Deep Vision-Based Framework for Coastal Flood Prediction Under Climate Change Impacts and Shoreline Adaptations

Areg Karapetyan Aaron C.H. Chow Samer Madanat
Division of Engineering
New York University Abu Dhabi
Abu Dhabi, United Arab Emirates
{areg.karapetyan,cc6307,samer.madanat}@nyu.edu

Abstract

In light of growing threats posed by climate change in general and sea level rise (SLR) in particular, the necessity for computationally efficient means to estimate and analyze potential coastal flood hazards has become increasingly pressing. Data-driven supervised learning methods serve as promising candidates that can dramatically expedite the process, thereby eliminating the computational bottleneck associated with traditional physics-based hydrodynamic simulators. Yet, the development of *accurate* and *scalable* coastal flood prediction models, especially those based on Deep Learning (DL) techniques, has been plagued with two major issues: (1) the scarcity of training data and (2) the high-dimensional output required for detailed inundation mapping. To remove this barrier, we present a systematic framework for training high-fidelity and high-resolution Deep Vision-based coastal flood prediction models in low-data settings. We demonstrate the proposed workflow on different existing DL architectures, including a fully transformer-based architecture and a Convolutional Neural Network (CNN) with additive attention gates. Additionally, we introduce a deep CNN architecture tailored specifically to the coastal flood prediction problem at hand. The model was designed with a particular focus on its compactness (only about 0.36 Mil. trainable parameters) so as to cater to resource-constrained scenarios and accessibility aspects. The performance of the developed DL models is validated against commonly adopted geostatistical regression methods and traditional Machine Learning (ML) approaches, demonstrating substantial improvement in prediction quality (ranging from 100% to 400% across key metrics). Lastly, we round up the contributions by providing a meticulously curated dataset of simulated inundation maps for the coast of Abu Dhabi, available at https://doi.org/10.7910/DVN/M9625R, which can serve as a benchmark for evaluating future coastal flood prediction models. The complete source code of the proposed framework, including the trained models and the evaluation scripts, can be accessed at https://github.com/Arnukk/CASPIAN.

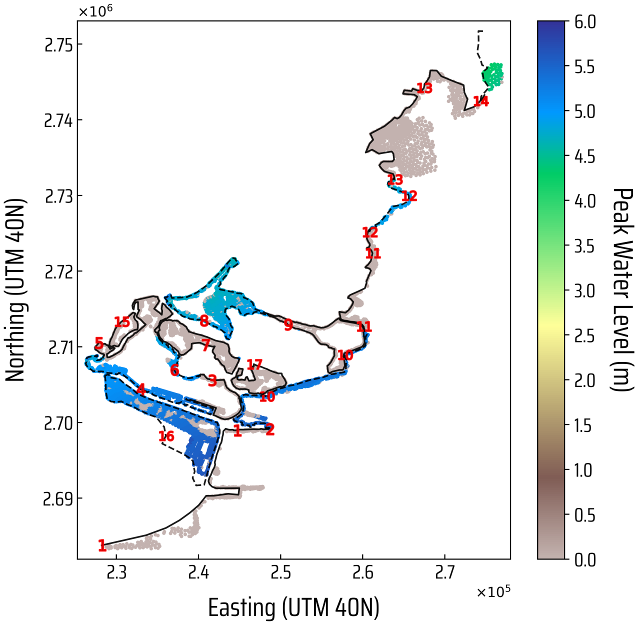

Figure 1: Flood depth map of Abu Dhabi's coast produced with a physics-based high-fidelity hydrodynamic simulator (presented in Sec. A) under a sample shoreline fortification scenario and an SLR projection of 0.5 meters. The coastline was partitioned into 17 segments (precincts) of which those protected by seawalls are delineated by black solid lines, whereas the unprotected are marked with dashed lines. The colorbar measures the magnitude of estimated inundations at 12066 nearshore locations of interest, plotted as scattered points.

1 Introduction

More than 60% of the world's population resides in coastal areas that are within 60 km from the shore (UN Atlas of the Oceans 2016). The looming global warming and its byproducts, such as rising sea levels and more frequent and severe storm surges (Garner et al. 2017; Sweet et al. 2022), render these regions increasingly susceptible to flooding incidents, which can inflict massive humanitarian, ecological, and economic devastation. To put this into perspective, in 2021 alone, hydro-related disasters, including tidal and pluvial flooding and storm surges, caused worldwide more than \$ 224 billion in loss and are projected to cost the global economy \$ 5.6 trillion by 2050 (Dickie 2022). Ensuring the safety and longevity of coastal zones against rising flood waters will require various interventions (Jongman 2018), among which construction of protective structures like seawalls, levees, and storm barriers is set to play a major role (Hummel et al. 2021). Although beneficial for *local flood protection*, these engineered fortifications may alter the hydrodynamics along the coast, amplifying water levels outside of their perimeter and spreading inundations to other (otherwise unaffected) regions (Wang et al. 2018; Haigh et al. 2020; Hummel et al. 2021). For instance, as demonstrated in (Hummel et al. 2021), protection of certain individual stretches of the coast in San Francisco Bay can exacerbate flooding in other zones by up to 36 million cubic meters owing to the effects of the *modified shoreline geometry*.

Progress in computing capabilities and numerical modeling techniques enabled the development of physics-based high-fidelity simulators capable of resolving the detailed hydrodynamics around seawalls. While these tools can accurately simulate nearshore hydrodynamics, providing fine-grade time-series data on estimated depth, duration, and velocity of floods, they are notoriously expensive in terms of computational time and resources. As a result, direct adoption of these high-fidelity simulators in studies requiring extensive number of model realizations (e.g., optimal shoreline protection planning, sensitivity analysis) remains impractical. To exemplify, with such a hydrodynamic model in place (described in Sec. A), simulation and post-processing of a single protection scenario for the coast of Abu Dhabi, illustrated in Fig. 1, demands approximately 72 hours on 128 HPC cores.

Data-driven supervised learning methods have emerged as promising alternatives that can unlock substantial speed-up gains. The principle behind these methods, which are also known as *surrogate models* or *metamodels* in the coastal engineering literature, is to emulate the high-fidelity numerical simulators by inferring the relationships between their inputs and outputs without explicit representation of the underlying sophisticated physical processes and the differential equations that govern their behavior. A prominent issue in the design of coastal flooding metamodels, as also highlighted in (Jia, Wang, and Stacey 2019; Kyprioti et al. 2021; Rohmer et al. 2023), concerns the dimensionality of the output predictions. The geographical domains considered in the analyses of nearshore areas are typically vast, leading to *high-dimensional prediction targets* with tens of thousands of points. Another major obstacle for surrogate hydrodynamic models, especially those employing Deep Learning (DL) methods (Stanton 2023), lies in the *scarcity of training data* since, as noted above, generating annotated samples via high-fidelity simulators consumes significant time and resources. These two factors combined have posed a solid challenge, leaving the development of efficient DL-based surrogate models for high-resolution coastal flood prediction problems elusive.

To address this concern, we present a strategy for synthesizing scalable and performant DL-powered coastal flood prediction models in *data-scarce learning settings*. For comprehensiveness, the flood prediction problem is studied considering both anticipated *climate change-induced effects* and *shoreline adaptations*. The core idea behind the framework is to recast the underlying multi-output regression problem as a computer vision task of translating a two-dimensional segmented grid into a matching grid with real-valued entries corresponding to water depths. This, in turn, enables the incorporation of effective data augmentation techniques, thereby facilitating the training of high-performance DL-based surrogate models. More concretely, the key contributions of the present work are three-fold:

- First, we provide a systematic approach for training efficient Deep Vision-based end-to-end coastal flood prediction models in low-data settings. We test the proposed methodology on different neural networks, including two existing vision models: a fully Transformer-based architecture (SWIN-Unet (Cao et al. 2022)) and a Convolutional Neural Network (CNN) with additive attention gates (Attention U-net (Oktay et al. 2018)). The developed models are benchmarked against existing geostatistical regression methods and traditional ML approaches commonly employed in the field. The comparison results reveal significant gains in predictive performance, with improvements from the developed DL-based surrogate models ranging from 100% to 400% across key metrics. The complete source code for reproducing the conducted analysis, including the trained models, can be accessed at https://github.com/Arnukk/CASPIAN.
- Next, we introduce a deep CNN architecture, dubbed Cascaded Pooling and Aggregation Network (CASPIAN), stylized explicitly for the coastal flood prediction problem at hand. The introduced model was designed with a particular focus on its compactness and practicality to cater to resource-constrained scenarios and accessibility aspects. Specifically, featuring as little as 0.36 Mil. parameters and only a few main hyperparameters, CASPIAN can be easily trained and fine-tuned on a single GPU. On the current dataset, the performance of CASPIAN approached remarkably close to the results produced by the physics-based hydrodynamic simulator (on average, with 97% of predicted floodwater levels having less than 10 cm. error), effectively reducing the computational cost of producing a flood inundation map from *days to milliseconds*.
- Lastly, we round up the contributions by providing a carefully curated database of synthetic flood inundation maps of Abu Dhabi's coast under 174 different shoreline protection scenarios. The maps were generated via a high-fidelity physics-based hydrodynamic simulator (see Sec. A) under a 0.5-meter SLR projection. The provided dataset, available at https://doi.org/10.7910/DVN/M9625R, to the best of our knowledge, is the first of its kind, and thus can serve as a benchmark for evaluating future coastal flooding metamodels.

