Deploying an Artificial Intelligence Application to Detect Flood from Sentinel 1 Data

Paolo Fraccaro,[1] Nikola Stoyanov,[1] Zaheed Gaffoor,[5] Laura Elena Cue La Rosa,[2] Jitendra Singh,[3] Tatsuya Ishikawa,[4] Blair Edwards,[1] Anne Jones,[1] and Komminist Weldermariam[5]
[1]IBM Research Europe, Hartree Centre, Daresbury, United Kingdom
[2]IBM Research Brazil, Sao Paolo, Brazil
[3]IBM Research India, Gurgaon, India
[4]IBM Research Japan, Tokyo, Japan
[5]IBM Research Africa, Nairobi, Kenya
paolo.fraccaro@ibm.com

Abstract

As climate change is increasing the frequency and intensity of climate and weather hazards, improving detection and monitoring of flood events is a priority. Being weather independent and high resolution, Sentinel 1 (S1) radar satellite imagery data has become the go to data source to detect flood events accurately. However, current methods are either based on fixed thresholds to differentiate water from land or train Artificial Intelligence (AI) models based on only S1 data, despite the availability of many other relevant data sources publicly. These models also lack comprehensive validations on out-of-sample data and deployment at scale. In this study, we investigated whether adding extra input layers could increase the performance of AI models in detecting floods from S1 data. We also provide performance across a range of 11 historical events, with results ranging between 0.93 and 0.97 accuracy, 0.53 and 0.81 IoU, and 0.68 and 0.89 F1 scores. Finally, we show the infrastructure we developed to deploy our AI models at scale to satisfy a range of use cases and user requests.

Introduction

Climate change is increasing the frequency and severity of extreme weather events, which cause a significant disruption, financial losses, and risk to life. One of the consequences of such extreme weather events is an increased likelihood of flood events (Matgen et al. 2019). These events already represent the most frequent natural disaster and pose significant risk to socio-economic systems (e.g. communities, civil infrastructure, food systems, supply chains, etc.) as well as a major threat to human life (Willner, Otto, and Levermann 2018), with projected increases in future exposure due to socio-economic factors and climate change (Wing et al. 2018). For these reasons, when these events occur, it is extremely important to have access to accurate information on the flood extent and evolution to design adaptation and resilience strategies such as organise and plan effective responses (Nemni et al. 2020). This is particularly important during the event itself, but also after the event to assess damage and more broadly to prioritise adaptation strategies (e.g., maintenance operations) as well as assess future risk.

Over the last decades, satellite imagery from remote sensing, due to the increased availability and coverage, has become the most used approach to identify water at scale and identify as well as track flood events. Optical and Radar are the two main types of satellite imagery that are currently used for this task (Matgen et al. 2019).

Optical imagery is collected from passive sensors that measure the reflected energy of the sun against the earth in a specific area. Examples of this type of optical imagery are the ones coming from the MODIS (NASA 1999), LandSat (NASA 1972) and Sentinel 2 (The European Space Agency 2015) satellites. With optical imagery, water is usually identified based on the calculation of a water index, from one or multiple frequency bands of the collected imagery (Gao 1996). These indexes have then to be contextualised to a specific region by picking a threshold that better separates water and land in the area of interest. Despite the good performance shown by using optical imagery to identify and track floods, this approach has some key limitations. First, it is not weather independent, as it is severely affected by clouds and therefore not usable in some whether conditions. Second, being a passive sensor, it requires sun light, therefore it cannot be used at any time of the day.

The second approach for flood detection is based on radar imagery. This type of imagery is collected by active sensors, which emit radiations towards a target area and measure the radiations reflected back by the target. The most established source of radar imagery to date is represented by the Sentinel 1 (S1) mission from the European Space Agency (The European Space Agency 2014). S1 has a high resolution (i.e. 10 meters) global coverage and on average cadence of around a week. One particular advantage of S1 is that it is not affected by clouds and can be used at any time of the day. However, the identification of water from this data source is not straightforward. Water detection algorithms from S1 data take advantage of low backscatter signals, which, in ideal conditions, show water as a distinctly smooth surface with distinct boundaries (Matgen et al. 2019).

