Geophysical Journal International

Geophys. J. Int. (2024) **238,** 382–399
Advance Access publication 2024 April 24
GJI Applied and Marine Geophysics

https://doi.org/10.1093/gji/ggae151

Machine learning emulation of high resolution inundation maps

Erlend Briseid Storrøsten [1], Naveen Ragu Ramalingam [2], Stefano Lorito [3], Manuela Volpe [3], Carlos Sánchez-Linares [4], Finn Løvholt [1] and Steven J. Gibbons [1]

[1] *The Norwegian Geotechnical Institute, Sognsveien 72, 0855 Oslo, Norway . E-mail: Erlend.Briseid.Storrosten@ngi.no*
[2] *The University School for Advanced Studies - IUSS Pavia, 27100 Pavia, Italy*
[3] *Istituto Nazionale di Geofisica e Vulcanologia, Osservatorio Nazionale Terremoti, Via di Vigna Murata, 605, 00143 Roma, Italy*
[4] *Departamento de Matemática Aplicada, Universidad de Málaga, 29010 Málaga, Spain*

Accepted 2024 April 12. Received 2024 March 4; in original form 2023 November 17

SUMMARY

Estimating coastal tsunami impact for early-warning or long-term hazard analysis requires the calculation of inundation metrics such as flow-depth or momentum flux. Both applications require the simulation of large numbers of scenarios to capture both the aleatory variability and the epistemic tsunami uncertainty. A computationally demanding step in simulating inundation is solving the non-linear shallow water (NLSW) equations on meshes with sufficiently high resolution to represent the local elevation accurately enough to capture the physics governing the flow. This computational expense is particularly challenging in the context of Tsunami Early Warning where strict time constraints apply. A machine learning (ML) model that predicts inundation maps from offshore simulation results with acceptable accuracy, trained on an acceptably small training set of full simulations, could replace the computationally expensive NLSW part of the simulations for vast numbers of scenarios and predict inundation rapidly and with reduced computational demands. We consider the application of an encoder–decoder based neural network to predict high-resolution inundation maps based only on more cheaply calculated simulated time-series at a limited number of offshore locations. The network needs to be trained using input offshore time-series and the corresponding inundation maps from previously calculated full simulations. We develop and evaluate the ML model on a comprehensive set of inundation simulations for the coast of eastern Sicily for tens of thousands of subduction earthquake sources in the Mediterranean Sea. We find good performance for this case study even using relatively small training sets (order of hundreds) provided that appropriate choices are made in the specification of model parameters, the specification of the loss function and the selection of training events. The uncertainty in the prediction for any given location decreases with the number of training events that inundate that location, with a good range of flow depths needed for accurate predictions. This means that care is needed to ensure that rarer high-inundation scenarios are well-represented in the training sets. The importance of applying regularization techniques increases as the size of the training sets decreases. The computational gain of the proposed methodology depends on the number of complete simulations needed to train the neural network, ranging between 164 and 4196 scenarios in this study. The cost of training the network is small in comparison with the cost of the numerical simulations and, for an ensemble of around 28 000 scenarios, this represents a 6- to 170-fold reduction in computing costs.

Key words: Machine learning; Tsunamis; Tsunami warning.

1 INTRODUCTION

Tsunamis pose potentially devastating consequences to coastal populations, and may inundate several kilometres inland far from their origin (e.g. Mori *et al*. 2022). Numerical simulations are essential to tsunami hazard assessment. Inundation is usually modelled by solving the non-linear shallow water (NLSW) equations (LeVeque & George 2008; de la Asunción *et al*. 2013a; Behrens & Dias 2015) on high-resolution digital elevation models (DEMs). This is typically performed using a nested or telescopic grid with increasingly fine spatial resolution and is normally the part of the simulation dominating the time-to-solution. In forecasting, such as

Probabilistic Tsunami Hazard Assessment (PTHA, e.g. Geist & Parsons 2006; Grezio *et al*. 2017) or tsunami early warning [e.g. Probabilistic Tsunami Forecasting (PTF), Selva *et al*. 2021], considerable uncertainty surrounds to the earthquake source. Quantifying this uncertainty demands a Monte Carlo-type analysis encompassing ensembles of many scenarios. An adequate representation of source variability can demand ensembles containing from thousands to millions of scenarios (e.g. Selva *et al*. 2016). For large ensembles, the necessary number of high resolution NLSW calculations needed can render the task computationally infeasible. It is only recently, with great advances in computational resources and efficient and optimized codes, that it has become at all possible (e.g. Gibbons *et al*. 2020).

The need to overcome the computational cost associated with solving the NLSW equations is long established. One option is to reduce the number of simulations by a careful subselection of scenarios. The properties of high-resolution inundation maps are linked to other observables in the tsunami modelling process, such as offshore wave heights or low-resolution inundation maps, and these parameters may guide the selection of scenarios (Lorito *et al*. 2015; Sepúlveda *et al*. 2017; Volpe *et al*. 2019; Williamson *et al*. 2020; Davies *et al*. 2022). For tsunami early warning, similar principles can be applied with databases of precomputed inundation scenarios. Predictions are made by selecting the most appropriate scenario in the database by matching offshore time-series (Gusman *et al*. 2014; Setiyono *et al*. 2017; Tanioka & Gusman 2018) or low resolution inundation grids (Mulia *et al*. 2018). Increasingly, there has been a move towards the application of machine learning (ML) to estimate directly near-shore time-series or inundation (e.g. Mulia *et al*. 2020; Fauzi & Mizutani 2020; Liu *et al*. 2021; Rodríguez *et al*. 2022; Kamiya *et al*. 2022). Direct approaches based on Gaussian Processes have also been applied (Salmanidou *et al*. 2017; Fukutani *et al*. 2021, 2023; Tozato *et al*. 2022). The potential of ML to predict inundation metrics rapidly from sensor data or simulation output has led to its implementation in 'end-to-end' workflows aimed at early warning based in closer-to-source measurements (e.g. Makinoshima *et al*. 2021; Núñez *et al*. 2022; Rim *et al*. 2022; Mulia *et al*. 2022).

Here we seek to use ML to reduce the cost of a single simulation so that sufficient numbers can be performed, either within an available time-frame or with the available computational resources. We assume that the benchmarked NLSW model simulations accurately reproduce the inundation for each tsunami scenario. The least expensive part of an NLSW calculation is the offshore wave propagation on the coarsest of the nested grids, from which we record offshore time-series at locations with a water depth close to 50 m (a depth at which the linear shallow water approximation holds reasonably well). The high resolution inundation simulation on the finer grids is significantly more expensive computationally (the computational time increases by a factor 8 for a factor 2 reduction in grid size). Fig. 1(a) displays an inundation calculation for the coastline of Eastern Sicily near Catania, resulting from a large subduction earthquake, using a 4-level system of nested grids. Our hypothesis is that, given an adequate training set of inundation calculations from the complete nested-grid simulations, we can predict inundation maps using ML from the offshore time-series alone (Fig. 1b). Given sufficiently accurate predictions, and a sufficiently rapid and efficient training process, we would be able to reduce greatly the time-to-solution for tsunami simulations. This is a goal in itself for PTF (Selva *et al*. 2021) in the Urgent Computing mode where a large set of numerical simulations are conducted on the fly (Løvholt *et al*. 2019; Ejarque *et al*. 2022; Folch *et al*. 2023). However, even without critical time constraints, a reduction of computational expense would increase the number of calculations that can be performed for the same computational cost. We emphasize that the ML approach presented here is specific to a given stretch of coastline as the inundation for a given offshore wave input is highly sensitive to local topo-bathymetry. Every stretch of coastline for which the hazard analysis is performed will therefore require a number of full simulations to be performed for generating training sets. For this approach to have a significant advantage, the training set must be far smaller than the number of inundation maps required for each specific site of interest.

We investigate encoder–decoder type models that can represent geometrically complex spatial patterns with a latent space of relatively small dimension (see Fig. 2). In the example displayed, our higher dimensional input is also a high resolution inundation map. However, this input could also be the offshore time-series calculated in the inundation calculations. The encoder transforms the time-series to the low-dimensional latent space and the decoder predicts the inundation map from these parameters. We establish a set of criteria by which success of such a process can be evaluated:

(i) We demand a significant improvement in the time-to-solution relative to the full numerical simulation: our primary motivation.

(ii) We demand that it be possible using a relatively modest training set. Each member of the training set is a complete numerical simulation and, given that the procedure is site-specific, an overwhelming number of necessary training events would defeat the objective.

(iii) We demand an acceptable level of accuracy in the predictions. Significant underestimates or overestimates of the inundation are equally undesirable.

(iv) We demand that the model works well in the tails of the distribution (the long-tail problem). The tsunamis that generate the most significant inundation are at the high-impact, low-probability, end of the distribution. An ML model trained on an event set culled from the higher probability portion of the scenarios would likely lack a basis on which to estimate the more extreme inundation.

Our tasks are to determine the extent to which an ML tsunami inundation model or emulator can fulfill these criteria, to find optimal methods and model architectures, and to determine the demands on the training sets. We anticipate criterion (i); for a trained model, an ML prediction will likely be rapid compared with the numerical simulation. Criteria (ii) and (iii) are competing aims; gains in one will likely lead to losses in the other. (The predictive capability of any learning algorithm is subject to the Bias-variance trade-off: overfitting versus underfitting.) Criterion (iv) puts even harsher demands on the size of the training set and the success of ML prediction of tsunami inundation will lie in finding an optimal balance between criterion (ii) and the demands (iii) and (iv).

The data set consists of the subduction earthquake scenarios for the seismic PTHA of Gibbons *et al*. (2020): 27 985 scenarios in total. The geographical distribution of the subduction earthquakes is displayed in Fig. 3 together with histograms for the magnitude distributions for the different source regions. The scenarios were selected based upon a hazard disaggregation from the NEAMTHM18 Tsunami Hazard Model (Basili *et al*. 2021) and feature stochastic slip distributions on triangular meshes modelling the subduction zones. For each of the main subduction scenarios, there are several stochastic realizations of the slip distributions; the form of the slip distribution can affect the impact significantly (e.g. Melgar *et al*. 2019; Davies 2019). The simulations were carried out using the Tsunami-HySEA model (de la Asunción *et al*. 2013b) within the

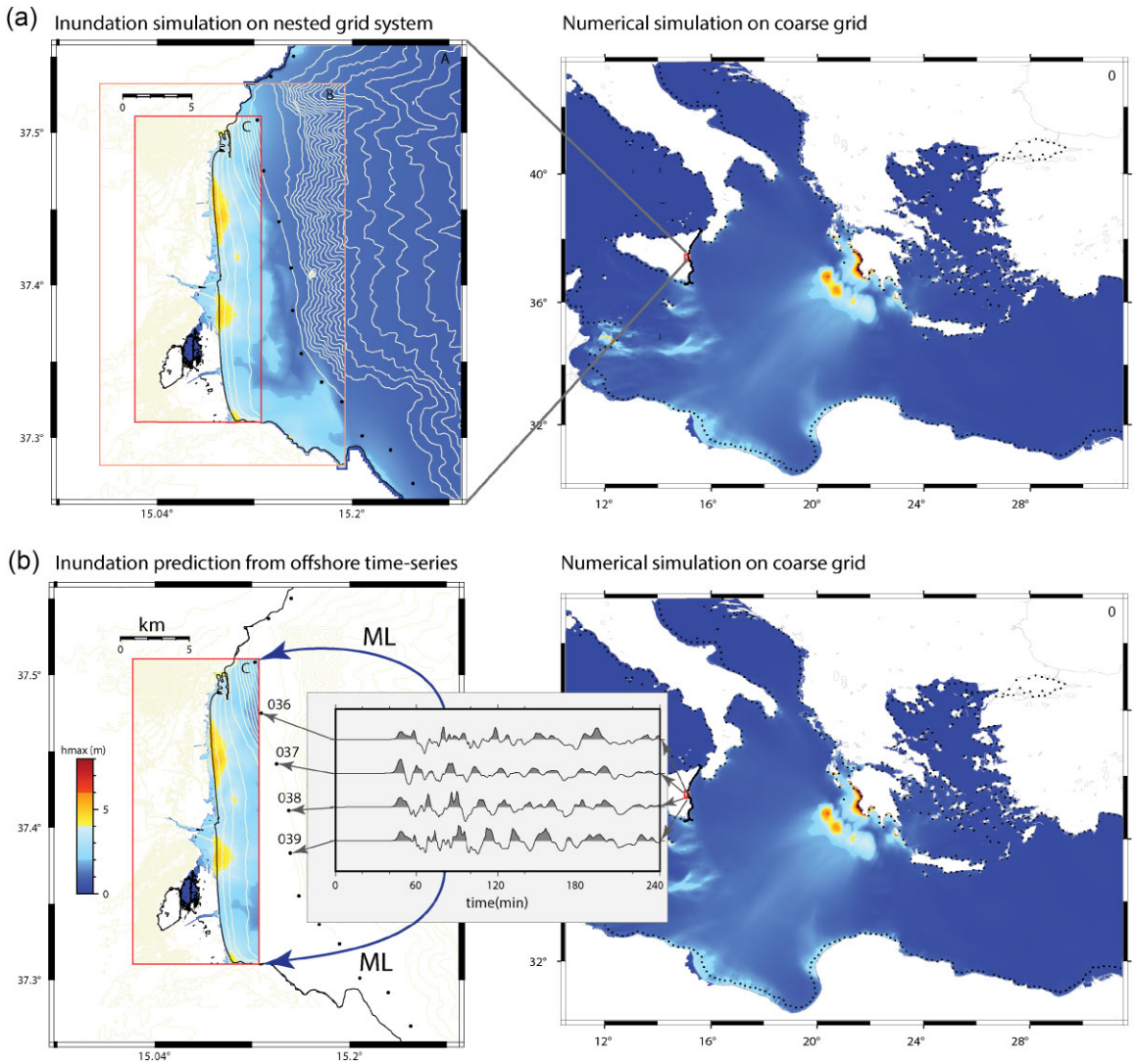

Figure 1. Estimation of high-resolution tsunami inundation using (a) a full numerical simulation on a four-level system of nested grids and (b) using a numerical simulation on a single level with the inundation prediction map calculated using Machine Learning, with the offshore time-series as input, where the inundation patterns have been learned from a training set of a limited number of full simulations. The colours in all panels indicate the maximum water height throughout the duration of the simulation of a subduction earthquake in the Hellenic Arc. This is a scenario from the PTHA study of Gibbons *et al.* (2020) and the black symbols indicate the locations of the time-series outputs for these simulations. These locations are separated by between 2 and 4 km and are approximately on the 50m depth isobath. Time-series from a total of 16 locations were exploited in the current study: the 13 locations visible and 3 just outside of the region displayed. The grids labelled 0, A, B and C have resolution 640 m, 160 m, 40 m and 10 m, respectively. The contour lines in panels (a) and (b) indicate elevation/depth with intervals of 200 m, 50 m and 5 m in grids A, B and C respectively. The numbers in panel (b) are the indices of the offshore time-series locations.