2 Literature Review

Data-driven surrogate models have been explored to predict various categories of floods, including pluvial (rainfall-induced), fluvial (riverine), and coastal (often linked to storm surges). From a design perspective, the existing models can be organized under two broad themes: end-to-end and multi-stage. Another layer of differentiation unfolds in the scope of targeted predictive outcomes: the risk, extent, intensity, or dynamics of flooding. In the paragraphs below, we provide a concise overview of recent flood prediction studies following the above categorization, while for a more exhaustive survey, the reader is referred to (Mosavi, Ozturk, and Chau 2018; Bentivoglio et al. 2022; Jones et al. 2023; Bomers and Hulscher 2023).

In (Chu et al. 2020), an ensemble data-driven method for emulation of fluvial inundations was investigated and applied to a river segment in Queensland, Australia. The area of interest was represented as a collection of $14,227$ discrete locations, and for each point, a separate feed-forward type artificial neural network was developed to regress the future state of water depth based on time-series data from past tidal levels and inflows. For pluvial floods, (Guo et al. 2021) designed a CNN-based surrogate modeling approach that supports end-to-end predictions in large-scale urban areas. To cope with the high spatial resolution, the images rasterized from the terrain properties of the catchment areas were split into fixed-size patches, which then, along with rainfall intensities, were fed to the CNN as inputs. Expanding the scope further, the work in (Hofmann and Schüttrumpf 2021) provided a pluvial flooding metamodel that accommodates spatially non-uniform precipitation events. The metamodel, termed floodGAN, employs a conditional Generative Adversarial Network (cGAN) to predict 2D inundation depth maps directly from raw images of rainfall distribution data.

In nearshore domains, flood prediction has often been studied under one specific triggering factor (e.g., wind or a short-term extreme event). For the latter setting, the authors of (Al Kajbaf and Bensi 2020) explored and summarized the frequently employed surrogate modeling techniques in the literature. Three techniques were covered, namely Artificial Neural Networks, Gaussian Process Regression (a.k.a., Kriging), and Support Vector Regression. Based on each, a metamodel was trained to predict storm surge heights at four selected locations with different coastal characteristics, and their performance was assessed.

To support high-resolution predictions over extended coastal/urban domains, prior works have alternatively considered two-stage modeling approaches that involve a dimensionality reduction step. For instance, in (Kyprioti et al. 2021), the landfall locations of storms along the coasts of New Jersey and New York were first grouped with the K-means algorithm, then the hazard curves produced for the centroids of these clusters were interpolated over the original grid via a Kriging-based surrogate model. In (Jia et al. 2016; Jia, Wang, and Stacey 2019; Kyprioti et al. 2022; Rohmer et al. 2023), different combination strategies of Kriging metamodeling with Principal Component Analysis (PCA) and clustering were explored. The study in (El Garroussi et al. 2022) proposed a two-step surrogate hydraulic model in which the low-dimensional latent representation of the output variables is inferred through an Autoencoder (AE) as opposed to PCA. The results based on AE reduction were found to yield more accurate predictions. Unlike their end-to-end counterparts, two-step approaches tend to propagate errors due to their sequential structure, thereby potentially limiting the predictive power of the resulting surrogate model.

With the exception of the work in (Jia, Wang, and Stacey 2019), to the best of our knowledge, flood prediction has not been studied *under the joint consideration* of future climate change impacts and shoreline armoring scenarios. In (Jia, Wang, and Stacey 2019), the authors developed a two-stage surrogate hydrodynamic model, based on Kriging and PCA, to investigate the sensitivity of coastal hydrodynamics to shoreline alterations caused by seawall installations under a projected SLR of 1.5 meters. The model was designed for the county-level protection of the San Francisco Bay area and evaluated (under leave-one-out cross-validation) on a dataset of 40 scenarios simulated with a high-fidelity hydrodynamic model Delft3D. Here, we approach this coastal flood prediction problem in a different fashion by first reformulating the problem as a computer vision task, then designing end-to-end DL-based surrogate models that directly support high-resolution predictions. Compared to the aforementioned common two-stage approach of Kriging with PCA, the proposed DL models confer measurable gains in terms of predictive performance and generalizability, as verified by extensive evaluations reported in Sec. 5.

3 Problem Statement

In this section, we formalize the coastal flood prediction problem under study and discuss the associated challenges.

Notational Convention: In what follows, unless stated otherwise, constants or variables are denoted in normal font (e.g., H, n), vectors and matrices are distinguished by boldface lowercase and uppercase letters, respectively (e.g., $\boldsymbol{x}$, $\boldsymbol{X}$), and sets are written in calligraphic or blackboard fonts (e.g., $\mathcal{X}$, $\mathbb{R}$). We let $\mathbf{0}$ and $\mathbf{1}$ symbolize the vectors of all zeros and ones, respectively. Lastly, for a given positive integer n, the notation $[n]$ shall serve as a shorthand for $\{1, 2, \ldots, n\}$.

Problem Definition: As mentioned in Sec. 1, shoreline adaptations caused by the installation of protective engineering structures (e.g., seawalls) can alter coastal water levels and flood patterns. Specifically, depending on which segments of the coastline these seawalls are raised (i.e., *protection scenario*), the ensuing hydrodynamic interactions and feedbacks can elevate or decrease water levels along other (unprotected) parts of the coast. Accordingly, we focus on the following problem: given an input protection scenario, predict the maximum floodwater levels along the coast. To formalize, denote by $d_{\boldsymbol{x}}$ the number of candidate shoreline segments considered for fortification and let $x_i \in \{0, 1\}$ be the corresponding decision made for the segment $i \in [d_{\boldsymbol{x}}]$ with 1 indicating the placement of containments and 0 otherwise. Then, a protection scenario would be represented by a $d_{\boldsymbol{x}}$-dimensional binary vector $\boldsymbol{x}$ and the set of all possible protection scenarios ($2^{d_{\boldsymbol{x}}}$ in total) can be defined as $\mathcal{X} \triangleq \{\boldsymbol{x} \mid \boldsymbol{x} \in \{0, 1\}^{d_{\boldsymbol{x}}}\}$. Let $\boldsymbol{y}$ be a (non-negative) real-valued vector quantifying the peak water levels at $d_{\boldsymbol{y}}$ nearshore locations of interest. With this notation, the prediction problem at hand can be formulated as a regression task of learning a mapping function $f : \boldsymbol{x} \in \mathcal{X} \to \boldsymbol{y} \in \mathbb{R}^{d_{\boldsymbol{y}}}$ provided with a set $\{(\boldsymbol{x}^k, \boldsymbol{y}^k) \mid k \in [n], \boldsymbol{x}^k \in \mathcal{X}, \boldsymbol{y}^k \in \mathbb{R}^{d_{\boldsymbol{y}}}\}$ of n available training examples. Since the generation of these input-output pairs involves running high-fidelity hydrodynamic simulations, extensive data collection can prove prohibitively expensive in terms of both time and resources. Consequently, for double-digit values of $d_{\boldsymbol{x}}$, the cardinality of the training set can turn disproportionately small compared to that of the input space (i.e., $n \ll 2^{d_{\boldsymbol{x}}}$), enforcing an *extremely* *low-resource learning setting*. The inference of f is further complicated by its output size $d_{\boldsymbol{y}}$, which is typically in the order of tens of thousands (here, $d_{\boldsymbol{y}} = 12066$).

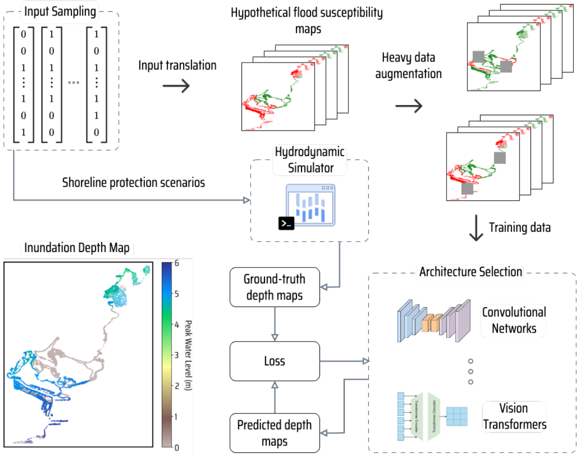

Figure 2: Schematic diagram of the proposed data-driven framework for training performant Deep Vision-based coastal flooding metamodels in low-data settings.

4 Proposed Deep Visual Learning Framework

The workflow of the proposed vision-based surrogate modeling framework, graphically summarized in Fig. 2, can be dissected into four parts, of which first is the generation of training tuples $(\boldsymbol{x}^k, \boldsymbol{y}^k)$. It is crucial to ensure a sufficiently representative selection of points $(\boldsymbol{x}^k)_{k\in[n]}$ for which f will be evaluated, especially under the imposed low-data regime. The scheme adopted herein relies on a combination of judicious manual selection and random sampling. From the former category, we include the following base scenarios: full protection (i.e., $\boldsymbol{x} = \mathbf{1}$), protection of the first and second halves, no protection (i.e., $\boldsymbol{x} = \mathbf{0}$), protection of single precincts (i.e., all binary unit vectors in $\mathcal{X}$) and the inverses thereof, resulting in a total of $4 + 2d_{\boldsymbol{x}}$ input instances. The remaining (out of n) random cases are constructed by drawing uniformly distributed random points from a $d_{\boldsymbol{x}}$-dimensional unit cube via Latin Hypercube Sampling (McKay, Beckman, and Conover 1979), then rounding their entries to the nearest integer value.

