Machine learning for post-storm profile predictions

MSc. Thesis

Koen van Asselt

Delft University of Technology

Machine learning for post-storm profile predictions

MSc. Thesis

by

Koen van Asselt

Student Name	Student Number
Koen van Asselt	4606434

TU Delft supervisors:	Dr. ir. J.A.A. Antolinez (Chair)
	Prof. dr. ir. A.J.H.M. Reniers
	Dr. ir. A. Heinlein
Deltares supervisors:	Dr. ir. P. Athanasiou
	Dr. ir. R. McCall
Project Duration:	Sep, 2022 - Jun, 2023
Faculty:	Faculty of Civil Engineering and Geosciences, Delft

Preface

In front of you lies my thesis "Machine Learning for Dune Erosion", written to obtain a master's degree in Hydraulic Engineering. I have been working on this document for the past nine months and I am proud to say that I have learned a lot. As the title suggests, I have attempted to combine two topics, dune erosion and machine learning. For a hydraulic engineering student, the former is a well-known topic and the latter is a big leap into the dark and an opportunity to learn something new. The balanced combination of the two has kept my interest and enthusiasm throughout the duration of this project.

While I am writing this preface from a first-person "I" perspective, the first-person "we" perspective might be more suitable. Because this research would have been quite a bit more difficult and especially less enjoyable without the help of a few people. First of all, I would like to thank Panos for his great help and enthusiasm for this subject. Our weekly catch-up sessions got increasingly more interesting by means of content and time management. Without your insightful comments and suggestions, this document would not have looked the same. I am looking forward to continuing with this topic together! Next to that, I would like to express my gratitude to the rest of my committee. José, Robert, Ad and Alexander, I appreciated your insightful feedback and critical questions throughout the duration of this project.

I would also like to thank my colleagues at Deltares for the opportunity to carry out my thesis in such a welcoming and inspiring environment. The shared motivation to make the world a better place is tangible during every meeting, brainstorm session, lunch and presentation. I would like to send a special thanks to my fellow intern Marloes. Throughout the last couple of months, I have enjoyed enabling the delta together.

Finally, I would like to thank my family and friends who have shown their relentless support during my studies. Peter and Greetje, I am grateful for the possibilities and support you offered me. Maartje, thank you for the critical feedback on the report and for simply being an amazing twin. Stijn, thank you for learning me to don't worry but be happy. Floor, thank you for the motivating *wie dan!*'s and for challenging me in making this project a little bit more comprehensible for medical students and the rest of the world. Next to that, I would like to thank the friends I have made during my studies at the civil engineering faculty, the Warmoezierstraat 46 and 't Entresol for a wonderful experience over the last 7 (!) years.

Koen van Asselt
Delft, February 2024

Summary

Sandy coasts cover about 30% of the world's coastline and offer several economic and ecological services. Dunes are a typical feature of sandy coasts and offer natural protection from the sea. In extreme storm conditions, the impact of the sea on these dunes results in sediment transport in seaward direction. To be able to guarantee the protection service of dunes and prevent the hinterland from flooding, models are developed to predict sediment transport processes at the dune (dune erosion).

Surrogate models, model of a model, have been developed to reduce computational times of dune erosion calculation. The state of art surrogate models provide a prediction based on a parameterized input and output (Athanasiou et al., 2022 and Gharagozlou et al., 2022). However, a desired surrogate model is able to deal with spatial input and the prediction of the actual shape of the post-storm sandy profiles.

Initially, dune erosion processes and existing surrogate modelling techniques are explored. Through studying theory on neural networks, U-Net, a CNN architecture developed for image segmentation, is chosen as a suitable convolutional neural network to process 1D pre-storm input profiles and predict 1D post-storm profiles. The potential of using the U-Net architecture is explored with a simplified dataset with known morphological dune response, stationary storm conditions and several performance metric. Through this exploration, the goal is to replicate these dune erosion processes using a surrogate model. The gained insights are used to scale up to a more realistic scenario for the Holland coast.

Parameter sensitivity analyses on the DEV showed that, in general, steeper slopes of submerged profile sections lead to an increase in DEV. Especially the beach- and nearshore slope have a large effect on the modelled DEV.

In the exploration phase, several techniques to improve the U-Net architecture are presented. For pre-and post-storm profiles, the signal of sediment transport is difficult to pick up for a U-Net architecture and it needs guidance to localize this signal. Therefore, the input grid should highlight the area of interest. Next to that, It was found that the network depth, network width and kernel size are crucial hyperparameters for the interpretation of the data by U-Net and the performance of the surrogate model. A shallow U-Net architecture is not able to gain an understanding of the processes of dune erosion and attempts to find the statistically optimal solution. In contrast, a deeper and more complex U-Net enables the surrogate model the mimic dune erosion processes and catches a wider range of dune erosion volumes (DEVs). However, it is important to note that due to the lack of alongshore variability in the test data, the improvements resulting from a deeper architecture may not be fully reflected in the performance indicator.

In the upscaling phase, the results on a realistic training and test dataset confirm the trends found in the exploration phase of this research. A multi-profile-based training dataset outperforms a single-profile-based training dataset. Accuracy and skill in post-storm profile shape prediction are obtained through either deeper networks or larger kernel sizes. However, it was found that the current surrogate model has trouble overcoming spatial alterations at the location of erosion processes. .

All in all, it can be concluded that U-Net shows potential for post-storm profile predictions. Taking into account the original purpose of U-Net and the consequences of the network's architecture on profile predictions, an appropriate surrogate model can be set up. It is recommended to carefully proceed to the upscaling phase and to include more realistic pre-storm profiles in the training dataset. This comes with a re-evaluation of the hyperparameters and performance metrics.

Contents

1
Introduction

1.1. Background

Coastal zones are the transition areas between sea and land. Sandy coasts cover about 30% of the world's coastline (Luijendijk et al., 2018). These dynamic areas offer numerous economic and ecological services. Dunes are a typical feature of sandy coasts and offer natural protection from the sea. In extreme storm conditions, these dunes are affected by the sea and sediment is transported from the dune to the sea. To be able to guarantee this protection service of dunes and prevent flooding, models are developed to predict sediment transport processes at the dune.

The concept of this research originates from the coastal defence system of the Netherlands. The coastline of the Netherlands stretches from Zeeuws-Vlaanderen up to Rottermerplaat and covers around 432 km. This coastline is typically divided into three districts with specific morphological features. From South to North, The South-West Delta, a large river delta that originates from the interaction between the North Sea and the outlets of the Meuse, Rhine and Scheldt rivers. The Holland Coast, a wave-dominated area with a few natural disruptions. About 60% of the primary sea-defense consists of dunes with a grain size of fine to coarse sands. In general, these dunes are fronted with a sub-aerial beach and a shallow foreshore.

Due to the low elevation and arising storm conditions, the Dutch hinterland is vulnerable to flooding. About 50% The Dutch hinterland, protected by these dune systems, is elevated below mean sea level. The potential hazard arising during floods are of enormous scales and bring along catastrophic consequences. Next to that, the system should provide enough robustness to withstand extreme storm conditions. These conditions come with an increase of the intensity of hydrodynamics (e.g. water level and wave height). These hydrodynamics can lead to erosion of the beach and especially dunes. This dune erosion (Figure 1.1) is an important aspect in the primary defense system and requires accurate warning system to forecast extreme conditions and consequential morphological changes.

Throughout the years, several methods have been developed to accurately predict dune erosion volumes. Currently, Rijkswaterstaat (RWS) is using a process-based model (XBeach) to predict dune erosion. These predictions are used to do safety assessments of the coast. While post-storm profile predictions with XBeach are accurate, they require a lot of computational effort for large areas.

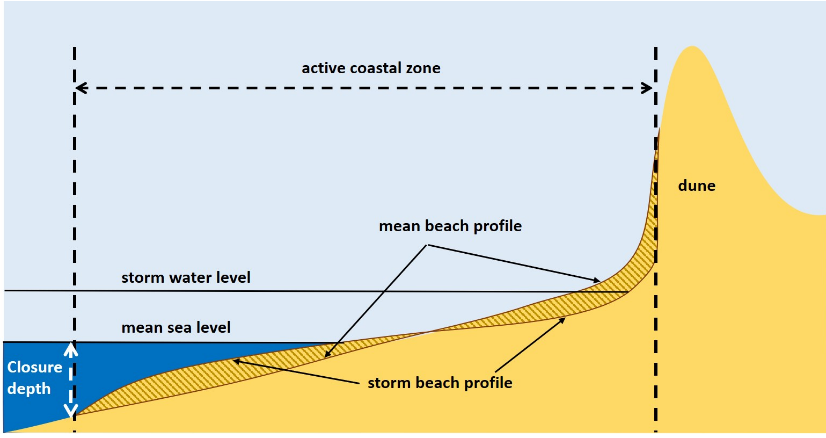


Figure 1.1: Schematic illustration of dune erosion at a sandy coast (Flanders Marine Institute, 2022).

To reduce computational efforts, surrogate models have been developed for dune erosion prediction. A surrogate model, a model from a model, can be used to describe the relationship between the input and output of a simulation model. This technique can reduce the computational time of a dune erosion prediction significantly (Athanasiou et al., 2022) and allows for a more in-depth exploration of uncertainty.

1.2. Problem statement

The state of art surrogate models for dune erosion provide a prediction of a morphological indicator based on a parameterized input and output. While skillful predictions can be made for distinct indicators, this limits the applicability of the surrogate model for other purposes. It would be favourable if the prediction of the post-storm dune response is not limited to a single indicator. This allows for more flexibility for the end-user of the model output and a wider range of post-processing possibilities. Therefore, a desired model is able to deal with spatial input and the prediction of actual shape of the post-storm sandy profiles. Gharagozlou et al. (2022) was able to make predictions of profile shapes using parameterized input (principal component analyses) and output (power-law function). While predicting the shape, still, pre-processing efforts were needed to reduce the dimensionality of the data. This research aims to limit the parameterization of the input- and output shapes, such that actual pre- and post-storm elevation points can be used.

1.3. Objectives and Research Questions

Based on the problem statement defined above, a research objective is formulated. This objective intends to describe the general purpose of this research and is supported by several sub-objectives. Each (sub-)objective is backed up by a research question.

Main objective

The main objective of this research can be formulated as: **Enabling fast prediction of actual post-storm sandy profiles along the Holland Coast using neural networks and XBeach.** State of art surrogate models are able to predict morphological indicators or parameterized input, but do not utilize the actual bed elevation points and profile shapes. Thus, this research explores the use of neural networks to set-up surrogate model and describe the relationship between modelled pre-storm and post-strom profiles. This method allows for faster prediction than process-based models.

Sub-objectives

Driving mechanisms of Dune Erosion

Before surrogate modelling steps are taken, the processes captured in the issued synthetic dataset should be properly understood. This synthetic dataset is supplied by the simulation model, the content of this dataset is of great importance. For this research, these data are supplied by the process-based model XBeach and will include several pre- and post-storm profiles. To judge whether surrogate models would be able to predict post-storm profiles and understand the dune erosion processes, the synthetic dataset is deliberately simplified to reduce its complexity. As a result, the pre-storm shape features in the input profiles can be isolated and analyzed independently. The modelled morphological response of the dune with XBeach is derived from the corresponding post-storm profiles. The shape features in the pre-storm profiles are characterized by the slope of different profile sections and the morphological response of the dune is quantified using the dune erosion volume (DEV). This is captured in the sub-objective: **obtain an understanding of the effects of different simplified pre-storm sandy profiles shapes on post-storm eroded dune profiles**.

Surrogate modelling

The input of the surrogate model originates from a synthetic dataset created with XBeach. This data should be pre-processed such that it is suitable and fits the neural network structure. Several neural network structures and set of hyperparameters can be applied for surrogate modelling. Each has its possibilities and limitations with respect to data processing and learning efficiency. **This research attempts to develop a neural network structure that is able predict post-storm profile shapes for a simplified set of synthetic pre-storm input profiles and stationary storm conditions**. Besides that, **a suitable standard performance metric should be developed to evaluate several surrogate modelling alternatives**.

Upscaling

After exploring the possibilities to predict post-storm profile shapes, the question arises whether the obtained insights can also scale-up to a realistic situation. This can be achieved by adjusting both the training and test data to a more adequate representation of the Holland Coast. Results obtained in the exploration phase of this research are evaluated for this realistic scenario. Therefore, **this research attempts to scale up findings from simplified post-storm profiles to a realistic situation, covering a wider range of profile types representing the Holland Coast.**.

Research questions

All sub-objectives are provided with a single or multiple research question(s).

Driving mechanisms of Dune Erosion

1. What response in dune erosion volumes is found in the post-storm profiles as a result of slope changes of sandy pre-storm profiles using a simplified dataset and XBeach?

Surrogate modelling

2. What performance metrics can be used to evaluate surrogate modelling using neural networks for post-storm profile shape prediction?

3. To what extent are pre-processing tools and neural networks able to make post-storm profile shape predictions for a simplified dataset?

Upscaling

4. Can the neural network structure, obtained in the exploration phase, be scaled-up to predict post-storm profile shapes of actual Holland Coast profile shapes?

1.4. General framework

The objectives formulated in section 1.3 involve aspects from both hydraulic and data engineering. Whereas the context is predominantly hydraulic engineering related, the proposed challenge and solution come with challenges regarding data engineering. The presented research framework has been set up to capture both facets but originates from a hydraulic engineering perspective.

Dealing with the challenge of developing new insight with respect to reducing computational efforts comes with a lot of possibilities (numerical schemes, machine learning, accuracy requirements, etc.). However, in this research, only the possibilities of dealing with this issue by applying surrogate modelling are explored. As described in section 1.2, one of the main challenges arising in this field is the embedding of the post-storm cross-shore profile in storm erosion surrogate modelling. To deal with this challenge, initially, the situation is simplified. Through this simplified situation, solutions to deal with this issue would not be limited by the complexity of the data, but mostly by the core of the issue itself. Once several neural network alternatives are explored, more complexity is introduced by considering a more realistic and wider range of profiles. This should bring the modelled situation closer to reality.

The term "machine learning techniques" is very broad. To specify, this research focuses on applying neural networks to set-up a surrogate model. This builds upon the work carried out by Athanasiou et al. (2022).

The general framework of this research is conceptually illustrated on the next page in Figure 1.2. The workflow throughout this thesis consists of two phases: (1) Exploration- and (2) Upscaling phase.

The exploration phase aims to understand the surrogate model's performance, identify any limitations, and gain valuable insights into the relationship between the input data and the predicted outcomes. The exploration phase is initiated by a literature study. This literature study consists of two main parts, storm erosion and neural networks. Based on this literature study, an initial synthetic dataset will be set-up with XBeach. This initial dataset represents a simplified situation of the Holland coast and will serve as a tool to explore surrogate modelling possibilities. Several neural network architectures will be evaluated and the performance of these techniques are assessed through different indicators. Based on these indicators, three steps can be taken: (1) Evaluate the neural network structure, (2) evaluate the training data or (3) proceed to the upscaling phase.

Upscaling involves extending the model's capabilities to handle larger and more diverse datasets. In general, the data which the surrogate model is trained on is scaled-up to a realistic scenario. This includes a better representation of the alongshore variability found a the Holland Coast. As the complexity of the dataset increases, the performance of the established U-Net structure may vary. The upscaling phase provides an opportunity to evaluate the model's scalability, robustness, and ability to handle increased variability and complexity in the input data.

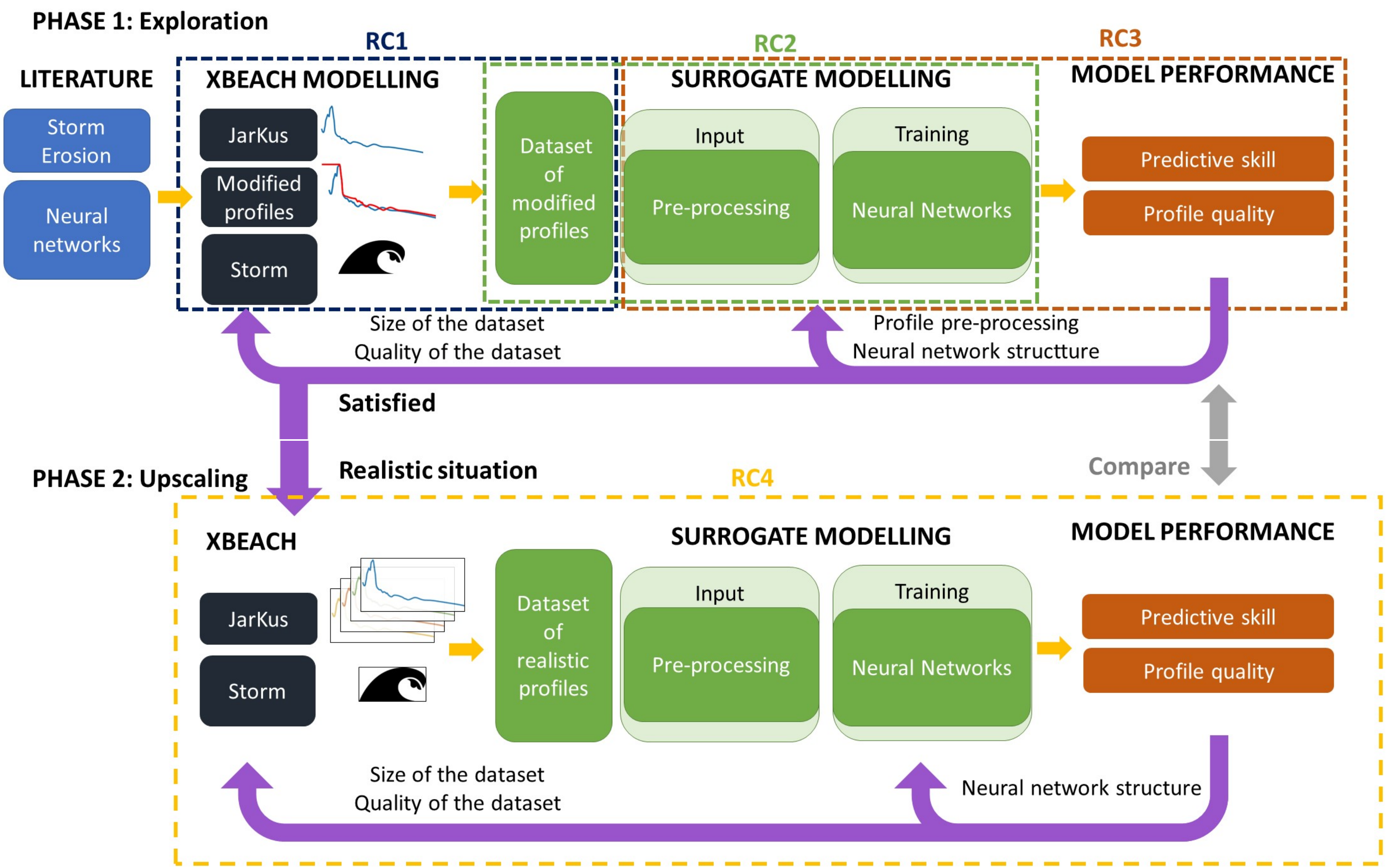


Figure 1.2: Flowchart of the general research framework indicating the two phases of the research. Different colours highlight the varying aspects of each phase and the arrows capture the direction of the flow.