Traditional approaches aim to identify the ideal threshold to separate the water from the background and go from classic binarization methods like (Otsu 1979) to more complex approaches that for example implement tiling and fuzzy logic (Twele et al. 2016). However, due to the negative effect that topography, land use (e.g. buildings in urban areas or vegetation) and wind can have on the signal, often an intervention is still required from the user to tune parameters, limiting their automation and generalisability. To overcome these issues, artificial intelligence (AI) via the use of deep neural networks has been used to identify water from S1 images (Nemni et al. 2020; Bonafilia et al. 2020). Particularly, Nemni et al (Nemni et al. 2020) and Bonafilia et al (Bonafilia et al. 2020) both adopted a Unet architecture (Ronneberger, Fischer, and Brox 2015) for this task and showed great potential of their approach to be used in production and at scale. However, these papers reported performances on small out-of-sample sets. They did not assess whether the use of other data sources relevant to water detection from S1 could further improve performance. Moreover, to the best of our knowledge, neither of them developed a solution that could be easily used globally to track and identify floods and combining results with other relevant information operationally. Not only such solution would be extremely relevant for near-real time disaster management, but also as a validation tool for flood risk forecast from simulation models, which are extremely relevant to climate change but very difficult to validate because of the general lack of ground truth data.

In this paper, we address the above limitations using AI on S1 data to detect water and floods with a threefold contribution. First, we investigate whether using a Unet architecture (Ronneberger, Fischer, and Brox 2015) with the addition of more input channels containing information relevant to identify water from S1 data (e.g., Digital Elevation Model (DEM), land use, and permanent water masks) could increase performance. Second, we perform a comprehensive validation of the sen1floods11 dataset (Bonafilia et al. 2020) with a leave-one-out approach for each of the historical events included in the dataset, providing the community with a better idea of the generalizability of AI models in this domain. Third, we deploy our flood detection models on a big geospatial data and analytic platform implementing an infrastructure that allows easy integration of multiple data sources, AI models incremental development, flood forecast, and flood extent validation as well as operational contextualisation of computed risk.

Materials and Methods

Data Sources

Sen1floods11 Dataset We used the sen1floods11 dataset from Bonafilia et al (Bonafilia et al. 2020) to perform our study and experiments. An in depth description of this dataset is available at the original paper (Bonafilia et al. 2020), with only its main characteristics that are reported here. Particularly, the sen1floods11 dataset includes different types of labelled data (i.e. water extent) for 11 historical flood events around the world. We included the *S1 weakly labeled* data, derived by applying the Otsu binarization method (Otsu 1979) on Vertical-Horizontal (VH) S1 polarisation data for each event. We also considered the *Hand labeled* data, which was manually derived based on S1 and S2 data. Overall, the dataset contains 4,385 S1 weakly labeled tiles and 446 hand labeled tiles, with a shape of 512x512 pixels.

The Copernicus Programme The Copernicus Programme is an European Union programme that aims to achieve global, high quality, and wide-ranging Earth observation capacity (https://www.copernicus.eu/en). From this initiative we included: *Joint Research Commission (JRC) Global Surface Water dataset*, which includes high-resolution mapping of global surface water and its long-term changes at a 30 meters resolution (Pekel et al. 2016); *Global land cover dataset*, including classification of land use around a range of high level categories at 100 meters resolution (see below for details) (Copernicus 1998); *Shuttle Radar Topography Mission*, which provides elevation data globally at a 30 meters resolution (NASA 2000).

Data Management and Preprocessing

We used the IBM PAIRS Geospatial (Lu et al. 2016) to manage our data. PAIRS is an easy-to-use platform that enables assembly and retrieval of large geospatial datasets, and complex spatio-temporal analytics using these datasets. For example, PAIRS aligns geospatial information coming from different data sources that use different coordinates systems (as we are using here) to a common coordinate projection and resolution, making it easier for the user to intersect across several datasets with a unique query. We used this approach to bring all datasets to the same coordinate system and resolution of the sen1floods11 dataset labels.

Below we report on the main preprocessing steps that we undertook to create our dataset.

Sentinel 1 Data Pre-processing and Color Composite Creation The sen1floods11 dataset included S1 input data that were originally downloaded from Google Earth Engine (Google 2014). In our study, we used VV and VH S1 polarisation layers (e.g. which represent how the radar signal is transmitted and received by the antenna) acquired in Interferometric Wide Swath (IW) and provided as Level-1 Ground Range Detected (GRD) downloaded directly from ESA and uploaded to the PAIRS platform. The following preprocessing steps were applied: apply orbit file; GRD border noise removal; thermal noise removal; radiometric calibration; terrain flattening and terrain correction.

For each label tile from the sen1floods11 dataset we created the related VV and VH S1 polarisation tiles from PAIRS. These were then used to create a RGB color composite by using the VV and VH as the Red and Green channels, respectively. The Blue channel was calculated as the division between the absolute values of the VV and VH polarisations. This was done to take full advantage of transfer learning when training our models (see below).

Land Use Categorisation The land use dataset from Copernicus includes 23 different categories. Considering the difference in resolution between the S1 data and this input data as well as the similarities between some of the land use categories, we decided to introduce a more high level categorisation, collapsing some of the categories together.