In the former category, the following base scenarios were included: full protection (i.e., $\boldsymbol{x} = \mathbf{1}$), protection of the first and second halves, no protection (i.e., $\boldsymbol{x} = \mathbf{0}$), protection of single precincts (i.e., all binary unit vectors in $\mathcal{X}$) and the inverses thereof, resulting in a total of $4 + 2d_{\boldsymbol{x}}$ input instances. For each selected input $\boldsymbol{x}^k$, the respective output $\boldsymbol{y}^k$ was computed by carrying out a numerical simulation with the coupled hydrodynamic model described in Sec. A.

Recall that every element of $\boldsymbol{y}$ corresponds to a specific geographical location parameterized by its latitude and longitude. In vectorial representation, however, this information is abstracted away, leaving the potential of exploiting the spatial correlations and interdependencies between these locations untapped. To enrich the data representation, the proposed pipeline remodels the input and output vectors into matrices as follows. From each $\boldsymbol{y}^k$, $k \in [n]$, we construct a corresponding *flood inundation map* $\boldsymbol{Y}^k \in \mathbb{R}^{H\times W}$ through a mapping $\Phi : \mathbb{R}^2 \to (i, j), i \in [H], j \in [W]$ that converts the geographic coordinates associated with the components of $\boldsymbol{y}$ into grid indices (i, j). This transformation Φ and the grid size $H \times W$ should be selected such that the existing spatial relationships among the output locations are minimally distorted. For the current application site, the coordinate conversion was performed by discretizing the axes of the geographical domain. The dimensions of the formed regular mesh grid, which underlies $\boldsymbol{Y}^k$-s, were equated for ease of processing, and the grid size was set to $H \times W = 1024 \times 1024$ to sustain the desired fine geographic granularity of predictions at a reasonable computational cost while maintaining the overall spatial structure of output locations. The mapping conflicts due to discretization were resolved according to the nearest neighbor principle. Subsequently, the established indexing is leveraged to translate the binary protection scenarios $(\boldsymbol{x}^k)_{k\in[n]}$ into hypothetical *flood susceptibility maps* $(\boldsymbol{X}^k)_{k\in[n]}$, where each $\boldsymbol{X}^k \in \mathcal{C}^{H\times W}$ and $\mathcal{C}$ stands for some discrete set of three predefined values that represent categories. Here, the latter was defined as $\mathcal{C} \triangleq \{-1, 0, 1\}$ and for $\forall\, k \in [n]$, $X^k_{i,j}$ was assigned -1 if the shoreline segment in $\boldsymbol{x}^k$ closest to the location tied to the (i, j)-th index was marked as unprotected, 1 if protected and the rest of the cells were filled with zeros[1]. In a sense, $\boldsymbol{X}^k$-s are segmented matrices in which the $d_{\boldsymbol{y}}$ nearshore locations are classified by their distance to unprotected parts of the coast, and the proximity is perceived as a proxy indication of flood risk. It should, nevertheless, be noted that these input matrices may not necessarily reflect the actual risk or susceptibility of flooding but are, instead, conceptual constructs devised for modeling input protection scenarios, hence the terming “hypothetical”.

Observe that with the remodeled input-output format, the initial regression model is effectively transformed into a problem of learning a mapping of the form $\boldsymbol{X} \in \mathcal{C}^{H\times W} \to \boldsymbol{Y} \in \mathbb{R}^{H\times W}$, where $\boldsymbol{X}$ and $\boldsymbol{Y}$ can be visualized graphically as grayscale (i.e., single channel) images. From a computer vision viewpoint, this problem generally falls under the umbrella of image-to-image translation tasks (Isola et al. 2017), however, it can also be deemed as a variant of monocular depth estimation from a single image (Yang et al. 2024; Fu et al. 2024) since the predicted output is a depth map (of floodwaters). While both of these directions have been extensively researched, to the best of our knowledge, the present problem of inferring depth information from a grayscale, segmented image has not been explored.

Capitalizing on the new image-like representation of inputs and outputs, as a third step of the proposed framework, we artificially increase the volume of training data through image augmentation. Let $\mathcal{D} \triangleq \{(\boldsymbol{X}^k, \boldsymbol{Y}^k) \mid k \in [n]\}$ be the dataset constructed as prescribed above. From each existing pair $(\boldsymbol{X}^k, \boldsymbol{Y}^k)$ in $\mathcal{D}$, m new training examples $(\boldsymbol{X}^{k(1)}, \boldsymbol{Y}^k), \ldots, (\boldsymbol{X}^{k(m)}, \boldsymbol{Y}^k)$ are generated via the Cutout

[1] The choice of values in $\mathcal{C}$ is intended for centering the input data around 0.

technique of (DeVries and Taylor 2017), which applies a fixed-size zero-mask to a random location(s) within the input. This method can be applied in conjunction with other image augmentation techniques, such as rotation, flipping, or shifting, although for the current case study, the application of the former method alone (yet in an excessive manner) was found to be sufficient.

As the final part of the proposed pipeline, it remains to select the type of neural network that will power the surrogate model and the loss function it will learn to minimize. Now that the problem has been transformed into an image processing task, one has a powerful arsenal of DL techniques at disposal, including both generative models, such as GANs (e.g., pix2pix (Isola et al. 2017)) and Diffusion models (e.g., GeoWizard (Fu et al. 2024)), as well as discriminative models, such as Vision Transformers (e.g., SWIN Transformer (Liu et al. 2021)) and CNNs. Drawing on the success of the U-Net architecture (Ronneberger, Fischer, and Brox 2015), we design (in Sec. 4.1) a minimalistic U-Net-like CNN model aligned to the priorities set forth in this work and the characteristics of the high-resolution flood prediction problem at hand. Additionally, we test the proposed surrogate modeling strategy on two existing architectures: a purely transformer-based model and a CNN with additive attention gates. The comparison results are reported in Sec. 5.

Turning to the selection of the loss function, a number of alternatives can be considered, including mean squared Error (MSE), mean absolute error (MAE), Huber loss (Huber 1992), and its reversed variant Berhu (Owen 2007). The choice can be informed by analyzing the distribution of water depth values in the dataset and through experimentation. For the current data, the best results were attained with the Huber loss function L_{Huber}, which sets the loss for each point in the output to

$$L_{\text{Huber}}(\delta) = \begin{cases} \frac{1}{2}\delta^2, & \text{If } |\delta| \leq \theta \\ \theta|\delta| - \frac{1}{2}\theta^2 & \text{otherwise} \end{cases}, \tag{1}$$

where δ denotes the error between the predicted and ground truth water depth values and $\theta \geq 0$ is a parameter. Recall that by construction, the predicted inundation maps will contain artificially added (background) points for which depth estimation is irrelevant. Therefore, the latter were masked out, and the loss was evaluated only on the valid points that correspond to the $d_{\boldsymbol{y}}$ locations of interest.

4.1 CASPIAN

The architecture of the introduced lightweight CNN model CASPIAN, a detailed breakdown of which is presented in Fig. 3, can be interpreted as a two-layered structure consisting of (i) a fully convolutional encoder-decoder network with a central bottleneck comprised of a series of ResNeXt (Xie et al. 2017) blocks, and (ii) a cascade of consecutive pooling operations and corresponding supervision blocks linked by skip connections and stacked on top of the encoder and decoder, respectively. In what follows, we discuss the constituents of the proposed model separately, elaborating on their structure, role, and key parameters.

The encoder part of CASPIAN consists of K successive downsampling blocks, which progressively filter and downscale (by a factor of 2 each) the input image (of size $H \times W$) to generate low-resolution hierarchical feature representations. To allow for efficient utilization of model parameters, we construct these blocks in a style similar to Xception (Chollet 2017). Specifically, each block, except the first, is built from depthwise convolutions (with stride 2) followed by concatenation (with the feature maps from the pooling path), then pointwise (i.e., 1×1) convolutions and a residual connection around them. The initial downsampling block, which for clarity is illustrated in a disassembled form in the topmost left corner of Fig. 3, instead employs a regular convolution with F filters. To save the number of parameters at higher resolutions, we keep the number of filters F constant across all downsampling blocks. The first block is additionally supplied with the output of a stack of operations from the pooling path, which collectively we refer to as *segregated pooling*. This unit filters the non-background points in the segmented input maps based on their class values into separate channels, which are then concatenated and fed into a pooling layer.

The central segment of CAPSIAN, which serves as a bridge between the encoder and decoder, is formed by M repeated ResNeXt blocks with identical configurations and fixed output size of $\frac{H}{2^K} \times \frac{W}{2^K} \times F$. Each block aggregates identity mapping with a set of transformations realized through grouped and pointwise convolutions, as illustrated in Fig. 3. As the low-resolution feature maps produced by the encoder undergo these transformations, the proposed network learns more complex and increasingly global (due to enlarging receptive field) feature representations. In addition to the depth M, this bottleneck path is parameterized by cardinality C and group width w, which control the size and extent of the transformations.