2

Literature review

2.1. Dune Erosion Modelling

2.1.1. Cross-shore profile

Before getting into the processes of storm erosion, some definitions are introduced. These definitions typically relate to the characteristics of a cross-shore profile that are used in this research. Figure 2.1 shows a typical cross-shore profile in a dune system along with several characteristics. From sea to land, the profile can be subdivided in the nearshore, inter-tidal area, beach and dune. This section defines these sections.

- *Nearshore* - The nearshore starts at the depth of closure (DoC) and ends at mean sea level (MSL). The DoC is defined as the deepest elevation for which sediment transport is still dominated by surface waves.

- *Inter-tidal area* - The inter-tidal area is bounded by the mean low water and mean high water. These are the lowest and highest elevation reached by the tide.

- *Beach* - The beach is the first dry section that can be found on the cross-shore profile. This section starts at MHW and ends at the dune toe. In normal, no storm, conditions, this profile section is not reached by the water.

- *Dune* - The dune starts at the dune toe and proceeds towards the hinterland. The dune crest is the highest point found in this area and (together with the dune toe) bounds the duneface.

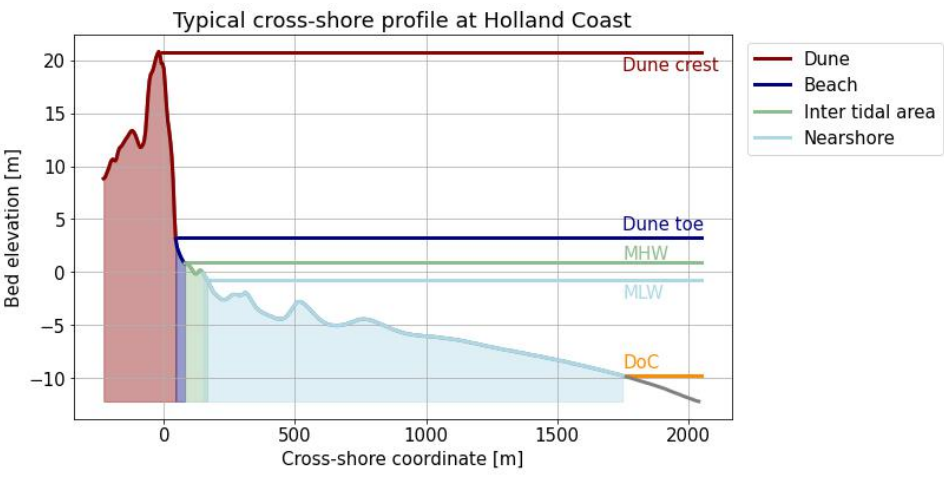


Figure 2.1: Schematization of a typical cross-shore profile at the Holland Coast. Different colours indicate the different profile sections along the profile.

2.1.2. Storm erosion

Coastal erosion processes can be characterized based on several temporal and spatial scales. The scope of this research is spatially bounded by the 1D cross-shore domain along the Dutch coast and considers the storm timescale (several hours/days). This section elaborates on these boundaries and discussed the involved processes.

In general, looking at the impact of a storm on a sandy coast, one finds a seaward transport of sediment and complementary a retreat of the coastline. An undertow current, caused by waves reaching the beach and/or dunes, in combination with the high suspended sediment concentration near the dune facilitates a large sediment transport capacity (Bosboom & Stive, 2021). As these waves are damped through bed friction, both the beach width and the storm surge level are found the be important parameters when considering coastal erosion (Bosboom & Stive, 2021). The net sediment loss can be captured in several morphological parameters such as the reduction of beach width, migration of the coastline and changes in beach volume. During storms, waves and currents cause sediment to be displaced in alongshore direction (Bosboom & Stive, 2021)). These, however, will not be taken into account for this 1D model study. Storm clusters, a sequence of several storms, can also result in (additional) coastal erosion events compared to a single storm. (Karunarathna et al., 2014).

Ruessink and Jeuken (2002) carried out data analyses to find trends in dunefoot dynamics along the Dutch Coast. Using the JarKus dataset, the importance of the pre-storm beach width and storm surge-level for dunefoot position were confirmed. Beuzen et al. (2019) carried out a study to identify controls of variability in storm erosion along the Australian coast. A dataset of laser measurements over 1700 cross-shore profiles enabled a data-driven Bayesian analysis to identify relationships between several morphological and hydrodynamic variables. Again, for dune erosion, the most important control variables were found to be wave height, beach width and wave run-up.

As for alongshore uniform coastlines, dune erosion is majorly caused by cross-shore processes, assessment of the storm impact can be done in 2D reference frame. A typical cross-shore pre- and post-storm profile as defined by Vellinga (1982) is depicted in Figure 2.2.

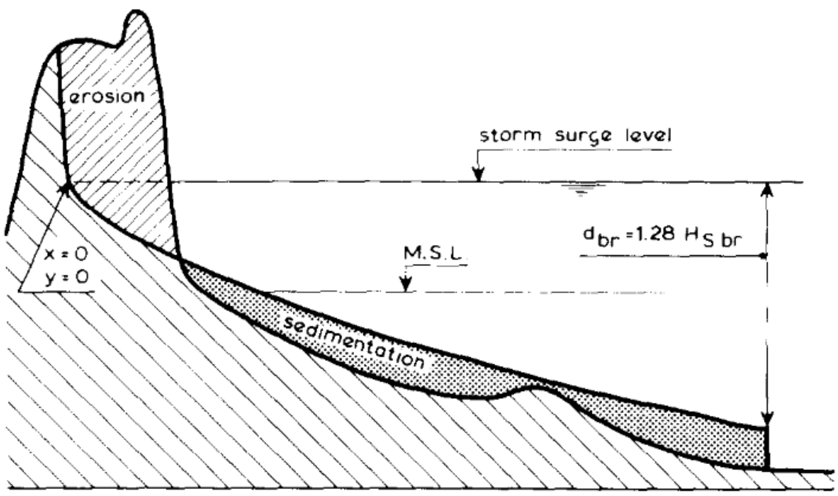


Figure 2.2: Typical pre- and post-storm profile shape as presented by Vellinga (1982)

This figure shows the bed level elevation (y) over the cross-shore distance (x). The mean sea level and surge level during storm conditions are illustrated as horizontal lines. In this case, the seaward transport of sediment results in erosion of the dune and deposition of sediment at the nearshore. As shown in the figure, the post-storm profile picks up at the toe of the dune (x = 0, y = 0) and continues in a uniform erosion profile. The mathematical expression (Equation 2.1) of this profile was derived by (physical) modelling attempts.

$$y = 0.415(x+4.5)^{0.5} - 0.88 \tag{2.1}$$

This relatively simple equilibrium approach (commonly referred to as the Duros model), was supplemented with additional variables on the significant wave height, wave peak period and grain size (Duros+) ((van Thiel de Vries et al., 2008)). Using this method, a post-storm profile can be obtained.

Profile adjustment schemes and critical slopes used in process-based models (see subsection 2.1.3) are in line with these expressions and show comparable geometrical features (Roelvink et al., 2009).

Sallenger (2000) developed a scale to categorize storms based on hydro/morphological impact. Recognizing several morphological developments as a result of the elevation of the water level with respect to the impacted beach profiles, four separate regimes can be distinguished:

- Swash regime

  During the swash regime, wave run up does not reach the toe of the dune ($R_{high} < D_{low}$) and is limited to the berm and beach face. As the dunes remain untouched by the hydrodynamic condition, the sediment transport is limited to the lower sub-aerial beach. As stated by Sallenger (2000), this can be compared to a typical winter storm with expected storm erosion on the foreshore and complete post storm recovery.

- Collision regime

  During the collision regime, wave run up exceeds the level of the dune toe and reaches the base of the dune ($D_{low} < R_{high} < D_{high}$). In addition to the transport the transport processes described for the swash regime, this level of impact results in a net transport of sediment from the dune to the foreshore (dune erosion). This sediment does not typically return to the re-established dune.

- Overwash regime

  During the overwash regime, wave run-up exceeds the dune crest ($R_{high} > D_{high}$). Overtopping and overwash of the dune top, results in placement of sediment over the top of the dune to a more landward position. In general, this regimes comes with a net landward migration of the coastline and a substantial loss of sediment.

- Inundation regime

  During the inundation regime, the dune system is fully submerged during the lowest surge ($R_{low} > D_{high}$. This results in breaching of the dune system and complete denudation of the beaches form sand.

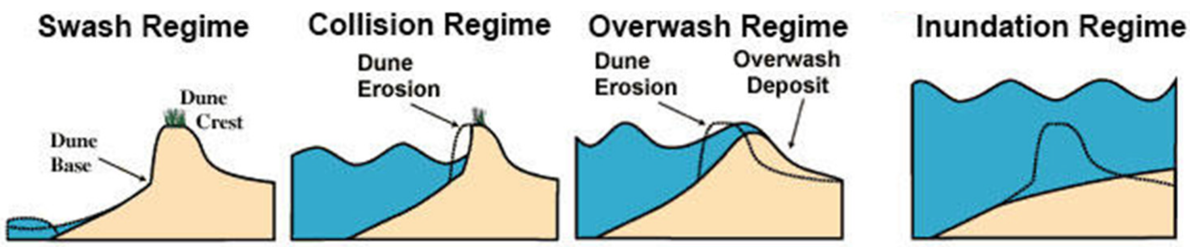


Figure 2.3: Storm impact regimes as defined by Sallenger, (2000), retrieved from Castelle and Harley (2020)

These regime are schematically illustrated in Figure 2.3. For the dune systems along Dutch coast, the swash and collision regime are most likely to occur during extreme storm conditions. Therefore, the processes involved in these regimes are studied in more detailed and are assumed to be dominant for the modelled sediment transport patterns.

During the collision regime, dune erosion is observed. According to experimental and mathematical modelling research carried out by van Rijn 2009, the dominant driving mechanisms for dune erosion are:

- The direct **wave impact forces** that act at a steep dune face. Generating high bed-shear stresses and erosion of the sediment.
- Unbounded **long waves** that are able to penetrate further up the cross-shore profile and reach the dune toe.
- **Increased turbulence** due to wave collision of incoming breaking waves and reflected broken waves
- Sliding of the duneface, known as **avalanching**, due to of the exceeding of critical bed slopes.

2.1.3. Modelling storm erosion

This section discusses the model that is currently used for strom erosion assessments along the Dutch coast. After a generic introduction of the model flow, input and output parameters, the limitations will be highlighted and put into perspective of this research scope.

Initiated by USACE-ERDC after devastating effects of hurricanes along coastal areas, the eXtreme Beach behaviour (XBeach) model has been developed (Roelvink et al., 2009). XBeach allows for the modelling of nearshore response of sandy coasts to extreme conditions. Since its development in 2009, the open-source model has become widely used for coasts assessments around the globe. The model is suitable for all storm regimes as defined by Sallenger (2000).

XBeach is a process-based model, meaning that it is the mathematical representation of several processes characterizing the storm induced erosion. Within XBeach, the wave propagating and the flow field are solved numerically. Allowing for examination of more complex coastal systems and the temporal behaviour of this system over time. A coupled sediment balance equation is used to compute concentrations, resulting bed level changes and bathymetry updates. The workflow of XBeach is illustrated below in Figure 2.4. All processes are incorporated separately, this makes it possible to study the effect and importance of single process on the total beach behaviour. All processes (waves, flow and sediment transport) are all computed online in a loop through time. The model loop starts with a bathymetry (bed elevation). Next, the hydrodynamics are computed, first the wave propagation in the domain is solved and subsequently the waves, tidal forcing and wind are used to generate currents and waterlevel variations by using the non linear shallow water equations. From the computed wave and flow field we can compute the sediment transport. Gradients in the sediment transport as a result of the deviation from the equilibrium concentrations give bottom change, which then can be used to update the bottom. This updated bottom morphology is provided to subsequent model loop as the new bathymetry.

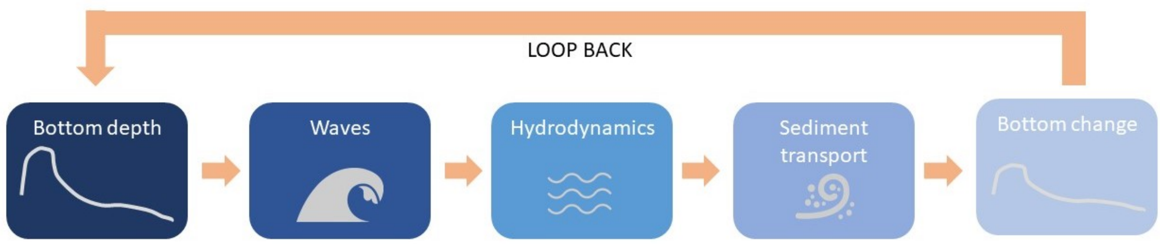


Figure 2.4: A schematic representation of the XBeach model set-up

For this research, the storm induced changes of the elevation profiles are of great interest. Therefore, the method used in XBeach to model these profile changes in a 1D environment is described more extensively. Gradients in the sediment transport ($\frac{\delta q_x}{\delta x}$) are obtained through the depth-averaged advection diffusion equation and using the formulation of Soulsby and van Rijn for the sediment transport formulations. Such that the sediment transport rates in a 1D situation are given by:

$$q(x,t) = \frac{\partial hCu^E}{\partial x} + \frac{\partial}{\partial x}\left(D_h h \frac{\partial C}{\partial x}\right) \tag{2.2}$$

The gradient in sediment transport rates obtained through this sediment transport formulations is used in the bed updating scheme. This scheme is based on continuity and uses the porosity (p) and morphological acceleration factor (f_{mor}) to update the bed level elevation (z_b):

$$\frac{\partial z_b}{\partial t} = \frac{-f_{mor}}{1-p}\left(\frac{\partial q_x}{\partial x}\right) = 0 \tag{2.3}$$

This formulation would be sufficient to describe bed elevation changes. However, within dune erosion processes, avalanching (the slumping of sandy material when it becomes wet) is an important process. XBeach solves this issue by locally looking at the slope of the profile and imposing a critical bed slope angle. Consistent with the equilibrium profiles according to Vellinga (1982), this critical slope value is set on 1 (dry) and 0.15 (wet).

$$\left|\frac{\partial z_b}{\partial x}\right| > m_{cr} \tag{2.4}$$

This process is triggered as soon as infra gravity waves (long waves) reach the dune front. Exceedance of the critical slope between two grid cells results in an exchange of sediment between the two cells to bring the slope back to the critical slope. In the subsequent step, this might cause a chain reaction to higher up the slope, such that sand is slumped from the full dunefront. This slumped sand is transported further seaward by currents and waves.

XBeach takes pre-storm elevation and hydrodynamic boundary conditions as input and returns elevation and hydrodynamic conditions during the storm. All these parameters are described extensively in the XBeach user manual (Roelvink et al., 2010). In this section the input and output are described qualitatively and the relevance for this research is considered. Bluntly, the input of XBeach consists of three main components: (1) Initial conditions, which describe the wave and flow conditions which the beach is exposed to, (2) boundary conditions that capture the bathymetry and sediment characteristics of the beach and (3) model parameterization coefficients. Within the 1D mode of XBeach, the input profile is defined as a set of elevation datapoints with respect to MSL (z_b) over a cross-shore distance (x). The output of the XBeach model is the evolution of the elevation profile and water surface over time. Through further analyses of this output, several morphological parameters such as the dune erosion volume (DEV) and beach width can be obtained. A typical pre- and post storm profile are shown in Figure 2.5. As you can see, the modelled profiles show resemblance to the profiles defined by Vellinga (1982) (Figure 2.2). The post-storm shape is at its steepest at the dune and flattens out towards MSL.

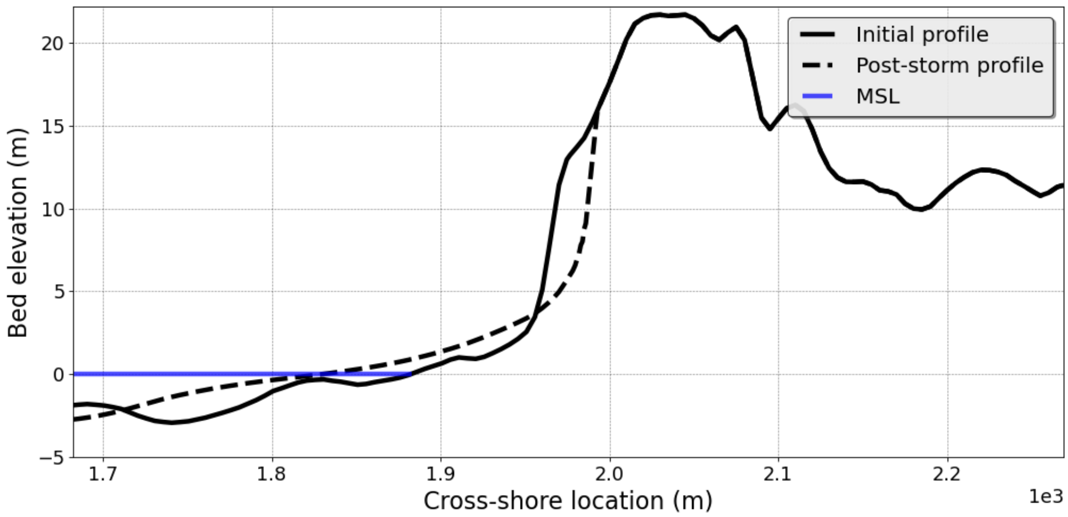


Figure 2.5: Example of a 1D XBeach profile. The initial (black) and post-storm (dotted) profile are illustrated. The blue line represents the mean sea level

XBeach enables the robust and physics based assessment of the impact extreme storm conditions on sandy coasts. Verification of XBeach with several field tests and real life observations confirmed the adequate performance of the model. Naturally, numerical modelling introduces several errors by means of truncation, roundoff and measurements (Zijlema, 2011). These introduced errors can be accounted for through calibration and be taken into account when assessing the impact of the storm. However, the introduced error is usually a trade-off between accuracy (reducing the error) and computational effort. Consequently, high fidelity numerical models come with large computational effort.