ChEESE project (Center of Excellence for Exascale in the Solid Earth: Folch *et al.* 2023). Its GPU-accelerated framework allows faster-than-real-time (FTRT) implementation, suitable for Tsunami Early Warning Systems (TEWS). Extensive testing and validation have previously been conducted (Macías *et al.* 2017, 2020a, b) against laboratory tests and benchmark problems (Synolakis *et al.* 2008) for its use in tsunami propagation and inundation studies. The relatively high number of high resolution inundation calculations make this data set ideal for studying sensitivity to the size of the training sets.

In Section 2, we outline the methodology, model architecture and considerations regarding parameter specifications and operational requirements. In Section 3, we evaluate the performance of a single model with a single set of parameter specifications and a single training set. In Section 4, we examine the sensitivity of the performance to changes in the model specification and in the size and requirements of the training set. Finally, in Section 5 we summarize findings and discuss subsequent strategy.

2 METHODOLOGY

In this study, we take the offshore simulated sea level time-series and apply a convolutional neural network (CNN) to predict an onshore intensity measure. We consider both the maximal inundation height (MIH; the maximum height of the water surface relative to the initial sea level) and the maximal flow depth ($d_{\max}$; the maximum height of the water surface over the ground). Note

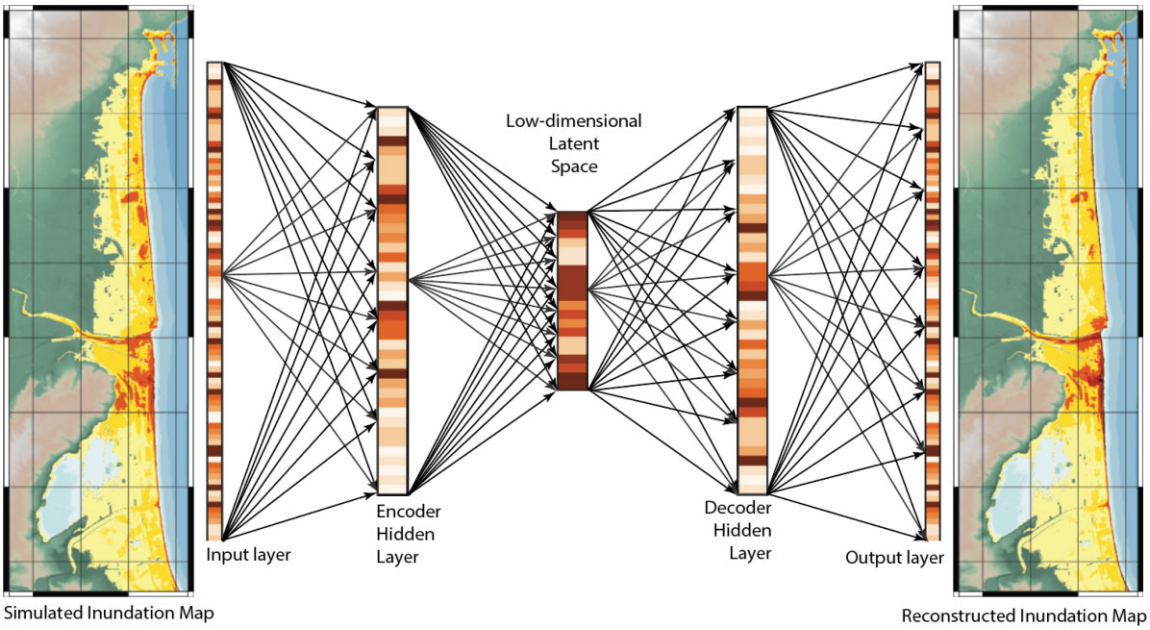

Figure 2. An encoder–decoder architecture for representing the high-dimensional inundation maps with vectors of parameters with far lower dimension. The dimensions of the latent space and the values depicted in the layers of the encoder and decoder are purely illustrative. Only a single hidden layer is displayed for both decoder and encoder; in practice there can be many. The maximum flow-depth map on the left is an actual simulation output. The map on the right depicts an imperfect reconstruction of this map.

that $d_{\max}$ and MIH have in principle a 1-1 mapping, but they may have different properties with regards to the ability to emulate them.

In the simulations, time-series were calculated at virtual tide-gauges around most of the coastlines: even in regions far from the inundation grids. Only the 16 virtual tide-gauges closest to the inundation grids were selected for the prediction of the inundation maps in the Bay of Catania; these 16 locations cover the extent of coastline over which incoming tsunami waves are likely to inundate the region of interest. The Tsunami-HySEA code writes out time-series based on the finest grid present at that location and we performed convergence tests to verify that time-series output using only the coarsest resolution at these locations showed a satisfactory similarity to those output using the fully nested grid. The full 4 hr of simulation time was exploited for the current study, which will also include reflected waves. Values were written out every 30 s of simulation time, resulting in 481 time-samples per simulation per virtual sensor. All time-series start at the origin time of the earthquake and so the arrival time of the first wave will increase with distance.

The highest-resolution grid in the PTHA study of Gibbons *et al*. (2020) has 912 pixels in the longitudinal direction and 2224 in the latitudinal direction: a total of just over 2 million 10 m by 10 m cells. Many points will never be inundated due to high elevation, and locations out at sea are not targets for inundation hazard assessment. 418 908 of these cells are flooded in at least one of the scenarios. Fig. 4 displays the mapping from wave height time-series to the inundation maps using an encoder–decoder, via the lower-dimensional latent space. The time-series and inundation maps are from simulations in the data set and provide an impression of the variability the model will need to accommodate. The following sections address the model architecture, the specification of the loss function, and selection of the training events.

2.1 Model architecture

Given the extensive flexibility in the design of a potential neural network, we limit the study by specifying a relatively simple generic model structure (Fig. 5) within which a few key parameters can be varied. The encoder consists of three consecutive layers of convolutions, each followed by a max pool layer (see Table 1). A convolutional layer (CL) computes the inner product between a set of kernels (the weights) and subwindows of the input with the same dimensions as the kernels. The CL is usually followed by the application of an activation function. We have used a leaky rectified linear unit (Leaky ReLU) with a coefficient of 0.01 (Xu *et al*. 2015). A max pooling layer records the maximum over subwindows of a specific size. Inspired by (Krizhevsky *et al*. 2012), the pooling layers are evaluated on overlapping windows. Through the application of convolutions and pooling layers, the output value depends only on a local part of its input, known as its receptive field. The stacking of multiple CLs with small kernels, interlaid with max pooling layers, is an efficient way of ensuring a large receptive field with respect to the input (in terms of the number of parameters) and is a common structure used for feature extraction in image analysis (Simonyan & Zisserman 2015). It is commonly understood that more complex features of the input are imaged by the deeper layers with a larger receptive field. In the current architecture, we expect important complex features associated with non-linear interactions of the inundation process, typically depending on different properties of the incoming wave and associated with different locations and times.

It is desirable for the network to map a zero input to zero output (a zero wave amplitude offshore should result in zero onshore flow-depth). This would be achieved readily by setting the bias of each layer to zero. However, non-zero biases are needed to ensure the flexible construction of non-linear features. As an intermediate approach, ensuring that the desired property is acquired easily in

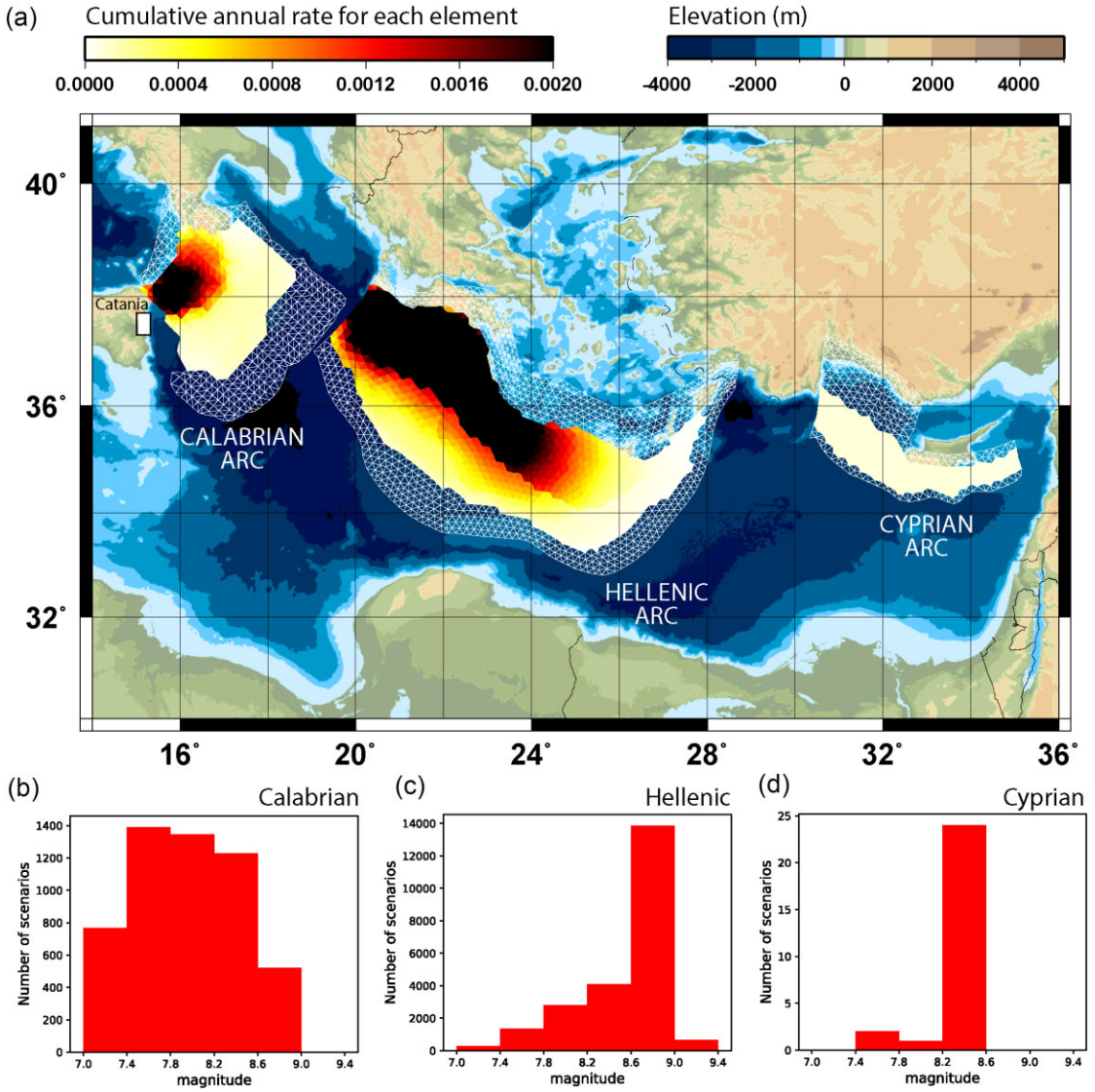

Figure 3. Representation of the tsunamigenic subduction earthquakes in the data set. (a) Geometry of cells that define the subduction zones with a relative measure of slip at each location integrated over the total number of scenarios. (b) Histogram of scenarios as a function of M_w (moment magnitude) for earthquakes in the Calabrian arc (5396 scenarios). (c) Corresponding histogram for the Hellenic arc (22 562 scenarios). (d) Corresponding histogram for the Cyprian arc (27 scenarios). Note that the vast majority of sources are in the Hellenic arc. The sets of scenarios for the Hellenic and Cyprian arcs are dominated by large magnitude earthquakes. The Calabrian arc is far closer to Catania and the data set includes relatively higher numbers of lower magnitude earthquakes here.

the training process, we skip the bias in the first three CLs. The 1 $\times$ 1 CL is there to increase the non-linearity of the encoder without affecting the receptive field (Simonyan & Zisserman 2015). It also acts to reduce the number of parameters in the model by reducing its output dimension.

Temporarily ignoring randomly selected neurons by setting their output to zero during training is known as dropout and is used to reduce overfitting and improve generalization (Krizhevsky *et al*. 2012; Srivastava *et al*. 2014). The number of ignored neurons is quantified by the dropout rate, counting the ratio between the number of ignored neurons to the total number of neurons. To reduce overfitting, we apply dropout before the first dense layer (see Table 1). The use of dropout in the fully connected layers induces a model-averaging effect, and is particularly useful in the case of small data sets (Brigato & Iocchi 2020). A dropout rate 0.5 is suggested to be close to optimal (Srivastava *et al*. 2014). The dropout rate could be optimized for the different network choices and with respect to the size of the training set. However, it is here kept fixed at 0.5.

The final output, $\hat{y}$, is related to the input of the final layer λ according to

$$\hat{y}^p(\lambda) = \text{leakyReLU}\left(\sum_i \lambda_i w_i^p + b^p\right), \quad (1)$$

where p denotes the pixel of the image and $\{w_1^p, \ldots, w_N^p, b^p\}_{p=1}^N$ are the weights of the layer. The final layer resembles a basis for the inundation maps. Approximately 99 per cent of the model parameters belong to this last layer. How many basis elements are necessary to obtain good accuracy? In most cases, we fix the number of basis elements at 64. (It is set to 32 in a few models.)