The decoding module in CASPIAN, structurally mirroring the encoder, is assembled from K blocks, which, relying on transposed convolutions (a.k.a., deconvolutions) and pointwise operations, learn to gradually upsample the feature maps distilled by the bottleneck back to the original input resolution $H \times W$. Similar to SegNet (Badrinarayanan, Kendall, and Cipolla 2017), instead of channeling the entire feature maps from the encoder to the decoder through skip connections as in U-Net, the proposed network transfers only the output of corresponding pooling layers as depicted in Fig. 3. Additionally, these feature maps are reused for modeling the hydrodynamic interactions among protected and unprotected parts of the coast and guiding the decoding process accordingly. In particular, we complement the first upsampling block with a Modulation block constructed similarly to Squeeze-and-Excitation (SE) unit (Hu, Shen, and Sun 2018). This block takes the propagated pooling maps as input and produces a set of F weights, one for each channel in the upsampled feature maps. Scaling the latter with these weights allows the network to recalibrate and rectify the decoding process, selectively emphasizing some channels and suppressing others. As illustrated in Fig. 3, in subsequent upsampling steps, the corresponding modulation blocks can be substituted by the output of the first block.

The output from the decoder is fed to a 1×1 convolution and simultaneously summed over channels. The resulting two $H \times W \times 1$ feature maps are summed, and a ReLU activation

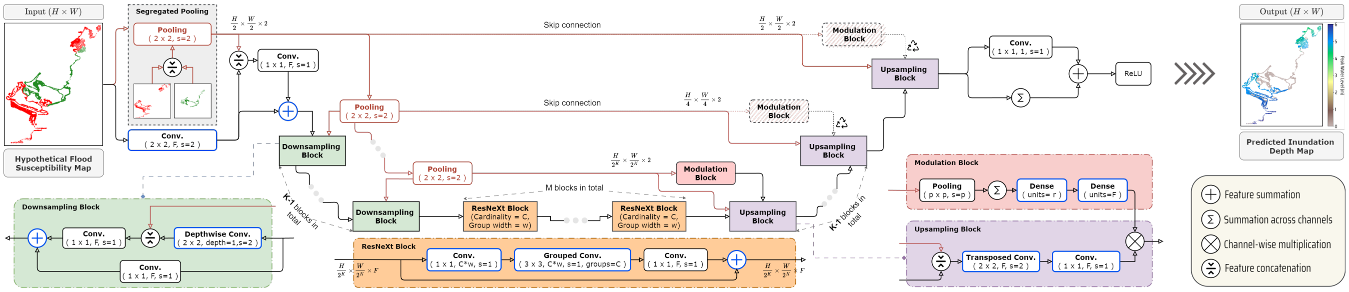

Figure 3: Detailed architecture of the proposed minimalistic CNN model, CASPIAN, for high-resolution coastal flood prediction under SLR and shoreline fortifications. The input image passes through two concurrent paths: a pooling path (colored in red) and a fully convolutional path. The modulation blocks drawn as sketches in dotted outlines are optional and could be substituted by the output from the initial block. The operations followed by non-linear activation functions are marked with a blue border.

is applied to it to produce the predicted inundation depth map. The effects of the summation operator (which incurs no additional trainable parameters) are examined in Sec. 5.3.

5 Flood Prediction Under SLR and Shoreline Armoring

5.1 Setup and Settings

Dataset: Following the proposed workflow presented in Sec. 4, a total of 142 input protection scenarios were generated, and the corresponding inundation maps were produced with the employed physics-based coupled hydrodynamic model to construct the dataset $\mathcal{D}$. To ensure the robustness of the results and the reliability of the evaluation, splitting of $\mathcal{D}$ into training, validation and testing sets was repeated multiple times. Specifically, $\mathcal{D}$ was randomly split thrice according to 112-12-18 partitioning, resulting in three different training, validation, and testing sets. For each split, it was ensured there were no overlaps among the three sets. On the training and validation sets, 19-fold data augmentation was applied through the Cutout technique with two patches, each of size 60×60. To test the generalizability of the developed models, a Holdout dataset consisting of 32 handcrafted protection scenarios was additionally constructed (see Sec. D).

Candidate Approaches: The pool of coastal flood prediction methods selected for evaluation involves two main groups. The first (benchmark) group includes two standard regression techniques, namely Linear Regression and Lasso Regression with polynomial features (referred to as Lasso with Poly.), and two commonly employed coastal flooding metamodels (as informed by the literature review in Sec. 2), namely Kriging with PCA and Support Vector Regression (SVR). The second group contains four end-to-end DL-based models developed under the proposed framework. Among these, three are based on two existing networks, Attention U-net (Oktay et al. 2018) and SWIN-Unet (Cao et al. 2022), originally designed for medical image segmentation. To adapt to the present settings, the segmentation heads in both networks were replaced by a 1×1 convolution with a ReLU activation. Additionally, to experiment with the transfer learning technique, we substitute the encoder stack in Attention U-net with the first 16 convolutional layers from the VGG19 network (Simonyan and Zisserman 2014) and consider two versions, one with the encoder weights initialized randomly while the other with those pre-trained on the popular ImageNet dataset, which contains more than a million images. The latter model is denoted as Attention U-net‡‡ to discern between these two. Accordingly, to conform to the three-channel input format of VGG19, for both models, the depth of input matrices was expanded by replacing the class values with RGB codes, resulting in an input size of $H \times W \times 3$. The final fourth model in this cohort is based on the introduced lightweight CNN network CASPIAN. To allow for an impartial inter-group comparison, predictions produced by the models in the first group were *post-processed* to replace the *negative values* with zeros. As an additional reference, we employ a naive regressor, termed Baseline Predictor, which outputs 0 if the corresponding coastal location in the input vector was (hypothetically) classified as inundation-safe or otherwise the average peak water level across the entire dataset.

Model Configurations and Implementation Details: Settings of the classical and generalized regression models comprising the first group were determined under experimentation and are listed in Table 2. Linear Regression, SVR, Lasso with Poly., and Kriging with PCA were implemented via Scikit-learn (Pedregosa et al. 2011) and SMT (Bouhlel et al. 2019) Python packages. The implementations (in Tensorflow Keras v2.1) of Attention U-net and SWIN-Unet were borrowed and adapted from (Sha 2021). The proposed model CASPIAN was built with Tensorflow Keras v2.1. The hyperparameters of Attention U-net were tuned manually and then transferred identically (except the weight initialization in the encoder) to Attention U-net‡‡. For SWIN-Unet and CASPIAN, the selection of hyperparameters was optimized through the Random Search algorithm provided as part of the Keras Tuner library (O'Malley et al. 2019) (see Sec. C for details). For brevity, Table 2 reports only the total number of trainable parameters of these models, whereas the hyperparameter values are relegated to Sec. B.

Training Specifications: All the four DL models were trained with Adam optimizer under the Huber Loss (as defined in Eq. 1) function with $\theta = 0.5$ and batch size of 2. The adopted learning schedule, determined through trials with

Model	Test Dataset (18 samples)							Holdout Dataset (32 samples)						
	(Lower is better)					*(Higher is better)*		*(Lower is better)*					*(Higher is better)*	
	AMAE	ARMSE	ARTAE (%)	$\delta > 0.5$ (%)	$\delta > 0.1$ (%)	R^2	ACC[0] (%)	AMAE	ARMSE	ARTAE (%)	$\delta > 0.5$ (%)	$\delta > 0.1$ (%)	R^2	ACC[0] (%)
Baseline Predictor	1.14 ± 0.46	2.19 ± 0.63	58.5 ± 32.3	27 ± 11.2	55.9 ± 25.7	-0.11 ± 0.25	57.9 ± 29.9	1.33 ± 0.28	2.49 ± 0.28	62.2 ± 19	29.9 ± 7	52.6 ± 12.3	-0.05 ± 0.23	70.9 ± 12
Linear Regression	0.20 ± 0.06	0.57 ± 0.15	34 ± 88.2	12.5 ± 2.85	19.3 ± 4.01	0.88 ± 0.17	41.4 ± 17.1	0.2 ± 0.09	0.58 ± 0.21	9.18 ± 2.69	12.3 ± 3.61	18.6 ± 4.12	0.93 ± 0.05	41.8 ± 15.6
Kriging with PCA	0.19 ± 0.07	0.55 ± 0.16	22.1 ± 49.5	11.3 ± 4.2	18.4 ± 5.26	0.91 ± 0.09	42.1 ± 16.4	0.21 ± 0.09	0.59 ± 0.22	9.31 ± 3.01	12 ± 3.83	19.3 ± 4.51	0.93 ± 0.05	42.8 ± 14.6
SVR**	0.17 ± 0.08	0.69 ± 0.26	19.9 ± 42.1	4.47 ± 1.68	6.27 ± 2.76	0.87 ± 0.08	21.3 ± 8.1	0.17 ± 0.12	0.7 ± 0.33	7.65 ± 3.94	4.44 ± 2.61	6.13 ± 3.01	0.9 ± 0.1	19.2 ± 6.84
Lasso with Poly.	0.14 ± 0.05	**0.42** ± **0.15**	22.5 ± 55.6	5.45 ± 2.65	16.3 ± 5.16	**0.95** ± **0.03**	10.8 ± 3.82	0.16 ± 0.08	0.46 ± 0.22	7.09 ± 2.42	6.72 ± 3.07	14.3 ± 4.58	0.95 ± 0.05	10.8 ± 4.6
Attention U-net	0.11 ± 0.07	0.64 ± 0.26	5.56 ± 5.73	1.9 ± 1.35	4.04 ± 2.4	0.89 ± 0.08	97.3 ± 3.08	0.13 ± 0.1	0.69 ± 0.32	5.38 ± 3.59	2.29 ± 2.08	4.71 ± 2.79	0.9 ± 0.08	**98.8** ± **1.49**
Attention U-net‡‡	0.10 ± 0.07	0.64 ± 0.26	5.41 ± 5.73	1.92 ± 1.37	**3.95** ± **2.38**	0.89 ± 0.08	97.2 ± 3.12	0.12 ± 0.1	0.69 ± 0.32	5.3 ± 3.57	2.28 ± 2.08	**4.52** ± **2.84**	0.9 ± 0.08	98.8 ± 1.51
SWIN-Unet	**0.07** ± **0.04**	**0.42** ± **0.18**	**3.21** ± **2.15**	**1.37** ± **0.89**	7.19 ± 3.69	**0.95** ± **0.03**	**97.3** ± **2.44**	**0.08** ± **0.06**	**0.45** ± **0.21**	**3.32** ± **2.05**	**1.70** ± **1.61**	7.58 ± 3.84	**0.95** ± **0.04**	98.1 ± 3.57
CASPIAN	**0.06** ± **0.03**	0.46 ± 0.18	**3.12** ± **1.89**	**1.06** ± **0.67**	**3.14** ± **1.65**	**0.94** ± **0.03**	**98.5** ± **1.84**	**0.06** ± **0.04**	**0.45** ± **0.19**	**2.80** ± **1.46**	**1.01** ± **0.87**	**3.99** ± **2.67**	**0.96** ± **0.03**	**99.1** ± **1.10**