The final categories were: Closed forest; Open forest; Other vegetation; Agricultural land; Urban. The Corpernicus land cover dataset includes also a permanent water layer, which we discarded. This was done because we included the JRC Global Surface Water dataset for characterising this piece of information, which is also openly available and with a higher resolution.

Joint Research Commission Global Surface Water Categorisation We selected the seasonality layer of the JRC Global Surface Water dataset, which represent the presence of water over a single year. The values of this data layer range from 0 to 12, with 0 being an area where no water was detected at any time during the year while 12 is an area where water was selected consecutively for 12 months over the year. We therefore binarized this data layer with all areas with a value equal to 12 representing permanent water, and considered all other pixels as background.

Implementation Details

Training, Validation and Test Datasets We followed a leave-one-out cross validation approach to structure our training, validation and test datasets. Particularly, we used each of the 11 events in the sen1floods11 dataset as an external hold out set (e.g. completely unseen data to the model), while training, validating and testing each time with a random 60:20:20 split on the rest of the data (e.g. data pertaining to the remaining 10 events). This means that if the hold out set would be the event in Bolivia, we would exclude this event completely, and use 60% of the remaining data for training our model, 20% for validation while training and 20% for testing on the training completed, alongside the data about the event in Bolivia. This is similar to the approach followed by Bonafilia et al (Bonafilia et al. 2020) in the original sen1floods11 paper, however they only used one of the events as a hold out set (i.e. historical event in Bolivia). We instead decided to use all events as a hold out sets to provide a more comprehensive evaluation of performance.

Model Structure and Training We used Pytorch (Paszke et al. 2019) via the fastai Python library (Howard and Gugger 2020) to perform all our analysis. Fastai provides modular components that can be easily taken out-of-the-box and customised to achieve state of the art results in standard deep learning domains. As Bonafilia et al (Bonafilia et al. 2020) and Nemni et al (Nemni et al. 2020), we used a Unet model architecture (Ronneberger, Fischer, and Brox 2015), with a Resnet backbone (He et al. 2016), as our model architecture. Conversely to what they did, we used the above mentioned S1 RGB color composite as our main input layer (i.e. S1 model), to which we added each of the extra input data layers described above in isolation (i.e. S1 + DEM model, S1 + Land use, S1 + JRC) and combination (i.e. S1 + all others). Heuristically, we derived a learning rate of $5 * 10^{-4}$, which we used across all analysis, alongside a weight decay of 0.1. We used a batch size of 16, and took advantage of the fit one cycle policy implemented by the fastai library (Howard and Gugger 2020). This has been shown to speed up training and increasing performance for small training sets (Smith and Topin 2019). To avoid overfitting and maximise information of our training set, to each of the input layers we applied a range of data augmentation transformations (i.e. random horizontal and vertical flip, random shift and scaling and random brightness contrast) via the albumentations library (Buslaev et al. 2020). Finally, we used the fastai Ranger optimiser, which has also been proven to outperform more classical optimizers in a range of scenarios (Wright 2019), and a Cross Weighted Entropy Loss function. This was used due to the higher prevalence of background pixels (i.e. 89.6% and 83.4% in the Hand labeled and S1 Weakly labeled), to which a lower weight was given following the approach from Nemni et al (Nemni et al. 2020).

For each model, we trained by using transfer learning from the Resnet Imagenet model provided by fastai for the weights of the first three data layers input (e.g. S1 RGB color composite) for 30 and 20 epochs for the models trained on the Hand Labeled and S1 Weakly labeled data, respectively. At the end of these epochs, the model scoring the best performance on the validation set was selected, and further trained for a following 15 and 10 epochs for the Hand labeled and S1 Weakly labeled data, respectively. We performed analyses on a single node with 4 GPUs and 16 cores, with an epoch time of around 90 seconds.

We calculated model performance in terms of accuracy, Intersection over Union (IoU) and F1 score, which for each iteration of the leave-one-out cross validation was assessed on the test set and hold out set. For each metric we calculated a weighted mean based on the number of pixels that each test and hold out set had. These metrics will be defined as All Water (AW) results. Furthermore, we used the JRC Global Surface Water dataset to stratify our results by Permanent Water (PW) and Flood Water (FW).