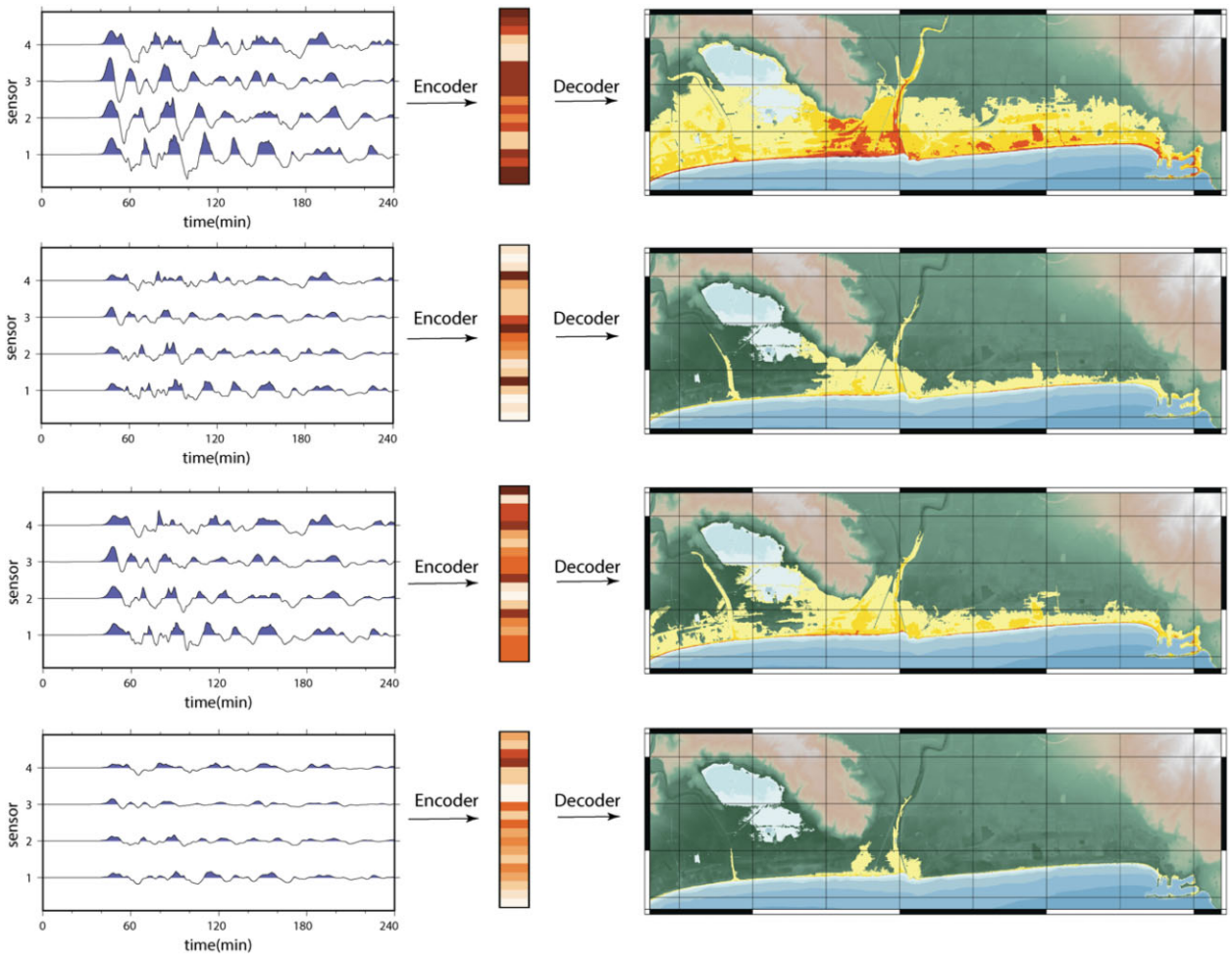

Figure 4. Transformation of waveforms to inundation maps via the latent space. The time-series on the left are from simulations carried out for the PTHA in Gibbons *et al*. (2020) and the inundation maps to the right show the corresponding flow depths. (22 km of the coastline of the Bay of Catania is displayed and the maps are rotated with the south-to-north direction horizontally for the purpose of display.) The representations of only four input waveforms out of the 16 used in the study and the latent space vectors in the middle are purely illustrative. The arrows labelled decoder and encoder can represent models of any type or complexity.

2.2 Loss function

The $d_{\max}$ and the MIH for each pixel are both potential tsunami intensity metrics for the neural network. At the boundary of the inundation, the MIH is equal to the topography, τ, and the $d_{\max}$ is zero. Beyond this, at greater elevation, the $d_{\max}$ can unambiguously be declared to be zero. A natural extension of MIH is given by letting MIH $:= \tau + d_{\max}$. Given the 1-1 mapping between $d_{\max}$ and MIH, for a given topography, τ, they provide an equivalent metric of the tsunami hazard. However, the application of $d_{\max}$ or MIH as a target gives rise to different loss functions. Let m be a model predicting the MIH, y, such that $\hat{y} = m(\eta, \theta)$, where η denotes the offshore time series and θ the model parameters. Let $\mathcal{I}$ denote the predefined set of pixels that are potentially inundated. Applying the ℓ^2 norm directly yields the following loss associated with the MIH.

$$L(\hat{y}, y) = \frac{1}{|\mathcal{I}|} \sum_{p \in \mathcal{I}} \left| \hat{y}_p - y_p \right|^2 =: \|\hat{y} - y\|_2^2, \tag{2}$$

where $|\mathcal{I}|$ denotes the number of pixels in $\mathcal{I}$, so that the ℓ^2 norm is normalized with respect to the size of the region. Alternatively, we can apply the fact that the $d_{\max}$, f, is always non-negative and correct the prediction according to $\hat{f} = (\hat{y} - \tau)^+$, where $(x)^+ := \max(x, 0)$ denotes the positive part. Applying the ℓ^2-norm to the corrected prediction yields a (relaxed) loss associated with the $d_{\max}$ given by

$$L_+(\hat{y}, y) = \|(\hat{y} - \tau)^+ - (y - \tau)\|_2^2 = \|\hat{f} - f\|_2^2 \tag{3}$$

The objective associated with the training set $\mathcal{T} = \{\eta_i, \tau_i, y_i\}_{i=1}^N$ and the loss L, is given by

$$\mathcal{L}(\theta, \mathcal{T}) = \frac{1}{N} \sum_{i=1}^{N} L(m(\eta_i, \theta), y_i) + R(\theta), \tag{4}$$

where $R(\theta)$ is a weight penalization term defined below. The objective $\mathcal{L}_+$ is defined similarly with respect to L_+ (for the $d_{\max}$ values). Note that $L \geq L_+$ so that $\mathcal{L}(\theta, \mathcal{T}) \geq \mathcal{L}_+(\theta, \mathcal{T})$. We have the choice of minimizing either the loss function $\mathcal{L}$ defined with respect to the MIH, or the loss function $\mathcal{L}_+$ defined with respect to the $d_{\max}$. How does the choice of loss function impact the trained model? Let us outline a few possible implications.

(i) As $\mathcal{L}_+$ does not penalize predictions below τ, minimizing $\mathcal{L}_+$ implies more flexibility for the model. This may lead to a better fit but could also make it more prone to overfitting.

(ii) Using $\mathcal{L}_+$ the gradient vanishes once $\hat{y}_p < \tau_p$. This makes it difficult to 'push' the predicted value upwards during training.

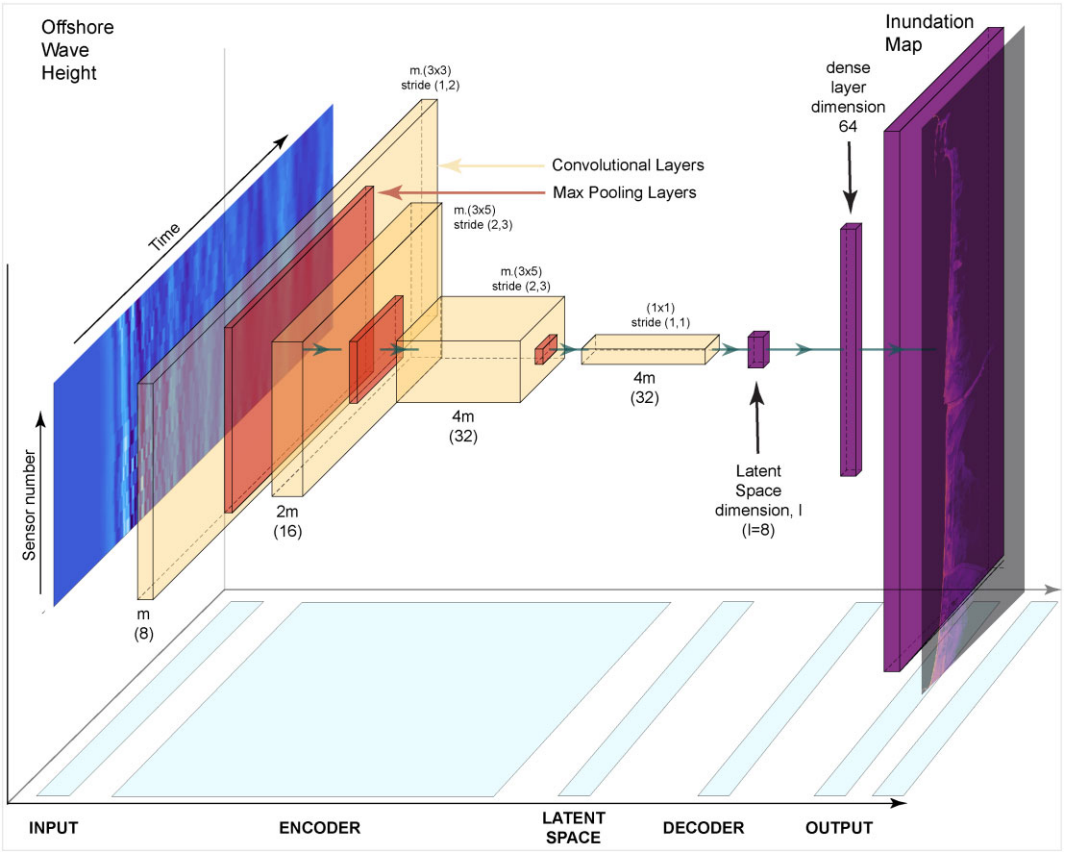

Figure 5. 3-D diagram of the network architecture for the model with $m = 8$ and $l = 8$ described in Table 1. Let the x-axis of a 3-D right handed orthogonal coordinate system be located along the propagation of information, marked by the grey arrows, from left- to right-hand side. On the left-hand side, in the yz-plane, the input data is represented by a heatmap of the wave amplitude measured at the offshore POIs. One unit length along the y-axis (Time), represents 10 times the number of pixels as one unit length on the z-axis (POIs). The transparent orange boxes represents the convolutional layers and are sized according to their output dimensions. The dimension along the x-axis, the depth, represents the number of kernels. Each of the three initial convolutional layers are followed by a max pooling layer represented by a thin red box. The pooling layers keeps the depth fixed, but shrinks the spatial and temporal dimensions. Therefore, they are only depicted according to their temporal and spatial output dimension. The dense layers are represented by purple boxes. As information propagates from every node, it is no longer meaning full with a spatial or temporal resolution. The number of nodes is loosely represented by the size of the boxes. In the final layer, each node represents the prediction at a specific location, as represented by the inundation map shown in the yz-plane at the right-hand side.

Table 1. The network architecture. Each line in the table represents the consecutive operations performed by the network starting at the first line. The conv-mp 3×3-m refers to a convolutional layer with m kernels of dimension 3×3 followed by a max pooling layer of the same size. While the convolutions are applied to every subwindow, the pooling layers are applied to subwindows displaced relative to each others according to the stride. The dense layers are simply recorded according to their output dimension. The architecture has been evaluated on combinations of (m, l) set to (32, 16), (8, 8), (8, 4) and (8,2).

Layer	Stride	Bias	Padding	
conv-mp 3×3-m	(1,2)	No	0	
conv-mp 3×5-2m	(2,3)	No	0	
conv-mp 3×5-4m	(2,3)	No	1	Encoder
conv-1×1-4m	(1,1)	Yes	0	
dropout - 0.5	-	-	-	
dense - l	-	Yes	-	
dense - 64	-	Yes	-	Decoder
dense - Output	-	Yes	-	

Note that L_+ and L may be seen as extremes. Using a Leaky-ReLU instead of simply taking the positive value enable us to choose something in between. To avoid the problems associated with (ii), $\mathcal{L}_+$ loss is henceforth defined by replacing $(\,\cdot\,)^+$ with a Leaky-ReLU with a coefficient of 0.01 in eq. (3).

To reduce overfitting we applied ℓ^2 weight penalization. In general,

$$R(\theta) := \rho_e \sum_i \|\theta_e^i\|_2^2 + \rho_d \sum_i \|\theta_d^i\|_2^2, \tag{5}$$

where θ_e^i and θ_d^i are the weights associated with the i-th layer of the encoder and the decoder, respectively, and $\|\cdot\|_2$ is the ℓ^2 norm, normalized according to the number of weights. For most of the models tested $\rho_e = \rho_d = 10^{-5}$, but some experiments with higher weight penalization have also been carried out (see Section 4).

2.3 Selection of training and test sets

Finding an appropriate and limited set of scenarios on which to train an ML model is a challenge. The 27 985 scenarios selected (the subduction earthquake sources) have sources located a significant distance from the coast, meaning that coseismic displacements at the shoreline are negligible. 15 000 scenarios were reserved for

selection of the training sets (basis set) while the remaining scenarios were used for testing. We cannot use the inundation maps from the simulation for selecting the training set since we assume that we only have the offshore time-series calculations at the outset. However, it is instructive to examine the ranges of potential inundation calculated by Gibbons *et al*. (2020) and the corresponding offshore wave heights in order to understand the selection criteria that might apply. Fig. 6 displays the percentage of scenarios in the PTHA that inundate a given map pixel with a given flow depth. Great regions of the map are inundated for either no scenarios or very few (e.g. significantly below 1 per cent). Other regions (along the shoreline and the river) are inundated in almost every scenario.

Fig. 7 displays relative histograms for metrics of both the onshore inundation (panels a and b) and the offshore time-series (panel c). The two metrics for the onshore inundation are the total area of inundation (i.e. the area that experiences a maximum flow depth greater than zero) and the global MIH. Both distributions show long tails: many scenarios resulting in minimal inundation and great inundation for very few scenarios. The offshore histograms (panel c) are exceptionally consistent from one offshore location to the next and have far less significant tails.

To select suitable training sets, the 15 000 scenarios contained in the basis set were binned according to the (square) maximum absolute amplitude at the selected offshore locations. A fixed maximal number of scenarios were selected randomly from each bin for training (Fig. 8a). This means that the proportion of scenarios selected rises in each bin with increasing maximum amplitude, ensuring that the largest scenarios are well-represented in each training set. The resulting training sets, applied in this study are shown in Table 2. Fig. 8(b) displays the effect of this subselection procedure on the distribution of the maximum flow depth, the maximum wave amplitude and the inundated area for the training set t591 (see Table 2).