2.1.4. Surrogate models

This section briefly describes the general definition of a surrogate model. Subsequently, the recent developments of surrogate modelling in coastal/hydraulic engineering in considered. The developments on storm erosion and profile adjustments are discussed in more detail.

A surrogate model can briefly be described as "a model of a model" and is a framework that allows for the bypass of large computational efforts that are needed for the original model. A surrogate model describes the relationship between inputs (i.e., model's adjustable parameters) and outputs (i.e., the predictor of the model). Surrogate modeling techniques are especially interesting for engineering purposes when computationally expensive numerical models are used (e.g. Computation Fluid Dynamics (CFD) or Computational Structural Dynamics (CSD)). In order to train an accurate data-driven surrogate model, adequate input and output data is required. This data is obtained by running the simulation model with different sets of parameters selected in the feasible parameter space. When referring to a surrogate model in this research, the same definition applies for response surface model, meta-model and emulator. A generic framework of a surrogate model is illustrated in Figure 2.6.

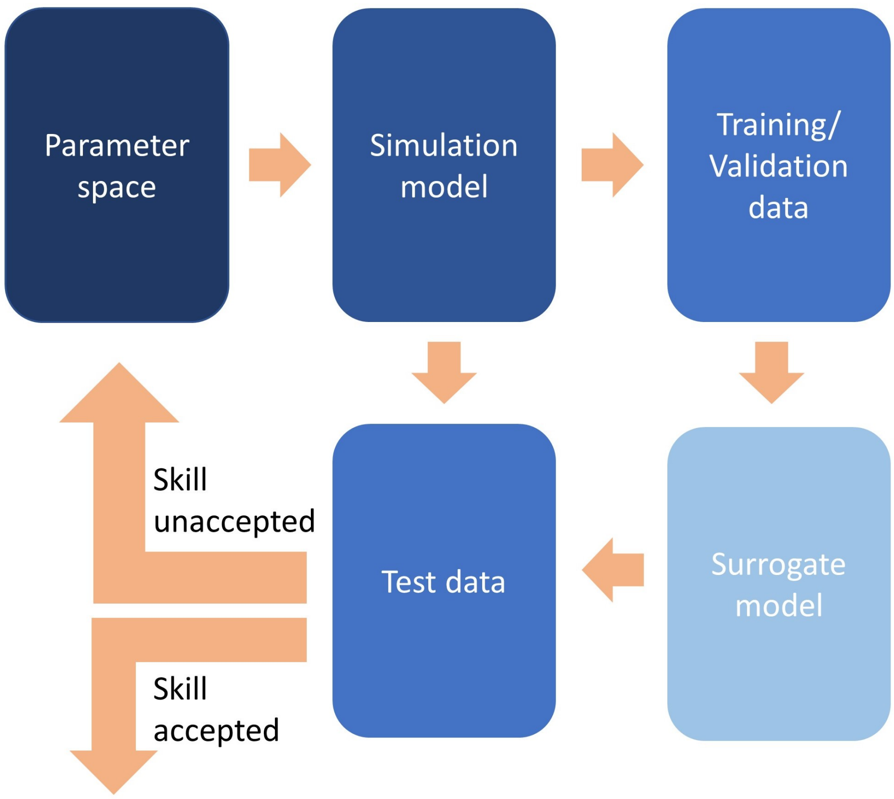


Figure 2.6: Schematic framework of a surrogate model. First the adjustable parameters are defined, secondly the simulation model is run with these parameters. The input and output of these runs is split into training and validation data. The training data is used for the surrogate model to train. During training, the model is validated with the validation data. After training, the skill of the model is assessed with test data. When the skill is not conforming to the pre-defined requirements, the parameter space and/or surrogate model learning technique is adjusted.

Several surrogate modelling techniques can be used to connect the input to the output of the simulation model. Well-known techniques are: Response Surface Model (RSM), Kriging, Radial Basis Functions (RBF), Multiple Linear Regression Model (MLRM). However, to proceed on findings by Athanasiou et al. and to be able to deal with non-linearity's, this research we will initially focus on artificial neural networks to be applied for surrogate modelling. Artificial Neural networks come in different forms, but are essentially a network of connected nodes that is based on the functionalities of the brain. Nodes are interconnected between layers and can be activated through an activation function.

The following subjects have already been studied by using ANN in the field of coastal and hydraulic engineering: beach seasonal changes (Hashemi et al., 2010), longshore sediment transport (Kabiri-Samani et al., 2011), sandbar characteristics (Kim et al., 2015) and (López et al., 2017), storm surge (Kim et al., 2015) and (Jia et al., 2016), sater resources (Razavi et al., 2012), overtopping (van Gent et al., 2007), (Verhaeghe et al., 2008) and (Chondros et al., 2021), dike breaching (Nourani et al., 2012), (Bomers, 2021) and (Serda et al., 2022).

Surrogate modelling for storm erosion processes

Santos et al. 2019 attempts to develop a meta-model for dune erosive processes. To explore possibilities to account for this geometrical erosion variable, this article presents several statistical models (MLRM, MARS) and machine learning tools (ANN, Random Forests) to develop a meta-model for Dauphin Island (USA). The morphodynamics is modelled using the triangular approach to describe the storm's temporal behaviour and the input variables consisting of 100 parameterized synthetic storms (η_{NTRF}, η_A, H_s, T_P, θ, and D (duration). Although the transects are considered independent, the 2-D of XBeach was used for this meta-model to account for the alongshore processes.

Crest height of both the primary and highest dune is used as a metric to assess the model's performance compared to XBeach. Some major findings in the research:

- ANN and MARS are the best-performing learning algorithms.
- Due to larger dynamics, the characteristics of the post-storm primary dune were harder to model than the highest dune.
- The best performance is achieved for dune toe elevation, dune crest elevation, area, and barrier island width.
- Changes in dune base and crest position are poorly predicted by the surrogate models.
- It was found that all meta-models overpredict erosion during events with small erosive potential, leading to worse results for both swash and collision regimes. Although changes in dune geometry are not that likely during these low-energy regimes, the model predicted a small morphological change.

It should be noted that the research carried out by Santos et al. (2019) was located in an area with other storm impact regimes. At Dauphin Island, the overwash and inundation regimes are more likely to occur than at the Dutch coast.

As described above, the model used by Santos et al. (2019) only used hydrodynamic input variables. Athanasiou et al. (2022) attempts to also include the pre-storm profile characteristics as input for the surrogate model to compute dune erosion volumes (DEV). The region considered in this article spans the total Dutch coast. To reduce computational effort of the 1-D XBeach model, 100 typical coastal profiles (TCPs) were created by using clustering techniques (Athanasiou et al., 2021). The profiles were parameterized through 10 morphological variables, such as dune volume, beach width and nearshore slope. 100 storms per profile were set-up to create a total of 10.000 XBeach runs that could be used as a training and validation dataset for the meta-model. For input of the model, the storm conditions were parameterized through H_s (significant wave height), T_p (peak wave period), SSL (sea surface elevation) and D (duration).

As Artificial Neural Networks (ANN) showed great potential to predict geometric erosion variables (Santos et al., 2019), this machine learning tool was also used for this meta-model. In total, the ANN consists of 14 input variables (10 morphological and 4 hydrodynamic), 2 hidden layers and one output layer. Two different ANNs were developed, one classification ANN to determine whether there would be any erosion at all (DEV = 0 or DEV > 0) and one regression ANN to quantify the DEV.

With an accuracy of 94%, the classification ANN to identify dune erosion performed properly. While adding more layers (deeper) did not result in a great increase of the performance, adding extra nodes in the layer (wider) did have a positive impact. Eventually, after several computations, a structure of 2 hidden layers with 32 neurons each turned out to be most efficient. For the regression ANN, a more complex structure with a higher number of neurons was required. Therefore, a regression ANN with 3 hidden layers and 32 neurons was selected. This structure obtained an average skill score of 0.82.

The most important insights obtained by Athanasiou et al. (2022) can be summarized as followed: (1) By applying the permutation importance approach, is was found that the pre-storm beach characteristics (volume, slope and width) is the most important input variable for predicting the DEV. This highlights the importance of the buffer services of the beach fronting the dune. Important hydrodynamic variables for dune erosion according to this technique are the SSL, T_P and D. This is in line with the understanding that the wave run-up above the dune toe and beach width are the most important drivers of observed dune erosion variability during a single storm (Beuzen et al., 2019). Unexpectedly, the significant wave height (H_s) has a small contribution to the predicted DEV. A possible explanation for the small significance of this variable is the possible correlation to other input variables (such as T_p). (2) The obtained skill level of the ANN can also be reached with half of the 10.000 scenarios that were included in this dataset. This could reduce the computational time that was used to develop the dataset.

Limitations of the meta-model described in this article touched upon the predictive capabilities and upscaling of the meta-model. In case of a meta-model, this capacity is greatly determined by the quality the synthetic data created with the numerical model. Within this numerical model, certain assumptions and simplifications have to be made to reduce the computational effort. These assumptions and simplifications include shore normal incident waves (no obliquity), simplification of storm input parameters and triangular storm evolution over time.

Concluding the article, the authors touch upon the applicability of the model and suggest that other erosion indicators could be studied as possible predictors/impact indicators in the meta-model. Using the full pre-storm and post-storm profile as input and output of the model could capture a variety of erosion indicators with a single model.

Post-storm beach profile shapes

Gharagozlou et al. (2022) is the first to develop a surrogate model to predict the actual post-storm profile shapes. Synthetic data created with 2D XBeach simulations spanning a total 18 km along the coastline of North Carolina and 1250 storm scenarios were used to collect beach profile changes for 105 different transect. The profiles were described using 9 empirical orthogonal functions (EOFs) obtained through Principal Component Analysis (PCA), catching 97 percent of the total variance of the considered profiles. The simulated sequential storms were described by 5 parameters (η, MSL, duration, H_s, T_p, θ). A simplified shape, defined by a power function ($z = A \cdot x^B$) and the scarp distance (D_s), was used to describe the post-storm profile. The input (9 EOFs and 6 storm parameters) and output (A, B and D_s) were used the train ANN.

The surrogate model was trained with 700 storm events and, compared to XBeach, could predict the dune erosion volumes quite accurately (average error of about 11%). More importantly, the developed emulator can account for pre- and post-storm beach profile changes in the subaerial beach and dune profile. The median absolute percent errors across the 105 considered profiles was 17.6%. Errors are largest because of the assumptions that are made in the process-based model. Other contributions to the observed errors are (1) the triangular distribution of the storm, (2) EOF parametrization and (3) the quality of the training.

Gharagozlou et al. (2022) is the first to attempt usage of the continuous elevation data of pre-storm profile and post-storm. By parametrization of the profile shape through pre- and post-processing efforts (EOF and power law function), it is possible to develop a surrogate for beach and profile shapes. It should be noted that the considered profiles are very different from the Dutch Coast and storm regimes are more extreme.

Setting up a surrogate model/meta-model is a balancing act between solving the involved complexities and preventing error development. Generally, an ANN is trained more efficiently when the amount of output parameters is reduced. However, an increase in the number of parameters would increase the applicability of the model. Gharagozlou et al. (2022) attempted to explore this aspect of surrogate modelling by varying in predicted output variables.

This showed that solely predicting the loss in dune volume (1 variable) comes with a greater predictive skill of the model than for three geometric variables. This one-dimensional framework for the output is similar to the meta-model set up by Athanasiou et al. (2022). As stated in the article: "*Although a direct prediction of DEV is useful, it does not preserve the exact shape of the beach, preventing any prediction of where that volume is lost along the profile and the associated geomorphic insights*". Showing the need for a multi-dimensional output framework.

Two other models were proposed. An emulator with (1) 9 EOF output variables for profile description. This model performed relatively poor compared to the one and three-variable output models. This can possibly be explained by the characteristics of the training dataset, containing little information on the sharp scarp features that are present after some storms. Next to that, (2) 220 parameters for the elevation at all cross-shore locations were used as output variables. While overall model skill was comparable to the three-variable model, a detailed analysis of several results shows that the skill for single transects is far less for a 220-variable model. This is shown by using the absolute error as an indicator, which resulted in a lower skill score (0.88) for the 220-variable model. Finally, this article stresses the importance of direct simulation of elevation and clustering techniques to create a surrogate model for specific event characteristics.

2.2. Neural Networks

2.2.1. General

The concept of neural networks originates from the connection and interaction between neurons in the brain. This connection was introduced in mathematics by McCulloch and Pitts (1943) in "A Logical Calculus Of The Ideas Immanent in Nervous Activity". In this paper, the nervous activity is treated by means of proportional logic, which allows for the conversion of continuous input to discrete output. This concept and its resemblance to a neural node are illustrated in Figure 2.7. The complex decision process in the brain (continuous) is explained by using a linear threshold gate. This McCulloch-Pitts neuron takes inputs and calculates the weighted sum, resulting in either an 0 or 1 as output. Donald Hebb (1949) build upon this idea in his book "The Organization of Behaviour", by proposing that neural nodes are strengthened over successive use. These two concepts, (1) Threshold Logic and (2) Hebbian Learning are considered the precursors to Neural Networks.

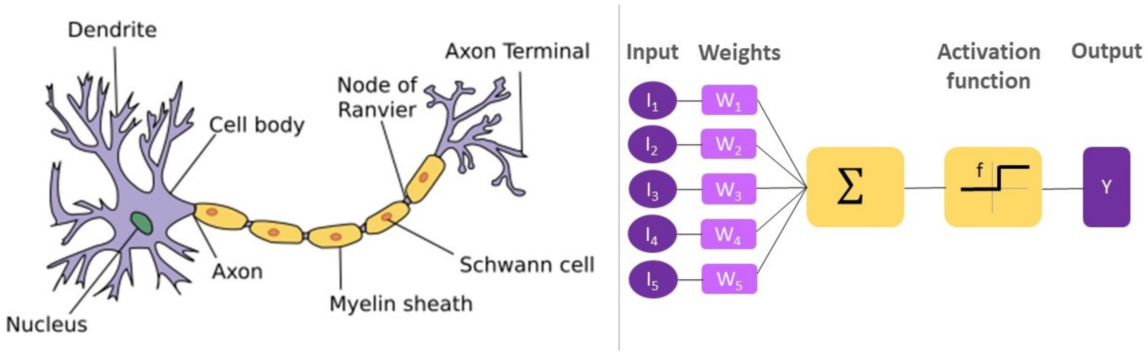


Figure 2.7: Computational illustration of neural node as described by McCulloch and Pitts (1943)

Figure 2.7 displays a single neuron with several weighted inputs and a single output ($\hat{y}$). This individual neuron is still the main component of all neural networks. The sum of all weighted inputs is calculated through:

$$\left(\sum_{i=1}^{n} W_{ij} I_i\right) + b_j$$

where n is number of layers in the previous layer, W_{ij} is the weight of the connection between the i_{th} neuron of the previous layer and the j_{th} in the current layer. I_i represents the input of the previous layer and b_j the bias of the current layer. The trainable parameters in this weighted sum are the weight (W) and bias (b). This weighted sum is inputted in a activation function (f). Mathematically this can be described such that:

$$\hat{y} = f\left(\left(\sum_{i=1}^{n} W_{ij} I_i\right) + b_j\right)$$

This formulation of $\hat{y}$ describes a forward pass for a single neuron. The activation function (f) can be chosen based on the task at hand. Common activation function used in the field of machine learning (tanh, ReLU, Sigmoid and Linear) are depicted below.

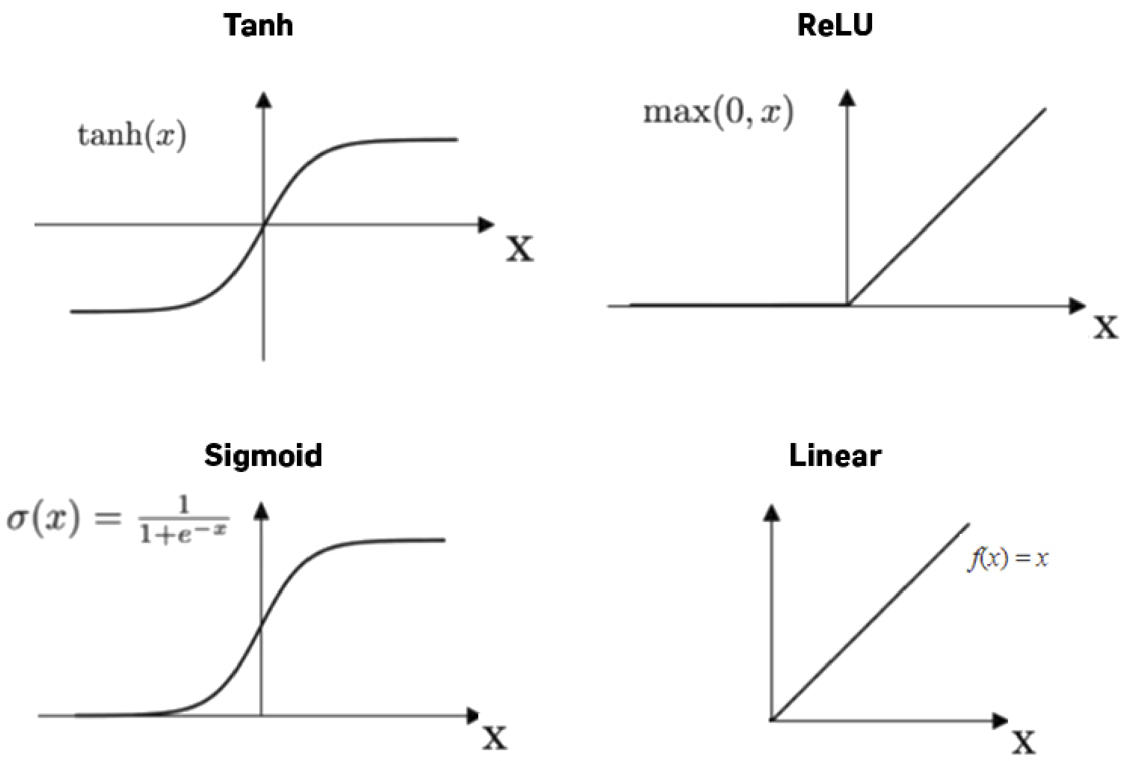


Figure 2.8: Common activation functions used in forward pass within neural network retrieved from machine-learning.paperspace.com, 2020

Frank Rosenblatt (1958), for his study on fly behaviour of flies, worked on the first network (Mark I Perceptron) for which the weights could be learned by successively passing inputs into the system and minimizing the difference between desired and actual output. This knowledge was put into practice by Bernard Widrow and Marcian E. Hoff (1962) for noise elimination in phone lines. The proposed ADALINE neuron, consists of a set of variable weights, a threshold and adaptation machinery for automatically adjusting its weights. It has analytically and empirically been demonstrated that a single ADALINE can be trained to recognize geometric patterns, perform logical functions, and store digital information to eliminate noise in phone lines. This experiment showed proofs of convergence of the learning processes and derivations of learning rates have been made.