Model Deployment Infrastructure

Figure 1 shows the high-level pipeline services we developed to operationalise our AI-based flood detection models. The solution can be used as a stand-alone flood extent map generator or to validate predictions from other models (e.g. flood risk forecast) against historical events or third-party ground truth. Users submit their requests via a REST API service in the form of jobs that run on a Red Hat Openshift cluster. Each job request consists of geospatial and temporal information about the area and time period of interest. First, the system always produces the flood map extent. This is done via a Docker container that implements a three-step workflow: preprocess with data coming directly from IBM PAIRS, inference and postprocess. This approach allows to easily swap one of these steps (e.g. because a better model is available or a different architecture is tested) for fast development and prototyping. The workload to produce the flood maps varies across use cases (e.g. because of the spatio-temporal requirements). The system deals with fluctuating computing requirements by implementing a batch processing architecture and on-demand scaling in Openshift. Second, the system stores the output to IBM PAIRS for downstream applications such as IBM Environmental Intelligence Suite (EIS) (IBM 2021) and IBM Cloud Object Storage (COS) for further processing, or to run a validation task.

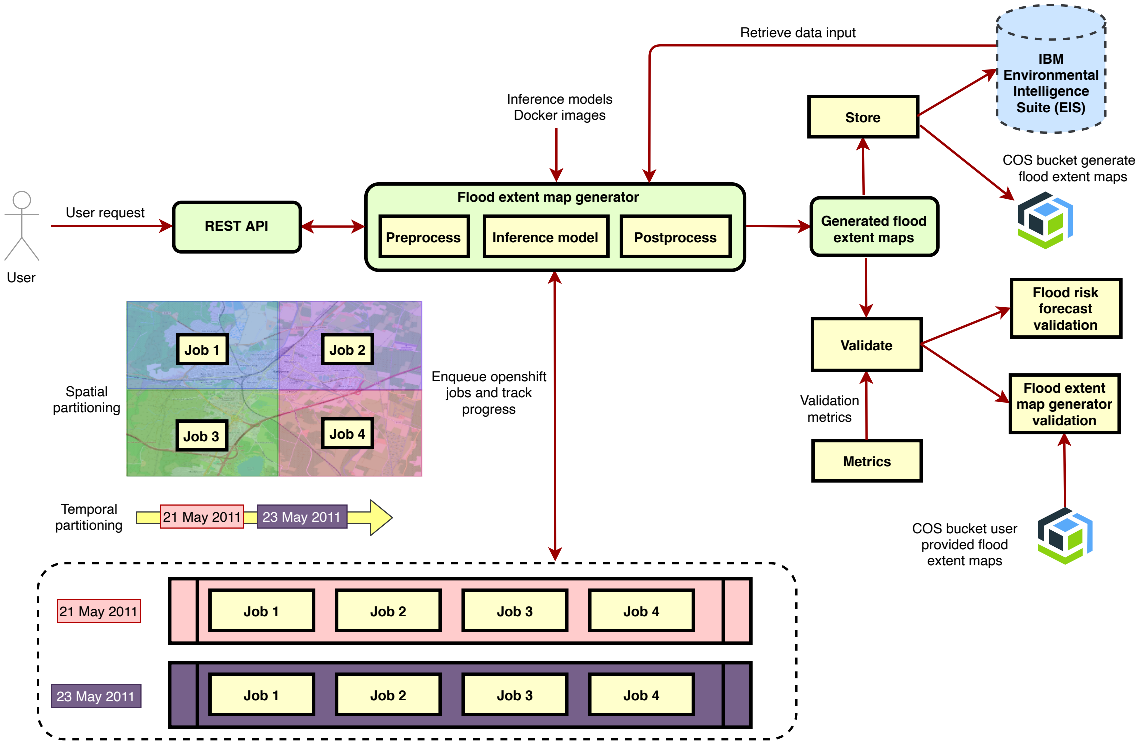

Retrieve data input
IBM Environmental Intelligence Suite (EIS)
Store
Inference models Docker images
COS bucket generated flood extent maps
User
User request
REST API
Flood extent map generator
Preprocess
Inference model
Postprocess
Generated flood extent maps
Job 1
Job 2
Job 3
Job 4
Spatial partitioning
Enqueue openshift jobs and track progress
Validate
Flood risk forecast validation
Flood extent map generator validation
Validation metrics
Metrics
Temporal partitioning
21 May 2011
23 May 2011
COS bucket user provided flood extent maps
21 May 2011
Job 1
Job 2
Job 3
Job 4
23 May 2011
Job 1
Job 2
Job 3
Job 4

Figure 1: The high-level architecture for the REST API service for inference with temporal and spatial partitioning.

Each validation task can use the generated ground truth as a target against the output of another flood risk model or user provided ground truth to test the performance of the flood extent map generator.

To showcase this infrastructure and further validate the generalizability of the developed models, we tested our AI models on a completely unseen historical event (i.e. not part of the sen1floods11 dataset), and calculated performance against ground truth obtained from the Copernicus Emergency Mapping Service (European Commission 2021). This event pertains to the area around Aude (France) on the 15 October 2018, and it was chosen because of its topological variety and presence of different types of flood during the event (e.g. pluvial and fluvial). In line with Copernicus that excludes PW areas, we focused only on the FW areas.