3 SYSTEMATIC EVALUATION OF AN ML INUNDATION EMULATOR

Before considering the impact of the size of training set, and model parameters, we inspect some results for a single model. To this end we select the model with $m = 8$ and $l = 8$ trained on the training set t591, using the $\mathcal{L}_+$ loss associated with the maximal flow depth. Fig. 9(a) displays the loss $\mathcal{L}_+$ as a function of the weight updates on the training and test set, respectively. The model was trained using a batch size 10 and the Adam optimizer (Kingma & Ba 2017) with (default) parameter settings $\eta = 0.001$ and $\beta = (0.9, 0.999)$. The loss displayed in Fig. 9(a) is the average over the scenarios in each batch. The training procedure was stopped after 80 000 weight updates. While the loss stabilized after about 20 000 weight updates on the test set, the training loss continued to decrease for the training set. To mitigate overfitting, we select the model obtained after 40 000 weight updates. As seen in Fig. 9(b), the frequency of large inundations is higher in the training set than the test set. Furthermore, the ℓ^2-error with respect to the flow depth (i.e. $\|\hat{f} - f\|_2$), scales approximately linearly with the ℓ^2-norm of the flow depth ($\|f\|_2$). This explains why the loss is in general higher on the training set. Fig. 9(b) indicates that the model is subject to a degree of overfitting (the ℓ^2-error is smaller for scenarios in the training set than for scenarios in the test set). This is particularly clear for the scenarios with larger inundations.

Fig. 10 displays the predicted flow depth (Prediction), the simulated flow depth (Target) and the residual (Target-Prediction), for a single scenario from the test set. The scenario is selected by arranging all the cases according to increasing ℓ^1-error and picking the scenario such that 99.9 per cent have a ℓ^1-error less than the selected case. (In other words, this is one of the predictions with the greatest error.) To best visualize how well the emulator-predicted flow-depths match those calculated in the numerical simulations, Fig. 11 displays scatter plots, with one symbol per tsunami scenario, for 12 selected locations in the Bay of Catania. A number of the locations are along the shoreline, many with an essentially zero elevation. Others are located further inland with elevations ranging from 1 meter to several meters. A symbol above the line $y = x$ in each of the panels represents a scenario for which the ML model underestimated the calculated inundation at that location and a symbol below the line represents a scenario for which the ML model overestimated the inundation. Each of the scatter plots in Fig. 11 is annotated with the corresponding Coefficient of Determination, r^2. The higher r^2 values for the scatter plots to the right of Fig. 11 indicate that the model predictions better explain the target values for the locations along the shoreline. For the near-shore locations that experience the greatest inundation (i.e. locations 10, 12, 20 and 24 in Fig. 11), we note that the accuracy of the predictions is better for the smallest and greatest inundation scenarios than it is for those scenarios between the extremes. The high accuracy associated with high flow depths may be explained by the high frequency of large inundations in the training set. The high accuracy for very small flow depths along the shoreline indicates that the model succeeds at mapping a zero signal to zero inundation.

Fig. 12 displays the r^2 value at all locations on the inundation grid using a colour scale. The r^2 values in Fig. 12(a) (the training set) are significantly higher than the corresponding values in Fig. 12(b). This confirms that the model is subject to a certain degree of overfitting. Fig. 12(b) shows how the quality of the predictions (r^2) diminishes as we move away from the shoreline and the r^2 map is qualitatively similar to the inundation count map (Fig. 6). This tells us that the quality of the prediction at a given pixel is likely directly related to the number of scenarios resulting in inundation at that pixel. In Fig. 12(c), we quantify the connection between the r^2 value for a given pixel for the test set and the number of inundations at that pixel in the training data. For r^2 to exceed 0.8 at a given pixel, we should have had over 100 scenarios in the training data that showed inundation at that pixel. For r^2 to exceed 0.9, we should have inundation at that point for over 200 scenarios in the training data.

The performance of simulated tsunamis relative to observations is frequently evaluated using Aida's number (Aida 1978). Here we use it to assess the accuracy of predictions versus simulations. Let P_i, S_i be the predicted and the simulated flow-depth at locations $1 \le i \le N$. Let $K_i = S_i/P_i$. Aida measured the accuracy in terms of a geometric mean ratio K given by

$$\log(K) = \frac{1}{N}\sum_{i=1}^{N} \log(K_i). \tag{6}$$

Note that K may be considered as a kind of correction factor. Its standard deviation

$$\log(\kappa) = \left(\frac{1}{N}\sum_{i=1}^{N}\left[(\log(S_i/P_i))^2 - (\log(K))^2\right]\right)^{1/2} \tag{7}$$

is a measure of the fluctuation of this correction factor.

To evaluate the models accuracy for different flow depths, pixels where classified according to simulated flow depths in the ranges [0, 0.2), [0.2, 1), [1, 3) and [3, ∞), labelled class 1–4, respectively. For

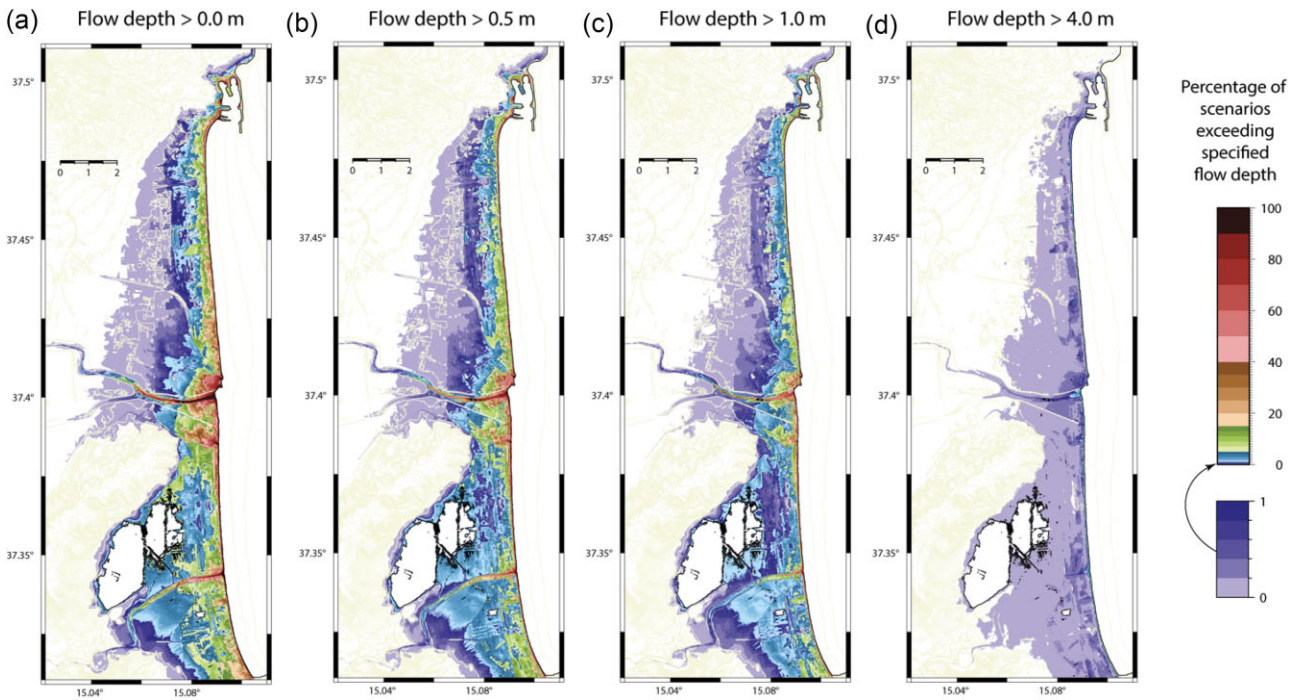

Figure 6. The proportion of earthquake tsunami scenarios from the PTHA study of Gibbons *et al*. (2020) that exceed the indicated flow depth as a function of location in the Bay of Catania. A total of 32 363 scenarios resulted from the hazard disaggregation and so lilac-coloured regions for example are locations at which fewer than 323 scenarios in the data set exceed the indicated inundation. Panels (a), (b), (c) and (d) correspond to flow depths exceeding 0 m, 0.5 m, 1 m and 4 m, respectively. Note that this figure indicates the inundation from all of the scenarios in the PTHA. Only a subset of these scenarios (those corresponding to subduction earthquake scenarios) are exploited in the current study.

each scenario in the test set, Aida's numbers K and κ were calculated for depth classes 2, 3 and 4, given that there were more than 100 pixels in the given class. Due to the definition of K, only pixels where the predictions where nonzero are considered. Similarly, the mean residual and the 95 per cent quantile of the absolute value of the residual was calculated. The results are displayed in Fig. 13. K is the geometric mean ratio of the relation between the simulated and the predicted flow depth. Consequently too high predictions are associated with $K < 1$ while too small predictions are associated with $K > 1$. While the model is in general unbiased, a slight tendency towards too high predictions for small flow depths, and too small predictions for large flow depths is visible both from Figs 13(a) and (c). The fluctuation of the ratio between simulated and predicted flow depth (for each prediction) is measured by κ. Fig. 13(d) shows that the ratio between predicted and simulated values is more spread out for the smaller flow depths. This is not surprising due to the increased sensitivity of the ratio K for smaller predicted flow depths. Note that small flow depths can be associated with the shoreline predictions for small inundations, or predictions further inland for larger inundations. Comparing with Figs 11 and 12, these two cases behave quite differently in terms of predictive accuracy, and perhaps also in terms of bias. Further comparison with Fig. 11 agrees well with the impression that the relative accuracy is higher for higher flow depths. Figs 13(a) and (b) reveals that the absolute values of the residuals are in general higher for larger flow depths. The reduction in $|K - 1|$ with increasing flow depth indicates sublinear growth of the residual with respect to flow depth. Note that this does not conflict with the fact that the ℓ^2 error increases approximately linearly with the ℓ^2-norm. This is because the large inundations have non-zero flow depths over a larger area.

4 SENSITIVITY OF PERFORMANCE TO MODEL PARAMETERS AND TRAINING SETS

Selecting model architecture, loss function, optimization procedure and training set is a challenging task. To this end, a range of different models have been fitted to the training sets described in Section 2.3. Here, we examine how prediction accuracy varies with the size of the training set, the choice of loss function and the model architecture. Adjusting the size of m (the number of Kernels in the convolutional layers) and l (the dimension of the latent space) is one way of adjusting the flexibility of the model. Another option is to adjust the loss function and the training procedure. We encode the model specifications into the coded model names with a core of the form `mc8_l4` meaning 8 Kernels in the convolutional layers and 4 parameters in the latent space (*cf*. Table 1). Specifying in addition the training set employed (*cf*. Table 2) leads to a model code of the form `t164_mc8_l4`. Applying a ReLU in the loss function (*cf*. eq. 3) results in appending `_rel` to the model code (i.e. `t164_mc8_l4_rel`) and an increased weight penalization for the loss $\mathcal{L}_+$ is denoted with an additional `_reg` (for increased regularization).

The adjustment of model parameters is not possible without some means of evaluation. While single number statistics may help to compare different models, it is frequently insufficient in terms of model adjustment. A statistic like the mean ℓ_2-error over the test set reveals the average accuracy, but tells us little about the properties of the model. (This is especially due to the dependence of the error on the size of the inundation.) It can be much more illuminating to use a size-error scatter plot of the training and test sets. Size-error plots form the basis for the following discussion, while ℓ_2-error statistics for different models are available in Table 3.

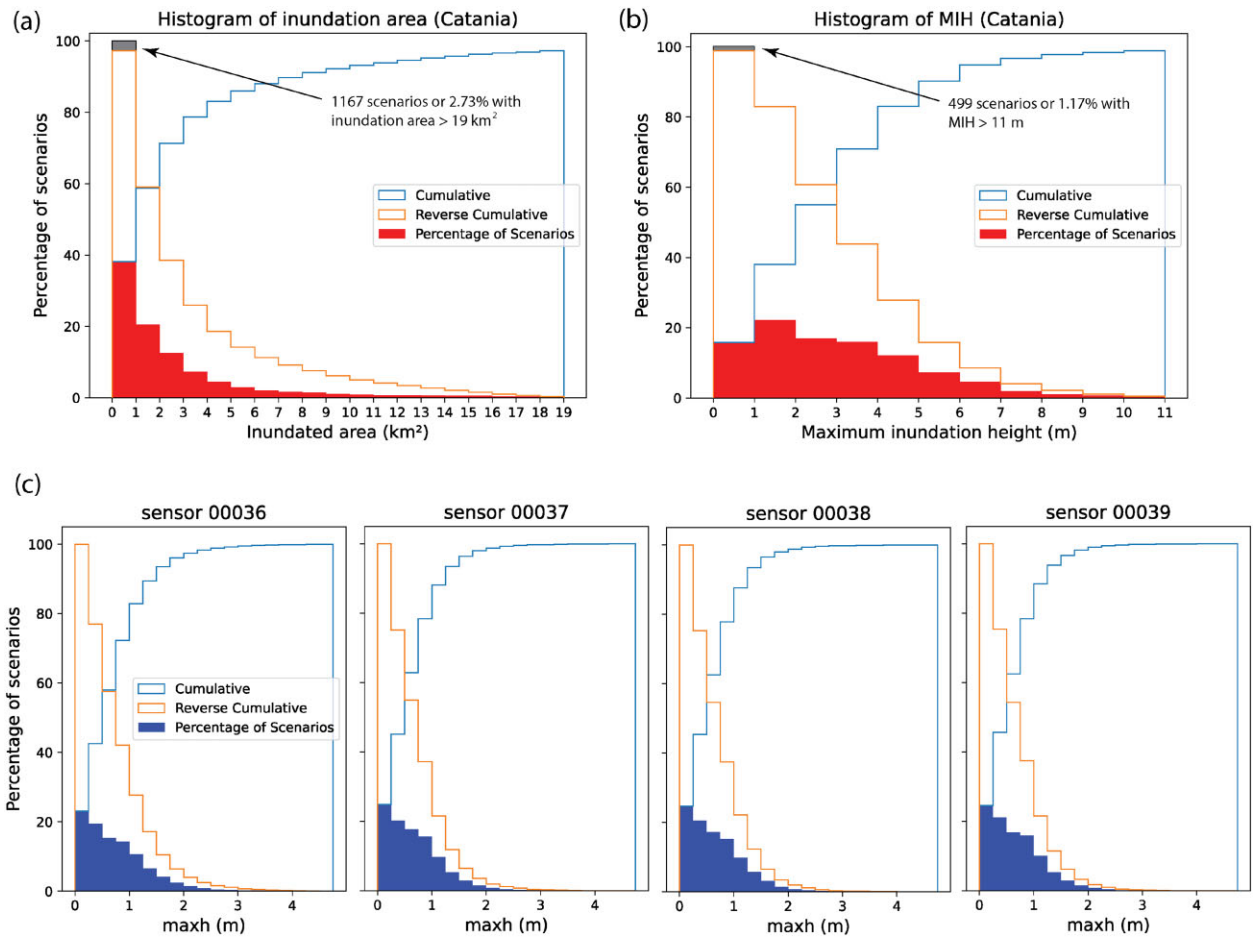

Figure 7. Histograms for (a) area in km^2 inundated, (b) Maximum Inundation Height in m over all initially dry locations and (c) maximum height at the offshore locations 36, 37, 38 and 39 as labelled in Fig. 1b). The basis for the plot is the same 32 363 scenarios from Gibbons *et al*. (2020) displayed in Fig. 6 and all bars are scaled to display the percentage of the total number of scenarios.