However, as addressed by Minsky and Papert (1969), a single neuron cannot learn a simple but non-linear exclusive or circuit (XOR). This limitation and others criticism on neural networks was documented in "Perceptrons" and was the beginning of a period referred to as "the AI winter". During this period, little to no new research was carried out in the field of Neural Networks.

In 1982 (!) the neural network community started building again. In his PhD. thesis, Paul Werbos (1994), showed the potential of backward propagation for artificial neural networks. **Backward propagation** along with **gradient descent** forms the backbone and powerhouse of neural networks. While Gradient Descent constantly updates and moves the weights and bias towards the minimum of the cost function, backward propagation evaluates the gradient of the cost w.r.t. weights and biases in the previous output layer. The magnitude and direction of this derivative are used by gradient descent to evaluate the size and direction of the corrections to weights and bias parameters. With respect to the original concept of neural networks (forward propagation), which allows for the flow of information from input to output, backward propagation allows for flow in the opposite direction. Through forward propagation, a prediction of variable ($\hat{y}$) is made. The loss compared to the actual value of this variable (y) is captured in a cost function (E). This is defined as:

$$E(\Theta) = \frac{1}{m}\sum L(y, \hat{y})$$

m is defined as the number of training samples and L is the loss sustained by between y and $\hat{y}$. The objective is to minimize the cost E. This is achieved by differentiating E with respect to the parameters and adjusting the parameters in the opposite direction of the computed gradient. This optimization effort is described as gradient descent. The loss function L can be picked based on the task at hand. For regression tasks, the Mean Squared Error (MSE) is a commonly used loss function. For a single training sample ($m = 1$), the cost function will resemble this:

$$E(\Theta) = (y - \hat{y})^2$$

The weight and biases within the network are updated with a certain learning rate (ϵ) through the scheme:

$$w_{t+1} = w_t - \epsilon \frac{\partial E(\Theta)}{\partial w}$$

$$b_{t+1} = b_t - \epsilon \frac{\partial E(\Theta)}{\partial b}$$

These update schemes require the computation of the derivatives of the current layer with respect to the previous layer. To do so, the concept of chain rule differentiation is used. For a weight (j) in a neural network this would appear in this form:

$$\frac{\partial E}{\partial w_{ij}^2} = \frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{z_i^2} \frac{\partial z_i^2}{\partial w_{ij}^2}$$

The principle of gradient descent is illustrated for a 2D situation in Figure 2.9. To determine the quantity and direction of the gradient descent for each dot, backward propagation is continuously carried out in every single black dot. After several iterations, the solution will converge towards a local minimum. There are several types of gradient descent algorithms. These mainly differ in the data handling procedure before the gradient is obtained.

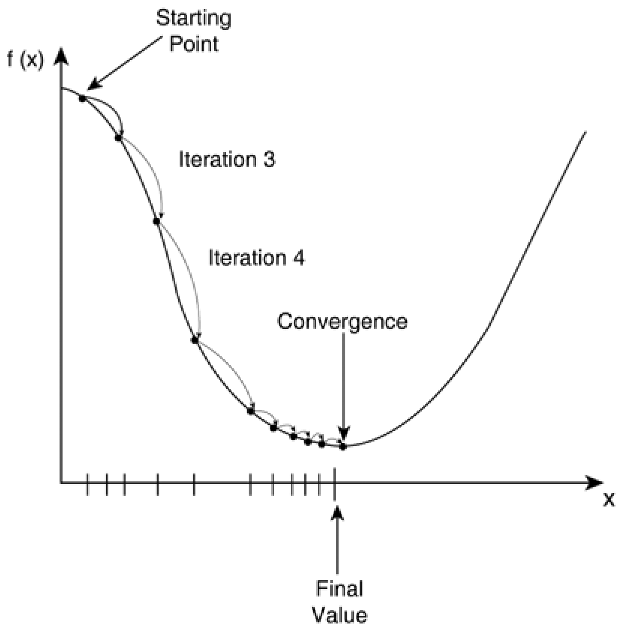


Figure 2.9: A visualization of the gradient descent towards a minimum. The magnitude and direction of the derivative at each point are calculated with backward propagation retrieved from machine-learning.paperspace.com, 2020

In the state of art neural networks, gradient descent has become a basic type of optimization procedure. More advanced schemes such as the Momentum, Nesterov Accelerated Gradient, Adagrad and Adam have been developed over the last years. These optimizers use addition characteristics, such as second-order derivative and momentum of the loss function to find the appropriate minimum.

Zooming out, there are two main types of machine learning; (1) supervised and (2) unsupervised learning. These types of learning can be distinguished by the type of data that is provided to the neural network. For supervised learning, the provided data contains both the input and the required output (labels). The goal of supervised learning is to learn the network that, given a sample of data and desired outputs, best approximates the relationship between input and labels provided by the data. For unsupervised learning, there is no prior knowledge about the outcome of the learning process. The data consists solely of input variables, without labels. This data can be clustered or explained by using machine learning techniques.

2.2.2. Convolutional Neural Networks

Essentially, neural networks assume linear independence of input features and low resolution of the input space. For parameterized input, these assumptions are generally met. However, for raw input data such as images, audio, timeseries and text these requirements are not accommodated. Convolutional neural networks (CNN) are a type of neural network which are specifically designed to deal with these kind of data. While the input features can not be assumed independent in these can of data, the features extracted by the CNN can. CNN can extract features from the data and decrease the dimensionality.

Introduced as a promising network structure (LeNet-5) for standard handwritten digit recognition tasks by Lecun et al. (1998), CNNs showed great skill in dealing with pattern recognition for several other applications. The convolutional layers applied in these networks, find their origins in the field of computer vision. In computer vision, a set of weights (filter) is systematically multiplied with the input by taking the dot product. While shifting the filter over the two-dimensional input, the output of this dot product (scalar product) is stored in the next layer (feature map). The filter is designed to detect a specific type of feature in the input, such that the application of that filter systematically across the entire input image allows the filter an opportunity to discover that feature anywhere in the image. This concept to discover features in a 2D input layer is schematized in Figure 2.10 and is also applied in CNNs.

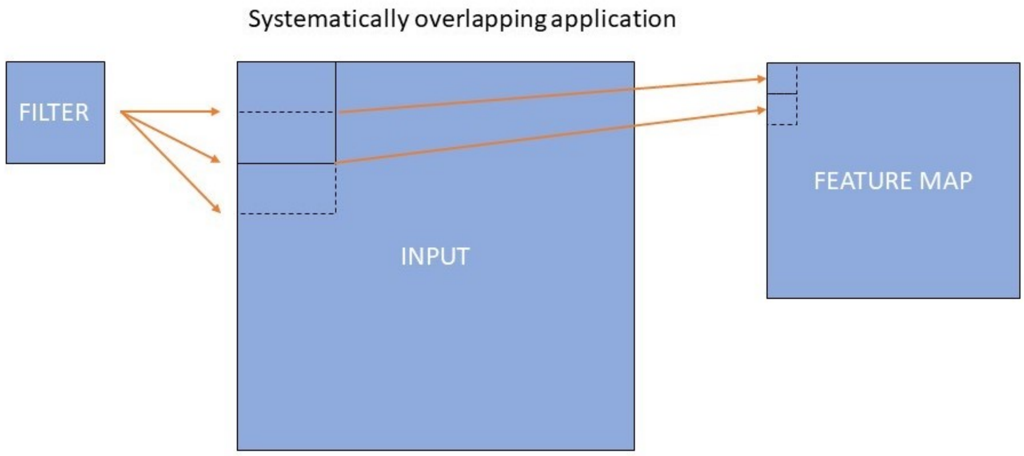


Figure 2.10: Schematized principle of computer vision and convolution

The filter is smaller than the input data and the type of multiplication applied between a filter-sized patch of the input and the filter is a dot product. A dot product is an element-wise multiplication between the filter-sized patch of the input and filter, which is then summed, always resulting in a single value. Because it results in a single value, the operation is often referred to as the scalar product. A typical dot product operation is depicted in Figure 2.11.

Using a smaller filter layer then the input matrix, allows for the shifting of the filter layer over the input matrix. By systematically carrying out his act and calculating the dot product, patterns and characteristics within the input layer can be signalized and/or adjusted. This capability is known as the translation invariance (the general interest in whether the feature is present rather than where it was present). This makes the systematic application of the same filter across an image a powerful concept. Common operations in computer vision are the sharpening, blurring and detection of edges of images.

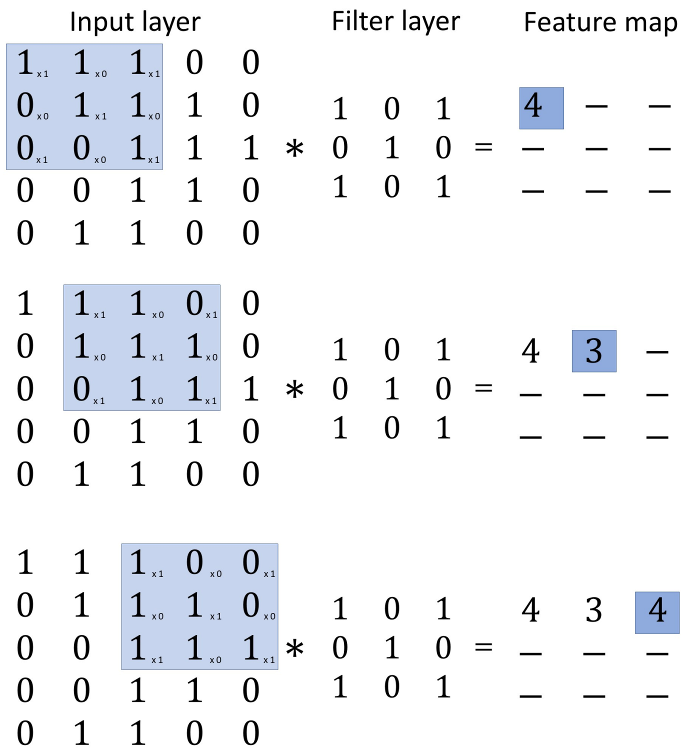


Figure 2.11: Typical dot product operation that is the base of every convolution. The kernel is systematically shifted over the input and multiplied with the values covered by the filter.

This operation is frequently used in convolutional neural networks to deal with 2D input and detect patterns within this input. Patterns regularly present in the input layer can be detected with filters and result in higher translation invariance after performing a computer vision operation. To make the process of pattern recognition more striking and efficient, pooling layers are applied to synthesize the "most important" patterns that are found in the image. In practice, this results in the downscaling the dimensions of the feature maps, which accelerates the performance of the neural network. A frequently used type of pooling is the max pooling technique, a technique that computes the largest value in a patch of the considered matrix. This can be illustrated in matrix form (Figure 2.12). The max pooling operation does not need any assigned weights, but solely applies an aggregation function. Although information about certain patches of the matrix might get lost, the pooling layer can significantly reduce the computational load and limits the risk of overfitting.

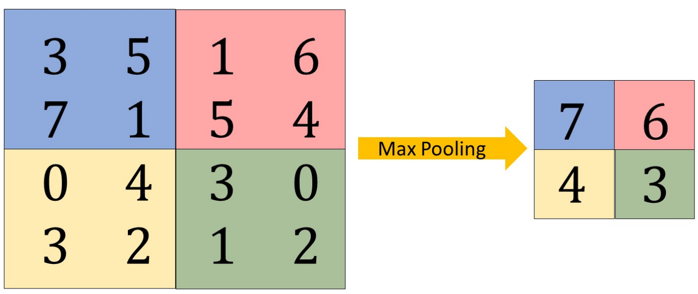


Figure 2.12: Maximum pooling operation to reduce the size of the input. Each colour represents a different patch.

Other tools to either avoid or promote downsampling of the input data are padding and striding. With padding, the boundaries of the input data are supplied with artificial data (zeros), to avoid loss of information. Adapting the stride of a convolutional operation, implies that the shifting of the filter over the input layer is done with less detail. With a larger stride, larger parts of the input data are skipped between two successive dot products.

As discussed above, computer vision is an interesting tool to detect patterns in an input matrix. However, the design of an appropriate filter layer for signalizing trends can be very time-consuming. Besides that, these patterns will probably appear in different forms for different inputs. This makes a single filter layer unsuitable for the detection of patterns in different input layers. This is where the potential of neural networks come in.

Essentially, the filter layer applied to the input matrix is a matrix of weights. These weights indicate whether a certain pattern (described by the filter layer) is present in the input layer. However, before training a CNN, the filters do "not know" which feature they should detect. The weights in the filters are chosen randomly and trained through backpropagation. Through training, the set of weights is assigned to a filter. This set of weights contains visual information of the input matrix.

Multiple filters can be applied to the same input matrix. All these filters can be trained by neural network to a set of weights with their own feature maps and visual information. The amount of applied filters layers determines the channel size of a convolutional layer. This channel size is considered as the "width" of a neural network. The depth of a CNN is determined by the number of convolutional layers present in the neural network.

Originally, CNN was introduced for classification tasks. The convolutional layers were followed up by a fully connected part which was able to make the classification of the image. This structure was also proposed by Lecun et al. (1998) for digit recognition:

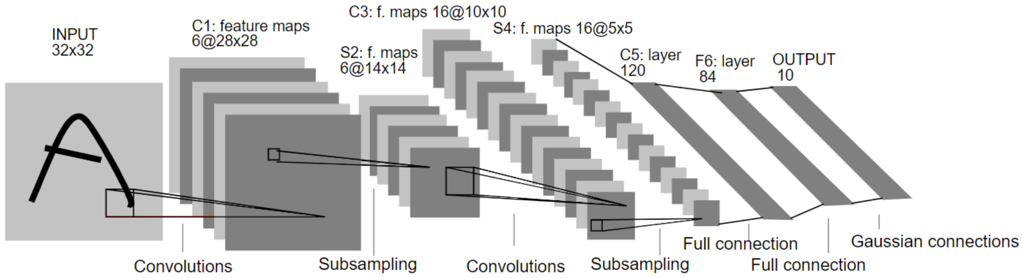


Figure 2.13: Original LeNet-5 structure as proposed by Lecun et al. (1998)

Throughout the depth of this network, the input layer is encoded and considered in more detail while losing the spatial information. Hence, more is learned about the features (what) and information about the locations (where) of these features is lost. This characteristic is easiest explained by means of an illustration (Figure 2.14).

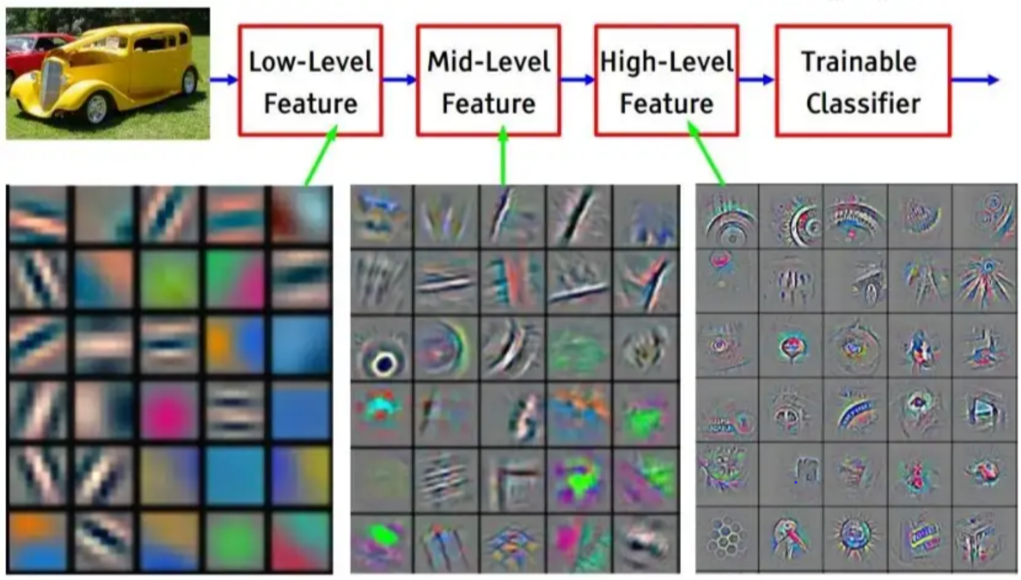


Figure 2.14: Hierarchical structure simple CNN. Feature maps in deeper parts of NN detect more detailed features of the input. Retrieved from Sun et al., 2021

As shown in the illustration, kernels in the deeper convolutional layers focus on detecting small-scale features while kernels in the shallow convolutional layers focus on detecting large-scale features. As you travel towards the deeper layers of the network, each pixel has more and more information about the input image encoded in it. In this way, the structure of CNNs is described as hierarchical. The CNN uses this hierarchical structure to learn information that is captured in the input layer. Weights in the filters in shallow parts of the neural network are trained by the composition of deeper-level features.