Results

Model Performance

Table 1 presents the results from testing the Resnet Unet models across the different sen1floods11 labels and input combination over the hold out sets (e.g. one event left out from training each time). Overall, adding more input layers did not improve performance of the baseline model for the Hand Labeled data, which in same cases had the best performance. For the S1 Weakly labeled data, some improvement could be seen by adding the PW layer to the S1 data that led to a 1% improvement for accuracy and F1 score and 2% for IoU. For context, our best model IoU performance on the same hold out set used by Bonafilia et al (Bonafilia et al. 2020) (i.e. historical event in Bolivia) was 0.59 and 0.80 in AW detection for Hand and S1 labeled data, respectively. All results pertaining to the other analyses we performed (e.g. on 20% test set) confirmed the trends described above, with performance on the test set that was generally higher than the one in the hold out sets. All results are available on request.

Model Deployment

Figure 2 shows the comparison between the ground truth data coming from Copernicus and our S1 only AI model for the Aude event, obtained by running our model. Overall, there is good correspondence between the two flood extent plots, with the AI model that seems to identify more continuous water areas and the Copernicus one being more fragmented. In terms of quantitative performance on FW, accuracy was 0.98, IoU 0.49 and F1 score 0.66. Figure 3 shows the AI model prediction used as a filter on the IBM EIS platform, where affected areas that are more densely populated can be easily identified in red.

Discussion

We investigated the added value of including additional geospatial datasets input layers to detect flood from S1 data using AI. These extra input layers only improved prediction marginally and for only one type of data labels we analysed. Overall, we showed that out of bag best performance across the 11 locations ranged between 0.93 and 0.97 accuracies, 0.53 and 0.81 IoU, and 0.68 and 0.89 F1 scores. Finally, we

Label	Metric	Type	S1	S1 + DEM	S1 + Land use	S1 + PW	S1 + all others
Hand labeled	Accuracy	PW	0.95 (0.08)	0.93 (0.06)	0.95 (0.06)	0.95 (0.07)	**0.96 (0.06)**
		FW	**0.93 (0.04)**	0.92 (0.03)	**0.93 (0.03)**	0.91 (0.04)	0.90 (0.05)
		AW	**0.93 (0.04)**	0.92 (0.03)	**0.93 (0.03)**	0.91 (0.04)	0.91 (0.05)
	IoU	PW	0.94 (0.09)	0.93 (0.06)	**0.95 (0.06)**	**0.95 (0.08)**	**0.95 (0.06)**
		FW	**0.43 (0.16)**	0.4 (0.15)	0.40 (0.14)	0.37 (0.17)	0.37 (0.18)
		AW	**0.53 (0.14)**	0.5 (0.14)	0.52 (0.13)	0.47 (0.19)	0.47 (0.19)
	F1 Score	PW	0.97 (0.06)	0.96 (0.04)	0.97 (0.04)	0.97 (0.04)	**0.98 (0.04)**
		FW	**0.58 (0.15)**	0.56 (0.15)	0.56 (0.14)	0.52 (0.18)	0.51 (0.2)
		AW	**0.68 (0.13)**	0.66 (0.12)	0.67 (0.11)	0.62 (0.18)	0.62 (0.19)
S1 weakly labeled	Accuracy	PW	0.94 (0.04)	**0.97 (0.02)**	**0.97 (0.02)**	**0.97 (0.02)**	**0.97 (0.02)**
		FW	**0.97 (0.02)**	**0.97 (0.02)**	0.96 (0.02)	**0.97 (0.02)**	0.96 (0.02)
		AW	0.96 (0.02)	**0.97 (0.02)**	0.96 (0.02)	**0.97 (0.02)**	0.96 (0.02)
	IoU	PW	0.94 (0.04)	**0.97 (0.02)**	**0.97 (0.02)**	**0.97 (0.02)**	**0.97 (0.02)**
		FW	0.73 (0.13)	0.72 (0.18)	0.71 (0.14)	**0.74 (0.13)**	0.72 (0.13)
		AW	0.79 (0.09)	0.79 (0.13)	0.79 (0.08)	**0.81 (0.08)**	0.80 (0.08)
	F1 Score	PW	0.97 (0.02)	**0.99 (0.01)**	0.98 (0.01)	0.98 (0.01)	0.98 (0.01)
		FW	**0.84 (0.09)**	0.82 (0.13)	0.82 (0.11)	**0.84 (0.09)**	0.83 (0.1)
		AW	0.88 (0.06)	0.88 (0.09)	0.88 (0.05)	**0.89 (0.05)**	0.88 (0.05)