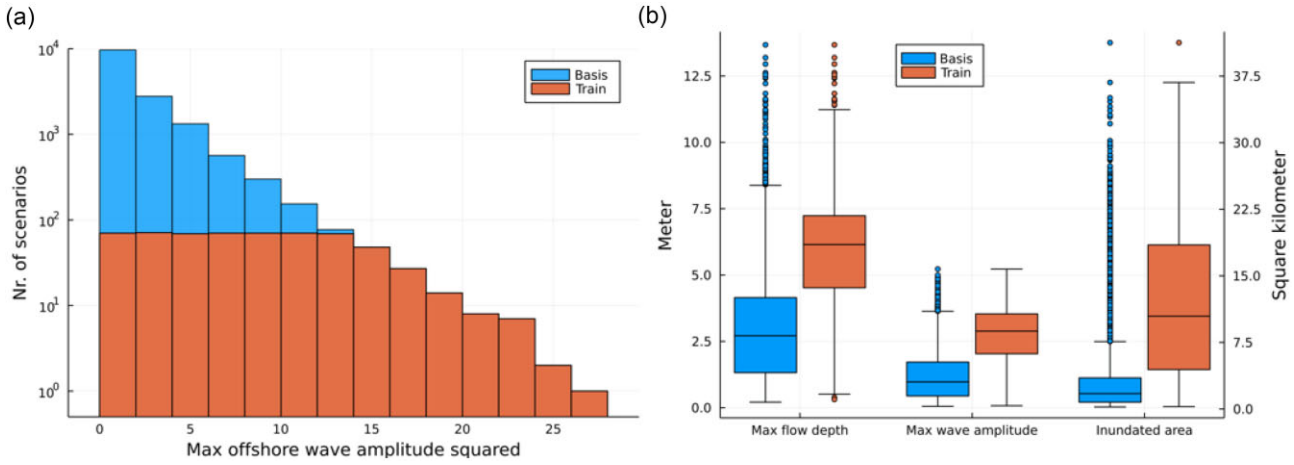

Figure 8. Selection of the training set t591. (a) Size of the 15 bins in the randomly chosen basis set and the number of randomly selected scenarios from each bin used in the training set. (b) Box plot of the maximum flow depth, the maximum wave amplitude at the offshore POIs and the size of the inundated area for the training set t591 and the randomly chosen basis set.

Fig. 14 compares the results of fitting the model with $m = 8$ and $l = 8$ (named `mc8_l8`) on differently sized training sets and with the loss functions $\mathcal{L}$ (MIH) and $\mathcal{L}_+$ ($d_{\max}$). Both models, independent of the loss function, perform far better on the training set than on the test set for the smaller training sets (t164 and t295). An improvement is evident for training set t591, but only for t1831 is there good overlap of the performance on the test and training sets. Along the dashed lines, the ℓ_2-error has the same size as the ℓ_2-norm.

Table 2. Selection of different training sets used in the paper. All scenarios are sorted by the maximum offshore wave amplitude squared prior to placement in bins.

Label	Max scenarios per bin	Total scenarios
t4196	1000	4196
t1831	300	1831
t591	70	591
t295	30	295
t164	15	164

We always want points to lie below this line (ℓ_2-error $<$ ℓ_2-norm) although the severity of the consequences of error increases with increasing ℓ_2-norm (along the x-axis).

The cusp-shaped contour for model `t164_mc8_l8_rel` indicates that the model maps a certain subset of the scenarios to a relatively small (almost) constant inundation map. This is confirmed by inspecting the pointwise predictions in scatter plots similar to the ones shown in Fig. 11. This may be seen as a kind of degeneracy of the model towards a mean prediction on a subset of the input. For a very small number of scenarios with low actual inundation, the model `t295_mc8_l8_rel` predicts some erroneously large inundations. This is a consequence of overfitting and results from the model recognizing certain features associated with large inundation-scenarios in the training set, that are not physically associated with large inundation. This is, in turn, the result of too flexible a model or an insufficient training set. Case-by-case inspection of these erroneous predictions reveals some occurrences of non-physical patterns in the predicted inundation maps (for example with low flow depth along the shore and higher flow depths further inland). Careful analysis of spurious emulations should help us both to find robust methods for automatically detecting them and for improving the models so that they can be avoided. The models trained using the $\mathcal{L}$ loss tend to yield a nearly constant error for scenarios below a certain size (seen as a flattening of the size-error plots in the top row of Fig. 14). The $\mathcal{L}$-loss leaves the model with less flexibility due to the penalization of predictions below the topography (*cf.* Section 2.2). This rigidity appears to make the model unable to fit the smaller scenarios. This hypothesis is supported by the fact that the 'kink' does not vanish using a larger training set. (If anything, the tendency is most pronounced for the largest training set, t1831.) Further inspection of the models trained using the $\mathcal{L}$ loss shows a tendency to predict small positive values far from the shore, even for scenarios with relatively small inundation. This non-physical behaviour is explained by the penalization of predictions below the topography by the loss function $\mathcal{L}$.

Fig. 15 displays size-error plots of the models `mc8_l4_rel` and `mc8_l2_rel`, with latent space dimensions of only 2 and 4, respectively, trained on the training sets t164, t295 and t591. Reducing the dimension of the latent space is expected to enforce more regularization. The clearly visible cusps appearing at slightly different locations for the training sets t164 and t295 reveals that the models shows signs of degeneracy as was also the case for the model `t164_mc8_l8_rel` in Fig. 14. For the larger training set t591, no cusps are visible. Although this degeneracy might only represent a local minima, it seems to be a feature associated with (very) low dimensional latent spaces. We note that the model `mc8_l2_rel` seems to perform well on the training set t591 compared with both `mc8_l8_rel` and in particular `mc8_l4_rel`.

Fig. 16 displays size-error plots for a single set of l and m parameters ($m = 32$ and $l = 16$), trained on four different training sets (t164, t295, t591 and t1831) using the loss function $\mathcal{L}_+$ with different degrees of weight regularization. For the smaller training sets, especially t164, the tendency to overfit for the model `mc32_l16_rel` is greater than we have observed so far. The model `t591_mc32_l16_rel` also erroneously predicts large inundations for some scenarios with very low actual inundation. These models tend to fit the data faster. While 40 000 weight updates is early stopping for `mc8_l8_rel`, this is not the case for these models. Note that there is no sign of a 'cusp' in the plots for `mc32_l16_rel` (center row). Due to the larger parameter space, a second round of training was carried out with an increased weight penalization for the loss $\mathcal{L}_+$ and early stopping for the data sets t164 and t295 (middle row). To this end, ρ_e, and ρ_d were set to 0.05 and 0.01 respectively, *cf.* eq. (5) on t591 and t1831, while ρ_e was set to 0.1 for t164 and t295. The model was labeled `mc32_l16_rel_reg` and trained using 60 000 weight updates for t591 and t1831, 20 000 weight updates for t295, and 10 000 weight updates for t164. On the data set t1831, the increased regularization does not have a big impact except the 'kink' introduced for very small inundations. This may be due to the loss being dominated by the weight penalization term for small-inundation scenarios. Considering the data set t591, the increased weight penalization has led to more stable predictions, and a better overlap of the training and test sets. There are fewer large inundation scenarios with relatively high error, and performance is better for the intermediate-inundation scenarios. Furthermore, there are no scenarios with low inundation that are erroneously ascribed high inundation. However, the general prediction quality for very small scenarios has been reduced. For the smaller training sets t295 and t164, there is a considerable improvement. It is most likely the early stopping that had the most regularizing effect. On t295, the regularized model appears to have a better fit than the `t591_mc8_l8_rel` visualized in Section 3. For the data sets t164 and t295, a couple of further adjustments were done to increase regularization of the model (bottom row); a reduction in the model parameters was introduced by reducing the output dimension of the 1×1 CL to $m = 32$ (*cf.* eq. 1). Furthermore, the weight penalization was further increased to $\rho_e = \rho_d = 10$. The batch size was also increased to 30 and training was stopped at 20 000 weight updates. For this model the loss stabilized both on the training and the test set after about 5000 weight updates. Fig. 16 shows that the training and test sets have good overlap. Furthermore, predictions are relatively accurate also for very small scenarios, indicating that weight penalization does not necessarily introduce a 'kink' as was observed for `mc32_l16_rel_reg` on t591 and t1831. It should however be mentioned that the accuracy is in general reduced. Closer inspection reveals that the overall reduced accuracy is associated with a bias towards underestimation.

5 DISCUSSION AND CONCLUSIONS

We have explored the capability of convolutional encoder–decoder based neural networks to predict high-resolution tsunami inundation maps based on simulated offshore time-series. The primary motivations are for increased speed and reducing the computational cost. This is relevant for deep-sea tsunami simulation if large numbers of scenarios need to be simulated to explore the natural tsunami source variability. It is especially important for the very expensive numerical calculation of local inundation; the calculation of the offshore time-series is far cheaper. If we can simulate the offshore time-series, and then use an ML-based model to predict the final outcome, we may process much larger parameter spaces for the

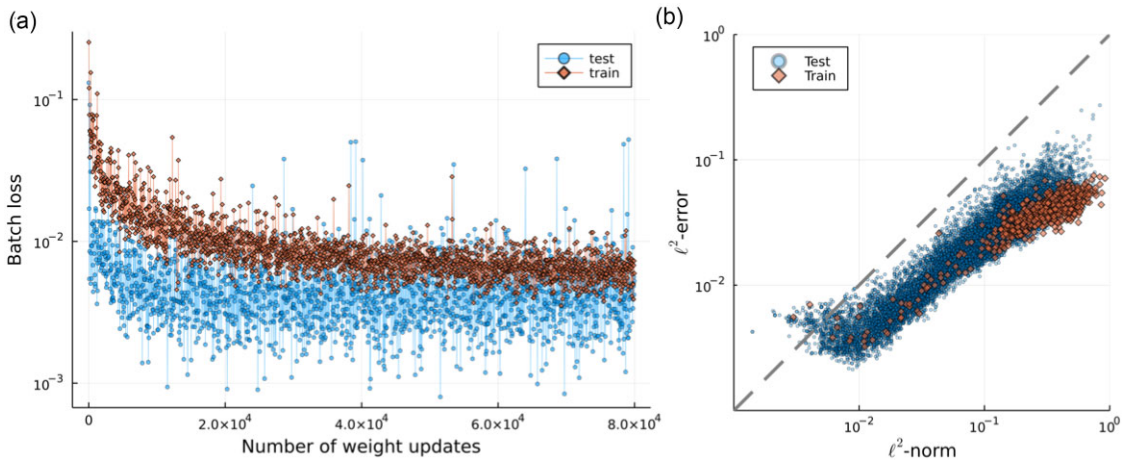

Figure 9. (a) The loss $\mathcal{L}_+$ evaluated on batches of size 10 on the training and test set during training of the model with $m = 8$ and $l = 8$. (b) Scatter plot of normalized ℓ^2-error against the normalized ℓ^2-norm of the flow depth for the trained model (40.000 weight updates) on the training and test set as indicated.

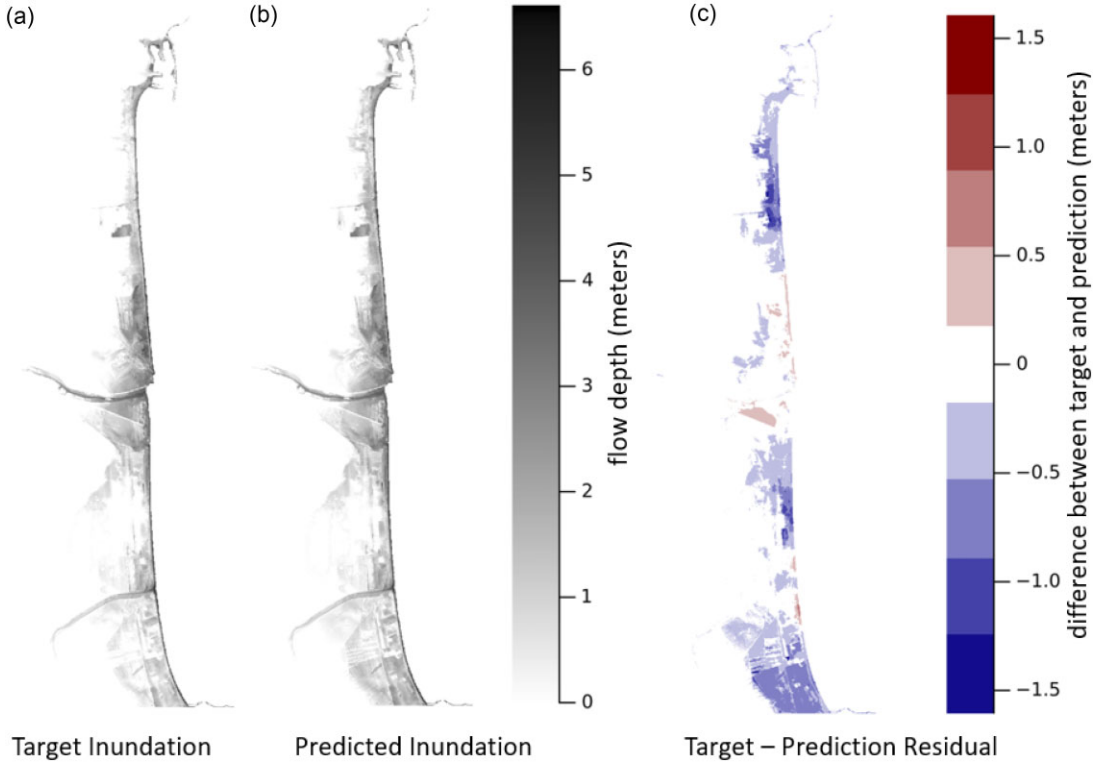

Figure 10. (a) Target $d_{\max}, f$, (b) predicted $d_{\max}$, $\hat{f}$ and (c) residual $(f - \hat{f})$ measured in meters for a given scenario in the test set. The scenario is selected by taking the 99.9 per cent quantile of the ℓ^1-error of the predictions on the test set. Panels (a) and (b) demonstrate the similarity of the simulated inundation map and that predicted by the emulator. The locations at which the emulator overestimates the inundations are coloured blue and the locations at which the emulator underestimates the modelled inundation are labelled red.

same computational cost, reduce time-to-solution in urgent tsunami computations, and simulate massive ensembles more cheaply. A tsunami hazard analysis can require tens of thousands of numerical simulations. A ML-based emulator will require a training set containing a sufficient number of inputs and outputs, representative of the range required. The cost of calculating the training examples, and the cost of training the model, should be significantly smaller than the cost of computing the complete set of numerical simulations. The model would be required to cope across the range of anticipated impact and provide predictions with a satisfactory level of accuracy.