Throughout the depth of a CNN, the receptive field of neurons increases. The receptive field is the region of the input that a particular neuron or layer in a neural network "sees" or is influenced by. In CNNs, the receptive field is determined by the size of the filters or kernels used in the convolutional layers and the stride. The receptive field size is important because it determines the spatial context that a neuron can capture and process. Neurons with small receptive fields focus on local details and low-level features, while neurons with larger receptive fields capture more global patterns and high-level features.

When it comes down to activation function, CNNs require some specification. As introduced earlier, several activation function can be applied within neural network (Figure 2.8). However, the sigmoidal functions such as the tanh and sigmoid function both reach into the negative domain $\mathbb{R}^-$. For unseen data, this would result in negative contribution to the output of the network if feature turns out to be irrelevant. However, since we assume independence between obtained features, a feature being irrelevant does not imply that other features also less relevant. Therefor, it is preferable for CNN's to use non-negative activation functions. Next to that, ReLU requires less computational load since it only involves a comparison between its input and the value 0 and it also has a derivative of either 0 or 1. This makes the gradient descent functions involved in the backpropagation easier to compute.

When training a (convolutional) neural network, the parameters whose values control the learning process are defined as the hyperparameters. The prefix hyper- indicates the more top-level character of these parameters. This includes parameters such as the earlier addressed learning rate in optimization algorithms, choice of optimization algorithm, choice loss function, number layers, number of iterations (epochs), kernel or filter size, pooling size and batch size (Nyuytiymbiy, 2020).

In the training procedure of a neural network, a dataset can splitted into a training and validation dataset. Backward propagation, optimization and actual training of the model only takes place for the training data. While, for each iteration, the established model parameters are also validated without being adjusted. This validation helps to tune the model hyperparameters.

When testing a model, the model is confronted with a dataset that is completely unknown. This is where a judgement of the performance is carried out.

2.2.3. Image segmentation and CNN

The problem at hand described in chapter 1 requires the prediction of the post-storm profile shape, indicating that the requested outcome of the model has spatial features that describe the morphological changes of the dune. As the input of XBeach is also a 1D profile (x,z), both the input and output of surrogate model have the same reference frame. Looking into other applications of convolutional neural networks, the operation of image segmentation shows similar requirements. Briefly, image segmentation is a function that takes an image as input and produces a masked image as output. Through encoding of the image important features are extracted. These features are decoded into the original image and presented in their original form. It can be compared to a simple coloring page that your parents gave you as a child. Pixels with the same label will get the same colour.

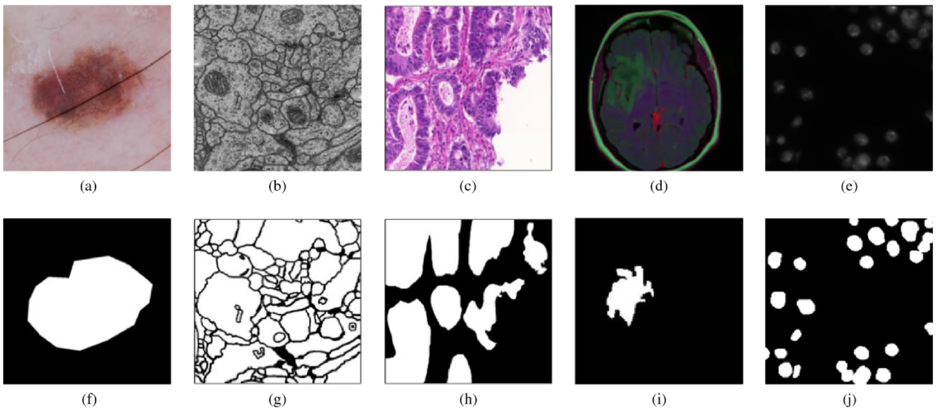


Figure 2.15: Segmentation applied on several types of medical images. From left to right: (a) dermoscopy, (b) electron microscope, (c) histopathology, (d) MRI and (e) nuclei microscopy and corresponding segmented images. retrieved from GUDHE 2021

A well-known application of image segmentation is the identification of deformations in medical pictures. Biomedical image segmentation takes a plain medical picture as input and produces an output image with a highlighted deformation or hazardous tissue. This can help medics to diagnose patients and make trustworthy judgements on their health. Typical biomedical image segmentation cases are presented in Figure 2.15. The profile shape prediction task is from a similar nature. Based on a certain input (pre-storm profile), an output in an identical reference frame (post-storm profile) should be predicted.

The encoding branch of the UNet architecture pursues the traditional structure of a convolutional neural network. It consists of several blocks connected with max pooling layers (2x2 and stride = 2) to downsample the input. Each individual block consists of two subsequent convolutional layers with a 3x3 filter and rectified linear unit (ReLU). At each downsampling, the number of feature channels is doubled. The decoding branch of the network consists of the same type of blocks as the encoding branch. However, to allow for upsampling, the blocks are connected with transverse convolutional layers which decrease the number of feature channels by half it's size. After the upsampling, the resulting feature map is concatenated with the corresponding feature map from the encoding branch (skip connection). At the final layer, a convolution layer with filter size 1 is used to map each feature vector that originates from the network to the desired number of classes. The number of blocks and channels considered in the network can differ, but the original network structure as proposed by Ronneberger et al. (2015) has 23 convolutional layers. The symmetrical characteristics of the network's architecture shows similarities to the letter U, thus given the name: UNet. The original UNet-structure proposed by Ronneberger et al. (2015) can be found in Figure 2.16.

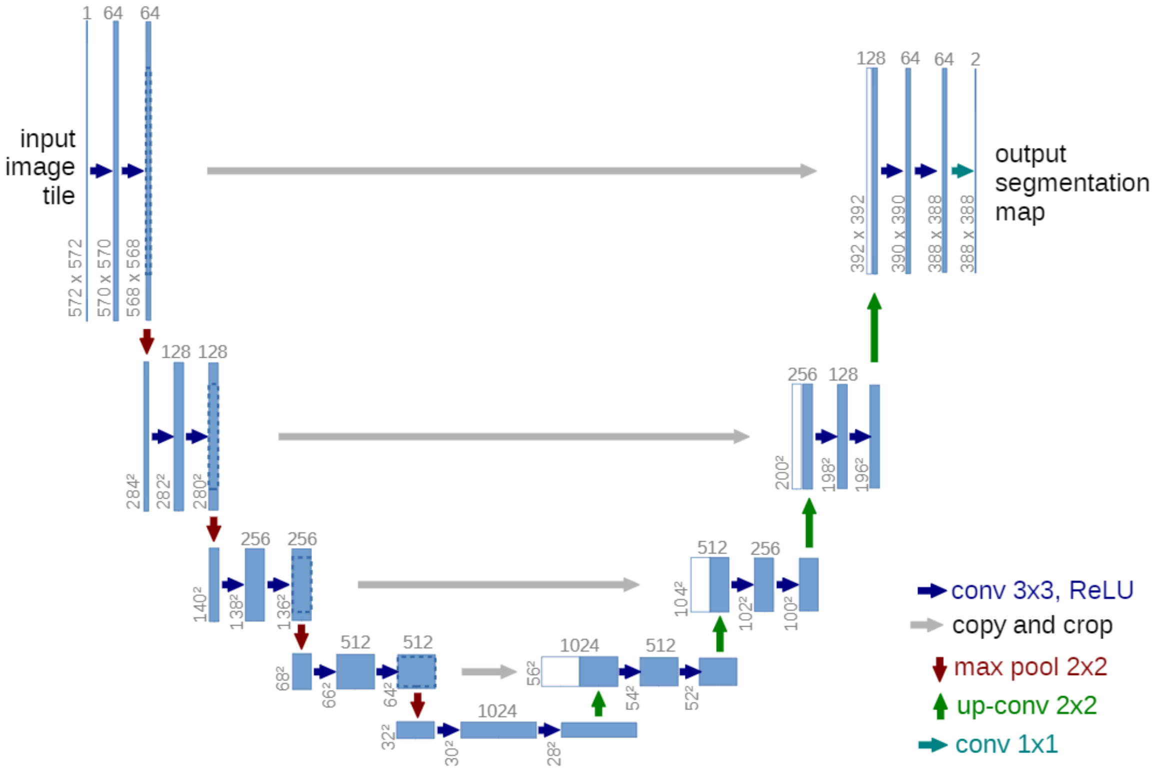


Figure 2.16: Proposed UNet-structure by Ronneberger et al. (2015). The U-shape is clearly visible. Several convolutional blocks are connected with max pooling (red arrow) and transverse convolutional operation (green arrow). Skip connections are depicted as grey arrows connecting the encoding and decoding branches.

Since the development of the U-Net architecture, it has been applied and reviewed for many purposes. Next to extensive use for biomedical segmentation, U-Net has also proven its value for analysing satellite imagery with respect to glacier retreat (Baumhoer et al., 2019), road extraction (Yang et al., 2019) and deforestation (Maretto et al., 2021). Next to that, it has been applied to predicting landslides (Prakash et al., 2020), ultrasound segmentation (Amiri et al., 2020), thermal crack detection (A et al., 2021) and fluid dynamic simulations (Eichinger et al., 2022).

2.2.4. Model performance

The performance of a model is typically judged based on comparison between the model predictions and the known/target responses. Performance metrics for regression tasks are typically based on the error between the modelled and known response values. Popular metrics include the mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE) and coefficient of determination (r^2) (Beuzen & Splinter, 2020). These metrics were also applied by Gharagozlou et al. (2022) to compare surrogate model predictions of erosion profile shapes.

As discussed by Gharagozlou et al. (2022), applying these metrics for geometric assessment of prediction is a delicate process. The RMSE is always positive and prevents that overpredictions and underpredictions cancel each other out in this cumulative metric. However, when interested in positive or negative contributions, the RMSE might not be the most suitable option. The mean error (ME) does include the sign of the error.

Besides that, the RMSE and MSE are a cumulative error metric. This is required to carry out bulk performance evaluation of models. However, for spatial data, it might be required to look into the spatial distribution of the error. In this case, the squared error (SE) can be used at individual gridpoints.

3
Methods

3.1. XBeach Dataset

This section describes the experimental setup of the XBeach runs that are carried out to create a synthetic dataset. This data set will subsequently be used for surrogate modelling purposes. First, the data pre-processing efforts to establish a simplified situation are highlighted. This includes modification and simplification of existing JarKus profiles. Second, the model set-up of XBeach will be discussed. This includes the characterization of the underlying assumptions and resulting boundary conditions.

3.1.1. Data Pre-Processing

JarKus

The cross-shore profiles used in this research are taken from the JarKus (Jaarlijkse kustmetingen) dataset. This dataset was set up to monitor the morphological changes along the Dutch Coast. Starting in the 1960s, yearly elevation measurements have been carried out. These measurements are taken from the primary dune up to 1000 m seaward at a perpendicular angle to the coastline with a separation of about 250 m. Measurements take place after the stormy winter period around April and are archived in the database (Rijkswaterstaat, 2021). Generally, all transects consist of a terrestrial (dry) and a submerged (wet) area, which are measured by laser altimetry from aircraft and soundings from vessels respectively. Timing these measurements such that the former is carried out during low tide and the latter during high tide enables the merging of the two datatypes. The cross-shore data resolution varies between 5 m on the beach and 20 m further offshore and the elevation is taken with respect to the NAP (Normaal Amsterdams Peil). In total, the Jarkus dataset consists of 2178 transects located in 15 different bounded regions (kustvakken). As this research is focused on sandy dune beaches, the dataset can be reduced to 1430 transect by taking out the locations that include hard structures or show extremely dynamic behaviour (Athanasiou et al., 2021).

Modifications

For the exploration phase, a dataset with known and fundamental dune erosion patterns should be set up. This makes the interpretation of the surrogate modelling efforts more straightforward. Therefore, initially, one cross-shore profile is selected and subsequently modified to simulate different morphological responses. These profiles are selected based on several requirements to standardize the modification operation. (1) The profile should be a typical dune-beach system, (2) the profile should have a smooth nearshore surface without any major disturbances besides the regularly appearing sandbars, (3) the primary dune should be the highest dune in the system. Based on visual inspection, using the Jarkus Analysis Toolbox (van IJzendoorn, 2021) and "De Kustlijnenkaart 2022" (Rijkswaterstaat, 2021), profile with transect ID 7200 (kustvak 8) was chosen.

Unique profiles will be created by modifying the original profile presented in Figure D.1. To simulate a different morphological response, the morphological features of the profile are altered. This is done by changing the width or height of specific profile sections. As introduced in subsection 2.1.2, the changes in bathymetry should result in different morphological responses. These imposed changes in bathymetry are illustrated in

Figure 3.3 and quantified in Table 3.1.

Applying this set of single modifications to a single transect would result in a dataset of around 60 unique profiles. However, based on other examples of surrogate models, this won't be enough to train a neural network and prevent overfitting (Santos et al., 2019). More profiles and variance are generated by imposing a set of second modifications (right column in Table 3.1) to several profiles. Profiles undergoing an additional second modification are printed boldly in the first column Table 3.1). These second modification factors are collected from a smaller range and are identical for all profile parameters. For example, a profile with a 1.25 beach width extension undergoes a second modification on all other profile sections, creating another 24 profiles. Note that the first and second modifications are never carried out on the same profile section.

Finally, to secure sediment availability, these profiles are extended landward from the crest of the dune. The workflow of a double profile modification is schematized in Figure 3.1.

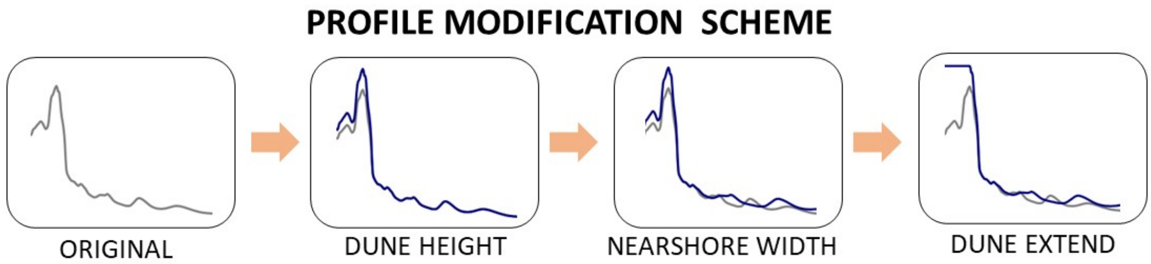


Figure 3.1: Scheme of modification of a single profile with two modifications. First, the height of the dune is increased with a certain factor. Subsequently, a second modification in the nearshore width is applied. Finally, the dune top is extended.

Profile parameter	Modification factors	Second modification factor
Beach width	0.1, 0.2, 0.5, **0.75, 0.9**, 1, **1.1, 1.25**, 1.5, 2, 5	0.5, 0.75, 0.9, 1.1, 1.25, 1.5
Inter-tidal area width	0.5, 0.6, **0.75, 0.9**, 1, **1.1, 1.25**, 1.4, 1.5, 2	0.5, 0.75, 0.9, 1.1, 1.25, 1.5
Nearshore width	0.5, 0.6, **0.75, 0.9**, 1, **1.1, 1.25**, 1.4, 1.5, 2	0.5, 0.75, 0.9, 1.1, 1.25, 1.5
Dune width	0.5, **0.75, 0.9**, 1, **1.1, 1.25**, 1.5, 2, 5	0.5, 0.75, 0.9, 1.1, 1.25, 1.5
Dune height	0.5, **0.75, 0.9**, 1, **1.1, 1.25**, 1.5	0.5, 0.75, 0.9, 1.1, 1.25, 1.5
Dunecrest height	0.5, **0.75, 0.9**, 1, **1.1, 1.25**, 1.5	

Table 3.1: Profile parameters and imposed modification factors. The bold values in the modification factor columns represent the profiles which undergo a second modification.

The original length and height of these profile sections are defined by means of the definitions presented in subsection 2.1.1. The methods used to compute these profile characteristics are listed below.

- *Depth of Closure* (DoC) - For the Dutch coast, the DoC is around a depth of 9 m and can be obtained by using different methods. The Hallermeier formulation (Hallermeier, 1981) is widely used to determine the DoC and based on the wave height. For this research, the DoC is extracted from a dataset set-up for global nearshore slope distributions (Athanasiou et al., 2019)

- *Mean Low Water (MLW) and Mean High Water (MHW)* - These elevations are measured for the Dutch coast and labelled to profiles in the JarKus dataset.

- *Dune toe* - There are several methods available to methods to determine the dune toe of a cross-shore profile. The method used in this research uses a fixed elevation of NAP + 3 m for the dune toe. Using a fixed value for the dune toe elevation allows for the observation of trends in the cross-shore location of the dune toe. This point marks the end of the beach and the beginning of the dune section.

- *Dune crest* - This point is determined by using the find.peaks function of the scipy package. From the range of points that is returned by this function, the first value is chosen. This point marks the end of the dune section.

For all profile parameters, the zero-crossing method is applied to find the intersection of the profile with the highlighted elevations. This results in the characterization of a profile as illustrated in Figure 2.1.

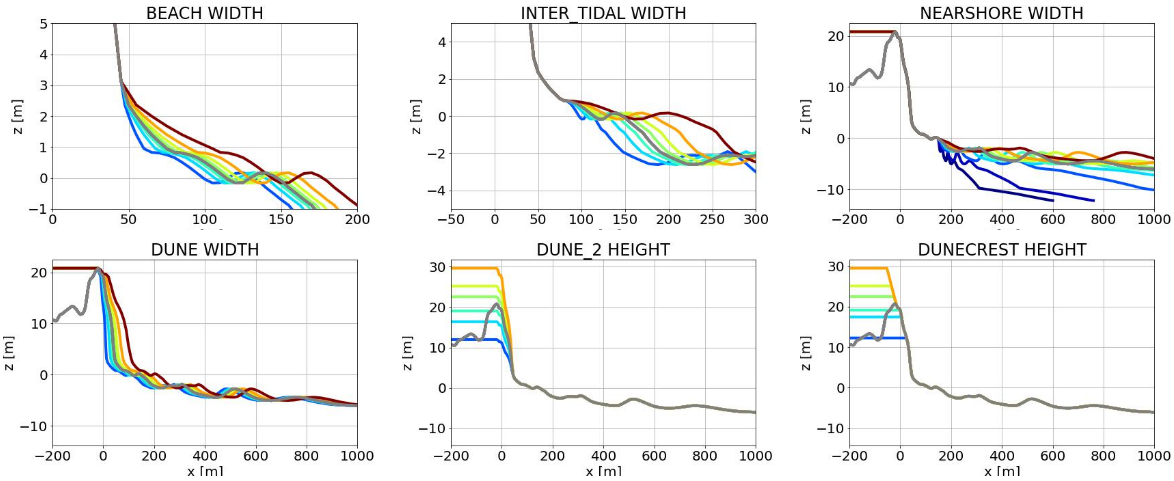


Figure 3.3: Several modifications of Jarkus profile 8007200. A blue colour indicates a reduction of a certain feature, while a red colour indicates an increase of that same feature. For some modifications, the scale on the x-axis is changed to make the alternations visible

Dataset of input profiles

This section presents the different datasets used in this research. As explained in subsection 2.2.2, datasets considered for neural networks can serve different purposes (training, validation and testing). For all purposes and research phases, a different dataset is generated.