Table 1: Performance from the different tested models on the hold out sets. In bold models that performed best performance for a specific metric and analysis. (PW: Permanent water; FW: Flood water; AW: All water)

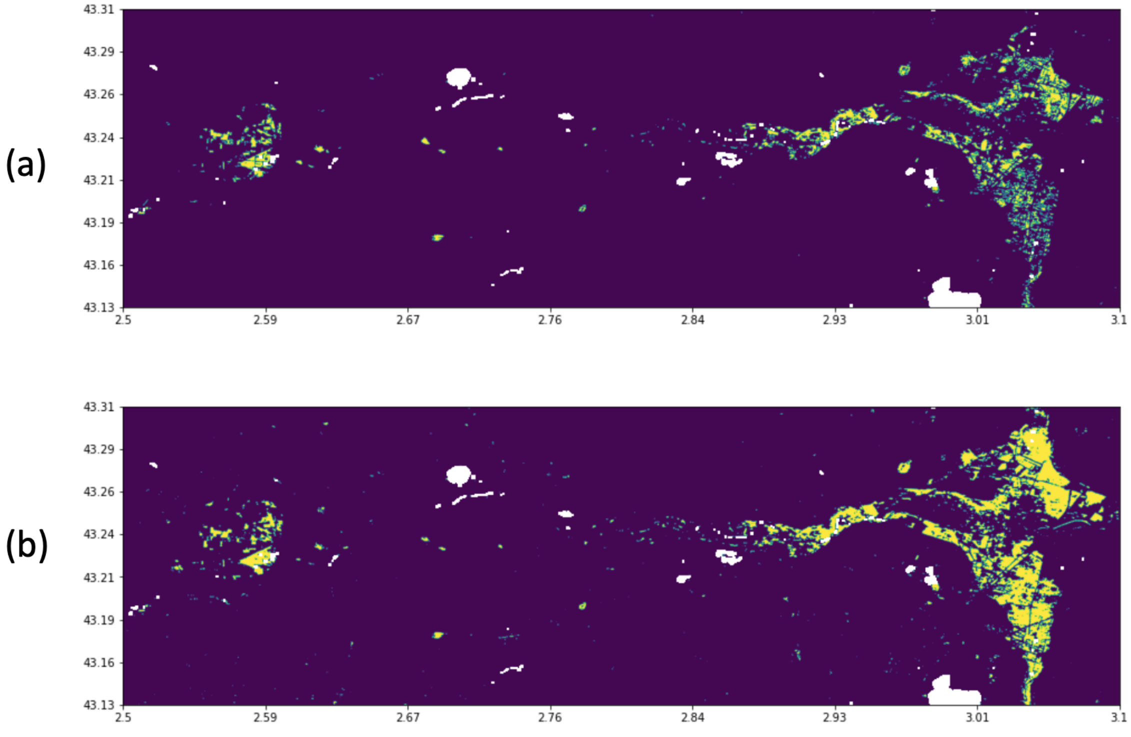
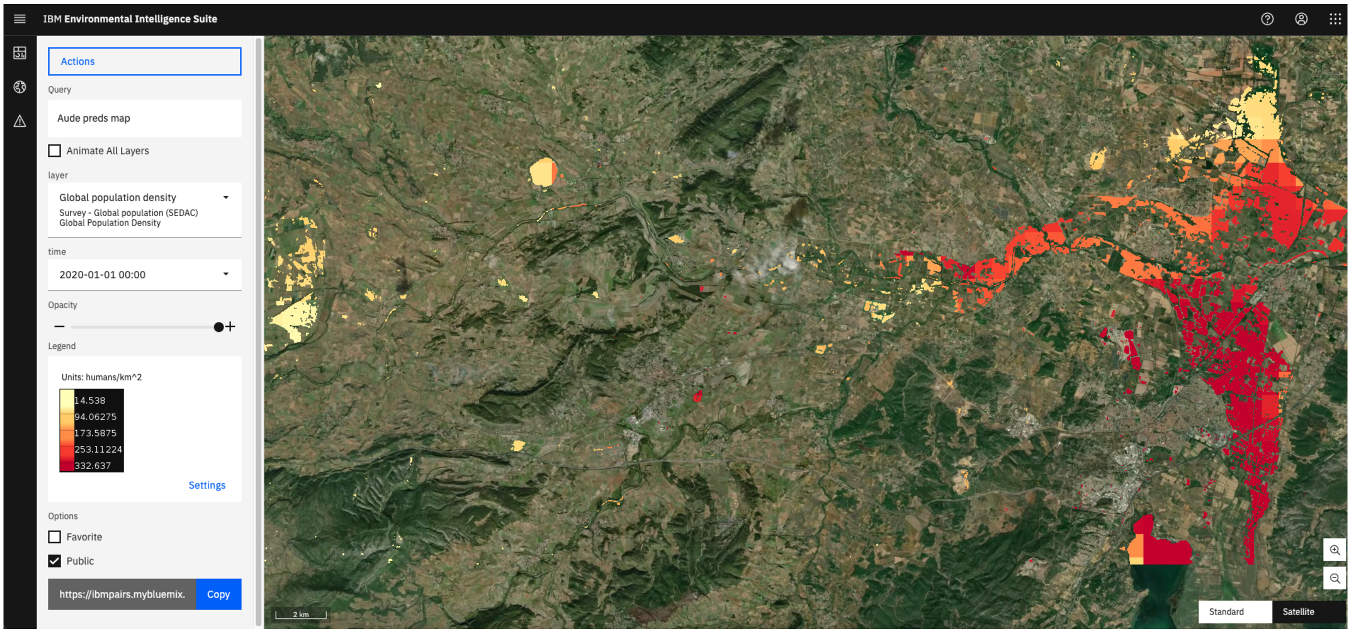