Table 1: Quantitative comparison of the candidate coastal flood prediction methods. The models developed with the proposed Deep Visual Learning framework are highlighted in green. The results for each metric are reported as the mean and standard deviation across the samples over the three data splits. The top scores are highlighted in blue, and the runner-ups are in orange. The superscript ** denotes an ensemble of individual models, each trained for one specific coastal location.

several alternatives, was set to start with a gradual warm-up that increases the learning rate from 0 to LR linearly for 20 epochs, followed by 200 epochs of the main training session wherein the learning rate was reduced (×0.85) whenever the validation loss plateaued (patience = 10). During the main training, early stopping was applied if no improvement in the validation loss was recorded for 40 consecutive epochs. LR was set to $1.5 \cdot 10^{-4}$ for Attention U-Net and Attention U-net‡‡, to $1.8 \cdot 10^{-4}$ for SWIN-Unet, and to $8 \cdot 10^{-4}$ for CASPIAN. All the models were trained and evaluated on a desktop machine with an Intel Core i9 3.00 GHz CPU, 64 GB of RAM and a single NVIDIA RTX 4090 GPU.

Model	Input size	Output size	Training time (hr.)	Inference time (ms.)	Parameters (# or settings)
Baseline Predictor	12066	12066	$\ll 0.001$	23	1
Linear Regression	17	12066	$\ll 0.001$	≈ 1	17
Kriging with PCA	17	12066	≈ 0.001	≈ 8.5	Reg. f. : linear Cor. f. : sq. exp.
SVR**	17	$1 \cdot (12066)$	≈ 0.001	≈ 3770	Kernel : linear $C = 5, \epsilon = 0.05$
Lasso with Poly.	154	12066	≈ 0.029	≈ 0.49	Int._only: True Degree : 2
Attention U-net	$H \times W \times 3$	$H \times W \times 1$	≈ 7.5	≈ 25	≈ 12 Mil.
Attention U-net ‡‡	$H \times W \times 3$	$H \times W \times 1$	≈ 8.6	≈ 25	≈ 12 Mil.
SWIN-Unet	$H \times W \times 1$	$H \times W \times 1$	≈ 2	≈ 50	≈ 8.3 Mil.
CASPIAN	$H \times W \times 1$	$H \times W \times 1$	≈ 3.5	≈ 25	≈ 0.36 Mil.

Table 2: Addendum to Table 1.

Evaluation Metrics: As emphasized in (Al Kajbaf and Bensi 2020), in most prior works studying coastal flood prediction, the performance of developed coastal metamodels was assessed considering only a few basic aggregate metrics, such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) or Coefficient of Determination (R^2), which may not adequately reflect the actual quality of predictions. To provide a more comprehensive evaluation, we consider 6 different metrics, including both error and accuracy measures, formally defined as follows:

$$\text{ARTAE} \triangleq \frac{1}{N}\sum_{k=1}^{N}\frac{\|\boldsymbol{y}^k - \hat{\boldsymbol{y}}^k\|_1}{\|\boldsymbol{y}^k\|_1}, \quad R^2 \triangleq \frac{1}{N}\sum_{k=1}^{N}(1-\Psi)$$

$$\text{ARMSE} \triangleq \frac{1}{N}\sum_{k=1}^{N}\sqrt{\sum_{i=1}^{d_{\boldsymbol{y}}}\frac{(y_i^k - \hat{y}_i^k)^2}{d_{\boldsymbol{y}}}}, \quad \delta > \Delta \triangleq \frac{1}{N}\sum_{k=1}^{N}\frac{|\mathcal{S}_\Delta|}{d_{\boldsymbol{y}}},$$

$$\text{AMAE} \triangleq \frac{1}{N}\sum_{k=1}^{N}\sum_{i=1}^{d_{\boldsymbol{y}}}\frac{|y_i^k - \hat{y}_i^k|}{d_{\boldsymbol{y}}}, \quad \text{ACC}[0] \triangleq \frac{1}{N}\sum_{k=1}^{N}\frac{|\mathcal{O}_k \cup \hat{\mathcal{O}}_k|}{|\mathcal{O}_k|}$$

where N is the number of evaluated samples; $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ correspond to the ground truth and predicted peak water levels of the $d_{\boldsymbol{y}}$ locations of interest, respectively; $\bar{\boldsymbol{y}}^k \triangleq \mathbf{1} \cdot \frac{1}{d_{\boldsymbol{y}}} \sum_{i=1}^{d_{\boldsymbol{y}}} y_i^k$ denotes the mean vector of actual peak water level values for the k-th sample; Δ is an error threshold (in meters); $\mathcal{S}_\Delta \triangleq \{i \in [d_{\boldsymbol{y}}] : |y_i^k - \hat{y}_i^k| > \Delta\}$; $\Psi \triangleq \dfrac{\|\boldsymbol{y}^k - \hat{\boldsymbol{y}}^k\|_2^2}{\|\boldsymbol{y}^k - \bar{\boldsymbol{y}}^k\|_2^2}$; $\mathcal{O}_k \triangleq \{i \in [d_{\boldsymbol{y}}] : y_i^k = 0\}$; $\hat{\mathcal{O}}_k \triangleq \{i \in [d_{\boldsymbol{y}}] : \hat{y}_i^k = 0\}$. Here, ARTAE, ARMSE, and AMAE stand for average relative total absolute error, average RMSE, and average MAE, respectively; $\delta > \Delta$ quantifies the fraction of cases where the error in predicted floodwater levels exceeds the specified threshold Δ – an important metric for assessing models' reliability and gaining a more nuanced understanding of their performance; and ACC[0] measures the zero detection rate (that is, models' accuracy of detecting non-inundated locations).

5.2 Results and Comparison

Table 1 summarizes the candidate models' performance on the Test and Holdout datasets averaged over the three data splits. Notable observations from the first group of benchmark models are as follows. The two-stage Kriging with PCA approach, commonly employed in prior works, slightly improves upon Linear Regression by achieving 22.1% ARTAE, but δ errors are comparable, indicating a similar prediction quality. On the other hand, we observe significant improvement with Lasso with Poly, which reached the highest ARMSE of 0.42% and $R^2 = 0.95$ across all the models while achieving a $\delta > 0.5$ error of 5-7%, about half of the error produced by Kriging with PCA. This signifies that the incorporation of engineered input features that model the interactions between the protected and unprotected precincts can improve the prediction quality. Among the first group of models, SVR achieved the lowest δ errors, which indicates a higher quality of predictions, yet ACC[0] is extremely low at around 20%. However, it should be noted that in the case of SVR, a separate model has to be trained for every output coastal location independently, which raises potential scalability issues.

The DL models trained with the proposed framework, comprising the second group, significantly outperformed the above benchmark models in terms of AMAE (by a factor of 2 on average), ARTAE (by a factor of 2-5) and δ errors (by a factor of 2-5), and especially for ACC[0] (more than two-fold). The version of Attention U-net with the pre-trained weights achieved only modest improvements over the version trained from scratch, specifically a marginal improvement of 0.1% of ARTAE and 0.01% of $\delta > 0.5$ error. This result could possibly be attributed to the stark difference in image modalities and sizes between the current dataset and ImageNet, aligning with the findings and conclusions drawn in (Zhuang et al. 2021). The two best-performing candidate models were SWIN-Unet and CASPIAN, the former a close runner-up to the latter for the majority of the metrics. Notably, CASPIAN attained the highest scores for all metrics in the Holdout dataset while requiring only a fraction of SWIN-Unet's model's size. As reported in Table 2, CASPIAN achieved $\delta > 0.5$ error of only 1% on both datasets, and the performance with respect to the other metrics is consistent on both datasets, demonstrating the generalizability of the introduced CNN model.

5.3 Ablation Experiments

To supplement the evaluation results reported in Sec. 5.2, this section ablates the key architectural components introduced in CASPIAN. In particular, we remove/truncate the blocks/modules individually in four steps, resulting in the following versions: (i) CASPIAN without the final channel-wise summation, abbreviated as CASPIAN_B, (ii) CASPIAN with the depth of the central bottleneck reduced to 2 (i.e., $M = 2$), denoted as CASPIAN_Γ, (iii) CASPIAN with the modulation block removed, denoted as CASPIAN_Z, (iiii) CASPIAN with the pooling path completely eliminated, referred to as CASPIAN_Ω.