We have designed an encoder–decoder based model in which the input time-series map to the output inundation maps via a latent space with a far lower dimension than either inputs or outputs. We have tested the performance of the model with respect to key parameters m (the number of Kernels in the convolutional layers of the encoder), l (the dimension of the latent space), the size and constitution of the training set and strategy for training the model. If a single most important requirement were to be isolated, it would be that every location at which inundation is to be estimated has to have been inundated in an adequate number of scenarios in the training set. Assessing the accuracy of predictions as a function of

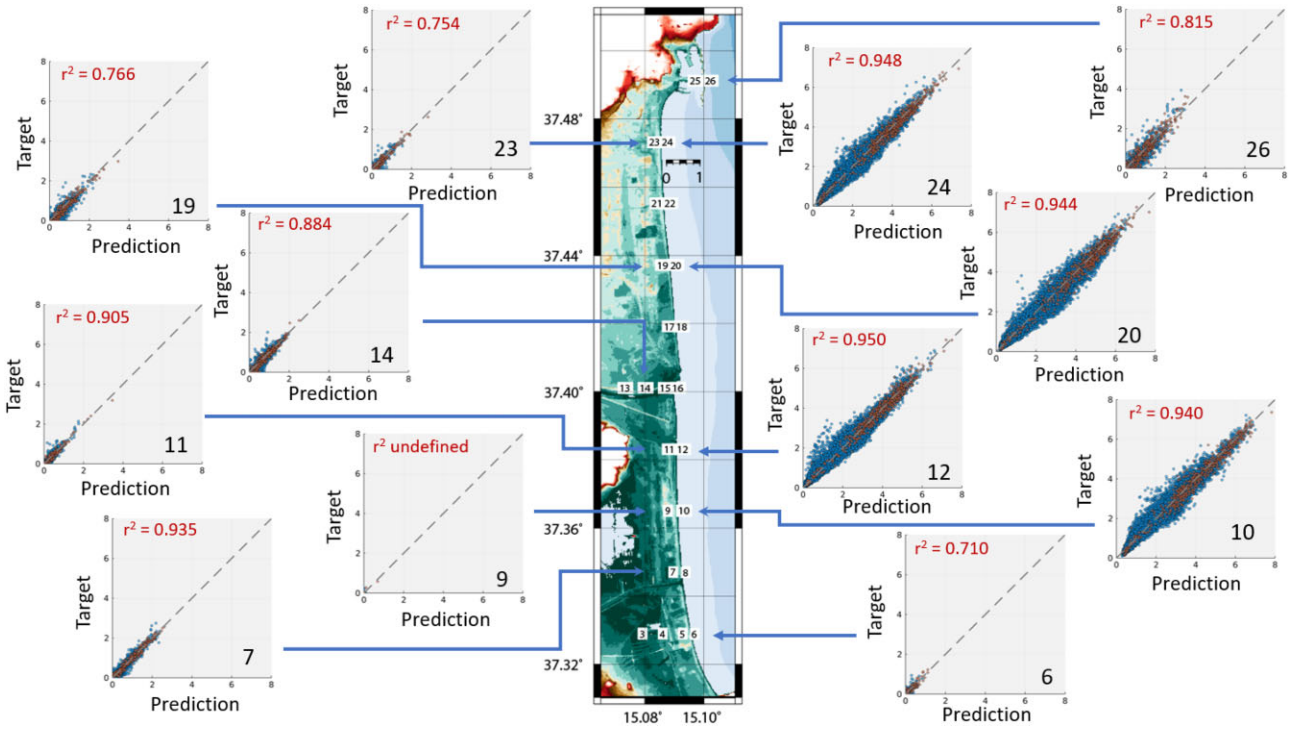

Figure 11. Predicted versus simulated (target) flow-depth estimates for selected locations using the `mc8_l8_rel` model. Each scatterplot is linked to the location on the map at which the inundation was evaluated. All red symbols on the scatter plots relate to the training data and all blue symbols relate to the test data. Scatter plots to the right of the map correspond to locations very close to the coast, mostly with elevation close to zero. Scatter plots to the left of the map correspond to inland locations. The r^2 value provided on each scatter plot is limited to the test data. Note that location 9 is at a location with exceptionally high local elevation and that there are almost no scenarios at which the flow-depth here is above zero.

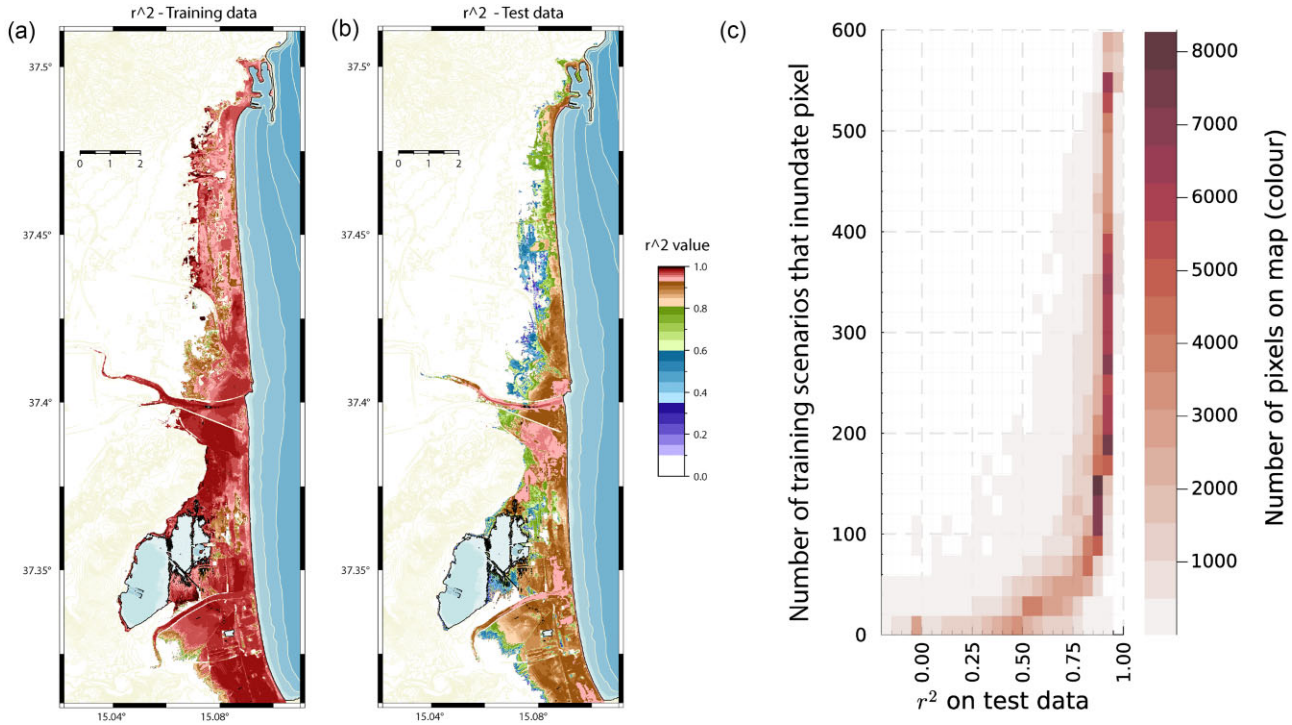

Figure 12. Coefficient of determination, r^2, for flow-depth predictions using model `mc8_l8_rel` at locations in the Bay of Catania for training scenarios (a) and test scenarios (b). Panel (c) is a heatmap of the 2-D histogram relating the number of scenarios in the training set that inundates (hit) a given pixel with the coefficient of determination r^2 evaluated on the test set. r^2 is a measure of prediction accuracy and panel (b) displays how the greatest uncertainty is found at the locations at high elevation, or far from the coastline, that are inundated only exceptionally. Panel (c) shows that around 100 training scenarios need to have inundated a given location before r^2 consistently exceeds a value of 0.8.

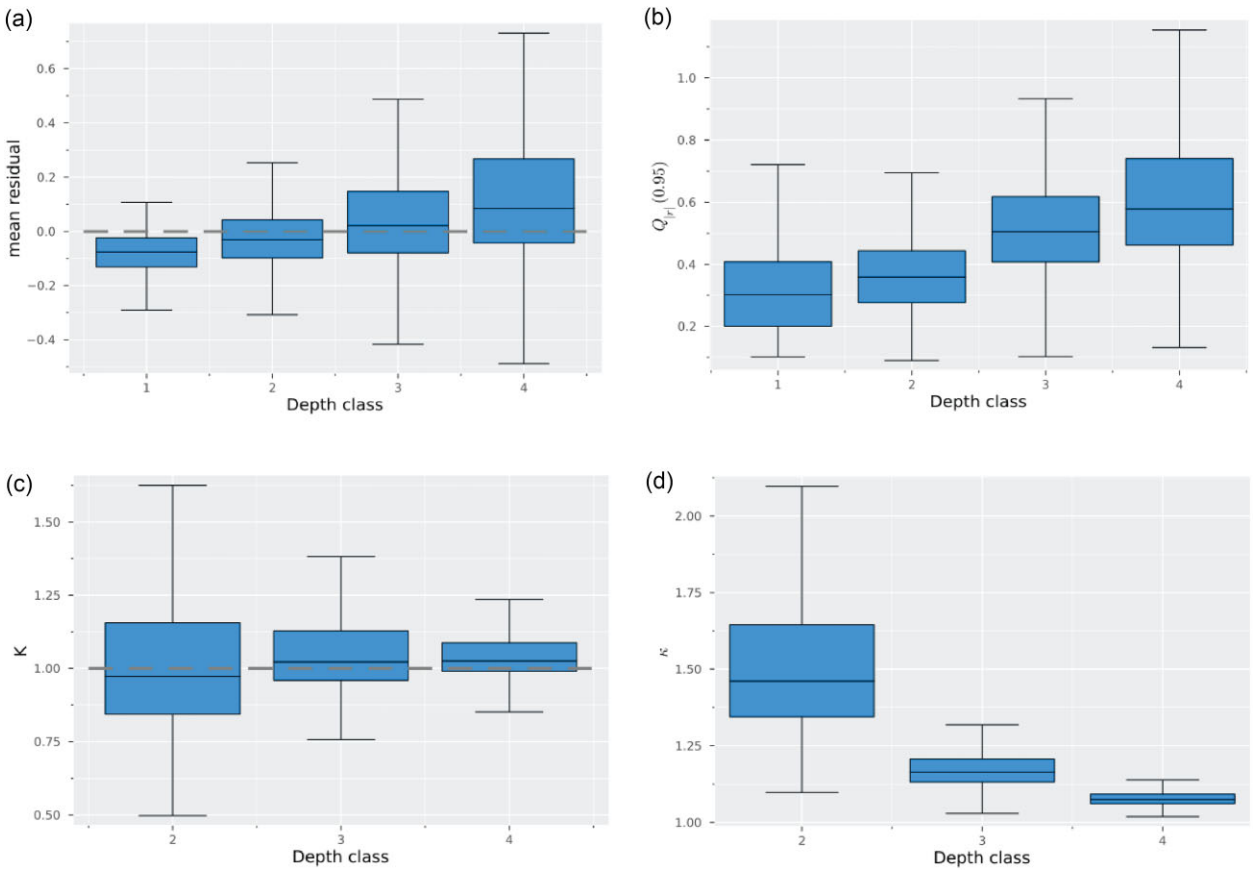

Figure 13. Evaluation metrics for the model described in Section 3. Box plots of mean residual flow depth (a), 95 per cent quantile of the absolute value of the residual (b), Aida's numbers K (eq. 6) and κ (eq. 7) on the test set estimated for different flow depth classes. Classes 1–4 corresponding to depths [0, 0.2), [0.2, 1), [1, 3) and [3, ∞) m, respectively.

Table 3. Table of mean (mean_l2) and 95 per cent-quantile (q95_l2) of the ℓ^2-error on the test set for different models presented in Section 4. The models with X in the code, for example `mc32X_l16_rel_reg` also involve a reduction in the dimension of the output of the 1×1-convolutional layer.

Model	Variable	t4196	t1831	t591	t295	t164
`mc32_l16_rel`	mean_l2	0.0168	0.0235	0.0220	0.0268	0.0358
	q95_l2	0.0385	0.0526	0.0529	0.0700	0.0948
`mc32_l16_rel_reg`	mean_l2		0.0191	0.0243	0.0267	0.0267
	q95_l2		0.0452	0.0619	0.0671	0.0710
`mc8_l8_rel`	mean_l2	0.0201	0.0214	0.0246	0.0285	0.0344
	q95_l2	0.0472	0.0480	0.0583	0.0668	0.0884
`mc8_l8`	mean_l2	0.0212	0.0246	0.0277	0.0299	0.0298
	q95_l2	0.0492	0.0514	0.0602	0.0684	0.0765
`mc8_l4_rel`	mean_l2			0.0261	0.0305	0.0354
	q95_l2			0.0607	0.0724	0.0872
`mc8_l2_rel`	mean_l2			0.0270	0.0319	0.0325
	q95_l2			0.0665	0.0720	0.0825
`mc32X_l16_rel`	mean_l2			0.0324	0.0316	0.0351
	q95_l2			0.0783	0.0771	0.0872
`mc32X_l16_rel_reg`	mean_l2				0.0344	0.0358
	q95_l2				0.0831	0.0875

location indicates that models perform far better at locations close to the shoreline (inundated for almost all scenarios with a wide range of inundation heights) than for locations further inland and at higher elevations (inundated for only the more extreme scenarios). An acceptable accuracy for prediction at a given location appears to require inundation from at least 100–200 scenarios in the training set. That this will apply to as large a part of the domain as possible requires training sets with the order of several hundred scenarios. This suggests a major reduction in computational cost compared with the number of earthquake scenarios needed for the high resolution PTHA at this specific site (Gibbons *et al*. 2020). Comparable ML based emulators like the ones presented in Makinoshima *et al*. (2021), Núñez *et al*. (2022) and Mulia *et al*. (2022) apply training sets on the order of several thousand scenarios.