Figure 2: Comparison between the ground truth data coming from Copernicus (a) and our S1 only AI model (b) for the Aude event. Purple indicates background, while yellow indicates water. White areas identify PW areas, which have been mapped out from the results.



Query: Aude preds map

□ Animate All Layers

layer: Global population density
Survey - Global population (SEDAC) Global Population Density

time: 2020-01-01 00:00

Opacity

Legend

Units: humans/km^2

14.538
94.06275
173.5875
253.11224
332.637

Options

□ Favorite

☑ Public

https://ibmpairs.mybluemix.

Figure 3: AI Model output used as a filter to identify highly populated areas affected by the flood on the IBM EIS. More intense red colour represents higher populated areas.

showed how we deployed our model at scale to meet requirements for a range of use cases.

Despite not leading to major improvements in performance, to the best of our knowledge, this is the first study assessing the introduction of additional open access input layers an AI model to detect water from S1 data satellite imagery. Although optimal performance was not their goal, our results improve on the performance reported by Bonafilia et al (Bonafilia et al. 2020) on the sen1floods11 dataset, where they reported an IoU of 0.35 and 0.39 in detecting all water for the Hand labeled and S1 weakly labelled data for their hold out set that was only based on the event in Bolivia. In addition to explore new approaches to improve performance, our study provided a more comprehensive overview of the models performance and their generalizability by implementing a full leave-one-out cross validation procedure.

The system we deployed via the IBM EIS provides some advantages over platforms that are currently available to map flood extent from satellite images. First, it uses S1 data at 10 meters resolution, which is higher than systems like FloodScan ran by the US company Atmospheric and Enviromental Research (i.e. 90 meters resolution) or NASA via the MODIS surface water product (i.e. 250 meters). Second, the infrastructure we adopted allows scalability as well as AI models incremental development. This is obtained by modularising each step of the AI flood detection model (i.e. preprocessing, model inference and postprocessing) that can be therefore easily be swapped for an improved version or an entirely different strategy (e.g. traditional remote sensing approach) being still part of the same overall system. Third, the system provides the possibility of easily validating flood risk forecast against flood extent for historical events, which is something that, to the best of our knowledge, is not available elsewhere. Finally, the deployment of the AI model as part of the IBM EIS enables the integration of additional data sources, both to improve model prediction accuracy and to better quantify flood impact: for example, by integrating the areas affected by the flood with specific assets location, population density or others via a user-friendly dashboard.

Despite the comprehensive validation of AI models to detect water and flood we provided, the addition of extra input layers did not substantially improve the performance of models using only S1 input data. One of the reasons why this might be is the difference in resolution between the S1 data itself and the extra input layers we added (e.g. 30 meters and 100 meters versus 10 meters), which led to information repetition in the AI model inputs. In principle, this should lead to a more significant impact on performance. Although such higher resolution extra layers are not available globally, refined versions of our approach could be implemented where they are present. Furthermore, additional features that are relevant to flood risk analysis could be added to the input layers. For example, slope or topographic wetness index could be explicitly derived from the DEM and passed to the model as an input or used as a post-processing tool to decrease false positives. Moreover, completely new information could be added to the input layers. This could be flood risk specific like soil type, which provides key information on the likelihood of an area to flood. Finally, different learning algorithms or deep network model architectures for semantic segmentation different from the Resnet-Unet could be tried and adapted to work with more than just three channels. For the future, we will pursue these avenues, and, thanks to the modular architecture we implemented, aim at deploying an improved model to increase impact on real world industrial applications.

Acknowledgements

This work was supported by the Hartree National Centre for Digital Innovation, a collaboration between STFC and IBM.