Exploration

To make sure that modifications and imposed scaling factors are realistic to the Dutch coastal system, the profile parameters found in the created dataset are compared to the parameters observed along the Holland coast (Rijkswaterstaat, 2021). The boxplot used for this comparison can be found in Appendix D. Outliers exceeding the ranges found on the Holland coast are excluded from the dataset. Next to that, profiles exceeding physical limits, such as the maximum dry slope of sand, are also eliminated. Therefore, not all beach parameters have the same range and density of modification factors. Hence, a dataset of 502 unique profiles is set up (Figure 3.4). All these profiles originate from a single profile (transect 8007200) and are used for the exploration phase of this research.

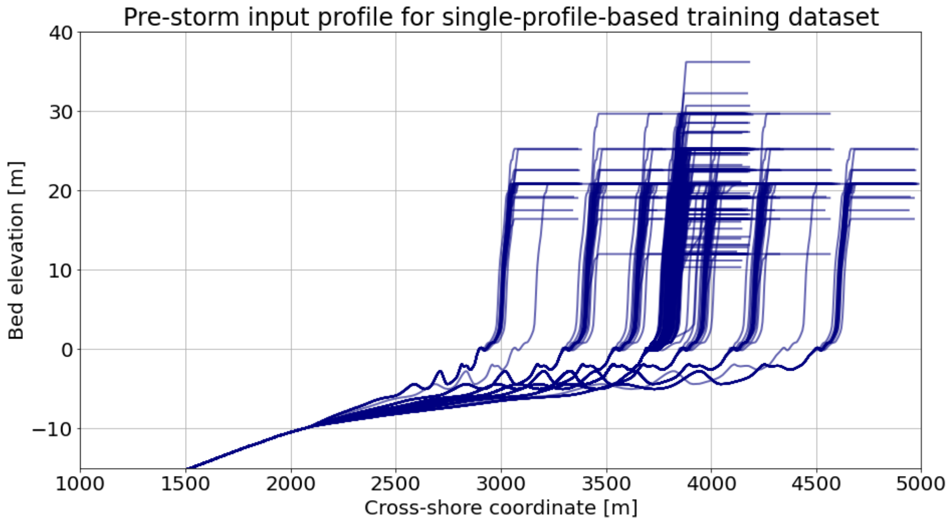


Figure 3.4: All input profiles in the single-profile-based dataset as inputted to XBeach

It should be stressed that the dataset highlighted above is used for training and validation during the exploration phase of this research. The test dataset to judge the performance of several surrogate modelling alternatives is generated by applying the same type of modification on a different JarKus profile (transect 7004250). These modifications also differ in scale compared to the ones used for the training dataset. The specifics of the test data can be found in Appendix F. The content of both datasets is briefly described in Table 3.2.

Upscaling
To gain further insight into the requirements for upscaling, the training and test data to set-up a surrogate model are upgraded. Based on the insights obtained in the exploration phase, different pre-storm profiles datasets are created for training, validation and testing purposes.

For the training dataset, this includes the consideration of multiple original profiles in the dataset. Instead of considering a single original profile, four profiles will be incorporated as the base of the dataset. The specifics of this data is highlighted in Appendix D.

Next to an enlarged training dataset, the test dataset will also be scaled up. For this test dataset, 21 profiles are extracted along the stretch of the Holland coast. These profiles will solely be extended in duneward direction, but not modified with respect to the profile parameters. This is the final test to analyse by which extent the U-Net is capable of making predictions for the Holland Coast. The specifics of this test dataset are presented in Appendix F.

An overview of the different datasets used in this thesis are presented in Table 3.2.

	Dataset	Description
Exploration	Single profile	Based on a single JarKus profile 8007200, 550 modifications
	Training	80% random samples of single profile dataset
	Validation	20% random samples of single profile dataset
	Test	Based on a single JarKus profile 7004250, 78 modifications
Upscaling	Multi profile	Based on four JarKus profiles, 440 sampled modifications
	Training	80% random samples of multi profile dataset
	Validation	20% random samples of multi profile dataset
	Test	21 actual profiles along stretch of the Holland Coast

Table 3.2: Datasets of input profiles used in different stages of this research.

3.1.2. XBeach Modelling

The input profiles for the XBeach simulations have been established. XBeach also requires the specification of the storm conditions and other boundary conditions to model extreme storm behaviour. Initially, to control the amount of input parameters in the surrogate model, one single storm for all 500 profiles is chosen. Such that, in the orientation stage of this research, we can solely focus on the pre-storm profiles shape.

Most importantly, to train a surrogate model on dune erosion, the hydrodynamic conditions imposed on the cross-shore profiles should be sufficient to enforce significant transport at the dune. In this case, the waves and waterlevel simulated in XBeach should have significant impact on the dunes, such that dune erosion takes place. Throughout the range of profiles, the collision scheme, as introduced by Sallenger (2000), should be reached and different quantities of dune erosion volume should be observed.

In reality, a storm has a certain build-up and breakdown phase. Peak conditions are observed during a relatively short duration within the overall storm event (Tijssen & Diermanse, 2010). Due to the unavailability of continuous storm timeseries stormhydrographs must be described by means of a synthetic shape. For example, the temporal evolution of storms at the Holland coast can be characterized as a triangular shape (Athanasiou et al., 2021). For this research, stationary conditions at the offshore boundary are assumed. This simplifies the storm input for later stages of this research. The conditions throughout the duration of the storm result from the same forcing. Since the tide is also stationary and no build-up of the storm conditions is involved, this is a highly unrealistic situation. However, the obtained data will be suitable to explore the possibilities of setting up a surrogate model for predicting profile shapes.

To simulate sufficient dune erosion, extreme hydrodynamic conditions should be imposed. To provide guidance in the process of modelling the appropriate storm, the 1953 storm is adopted as a starting point. However, using stationary conditions would result in unrealistic dune erosion quantities (DEV 400 m^2). Therefore, a smaller storm (1976) is used as a reference instead. Hydrodynamic input parameters for this storm are deducted from Athanasiou et al. (2022) and tweaked such that realistic dune erosion processes are modelled. Characteristics of this stationary storm are presented in Table 3.3 and illustrated in Figure 3.5.

	SSL (m)	MHW (m)	H_s (m)	T_p (s)	Duration (h)
Observed 1976	2.2	1.0	6.1	10.8	30
Modelled	1.5	1.0	5.5	9	32

Table 3.3: Observed storm conditions during 1976 storm and used storm conditions for surrogate model.

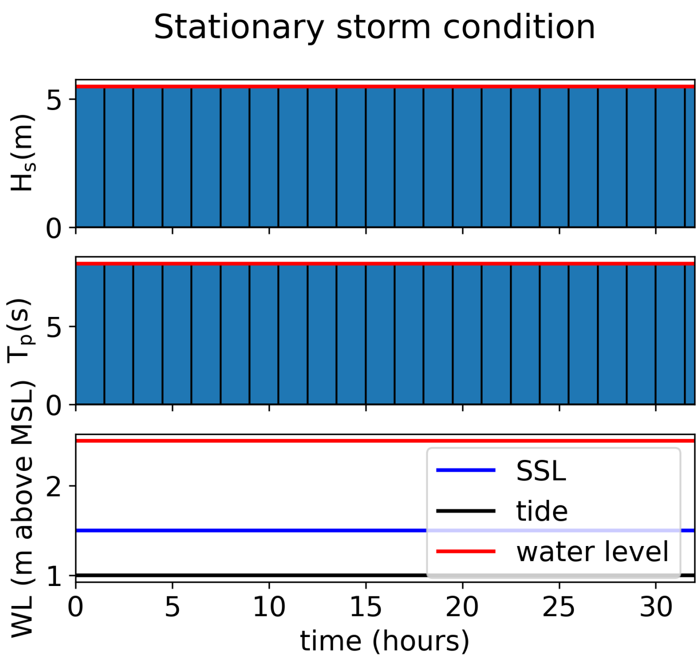


Figure 3.5: Modelled stationary storm conditions for XBeach runs

Next to sufficiently a large signal, the data supplied to the surrogate model should involve a certain range of variability. Data with a lack of variability would result in a surrogate model with limited applicability and potential overfitting issues. This variability is secured for the input (500 unique profiles) but does not necessarily imply that the simulated storm behaviour also shows this variability. The DEV is a suitable indicator of the impact of a storm on a dune system. Calculating this variable for each individual profile should give insight into the modelled storm behaviour within the dataset.

These hydrodynamic parameters are used to set-up a sea state by means of a Jonswap-type wave spectrum. This wave-spectrum summarizes intensities for several wave frequencies that frequently occur at the North Sea. This method has the advantage that the modelled wave field is more realistic and contains varying wave groups resulting in variation of hydrodynamic conditions. To set-up such a spectrum, next to the parameters listed in Table 3.3, a wave angle ($\alpha = 270°$, peak enhancement factor ($\gamma = 3.3$), directional spreading ($s = 6.3$) and timestep (1 s) need to be specified.

Already, some assumptions have been made with respect to the storm conditions. The 1D version of XBeach is used, this comes with the assumption of no alongshore processes and shore-normal waves. Any 2D or 3D processes are excluded. Next to that, the grain size of the sediment is assumed uniform along the profile. In reality, these differ along the cross-shore profile due to variations in dynamics (Bosboom & Stive, 2021).

XBeach grid

As discussed before, the 1D version of is used. Therefore, the model bathymetry input requires a definition of the x-grid and depth. As illustrated in Figure 3.6, the x-grid makes an orthogonal angle with the coastline and is drawn on both land and sea. To ensure offshore conditions and deep water, the depth of the profile is extended to -30 with a slope of 0.01. This allows the waves to build up to intermediate and shallow water. Finally, the grid is refined on certain, more dynamic, sections of the cross-shore profiles. The artificially extended part of the profile has a gridsize of 6 m and above +0 m NAP a gridsize of 1 is set. In between, the grid ranges from 5 to 6 meters. The refinement at the dune is in place to capture the high dynamics in the section of the grid (dune erosion).

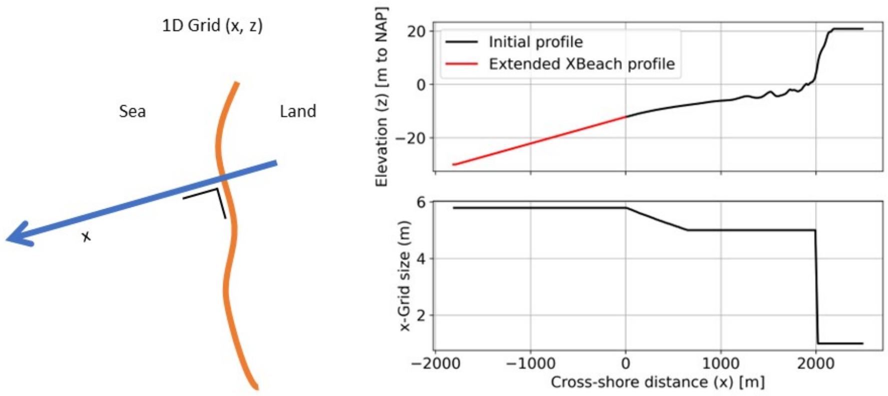


Figure 3.6: Illustration of grid reference frame (right) and grid spacing (left)

XBeach parameters

The XBeach model was set up with several other parameters. The parameters changed with respect to the regular settings are printed in Appendix A. Some tuned parameters in more detail:

- Physical processes

  XBeach is ran in the "surfbeat" mode. This mode indicates that both short waves and infragravity waves are resolved. These infragravity waves cause cyclonic patterns of run-up at the beach and are crucial in resolving the dune erosion patterns.

- Wave breaking

  Wave breaking is defined with the formulation as discussed by Daly et al. (2011), with γ and γ_2 defining the moment where waves start and stop breaking ($H > \gamma * h$). α defines the intensity of dissipation along the wave-breaking track. All values are based on combined empirical and modelling studies.

- Sediment transport

  One of the most important modelling choices within the sediment transport formulations is the definition of the equilibrium concentration. For this model, the formulations of van Rijn, with modifications by Van Thiel-de Vries (van Thiel-De Vries, 2009) are used:

$$C_{eq} = \frac{A_{sb}}{h}\left(\sqrt{v_{mg}^2 + 0.64u_{rms}^2} - U_{cr}\right)^{1.5} + \frac{A_{ss}}{h}\left(\sqrt{v_{mg}^2 + 0.64u_{rms}^2} - U_{cr}\right)^{2.4} \tag{3.1}$$

  In this formulation, the equilibrium sediment concentration depends on the depth-averaged velocities.

- Morphology

  The update of the bed due to the hydrodynamics is accelerated in the model with a factor 5 (morphological acceleration factor). This results in a lower computational time. The process of bed updating is started after a certain running time (morstart). Finally, the critical slope for avalanching in wet parts of the profile is set at 0.15.

  Using the pre-storm profiles, one single storm event and the boundary conditions described above, 1D extreme storm behaviour pre-storm profile can be modelled over the full duration of the storm.

Post-processing

To explore the driving mechanisms of dune erosion, several morphological indicators describing the sediment transport processes can be retrieved from the model output. For example, the post-storm location of the dune toe, the location of the dune crest and the slope of the duneface would be suitable indicators to demonstrate the response of the dune. In this research, the dune erosion volume (DEV) is chosen as the only morphological indicator to capture dune erosion. This is done to maintain simplicity and build upon the existing body of research (Athanasiou et al., 2022 and Gharagozlou et al., 2022). The DEV describes the volume that is eroded from the dune as a response to the storm.

The DEV is calculated by subtracting the area (1D) underneath the inputted pre-storm profiles and computed post-storm profiles. To compute these areas, the trapezoidal rule for integration is applied on a refined grid (cross-shore spacing = 1 m). These areas are bounded by two thresholds: 1) cross-shore boundary (cross-shore location of MHW) and 2) an elevation boundary (max surge + tide). The results of a typical DEV calculation are illustrated in Figure 3.7

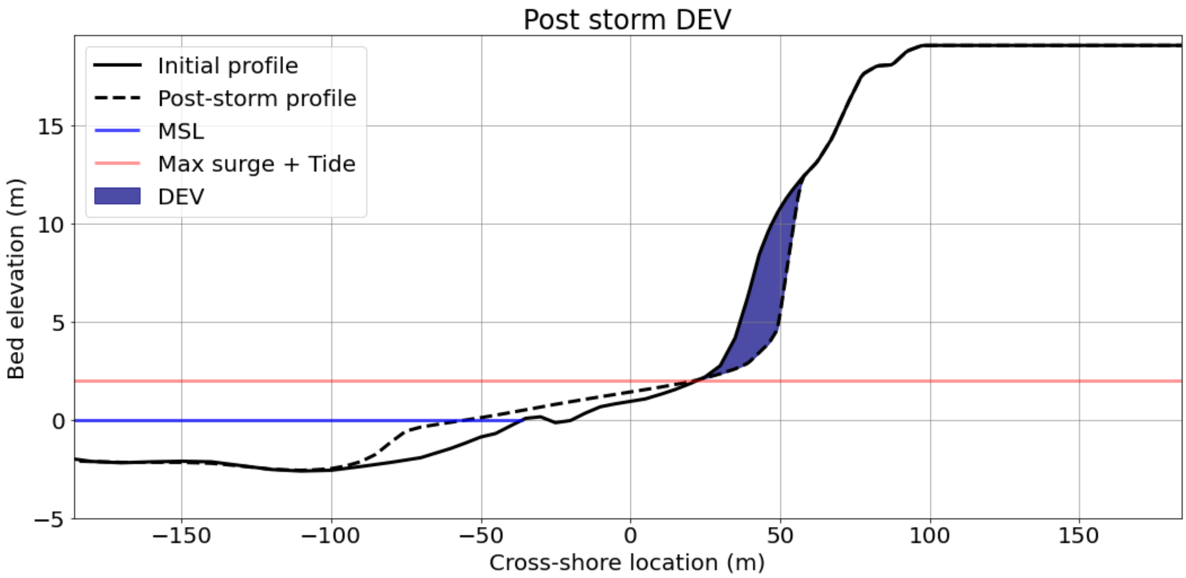


Figure 3.7: Typical DEV calculation for a post-storm profile. The blue area indicates the eroded volume from the dune, bounded by the elevation (Max surge + Tide).

In order to examine the influence of various input shapes on dune erosion, the previously discussed DEV metric is related to the slope of the profile sections that have been highlighted Figure 2.1. The slope of these sections are calculated by means of a linear fit of the endpoints of a section. This calculation is illustrated in Figure 3.8. As presented, the calculated slope is an approximation for the actual inputted bathymetry. Especially for profile sections with a non-linear trend, the endpoints calculation gives a highly consolidated result.