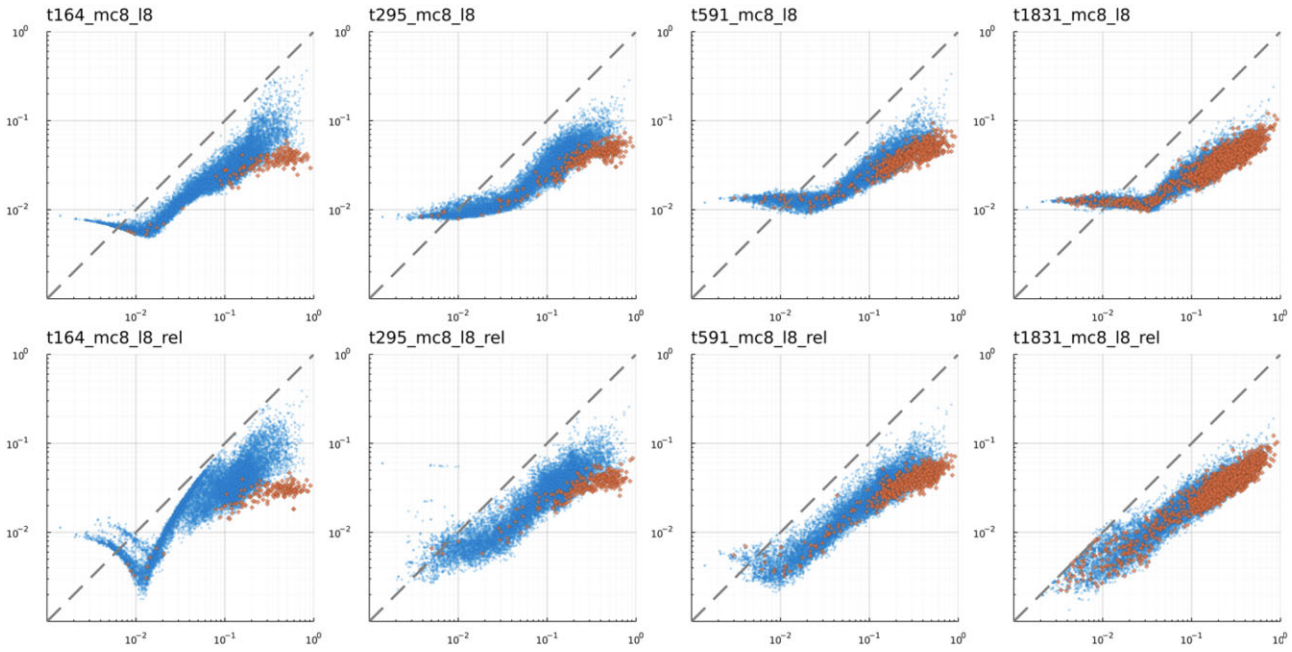

Figure 14. Scatter plots of the ℓ_2-norm (prediction size, horizontal axis) against the ℓ_2-error (prediction error, vertical axis) shown for the model `mc8_l8` and `mc8_l8_rel` trained on the training sets t164, t295, t591 and t1831. Each blue dot represents a single scenario in the test set, while each orange diamond represents a scenario from the training set. The models trained using the $\mathcal{L}_+$ loss are labelled `mc8_l8_rel` due to the application of a ReLU in the loss function (*cf.* eq. 3).

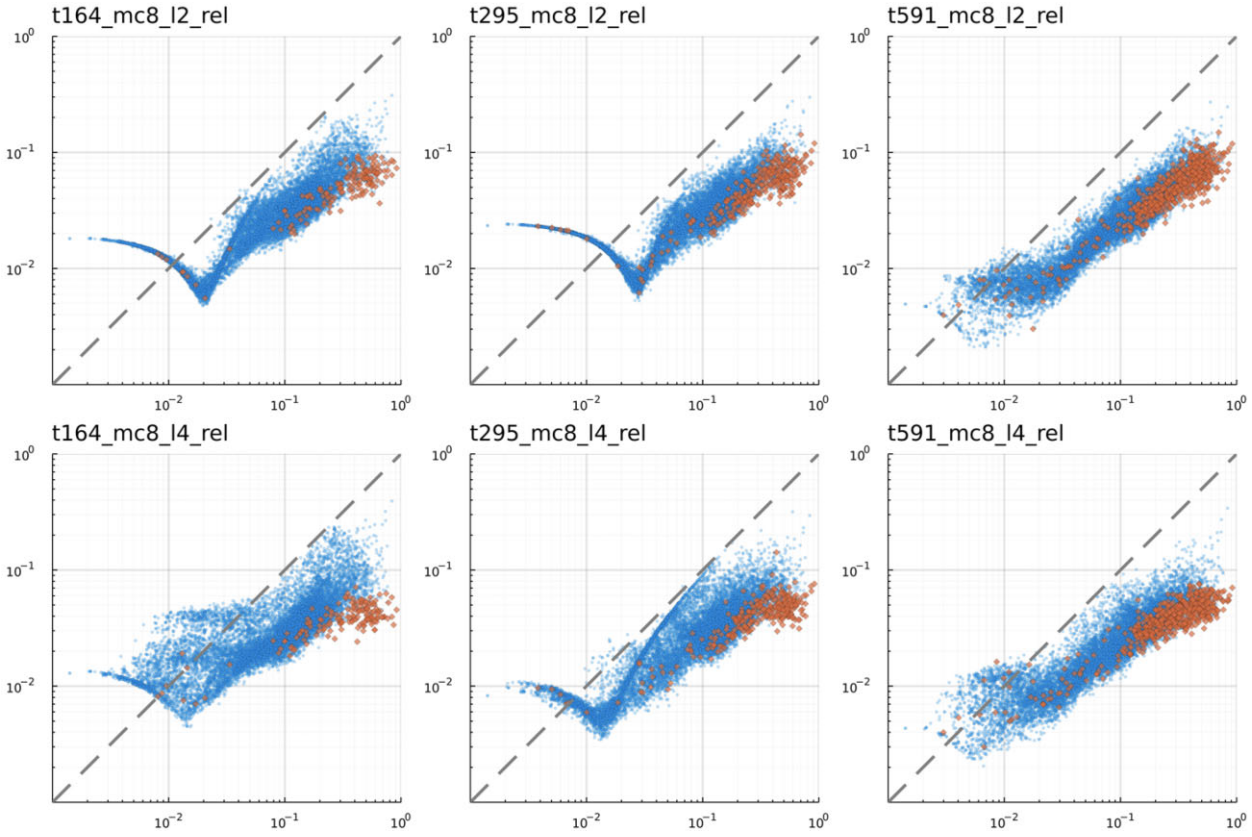

Figure 15. Scatter plots of the ℓ_2-norm (horizontal axis) against the ℓ_2-error (vertical axis) shown for the model `mc8_l4_rel` and `mc8_l2_rel` trained on the training sets t164, t295 and t591. Each blue dot represents a single scenario in the test set, while the orange diamonds represents scenarios from the training set.

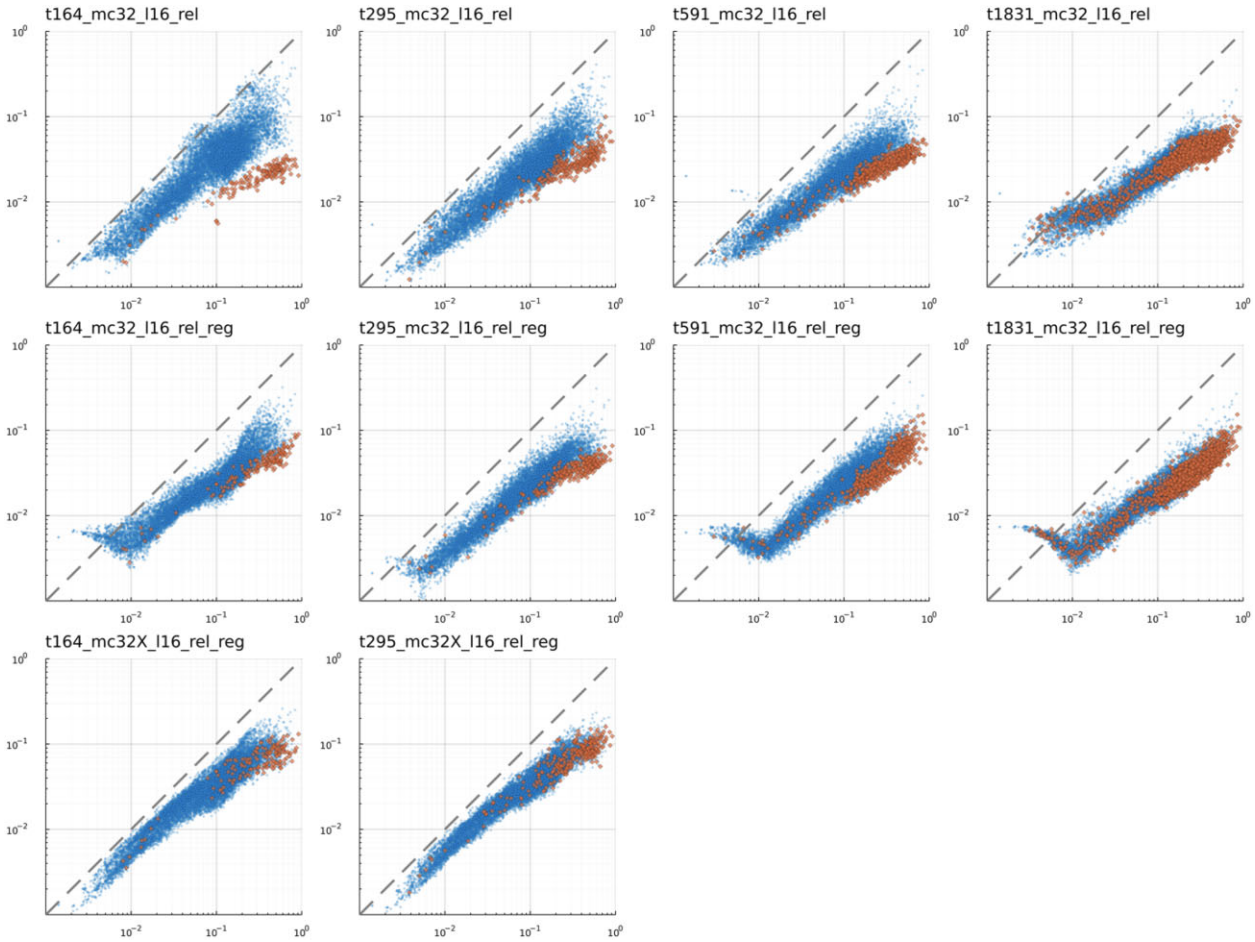

Figure 16. Scatter plots of the ℓ_2-norm (horizontal axis) against the ℓ_2-error (vertical axis) shown for the model `mc32_l16_rel` (top row) trained on the training sets t164, t295, t591 and t1831. A second round of optimization, increasing the weight penalization by setting $\rho_e = 0.05$ and $\rho_d = 0.01$, *cf.* eq. (5) was carried out. The resulting models are labeled `mc32_l16_rel_reg` (third row). The last row shows the result of further regularization ($\rho_e = \rho_d = 10$) and reducing the parameters in the model by reducing the dimension of the output of the 1×1-convolutional layer to $m = 32$. The models with X in the code, for example `mc32X_l16_rel_reg` also involve a reduction in the dimension of the output of the 1×1-convolutional layer. Each blue dot represents a single scenario in the test set, while the orange diamonds represents scenarios from the training set.

We acknowledge that the size of the training set is dependent on the local topography, and the variability of the scenarios. A more complex morphology could potentially result in more complex interactions between the incoming wave and the coast, and in turn necessitate a larger training set. In this paper, we restricted our attention to the scenarios with distant subduction earthquake sources. To apply the model to the full data set requires the inclusion of scenarios with perhaps different wavelengths originated by smaller crustal sources and/or significant coseismic displacement locally at the inundated site (Volpe *et al*. 2019). To cope with scenarios with significant local coseismic displacements, it will have to be included as an input to the neural network. As such, we acknowledge the need to extend the method to account for a wider set of sources, including local coseismic displacements.

Determining the scenarios to be selected for calculating the training set is challenging given that we will not know *a priori* how the inundation maps will look for each simulation; we need to choose them on the basis of the offshore time-series. The more extreme scenarios, likely to generate inundation at locations far inland need to be disproportionately well represented in the training set in order to provide adequate predictions for those locations only prone to inundation for the low-probability, high-impact, tail of the set of scenarios. In this paper, selection was carried out based on the binning of scenarios by their maximum offshore wave amplitude. This selection procedure could be improved by taking into account other types of variability. In applications, it is most likely difficult to assess *a priori* the number of training samples necessary to reach a required level of accuracy. A potential solution is to construct an adaptive selection procedure.

In this study, the raw time-series output from the simulations is provided as input to a convolutional neural network. 16 virtual tide-gauge locations were selected (those locations surrounding the Bay of Catania), and the entire simulation time (4 hr) was used for every scenario. These are rational choices to make but the consequences of these choices, and the sensitivity of the emulations to the input specifications, will need to be examined. Using the full duration of the simulation time will mean that the distance from the source is encoded implicitly in the input. Reducing the duration of the time-series used (with the arrival time accounted for), limiting the number of time-series exploited, and altering the treatment of the input data (for example by extracting waveform features rather than raw waveforms) are all candidates for comprehensive sensitivity studies, but beyond the scope of this paper.

Overfitting is observed in numerous situations where the freedom in the model is too great and/or the training set too limited. We find it beneficial to minimize a loss function based upon the maximal flow-depth rather than the maximal inundation height. The latter requires a penalization of negative values that can result in artificial non-zero emulated inundation at locations that should remain dry. Choosing the dimension of the latent space is a nontrivial problem. Our investigations show that a too low dimensional latent space, may 'disconnect' the encoder and the decoder over a substantial subset of the input space, producing 'degenerate behaviour' and poor prediction. It is also associated with relatively slow and unstable fitting of the model. Models with a slightly higher dimensional latent space and an increased number of kernels tend to fit the data faster and in a more consistent manner, but are more prone to overfitting. Applying dropout, early stopping and weight penalization are effective means to counteract overfitting. Increased weight penalization appears to increase the stability of the models, but also introduces a bias in the predictions.