References

Bonafilia, D.; Tellman, B.; Anderson, T.; and Issenberg, E. 2020. Sen1Floods11: a georeferenced dataset to train and test deep learning flood algorithms for Sentinel-1. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 210–211.

Buslaev, A.; Iglovikov, V. I.; Khvedchenya, E.; Parinov, A.; Druzhinin, M.; and Kalinin, A. A. 2020. Albumentations: Fast and Flexible Image Augmentations. *Information*, 11(2).

Copernicus. 1998. Copernicus Global Land Service. https://land.copernicus.eu/global/products/lc. Accessed: 2021-09-01.

European Commission. 2021. Emergency Copernicus Services. https://emergency.copernicus.eu/. Accessed: 2021-09-01.

Gao, B.-C. 1996. NDWI—A normalized difference water index for remote sensing of vegetation liquid water from space. 58(3): 257–266. Publisher: Elsevier.

Google. 2014. Google Earth Engine - Sentinel 1 preprocessing. https://developers.google.com/earth-engine/guides/sentinel1. Accessed: 2021-09-01.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Howard, J.; and Gugger, S. 2020. Fastai: A layered API for deep learning. *Information*, 11(2): 108. Publisher: Multidisciplinary Digital Publishing Institute.

IBM. 2021. IBM Environmental Intelligence Suite (EIS). https://www.ibm.com/products/environmental-intelligence-suite. Accessed: 2021-12-01.

Lu, S.; Shao, X.; Freitag, M.; Klein, L. J.; Renwick, J.; Marianno, F. J.; Albrecht, C.; and Hamann, H. F. 2016. IBM PAIRS curated big data service for accelerated geospatial data analytics and discovery. *2016 IEEE International Conference on Big Data (Big Data)*, 2672–2675.

Matgen, P.; Martinis, S.; Wagner, W.; Freeman, V.; Zeil, P.; and Mc-Cormick, N. 2019. Feasibility assessment of an automated, global, satellite-based flood-monitoring product for the Copernicus Emergency Management Service. *European Commission, Ispra, JRC119812, EUR*, 30073: 1–47.

NASA. 1972. Landsat. https://landsat.gsfc.nasa.gov/about. Accessed: 2021-09-01.

NASA. 1999. Moderate Resolution Imaging Spectroradiometer. https://modis.gsfc.nasa.gov/about/. Accessed: 2021-09-01.

NASA. 2000. Shuttle Radar Topography Mission. https://www2.jpl.nasa.gov/srtm/. Accessed: 2021-09-01.

Nemni, E.; Bullock, J.; Belabbes, S.; and Bromley, L. 2020. Fully convolutional neural network for rapid flood segmentation in synthetic aperture radar imagery. *Remote Sensing*, 12(16): 2532. Publisher: Multidisciplinary Digital Publishing Institute.

Otsu, N. 1979. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1): 62–66. Publisher: IEEE.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; and Antiga, L. 2019. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*.

Pekel, J.-F.; Cottam, A.; Gorelick, N.; and Belward, A. S. 2016. High-resolution mapping of global surface water and its long-term changes. *Nature*, 540(7633): 418–422. Publisher: Nature Publishing Group.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*, 234–241.

Smith, L. N.; and Topin, N. 2019. Super-convergence: Very fast training of neural networks using large learning rates. *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, 11006: 1100612.

The European Space Agency. 2014. Sentinel-1 Mission. https://sentinel.esa.int/web/sentinel/missions/sentinel-1. Accessed: 2021-09-01.

The European Space Agency. 2015. Sentinel-2 Mission. https://sentinel.esa.int/web/sentinel/missions/sentinel-2. Accessed: 2021-09-01.

Twele, A.; Cao, W.; Plank, S.; and Martinis, S. 2016. Sentinel-1-based flood mapping: a fully automated processing chain. *International Journal of Remote Sensing*, 37(13): 2990–3004. Publisher: Taylor & Francis.

Willner, S. N.; Otto, C.; and Levermann, A. 2018. Global economic response to river floods. *Nature Climate Change*, 8(7): 594–598. Publisher: Nature Publishing Group.

Wing, O. E. J.; Bates, P. D.; Smith, A. M.; Sampson, C. C.; Johnson, K. A.; Fargione, J.; and Morefield, P. 2018. Estimates of present and future flood risk in the conterminous United States. *Environmental Research Letters*, 13(3): 034023.

Wright, L. 2019. New Deep Learning Optimizer, Ranger: Synergistic combination of RAdam + LookAhead for the best of both. https://lessw.medium.com/new-deep-learning-optimizer-ranger-synergistic-combination-of-radam-lookahead-for-the-best-of-2dc83f79a48d. Accessed: 2021-09-01.