The results of ablation experiments are tabulated in Table 3. CASPIAN_B and CASPIAN_Γ achieved similar results on the test dataset compared to CASPIAN, yet $\delta > 0.1$ error of the produced predictions on the holdout dataset nearly doubled, indicating poorer generalizability. This observation corroborates the importance of the proposed deep central bottleneck and the final summation operator. In the case of CASPIAN_Z and CASPIAN_Ω, a significant drop in performance was observed on both datasets approaching that of Attention U-Net. This outcome can be expected since, after the elimination of the pooling path, the architecture of the latter two more closely resembles that of Attention U-Net.

Model	**Test Dataset (18 samples)**			**Holdout Dataset (32 samples)**			**# of parameters**
	AMAE	ARMSE	$\delta > 0.1$ (%)	AMAE	ARMSE	$\delta > 0.1$ (%)	
CASPIAN_B	0.06 ± 0.03	0.44 ± 0.19	3.61 ± 2.12	0.07 ± 0.06	0.45 ± 0.20	6.27 ± 10.5	= CASPIAN
CASPIAN_Γ	0.06 ± 0.04	0.45 ± 0.19	5.62 ± 4.91	0.07 ± 0.05	0.47 ± 0.20	5.59 ± 4.41	≈ 0.215 Mil.
CASPIAN_Z	0.10 ± 0.07	0.58 ± 0.30	7.04 ± 3.40	0.11 ± 0.09	0.62 ± 0.29	7.46 ± 2.81	≈ 0.344 Mil.
CASPIAN_Ω	0.10 ± 0.07	0.55 ± 0.31	6.21 ± 2.97	0.11 ± 0.09	0.62 ± 0.29	6.93 ± 2.81	≈ 0.341 Mil.

Table 3: Results of the ablation studies.

6 Concluding Remarks

In this paper, we presented a data-driven surrogate modeling framework for developing accurate and reliable coastal flooding metamodels powered by vision-based DL techniques. The proposed framework was tested on three different DL architectures, including a lightweight CNN model CASPIAN introduced in this work. The developed models were shown to significantly outperform existing geostatistical methods and standard regression techniques commonly employed in prior studies. The best-performing model, CASPIAN, closely and consistently emulated the results of the high-fidelity hydrodynamic simulator, on average achieving an AMAE error of 0.06 and $\delta > 0.5$ error of only around 1% on both Test and Holdout datasets.

One limitation of the developed coastal metamodels is that they are currently domain-specific since the training was performed considering one specific shoreline, SLR scenario, and fixed set of wind parameters. However, without major modifications to the proposed framework, one can extend the proposed framework to account for different coastal areas, for example, by including other geographical data such as local slopes, and hydraulic connectivities in the input maps. Another possible extension would be to expand the predictive scope and, in addition to peak water levels, also estimate the maximum velocities of floodwaters, a key prediction needed for coastal damage assessment.

References

Abu Dhabi Urban Planning Council. 2007. Plan Abu Dhabi 2030: Urban Structure Framework Plan.

Al Ahbabi, F. 2017. Department of Urban Planning and Municipalities. *Plan Maritime: Abu Dhabi coastal and marine framework plan. Abu Dhabi*, 238.

Al Kajbaf, A.; and Bensi, M. 2020. Application of surrogate models in estimation of storm surge: A comparative assessment. *Applied Soft Computing*, 91: 106184.

authors, A. 2022. Anonymous Title. *Hydrology*.

Badrinarayanan, V.; Kendall, A.; and Cipolla, R. 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12): 2481–2495.

Barnard, P. L.; van Ormondt, M.; Erikson, L. H.; Eshleman, J.; Hapke, C.; Ruggiero, P.; Adams, P. N.; and Foxgrover,

A. C. 2014. Development of the Coastal Storm Modeling System (CoSMoS) for predicting the impact of storms on high-energy, active-margin coasts. *Natural hazards*, 74: 1095–1125.

Bentivoglio, R.; Isufi, E.; Jonkman, S. N.; and Taormina, R. 2022. Deep learning methods for flood mapping: a review of existing applications and future research directions. *Hydrology and Earth System Sciences*, 26(16): 4345–4378.

Bomers, A.; and Hulscher, S. J. M. H. 2023. Neural networks for fast fluvial flood predictions: Too good to be true? *River Research and Applications*, 39(8): 1652–1658.

Bouhlel, M. A.; Hwang, J. T.; Bartoli, N.; Lafage, R.; Morlier, J.; and Martins, J. R. R. A. 2019. A Python surrogate modeling framework with derivatives. *Advances in Engineering Software*.

Cao, H.; Wang, Y.; Chen, J.; Jiang, D.; Zhang, X.; Tian, Q.; and Wang, M. 2022. Swin-unet: Unet-like pure transformer for medical image segmentation. In *European conference on computer vision*, 205–218. Springer.

Chollet, F. 2017. Xception: Deep Learning with Depthwise Separable Convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1800–1807. Los Alamitos, CA, USA: IEEE Computer Society.

Chu, H.; Wu, W.; Wang, Q. J.; Nathan, R.; and Wei, J. 2020. An ANN-based emulation modelling framework for flood inundation modelling: Application, challenges and future directions. *Environmental Modelling & Software*, 124: 104587.

Delft University of Technology. 2022. Simulating Waves Nearshore (SWAN). https://swanmodel.sourceforge.io.

DeVries, T.; and Taylor, G. W. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

Dickie, G. 2022. Floods, other water-related disasters could cost global economy $5.6 trillion by 2050 -report. Reuters [Online: https://bit.ly/3RcTMOf].

Egbert, G. D.; and Erofeeva, S. Y. 2002. Efficient inverse modeling of barotropic ocean tides. *Journal of Atmospheric and Oceanic technology*, 19(2): 183–204.

El Garroussi, S.; Ricci, S.; De Lozzo, M.; Goutal, N.; and Lucor, D. 2022. Tackling random fields non-linearities with unsupervised clustering of polynomial chaos expansion in latent space: application to global sensitivity analysis of river flooding. *Stochastic Environmental Research and Risk Assessment*, 36(3): 693–718.

Fu, X.; Yin, W.; Hu, M.; Wang, K.; Ma, Y.; Tan, P.; Shen, S.; Lin, D.; and Long, X. 2024. GeoWizard: Unleashing the Diffusion Priors for 3D Geometry Estimation from a Single Image. *arXiv preprint arXiv:2403.12013*.

Garner, A. J.; Mann, M. E.; Emanuel, K. A.; Kopp, R. E.; Lin, N.; Alley, R. B.; Horton, B. P.; DeConto, R. M.; Donnelly, J. P.; and Pollard, D. 2017. Impact of climate change on New York City's coastal flood hazard: Increasing flood heights from the preindustrial to 2300 CE. *Proceedings of the National Academy of Sciences*, 114(45): 11861–11866.

Guo, Z.; Leitão, J. P.; Simões, N. E.; and Moosavi, V. 2021. Data-driven flood emulation: Speeding up urban flood predictions by deep convolutional neural networks. *Journal of Flood Risk Management*, 14(1): e12684.

Haigh, I. D.; Pickering, M. D.; Green, J. A.; Arbic, B. K.; Arns, A.; Dangendorf, S.; Hill, D. F.; Horsburgh, K.; Howard, T.; Idier, D.; Jay, D. A.; Jänicke, L.; Lee, S. B.; Müller, M.; Schindelegger, M.; Talke, S. A.; Wilmes, S. B.; and Woodworth, P. L. 2020. The Tides They Are A-Changin': A Comprehensive Review of Past and Future Nonastronomical Changes in Tides, Their Driving Mechanisms, and Future Implications. *Reviews of Geophysics*, 58(1): e2018RG000636.

Hereher, M. E. 2020. Assessment of climate change impacts on sea surface temperatures and sea level rise—The Arabian Gulf. *Climate*, 8(4): 50.

Hofmann, J.; and Schüttrumpf, H. 2021. floodGAN: Using Deep Adversarial Learning to Predict Pluvial Flooding in Real Time. *Water 2021, Vol. 13, Page 2255*, 13(16): 2255.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-Excitation Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7132–7141.

Huber, P. J. 1992. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, 492–518. Springer.

Hummel, M. A.; Griffin, R.; Arkema, K.; and Guerry, A. D. 2021. Economic evaluation of sea-level rise adaptation strongly influenced by hydrodynamic feedbacks. *Proceedings of the National Academy of Sciences of the United States of America*, 118(29): e2025961118.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-To-Image Translation With Conditional Adversarial Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jia, G.; Taflanidis, A. A.; Nadal-Caraballo, N. C.; Melby, J. A.; Kennedy, A. B.; and Smith, J. M. 2016. Surrogate modeling for peak or time-dependent storm surge prediction over an extended coastal region using an existing database of synthetic storms. *Natural Hazards*, 81(2): 909–938.

Jia, G.; Wang, R. Q.; and Stacey, M. T. 2019. Investigation of impact of shoreline alteration on coastal hydrodynamics using Dimension REduced Surrogate based Sensitivity Analysis. *Advances in Water Resources*, 126: 168–175.

Jones, A.; Kuehnert, J.; Fraccaro, P.; Meuriot, O.; Ishikawa, T.; Edwards, B.; Stoyanov, N.; Remy, S. L.; Weldemariam, K.; and Assefa, S. 2023. AI for climate impacts: applications in flood risk. *npj Climate and Atmospheric Science*, 6(1): 63.

Jongman, B. 2018. Effective adaptation to rising flood risk. *Nature Communications*, 9(1): 1986.