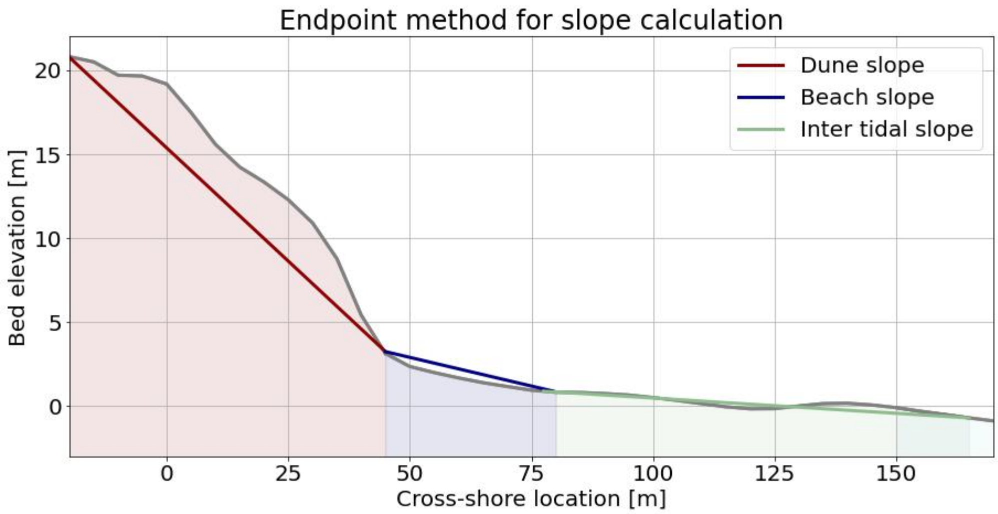


Figure 3.8: Slope calculation for profile sections, using the endpoint method. The computed slopes are illustrated in different colors.

3.2. Surrogate modelling

In this section the experimental set-up of the surrogate model will be discussed. Using the synthetic dataset, set-up in section 3.1, a surrogate model can be trained. This surrogate model uses the convolutional neural network theory and U-Net structure as discussed in section 2.2. This type of neural network can be established with different pre-processing methods, network architectures, training algorithms and hyperparameter alternatives. The methods and tools to set up these alternatives will be discussed in this chapter. Next to that, several erosion scenarios will be highlighted. These scenarios differ in forcing severity. Finally, the methods to validate and judge the performance of the surrogate model are specified.

3.2.1. Exploration

Input data

Grid

As discussed before, the input data and output data of XBeach runs are one-dimensional. For the U-Net structure to capture the same information for each input profile, the x-grid should be identical for all input profiles. Each gridpoint in this x-grid should contain the same spatial information for each profile. Therefore, the input processing start by defining a uniform grid that captures the range and characteristics of all input profiles.

The process of grid definition starts with the operation of getting all profiles in the same reference frame. As illustrated in Figure 3.4, the input profiles in the XBeach runs differ in length. Next to that, to reduce computational efforts, the profile has a varying grid spacing along its length (Figure 3.6). To secure uniformity along the cross-shore grid, both the length of the profiles and the spacing should be taken care of to set up a 1D U-Net architecture.

To obtain this uniformity, first, the cross-shore coordinates of the profiles are subtracted with cross-shore location MSL (+ 0m NAP). Second, a cross-shore value of a minimum bed elevation (z_{min}) is sought. To capture this minimum elevation of the profile with the gentlest slope, the minimum cross-shore value of this profile is employed as the baseline value for the uniform grid. Applying the same approach for the maximum value, a uniform cross-shore grid is set-up that fits all profiles in the dataset.

Next, a fixed elevation (z_{fix}) is chosen as a point on the grid where all profiles intersect at the same height and the grid is re-interpolated on a constant spacing (Δx). All grid parameters are illustrated in Figure 3.9.

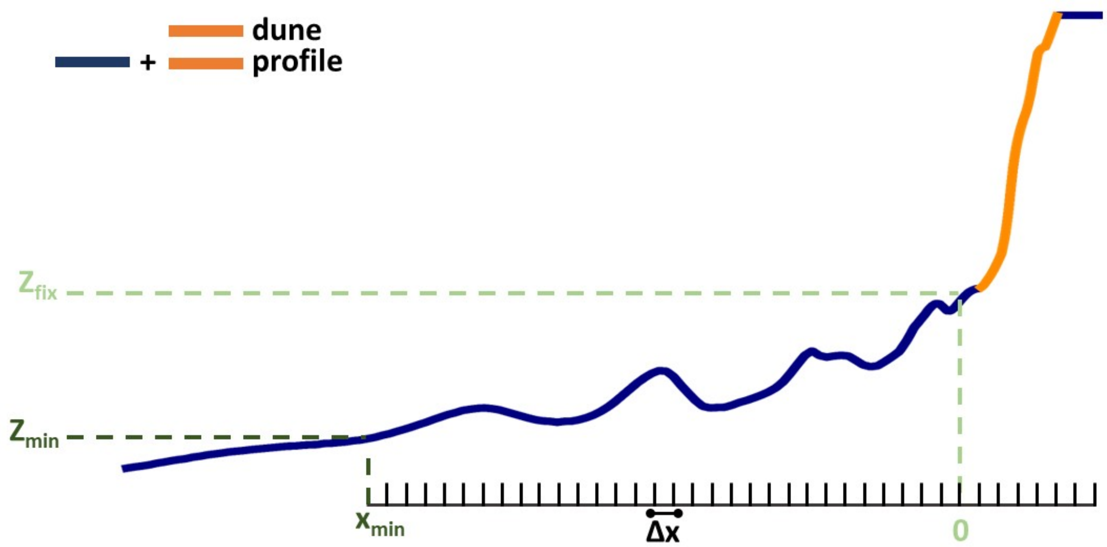


Figure 3.9: Grid standardization for pre-and post-storm profiles. Defined by z_{min}, z_{fix} and Δx. The orange colour indicates the section for which the MSE_{dune} is calculated.

Whereas, these grid parameters secure a uniform cross-shore grid, they are also important parameters that can influence the performance of the U-Net structure. For example, the z_{min} determines which part of the profile is observed by the model. Therefore, optimization of these parameters will be needed. Different grid alternatives are inputted in the same U-Net network. This process is referred to as grid optimization.

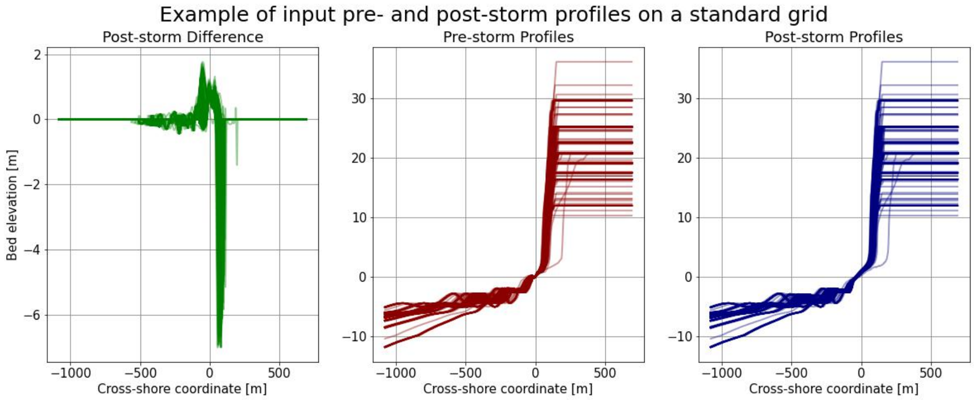


Figure 3.10: Input and outputs profiles in standard grid configuration. The left figure indicates the post-storm difference, middle and right indicate the representative pre- and post-storm profiles.

Difference

As described in section 2.1, dune erosion, as modelled by XBeach, is a local process that occurs only at a certain section of the profile. This is also found when computing the difference between pre- and post-storm profiles (Figure 4.9). Whereas the difference between the pre- and post-storm profiles is hard to spot based on the absolute values it becomes clear when considering the differences between the two. Essentially, the human eye is able to pick up the signal better when looking at differences instead of absolute values. Based on this analogy, U-Net will be trained on both the absolute elevation values and the differences between the elevation values. By training it on the differences, it is expected that the signal is easier picked up by U-Net.

Normalization

Normalization of the input data is a tool frequently used in the field of machine learning (section 2.2). By normalizing all input parameters, the model will be less sensitive to scale differences between the parameters. For the problem at hand, normalization might not be needed. Since the input data solely consists of elevation data in the same reference frame, normalization will not be needed to account for scale differences. However, it could be beneficial when other parameters are introduced and interesting to observe how the model responds to normalized data. The normalization is carried out by scaling and translating the bed elevation of each profile on a scale of 0 to 1, by the minimum and maximum elevation found in the dataset. The pre-processing module of scikit learn (Pedregosa et al., 2011) is used to normalize the data. The resulting input and output data can be found in Figure 3.11

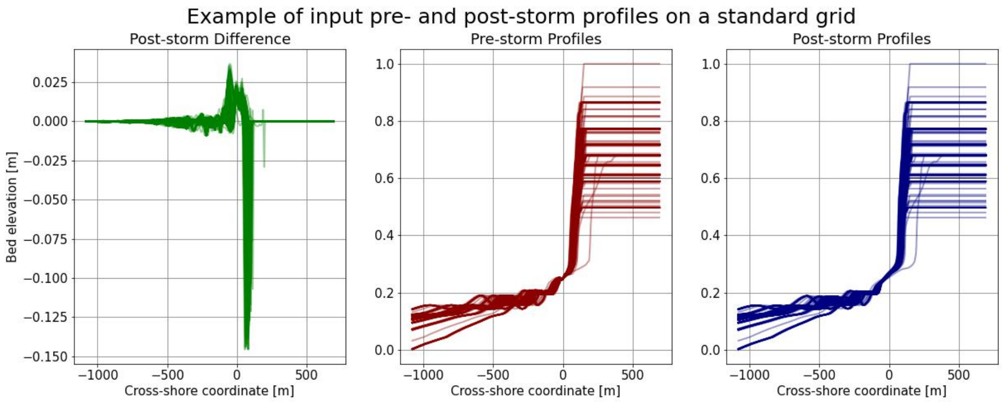


Figure 3.11: Normalized input and outputs profiles in standard grid configuration. The left figure indicates the post-storm difference, middle and right indicate the representative pre- and post-storm profiles.

Network structure

Internally, within the neural network, the structure and its hyperparameters can be optimized. This implies the optimization of several features of the neural network. Whereas the last section was dedicated to the input of the neural network, this section focuses on the parameters that regulate the overall outlook and internal processes of the neural network.

Based on findings in the literature review (section 2.2), the U-Net architecture is chosen as a suitable neural network architecture for the issued datatype. As discussed before, U-Net has been applied in the field of image segmentation and shows great potential in trend detection for linear dependent input types. Besides that, U-Net has previously been trained for issues that deal with input and output data having the same reference frame.

The original U-Net structure (Figure 2.16) was set-up for image segmentation (2D input and output). While the problem of predicting post-storm profiles can be formulated as a one-dimensional (1D) task. For the network architecture, this does not result in drastic changes. The 2D convolutional layers and pooling layers are replaced with 1D adaptations. These 1D variants use 1D filters and operations, but the underlying processes remain the same. Applying the same network depth and channel size as the original U-Net structure, the network will have the structure illustrated in Figure 3.12.

The 1D version of U-Net does not reform the network's architecture drastically, the major difference is found in the input and output size. The channel sizes are highlighted in grey underneath the respective layer. The input size depends on the data pre-processing operations, as defined in subsection 3.2.1 this will vary substantially. In general, no padding is applied, so the convolutional operations with filter size 3x1 (blue) come with a size reduction of two pixels. Besides that, the max pooling operation (size = 2) reduces the size of the input by half its size. These operations will result in a reduction in the size of the data throughout the network. This output can be rescaled to its original size. It should be noted that changing around hyperparameters such as the kernel size, stride and pooling size will result in different dimensions throughout the network.

The U-Net architecture is set up and trained in PyTorch (Paszke et al., 2019). PyTorch is an optimized tensor library for deep learning using GPUs and CPUs. This deep learning research platform provides maximum flexibility and speed. The python code for U-Net is documented in Appendix B and is an adapted version of an example provided by Arora (2020). The code is structured in classes, defining several parts of the network (Blocks, Encoder, Decoder and U-Net).

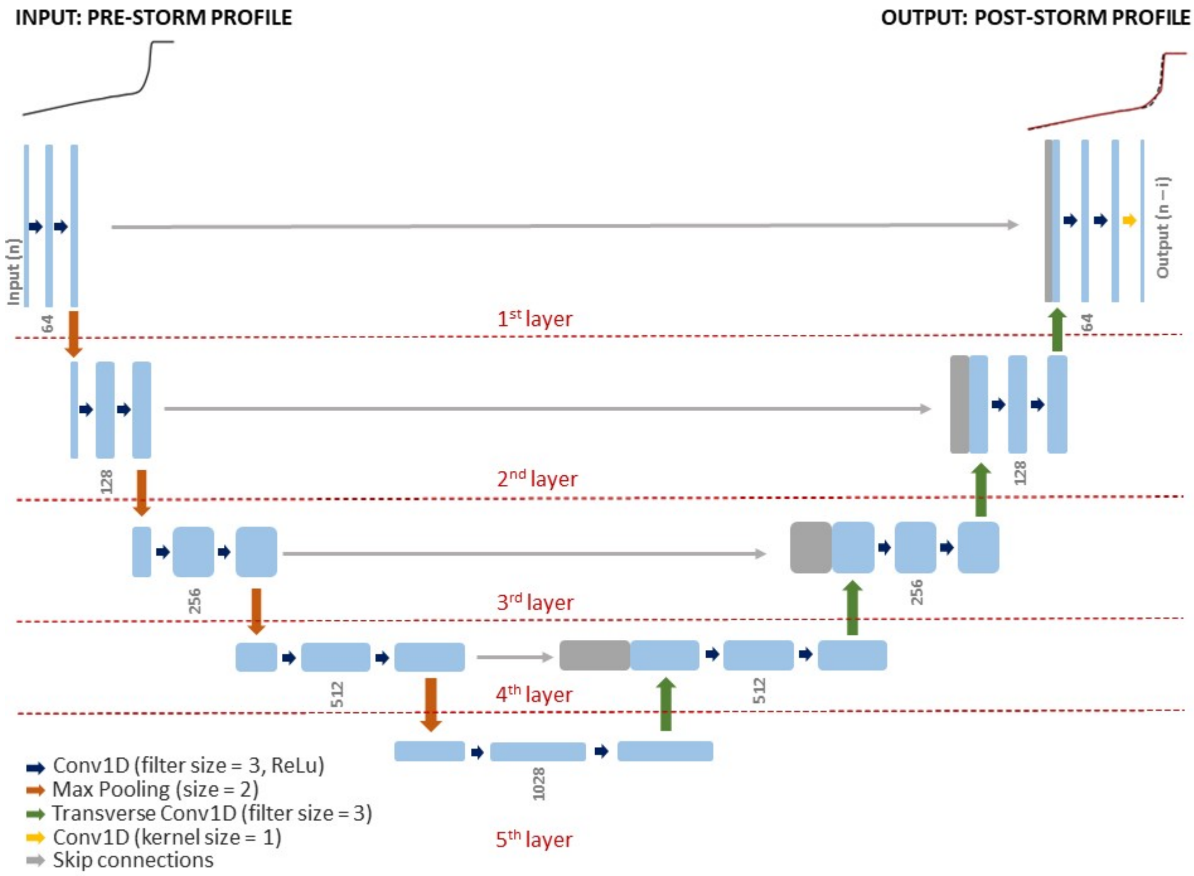


Figure 3.12: U-Net structure applied for 1D input profile data. The structure consists of two branches (encoding and decoding), which are built up from blocks of convolutional layers and connected with either pooling layers or transverse convolutional layers. The dotted red lines indicate the different layer depths

Network depth

The depth of a neural network is defined by the number of layers that the neural network is composed of. Generally, as addressed in section 2.2, this is a standard for the network's ability to capture greater detail and extract features from the data. This, however, comes with a loss of spatial information that is present in the data. The boundaries ranging from one to five as illustrated in Figure 3.12 indicate the considered layer depths in this analysis.

Skip-connections

U-Net architectures are characterized by skip-connections, whereas conventional CNN's do not have this feature. It is studied whether these skip-connections are needed and what the result on the predicted profiles might be. The skip-connections are indicated in gray in Figure 3.12

Network width

Another important parameter in the network architecture is the number of channels that is considered in the first layer. The amount of channels account for the number of filters/kernels that are considered in the first convolution. In Figure 3.12, 64 channels are used in the first layer, but for this analysis, other sizes in the geometric sequence of 2 will be applied. Starting with 8 and proceeding to 64 for the first channel size. As the next layer in the neural network is always the next doubling of this number, this requires the specification of the first channel size.

Kernel size

For the original U-Net architecture, a kernel size of 3 is used. This size has been used frequently in other U-Net applications. This choice is validated for the storm profile data that is dealt with here by applying different kernel size: 1, 2, 3, 4 and 5.

Pooling size

For the original U-Net architecture, the pooling operation downscales the dimension of the feature maps by two. In other words, this pooling size (2), halves the size of the input by taking the maximum on a patch of 2. The size of this patch in an important parameter for what resolution the encoder is picking up. The pooling sizes considered are: 1, 2, 3 and 4.