In this paper, attention has been on the construction of reliable emulators suitable for relatively small training sets. As such, no weighting of the scenarios has been applied for training or evaluation. In the perspective of risk analysis, it could make sense to tailor the fitting of the emulator to optimize the accuracy of the predicted risk. This may be done both by selection of the training set or by weighting of the selected scenarios in the loss function. We note that this is most important considering less flexible models trained on relatively small data sets.

ACKNOWLEDGMENTS

The code accompanying this paper has been implemented in Julia (Bezanson *et al*. 2017) using the ML stack Flux (Innes *et al*. 2018; Innes 2018). Maps and figures in this paper are created using GMT software (Wessel *et al*. 2019) or Plots.jl (Christ *et al*. 2023). Fig. 5 was created with the help of the code repository *PlotNeuralNet* (Iqbal 2020).

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955558. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Spain, Germany, France, Italy, Poland, Switzerland, Norway. Additional funding was received from the Norwegian Research Council, Norway, under project number 323825.

We are grateful to two reviewers whose comments helped us to improve the paper significantly.

DATA AVAILABILITY

The data applied and generated in this research will be shared on reasonable request to the corresponding author. The code associated with this work is available on GitHub at `https://github.com/norwegian-geotechnical-institute/tsunami-inundation-emulator` (Storrøsten 2023).

REFERENCES

Aida, I., 1978. Reliability of a tsunami source model derived from fault parameters, *J. Phys. Earth,* **26**(1), 57–73.

Basili, R. *et al.*, 2021. The making of the NEAM tsunami hazard model 2018 (NEAMTHM18), *Front. Earth Sci.,* **8,** doi:10.3389/feart.2020.616594.

Behrens, J. & Dias, F., 2015. New computational methods in tsunami science, *Phil. Trans. R. Soc., A,* **373**(2053), doi:10.1098/rsta.2014.0382.

Bezanson, J., Edelman, A., Karpinski, S. & Shah, V. B., 2017. Julia: a fresh approach to numerical computing, *SIAM Rev.,* **59**(1), 65–98.

Brigato, L. & Iocchi, L., 2020. A close look at deep learning with small data, preprint (arXiv:2003.12843).

Christ, S., Schwabeneder, D., Rackauckas, C., Borregaard, M. K. & Breloff, T., 2023. Plots.jl – A User Extendable Plotting API for the Julia Programming Language *Journal of Open Research Software,* **11**(5), doi: 10.5334/jors.431.

Davies, G., 2019. Tsunami variability from uncalibrated stochastic earthquake models: tests against deep ocean observations 2006–2016, *Geophys. J. Int.,* **218**(3), 1939–1960.

Davies, G., Weber, R., Wilson, K. & Cummins, P., 2022. From offshore to onshore probabilistic tsunami hazard assessment via efficient Monte Carlo sampling, *Geophys. J. Int.,* **230**(3), 1630–1651.

de la Asunción, M., Castro, M.J., Fernández-Nieto, E., Mantas, J.M., Acosta, S.O. & González-Vida, J.M., 2013. Efficient GPU implementation of a two waves TVD-WAF method for the two-dimensional one layer shallow water system on structured meshes, *Comp. Fluids,* **80,** 441–452. doi: 10.1016/j.compfluid.2012.01.012.

de la Asunción, M., Castro, M.J., Fernández-Nieto, E.D., Mantas, J.M., Acosta, S.O. & González-Vida, J.M., 2013. Efficient GPU implementation of a two waves TVD-WAF method for the two-dimensional one layer shallow water system on structured meshes, *Comp. Fluids,* **80,** 441–452. doi: 10.1016/j.compfluid.2012.01.012.

Ejarque, J. *et al.*, 2022. Enabling dynamic and intelligent workflows for HPC, data analytics, and AI convergence, *Future Generat. Comp. Syst.,* **134,** 414–429. doi: 10.1016/j.future.2022.04.014.

Fauzi, A. & Mizutani, N., 2020. Machine learning algorithms for real-time tsunami inundation forecasting: a case study in Nankai Region, *Pure appl. Geophys.,* **177**(3), 1437–1450.

Folch, A. *et al.*, 2023. The EU center of excellence for exascale in solid earth (ChEESE): implementation, results, and roadmap for the second phase, *Future Generat. Comp. Syst.,* **146,** 47–61. doi: 10.1016/j.future.2023.04.006.

Fukutani, Y., Moriguchi, S., Terada, K. & Otake, Y., 2021. Time-dependent probabilistic tsunami inundation assessment using mode decomposition to assess uncertainty for an earthquake scenario, *J. geophys. Res.,* **126**(7), e2021JC017250, doi:10.1029/2021JC017250.

Fukutani, Y., Yasuda, T. & Yamanaka, R., 2023. Efficient probabilistic prediction of tsunami inundation considering random tsunami sources and the failure probability of seawalls, *Stoch. Environ. Res. Risk Assess.,* **37,** 2053–2068. doi: 10.1007/s00477-014-0966-4.

Geist, E.L. & Parsons, T., 2006. Probabilistic analysis of tsunami hazards∗, *Nat. Hazards,* **37**(3), 277–314.

Gibbons, S.J. *et al.*, 2020. Probabilistic tsunami hazard analysis: high performance computing for massive scale inundation simulations, *Front. Earth Sci.,* **8**(December), 1–20.

Grezio, A. *et al.*, 2017. Probabilistic tsunami hazard analysis: multiple sources and global applications, *Rev. Geophys.,* **55,** 1158–1198. doi: 10.1002/2017RG000579.

Gusman, A.R., Tanioka, Y., MacInnes, B.T. & Tsushima, H., 2014. A methodology for near-field tsunami inundation forecasting: application to the 2011 Tohoku tsunami, *J. geophys. Res.,* **119**(11), 8186–8206.

Innes, M., 2018. Flux: elegant machine learning with Julia, *J. Open Source Softw.,* **3**(25), 602.

Innes, M. *et al.*, 2018. Fashionable modelling with Flux, CoRR, preprint (arXiv:1811.01457). doi: 10.48550/arXiv.1811.01457.

Iqbal, H., 2020. *Plotneuralnet (software),* https://github.com/HarisIqbal88/PlotNeuralNet. doi: 10.5281/zenodo.2526395.

Kamiya, M., Igarashi, Y., Okada, M. & Baba, T., 2022. Numerical experiments on tsunami flow depth prediction for clustered areas using regression and machine learning models, *Earth, Planets Space,* **74,** doi:10.1186/s40623-022-01680-9.

Kingma, D.P. & Ba, J., 2017. Adam: A Method for Stochastic Optimization. doi: 10.48550/arXiv.1412.6980.

Krizhevsky, A., Sutskever, I. & Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks, in *Advances in Neural Information Processing Systems,* Vol. **25,** Curran Associates, Inc.

LeVeque, R.J. & George, D.L., 2008. High-resolution finite volume methods for the shallow water equations with bathymetry and dry states, in *Advanced Numerical Models for Simulating Tsunami Waves and Runup,* Vol. 10: Advances in Coastal and Ocean Engineering, pp. 43–73, World Scientific.

Liu, C.M., Rim, D., Baraldi, R. & LeVeque, R.J., 2021. Comparison of machine learning approaches for tsunami forecasting from sparse observations, *Pure appl. Geophys.,* **178,** 5129–5153. doi: 10.1007/s00024-021-02841-9.

Lorito, S., Selva, J., Basili, R., Romano, F., Tiberti, M. & Piatanesi, A., 2015. Probabilistic hazard for seismically induced tsunamis: accuracy and feasibility of inundation maps, *Geophys. J. Int.,* **200**(1), 574–588.

Løvholt, F., Lorito, S., Macias, J., Volpe, M., Selva, J. & Gibbons, S., 2019. Urgent tsunami computing, in *019 IEEE/ACM HPC for Urgent Decision Making (UrgentHPC),* pp. 45–50, IEEE.

Macías, J., Castro, M.J. & Escalante, C., 2020. Performance assessment of the tsunami-hysea model for nthmp tsunami currents benchmarking. laboratory data, *Coast. Eng.,* **158,** doi:10.1016/j.coastaleng.2020.103667.

Macías, J., Castro, M.J., Ortega, S. & González-Vida, J.M., 2020. Performance assessment of tsunami-hysea model for nthmp tsunami currents benchmarking. field cases, *Ocean Modell.,* **152,** doi:10.1016/j.ocemod.2020.101645.

Macías, J., Castro, M.J., Ortega, S., Escalante, C. & González-Vida, J.M., 2017. Performance benchmarking of tsunami-HySEA model for NTHMP's inundation mapping activities, *Pure appl. Geophys.,* **174**(8), 3147–3183.

Makinoshima, F., Oishi, Y., Yamazaki, T., Furumura, T. & Imamura, F., 2021. Early forecasting of tsunami inundation from tsunami and geodetic observation data with convolutional neural networks, *Nat. Commun.,* **12,** doi:10.1038/s41467-021-22348-0.

Melgar, D., Williamson, A.L. & Salazar-Monroy, E.F., 2019. Differences between heterogenous and homogenous slip in regional tsunami hazards modelling, *Geophys. J. Int.,* **219**(1), 553–562.

Mori, N. *et al.*, 2022. Giant tsunami monitoring, early warning and hazard assessment, *Nat. Rev. Earth Environ.,* **3**(9), 557–572.

Mulia, I.E., Gusman, A.R. & Satake, K., 2018. Alternative to non-linear model for simulating tsunami inundation in real-time, *Geophys. J. Int.,* **214**(3), 2002–2013.

Mulia, I.E., Gusman, A.R. & Satake, K., 2020. Applying a deep learning algorithm to tsunami inundation database of megathrust earthquakes, *J. geophys. Res.,* **125**(9), e2020JB019690.

Mulia, I.E., Ueda, N., Miyoshi, T., Gusman, A.R. & Satake, K., 2022. Machine learning-based tsunami inundation prediction derived from offshore observations, *Nat. Commun.,* **13**(1), doi:10.1038/s41467-022-33253-5.

Núñez, J., Catalán, P.A., Valle, C., Zamora, N. & Valderrama, A., 2022. Discriminating the occurrence of inundation in tsunami early warning with one-dimensional convolutional neural networks, *Sci. Rep.,* **12**(1), 1–20.

Rim, D., Baraldi, R., Liu, C.M., LeVeque, R.J. & Terada, K., 2022. Tsunami early warning from global navigation satellite system data using convolutional neural networks, *Geophys. Res. Lett.,* **49**(20), e2022GL099511, doi:10.1029/2022GL099511.

Rodríguez, J.F., Macías, J., Castro, M.J., de la Asunción, M. & Sánchez-Linares, C., 2022. Use of neural networks for tsunami maximum height and arrival time predictions, *GeoHazards,* **3**(2), 323–344.

Salmanidou, D.M., Guillas, S., Georgiopoulou, A. & Dias, F., 2017. Statistical emulation of landslide-induced tsunamis at the Rockall Bank, NE Atlantic, *Proc. R. Soc., A,* **473**(2200), doi:10.1098/rspa.2017.0026.

Selva, J. *et al.*, 2016. Quantification of source uncertainties in Seismic Probabilistic Tsunami Hazard Analysis (SPTHA), *Geophys. J. Int.,* **205**(3), 1780–1803.

Selva, J. *et al.*, 2021. Probabilistic tsunami forecasting for early warning, *Nat. Commun.,* **12**(1), doi:10.1038/s41467-021-25815-w.

Sepúlveda, I., Liu, P. L.-F., Grigoriu, M. & Pritchard, M., 2017. Tsunami hazard assessments with consideration of uncertain earthquake slip distribution and location, *J. geophys. Res.,* **122**(9), 7252–7271.

Setiyono, U., Gusman, A.R., Satake, K. & Fujii, Y., 2017. Pre-computed tsunami inundation database and forecast simulation in Pelabuhan Ratu, Indonesia, *Pure appl. Geophys.,* **174**(8), 3219–3235.

Simonyan, K. & Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition, preprint (arXiv:1409.1556). doi: 10.48550/arXiv.1409.1556.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting, *J. Mach. Learn. Res.,* **15**(56), 1929–1958. https://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf?utm_content=buffer79b43&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer,

Storrøsten, E., 2023. Tsunami-inundation-emulator. https://github.com/norwegian-geotechnical-institute/tsunami-inundation-emulator.

Synolakis, C., Bernard, E., Titov, V., Kanoglu, U. & Gonzalez, F., 2008. Validation and verification of tsunami numerical models, *Pure appl. geophys,* **165,** 2197–2228. doi: 10.1007/s00024-004-0427-y.

Tanioka, Y. & Gusman, A.R., 2018. Near-field tsunami inundation forecast method assimilating ocean bottom pressure data: a synthetic test for the 2011 Tohoku-oki tsunami, *Phys. Earth planet. Inter.,* **283,** 82–91.

Tozato, K. *et al.*, 2022. Rapid tsunami force prediction by mode-decomposition-based surrogate modeling, *Nat. Haz. Earth Syst. Sci.,* **22**(4), 1267–1285.

Volpe, M., Lorito, S., Selva, J., Tonini, R., Romano, F. & Brizuela, B., 2019. From regional to local SPTHA: efficient computation of probabilistic tsunami inundation maps addressing near-field sources, *Nat. Haz. Earth Syst. Sci.,* **19**(3), 455–469.

Wessel, P., Luis, J.F., Uieda, L., Scharroo, R., Wobbe, F., Smith, W. H.F. & Tian, D., 2019. The Generic Mapping Tools Version 6, *Geochem., Geophys., Geosyst.,* **20,** 5556–5564. doi: 10.1029/2019GC008515.

Williamson, A.L., Rim, D., Adams, L.M., LeVeque, R.J., Melgar, D. & González, F.I., 2020. A source clustering approach for efficient inundation modeling and regional scale probabilistic tsunami hazard assessment, *Front. Earth Sci.,* **8,** doi:10.3389/feart.2020.591663.

Xu, B., Wang, N., Chen, T. & Li, M., 2015. Empirical evaluation of rectified activations in convolutional network, preprint (arXiv:1505.00853). doi: 10.48550/arXiv.1505.00853.