Kyprioti, A. P.; Taflanidis, A. A.; Nadal-Caraballo, N. C.; and Campbell, M. 2021. Storm hazard analysis over extended geospatial grids utilizing surrogate models. *Coastal Engineering*, 168: 103855.

Kyprioti, A. P.; Taflanidis, A. A.; Nadal-Caraballo, N. C.; Yawn, M. C.; and Aucoin, L. A. 2022. Integration of node classification in storm surge surrogate modeling. *Journal of Marine Science and Engineering*, 10(4): 551.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

McKay, M. D.; Beckman, R. J.; and Conover, W. J. 1979. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2): 239–245.

Melville-Rea, H.; Eayrs, C.; Anwahi, N.; Burt, J. A.; Holland, D.; Samara, F.; Paparella, F.; Al Murshidi, A. H.; Al-Shehhi, M. R.; and Holland, D. M. 2021. A Roadmap for Policy-Relevant Sea-Level Rise Research in the United Arab Emirates. *Frontiers in Marine Science*, 8: 670089.

Mosavi, A.; Ozturk, P.; and Chau, K.-w. 2018. Flood Prediction Using Machine Learning Models: Literature Review. *Water*, 10(11).

Oktay, O.; Schlemper, J.; Le Folgoc, L.; Lee, M.; Heinrich, M.; Misawa, K.; Mori, K.; McDonagh, S.; Hammerla, N. Y.; Kainz, B.; et al. 2018. Attention U-Net: Learning Where to Look for the Pancreas. In *Medical Imaging with Deep Learning*.

O'Malley, T.; Bursztein, E.; Long, J.; Chollet, F.; Jin, H.; Invernizzi, L.; et al. 2019. KerasTuner. https://github.com/keras-team/keras-tuner.

Owen, A. B. 2007. A robust hybrid of lasso and ridge regression. *Contemporary Mathematics*, 443(7): 59–72.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of Machine Learning Research*, 12: 2825–2830.

Rohmer, J.; Sire, C.; Lecacheux, S.; Idier, D.; and Pedreros, R. 2023. Improved metamodels for predicting high-dimensional outputs by accounting for the dependence structure of the latent variables: application to marine flooding. *Stochastic Environmental Research and Risk Assessment*, 1–23.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Navab, N.; Hornegger, J.; Wells, W. M.; and Frangi, A. F., eds., *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 234–241. Cham: Springer International Publishing.

Sha, Y. 2021. Keras-unet-collection. https://github.com/yingkaisha/keras-unet-collection.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Stanton, C. Y. 2023. *Overcoming Data Limitation Challenges in Predicting Tropical Storm Surge with Interpretable Machine Learning Methods*. Ph.D. thesis, Texas A&M University-Corpus Christi.

Subraelu, P.; Sefelnasr, A.; Yagoub, M.; Sherif, M.; Ebraheem, A.; et al. 2022. Global Warming Climate Change and Sea Level Rise: Impact on Land Use Land Cover Features along UAE coast through Remote Sensing and GIS. *J Ecosys Ecograph*, 12(329): 2.

Sweet, W. V.; Hamlington, B. D.; Kopp, R. E.; Weaver, C. P.; Barnard, P. L.; Bekaert, D.; Brooks, W.; Craghan, M.; Dusek, G.; Frederikse, T.; et al. 2022. *Global and regional sea level rise scenarios for the United States: Updated mean projections and extreme water level probabilities along US coastlines*. National Oceanic and Atmospheric Administration.

UN Atlas of the Oceans. 2016. United Nations Atlas of the Oceans. http://www.oceansatlas.org/facts/en/.

Wang, R. Q.; Stacey, M. T.; Herdman, L. M. M.; Barnard, P. L.; and Erikson, L. 2018. The Influence of Sea Level Rise on the Regional Interdependence of Coastal Infrastructure. *Earth's Future*, 6(5): 677–688.

Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K. 2017. Aggregated Residual Transformations for Deep Neural Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5987–5995.

Yang, L.; Kang, B.; Huang, Z.; Xu, X.; Feng, J.; and Zhao, H. 2024. Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data. arXiv:2401.10891.

Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; and He, Q. 2021. A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, 109(1): 43–76.

A Coastal Area and its Hydrodynamic Model

The geographical area selected for the application of the proposed surrogate modeling framework stretches along the coast of Abu Dhabi, which is the capital of the United Arab Emirates (UAE) situated inside the Persian Gulf. UAE's coastline features a low-lying and shallow-sloping (about 35 cm per km) topography (Melville-Rea et al. 2021). Over 85% of the population and more than 90% of the local infrastructure of the UAE is within a few meters of the present-day sea level (Al Ahbabi 2017). Notably, Abu Dhabi is comprised of a system of coastal mangrove islands, coral reefs and artificial islands, and 50% of its area lies only within 1 m above sea level (Hereher 2020; Subraelu et al. 2022). Considering that possible SLR estimates are on the order of 1.0 to 1.5 m by the end of 2100, most of the built and natural mangrove ecosystem of Abu Dhabi, along with its coastal communities, will potentially be exposed to flood hazards and subsequent damages.

Given the complex structure of Abu Dhabi's coastline, it is necessary to consider the protection of different sections. The partitioning scheme chosen for the purposes of this study was informed by the precincts defined in the 2030 Urban Structure Framework Plan of Abu Dhabi (Abu Dhabi Urban Planning Council 2007). For further refinement of partitions, we divided the main island of Abu Dhabi, grouped other islands, and delineated the boundaries between some precincts, which yielded 17 individual coastal segments that constitute the candidate sites for installation of engineered fortifications, as depicted in Fig. 1.

For simulating the tidal dynamics of the considered region, we utilized Delft3D, which is a hydrodynamic model that solves the time-dependent Reynolds Averaged Navier Stokes differential equations. That is, Delft3D is a physics-based numerical model that considers the time-varying forces exerted on a water body (such as the entire Persian Gulf) due to hydrostatic pressures (such as SLR), tidal forcing, wind and storm stresses, bottom (seabed) friction, and river inflows over a finite-element computational grid (up to 30 m in horizontal resolution) spread over variable bathymetry. For any point in this grid, it can provide time series outputs, with 30-minute intervals, of water levels and local water circulation velocities throughout the specified simulation period. Importantly, Delft3D can handle computational grid cells that alternate between dry and wet states (Barnard et al. 2014).

The tidal model was validated by running the Delft3D simulator over a 3-month period between 1 January and 31 March 2017 (without wind forcing) and computing the root mean squared error between the model outputs of hourly water levels at 194 locations throughout the gulf and hourly tidal gauge water level data obtained from the TPXO8 Ocean Atlas for the same period (https://www.tpxo.net/global/tpxo8-atlas. Also see (Egbert and Erofeeva 2002)). The model was calibrated by adjusting the bottom Manning's roughness coefficient for the entire gulf domain from 0.015 to 0.030. The lowest overall error was attained under the coefficient of 0.02, which was taken as the calibrated roughness value for the gulf model going forward. Fig. 4 demonstrates a typical fit between water level values (relative to the mean sea level) outputted by the model and the tidal gauge data at two representative locations near the UAE shore. Further details concerning the employed hydrodynamic model and its validation can be found in (authors 2022).

To account for wind-induced wave activity in the vicinity of Abu Dhabi's coast, the validated Delft3D model was rerun with wind forcing from the ERA5 database, and the results were fed to an additional spectral wave model, Simulating Waves Nearshore (Delft University of Technology 2022), which allows capturing wind-wave generation, wave diffraction, amplification and refraction of water surface waves as they approach the shoreline. The SWAN model was applied at a scale of about 100 km

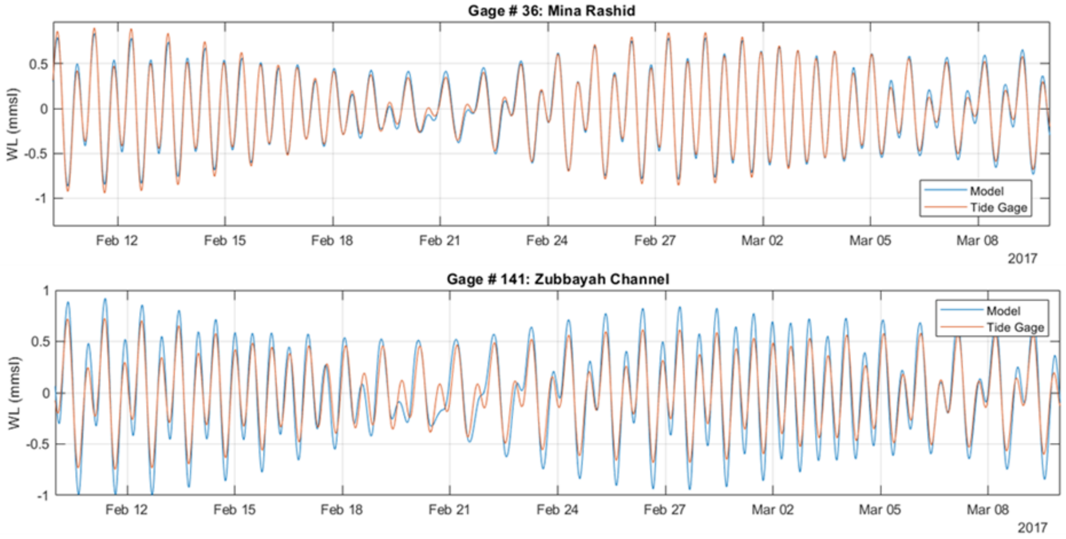

Figure 4: Comparison of the simulated tidal model's output and TPXO8 Atlas' time-series data on water levels from 10 Feb to 10 March 2017 at Mina Rashid (on top) and Zubbayah Channel (on bottom). The units of vertical axes are in meters relative to the mean sea level.

along the shoreline to about 50 km offshore under the same forcing from the ERA5 database. Finally, along the interface of the waves with the coastline, the SWAN-computed significant wave heights and the local shoreline slope were used to compute the run-up elevations along the coastline where the waves hit the shore.

