, Dong Zhang, Jieping Zhou, Ling Guo, Shen Shen, and Bingxiao Yin. An efficient flood dynamic visualization approach based on 3d printing and augmented reality. *International Journal of Digital Earth*, 13(11):1302–1320, 2020.