With the above coupled hydrodynamic model in place, illustrated in Fig. 5, one can run a reference case with no shoreline armoring (except those already existing in Abu Dhabi) to evaluate the maximum extent of flooding due to SLR and storms. For every protection scenario of the shoreline, the raw output of the model will include 3 months worth of hourly water levels for more than $400,000$ grid point locations throughout the Persian Gulf. To filter the nearshore inland locations of interest, based on which the effectiveness of protection scenarios can be appraised, the following two steps were taken: (i) the points lying outside the urban region of Abu Dhabi were excluded; (ii) the inland cells that never experienced flooding even in the case of no coastal protection (i.e., are not hydraulically connected to the Gulf or bear no correlation with the input) were removed. This resulted in the final set of 12066 locations along the coastline, which appear in Fig. 1.

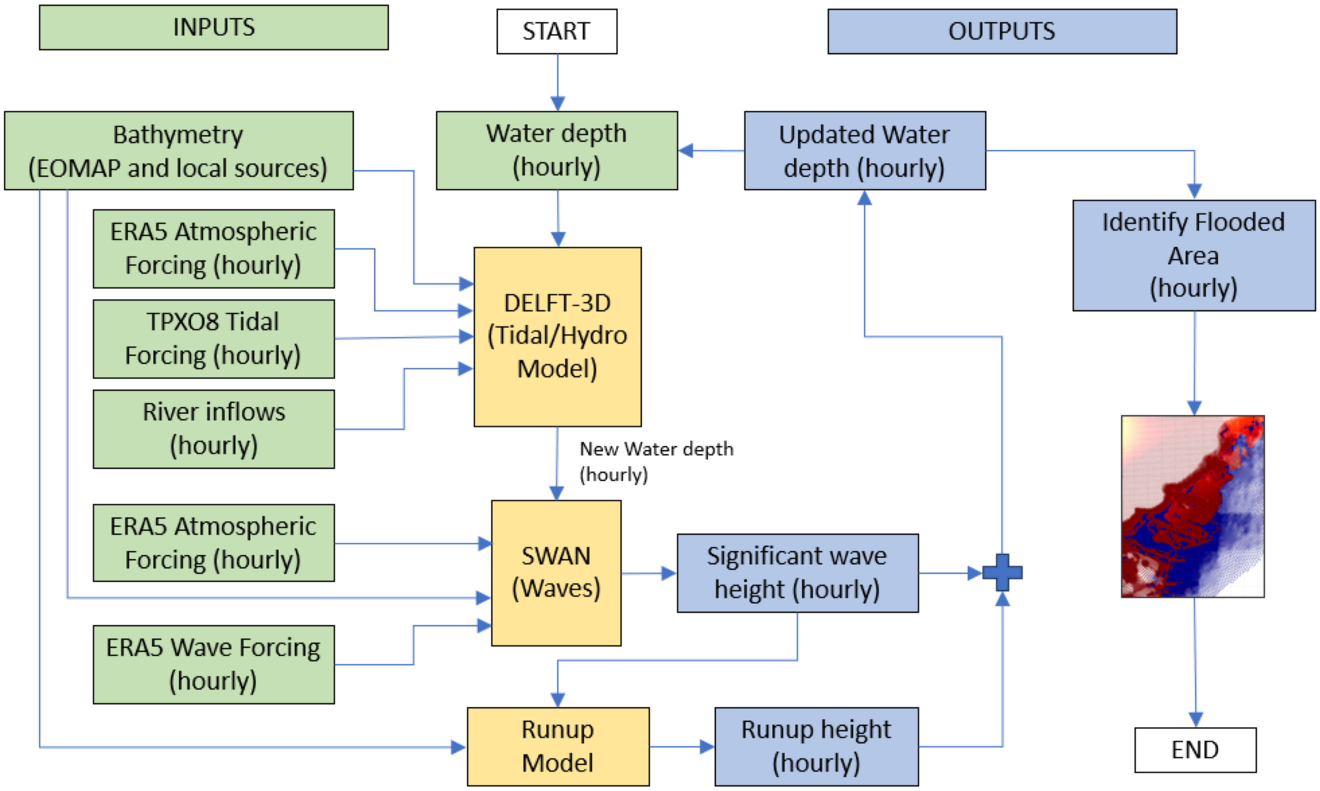
INPUTS
START
OUTPUTS
Bathymetry (EOMAP and local sources)
Water depth (hourly)
Updated Water depth (hourly)
Identify Flooded Area (hourly)
ERA5 Atmospheric Forcing (hourly)
TPXO8 Tidal Forcing (hourly)
River inflows (hourly)
DELFT-3D (Tidal/Hydro Model)
New Water depth (hourly)
ERA5 Atmospheric Forcing (hourly)
SWAN (Waves)
Significant wave height (hourly)
ERA5 Wave Forcing (hourly)
Runup Model
Runup height (hourly)
END

Figure 5: Schematic of hydrodynamic model elements. Green elements denote input parameters; Yellow are the constituent sub-models; Blue are model outputs. Running one cycle of the above model will generate an hourly update of the water depth throughout the UAE coastline.

B Hyperparameters of the Trained DL Models

Attention U-net and **Attention U-net**[‡‡]**:**

- `input_size` = (1024, 1024, 3)
- `filter_num` = [32, 64, 128, 256] (number of filters for each down- and up-scaling level)
- `stack_num_down` = 2 (number of layers per downsampling level/block)
- `stack_num_up` = 2 (number of layers (after concatenation) per upsampling level/block
- `activation` = 'ReLU'
- `atten_activation` ='ReLU'
- `output_activation` ='ReLU'
- `batch_norm` = False

- backbone = ’VGG19’ (the backbone model name)
- encoder_weights = ’random’ (**for Attention U-net**) and ’imagenet’ (**for Attention U-net**[‡‡])
- freeze_backbone = False

SWIN-Unet:

- filter_num_begin = 64 (number of channels in the first downsampling block)
- depth = 4 (the depth of Swin U-net, 4 means 3 down/upsampling levels and a bottom level)
- stack_num_down = 2 (number of Swin Transformers per downsampling level)
- stack_num_up = 2 (number of Swin Transformers per upsampling level)
- patch_size = 8
- att_heads = 4 (number of attention heads per down/upsampling level)
- w_size = 4 (the size of attention window per down/upsampling level)
- mlp_ratio = 4 (ratio of MLP hidden dimension to embedding dimension)

CASPIAN:

- input_shape = (1024, 1024, 1)
- F = 72
- K = 4
- C = 34
- M = 8
- modulation_level = 1
- p = 4
- r = 0.85 × F
- activation = ’tanh’
- init = "glorot_normal"

C Details of the Hyperparameter Tuning

The hyerparameters of the trained DL models reported in the previous section were selected as follows. Since Attention U-net and Attention U-net[‡‡] were implemented with the VGG19 backbone, we tuned the available remaining hyperparameters (learning rate and activations) manually. For SWIN-Unet and CASPIAN, the selection of hyperparameters was optimized through the Random Search algorithm provided as part of the Keras Tuner library. The considered search space of hyperparameter values for both models is listed hereunder.

Hyperparameter Search Space for CASPIAN:

- F : $\{min = 64, max = 128, step = 4\}$
- K : $\{min = 3, max = 6, step = 1\}$
- C : $\{min = 12, max = 64, step = 4\}$
- M : $\{min = 4, max = 12, step = 2\}$
- modulation_level : $\{min = 1, max = 3, step = 1\}$
- p : $\{min = 2, max = 8, step = 2\}$
- r : $\{min = 0.5, max = 0.95, sampling = linear\}\cdot$ F
- activation : [’relu’, ’tanh’, ’swish’]
- learning_rate : $\{min = 5e-5, max = 5e-3, sampling = log\}$

Hyperparameter Search Space for SWIN-Unet:

- filter_num_begin : $\{min = 64, max = 160, step = 16\}$
- depth : $\{min = 3, max = 4, step = 1\}$
- stack_num_(up down) : $\{min = 1, max = 2, step = 1\}$
- patch_size : [4, 8, 16]
- att_heads : $\{min = 2, max = 4, step = 2\}$
- dropout : [0.0, 0.05, 0.1, 0.2]
- learning_rate : $\{min = 1e-5, max = 5e-4, sampling = log\}$

D Holdout Dataset

Table 4: Protection scenarios considered in the Holdout dataset.

Holdout protection scenarios (32 in total)				
00110011001100110	11100000000000111	00000111100000111	00011000110001100	11110000111100001
00000011111100000	11110000000001111	00000111111100000	11111100000111111	00001111111110000
11111000001111100	00001110000111000	10101010101010101	11111110000001111	00000001111110000
11111000000011111	11111110000000111	00000111110000011	00011100011100011	00000001111111000
11000000000000011	00111111111111100	01010101010101010	11111100000011111	11111000011111000
00000011111000000	11110001111000111	11100011100011100	00001111000011110	11001100110011001
11100111001110011	00011111111111000			