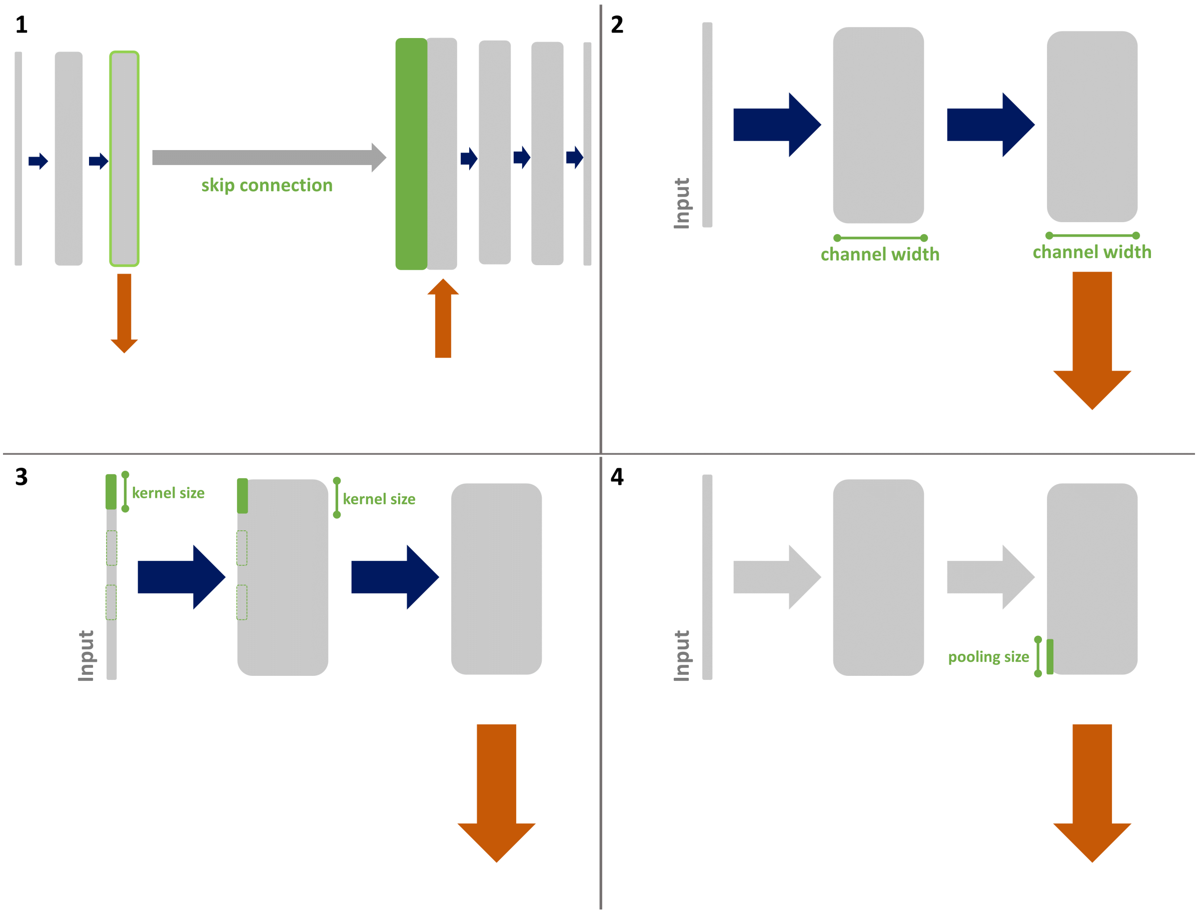


Figure 3.14: Illustration of parameters that determine the architecture of U-Net. (1) Skip connections, (2) First channel size, (3) Kernel size and (4) Pooling size

Hyperparameters

- Batch size

  The batch size that is used to train the neural network, implies the size of the batches for which the data is inputted into the neural network. To train the model, the profiles are inputted into the neural network in batches of profiles. The weights in the neural network are updated after each batch. The batch size should be considered as a fraction of the training dataset. The batch size, therefore, relies partly on the size of the dataset.

- Epoch size

  The epoch size is the equivalent for the number of times that the total dataset is inputted to the neural network. Basically, the epoch size determines the number of iterations that is used to train the neural network. Too few epochs might result in inaccurate model, while too many epochs might result in overfitting. As discussed in section 2.2, an increasing validation loss is a common indicator for overfitting processes. Therefore, an early stopping tool is applied to stop the training of the model when the validation loss has not decreased over a span of 7 epochs.

- Learning rate

  The learning rate determines the step size to reach the global minimum that is illustrated in Figure 2.9. The evaluated learning rates, range between 0.01 and 0.0001.

Loss functions

Alongside pre-processing the input data and designing the neural network architecture, the selection of loss functions for computing gradients during backpropagation is an important aspect of the neural network. To investigate the potential of such a loss function several alternatives are highlighted. All alternatives will be computed by computing the mean square error as shown in subsection 2.2.1.

Elevation error

This loss function has the most straightforward form. The loss is calculated by taking the mean square error on the full profile. There is no induced amplification or weights on specific parts or variables. The loss is an averaged value over a batch (m profiles), the calculated values for these profiles are the MSE over all elevation points (y_n) on the profile.

$$E = MSE_{elevation} = \frac{1}{m}\sum(y_n - \hat{y}_n)^2 \tag{3.2}$$

Elevation error with target section

This loss function is similar to the previous one, however, the errors computed from the dune toe to the dunecrest are amplified. By applying a weight to the errors occurring on this section, they are prioritised over the others. This technique is referred to as targeting and is outlined in the equation below. The subscript r and s refer to the points on the regular profile and section respectively. The latter is multiplied with a weight (W). To clarify, this method is illustrated in Figure 3.15.

$$E = MSE_{elevation,weighted} = \frac{1}{m}\left(\sum(y_r - \hat{y}_r)^2 + W * \left(\frac{1}{m}\sum(y_s - \hat{y}_s)^2\right)\right) \tag{3.3}$$

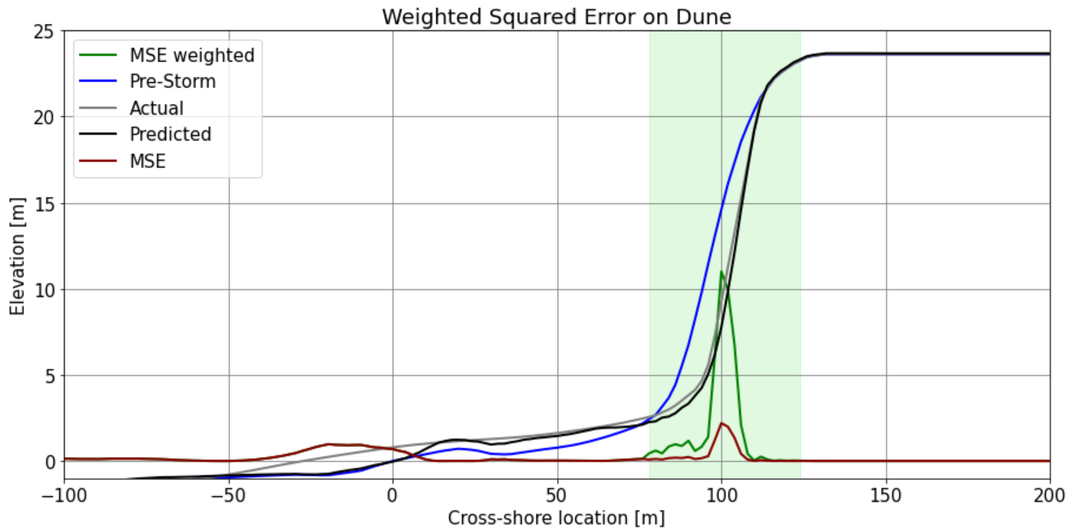


Figure 3.15: Example of weighted MSE for dune section. The dune section for this profile is highlighted in light green. At this section the MSE is multiplied by a weight, resulting in a weighted MSE (green).

Surrogate model applicability

To test the applicability and predictive skill of the model, the DEV and other beach characteristics will be extracted from the predicted profiles and compared to the target profiles. Next to comparative scatter plots of the extracted parameters, the skill parameter (subsection 3.2.3) will be an important indicator for the predictive skill.

Data quantity

The surrogate model will be tested for several quantaties of profiles. A subset of the training data is taken based on a certain percentage. The selection process of the subset of profiles included in the dataset will be done at random. It will be analysed at which data quantity the result converges towards a stable result as found for the complete data.

3.2.2. Upscaling

Considering a surrogate model is established, capable of predicting profile shapes for a simplified scenario, the next phase in the research process involves upscaling these outcomes. As previously mentioned, this is accomplished by setting-up a multi-profile training dataset and a realistic test dataset. This set-up is highlighted in Figure 3.1.1. Building upon findings in the exploration phase, additional performance metric are incorporated into the testing process. Besides that, sampling methods are explored to reduce the size of dataset and prevent overfitting issues.

Throughout this upscaling phase, all model alternatives are tested on a realistic test case. Initially, the performance of both the single- and multi-profile-based datasets are compared. This comparison aims to verify whether the process of upscaling the training dataset is indeed beneficial for post-storm profile prediction. Next, the multi-profile dataset is utilized to analyse several U-Net structures. Analyses on the network depth, network width and kernel size are carried out for a selected profile in the test data. These analyses are again carried out on the applicability of the surrogate model to extract the DEV, but also include a visual inspection. Finally, a selected U-Net structure is evaluated based on the capability to mimic dune erosion processes as captured in the full test dataset and highlighted in Equation 3.1.2. These different stages of the upscaling phase are illustrated in Figure 3.16.

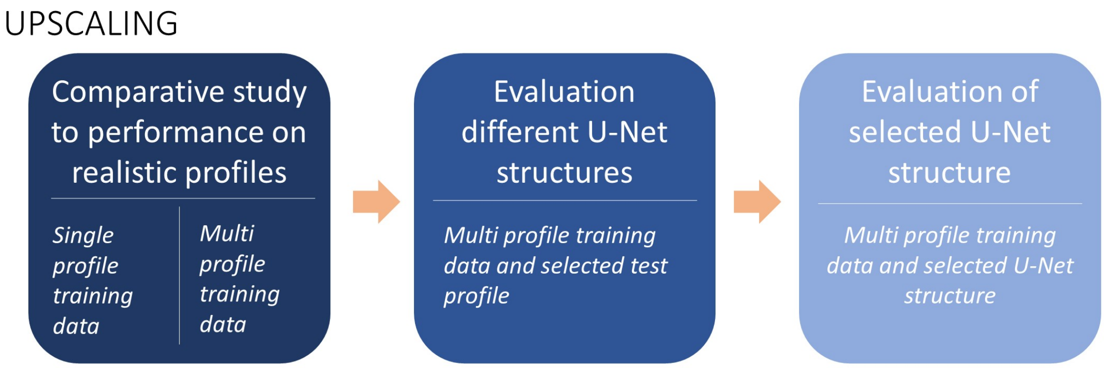


Figure 3.16: Different stages of the upscaling phase. Start with a comparative study of two training dataset (single- and multi-profile based) on a realistic test cast. Next, several U-Net structures are evaluated for a selected test profile. Finally, a selected U-Net profile is evaluated for the dune erosion processes captured in the full test dataset.

3.2.3. Testing

Data

To be able to test the neural network performance, a test dataset is set-up. In the orientation phase, this test dataset is based on a different JarKus profile and has also been modified to generate multiple pre-strom profiles. The specifics of this test dataset can be found in Appendix F.

As explained before, in the upscaling phase, the test dataset consists of actual profiles along the Holland coast. These profiles have solely been extended at the dunecrest to secure sediment availability.

Metric

To judge the performance of the Neural Network, an appropriate testing scheme is needed. As discussed in subsection 2.2.4, several tools are available. For the goodness of the fit, the Mean Square Error (MSE) is an appropriate indicator. However, it comes with some shortcomings that should be taken care of through another metric.

To make sure that all grid alternatives are judged correctly, grids need to be re-interpolated on a fixed amount of points. This is done for the full profile ($MSE_{profile}$) and for the dune (MSE_{dune}). The boundaries of both ranges are indicated in Figure 3.9 The amount of points on this interpolated grid is based on the largest amount of points found in the computed grids. Next to the MSE, to account for the sign of the error, the Mean Error (ME) is computed for the full profile ($ME_{profile}$) and the dune (ME_{dune}).

Since the computed profiles should be useful for the extraction of morphological parameters, this will be tested by calculating the dune erosion volume (DEV). The quality of the prediction for the DEV will be assessed using a skill score (Murphy, 1988)

Finally, to interpret the quantitative metrics introduced above, a visual inspection of the computed profiles will be carried out. This detailed analysis of the profile shapes is introduced to overcome spatial averaging and include the shape of the predicted profile. To reduce the amount of profiles that need visual inspection, a subset of the original training dataset is taken. This subset is selected by carrying out a Maximum Dissimilarity Algorithm (MDA) on the elevation data.

Metric	**Purpose**	**Equation**	**Range**	Best value
$MSE_{section}$	General fit on profile section (profile and dune)	$\frac{1}{m}\sum(y_n - \hat{y}_n)^2$	0 – inf	0
$ME_{section}$	General fit on profile section (profile and dune) with sign	$\frac{1}{m}\sum(y_n - \hat{y}_n)$	$-\infty - \infty$	0
$skill_{DEV}$	Extraction of morphological parameter	$1 - \frac{MSE_{DEV}}{\sigma_{DEV_{target}}}$	$-\infty - 1$	1
Visual inspection	Spatial distribution of error	-	-	-

Table 3.4: Used metric to judge model performance.

4

Results

4.1. Driving mechanisms of dune erosion

This section gives further insight into the modelled storm behaviour in the synthetic dataset set up with XBeach (section 3.1). To find out what morphological response is found in dune profiles as an effect of imposing storm conditions, several analyses are carried out.

4.1.1. Post storm profile

Figure 4.1 illustrates a typical pre- and post-storm profile that has been modelled with XBeach. The Dune Erosion Volume is highlighted in blue. A shown in the figure, the DEV is calculated above the threshold of the maximum water level. In the initial training dataset, the dune erosion remains in the collision scheme and therefore never reaches the top of the dune. The temporal evolution of the DEV in somewhat linear, but tends to decrease in trend over the duration of the storm.

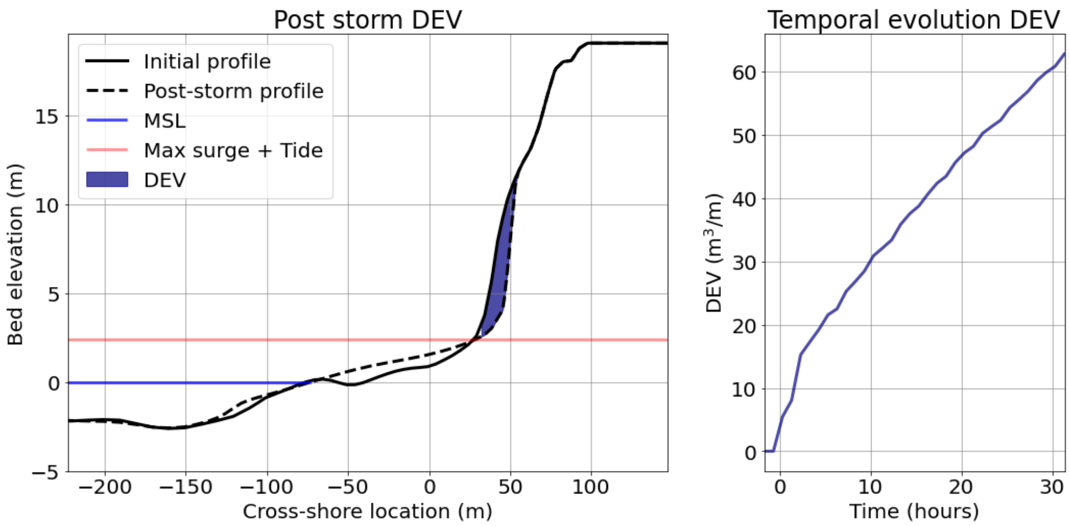


Figure 4.1: Example of the modelled DEV using XBeach. The solid line indicates the pre-storm profile and the dotted line the post-storm profile. The eroded volume from the dune is highlighted in dark blue (DEV). The figure at the right indicates the temporal evolution of the DEV over the duration of the storm.

4.1.2. Variability in storm response

To show the modelled signal and variability, the calculated DEV with respect to the index of the profiles is illustrated in Figure F.6. Next to that, a histogram plot of the same data is shown.

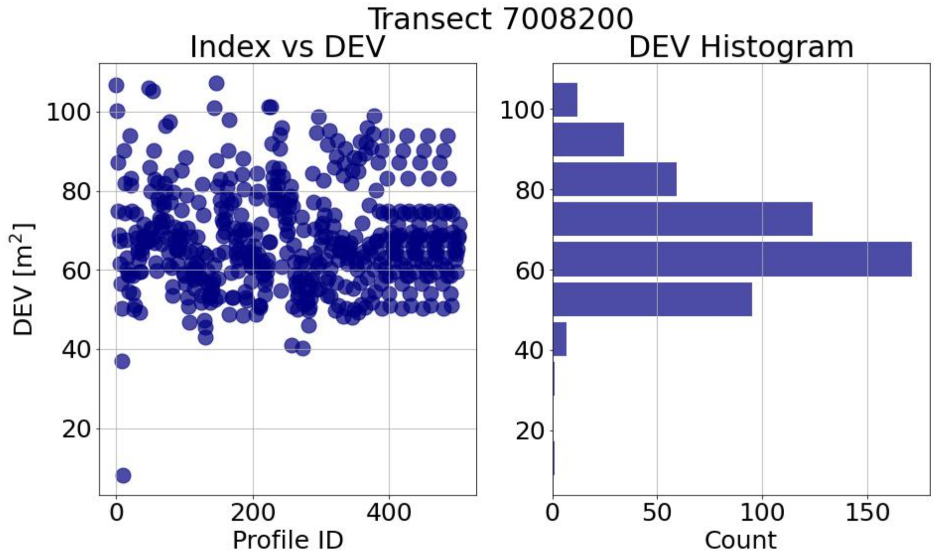


Figure 4.2: The variability of DEV captured in the dataset. The DEV per profile id (left) and a histogram plot of the DEV data (right)

In both figures, a variability in DEV can be observed. The quantities of DEV are ranging between 0 and 110 m^2 and have an average of 67 m^2 and a standard deviation of 12.6 m^2. Next to that, it can be seen that the variability in DEV is quite small for the last profile IDs in the dataset. This can be explained by the fact that these indices represent profiles with modifications of the dune height. As discussed in the next section, the DEV is not that sensitive to this parameter.

The same analysis has been carried out for the test dataset. This can be found in Appendix F.

4.1.3. Parameter sensitivity

To obtain an understanding of the sensitivity of the model to different input profile shapes, a parameter sensitivity analysis on the morphological change is carried out. To do so, profiles with a single modification are isolated and will be analysed. This allows for the required isolation of the adjusted parameter. The modelled dune erosion volume is used as an indicator for morphological change to carry out this parameter sensitivity analysis.

The sensitivity of the DEV to the slope of the studied parameters (profile modification) is printed in the same figure. Note that the values represented on both axes are changes in percentage compared to the reference situation (no modifications). The results are shown in Figure 4.3

At first, it should be noted that the ranges for which the different parameters are analysed are different. To keep the modifications in between the physical and realistic boundaries of the Holland coast, different ranges of modification factors have been applied. This is explained in more detail in subsection 3.1.1.

Figure 4.3 shows the sensitivity of the model to different slope parameters. As shown in the figure, the reference profile is (logically) found at the point where no change is observed. In general, a steep line in this figure suggests a higher sensitivity to this parameter and a gentle line suggests a lower sensitivity. It can be seen that, for all parameters, a steeper slope leads to a higher DEV. This effect is of different severity for the analysed parameters. The smallest change in DEV is observed for the changes in dune slope. The sensitivity of the DEV to the beach- and inter-tidal slope is of the same order for slope increases. While, for slope decreases, the nearshore and beach show a similar trend.

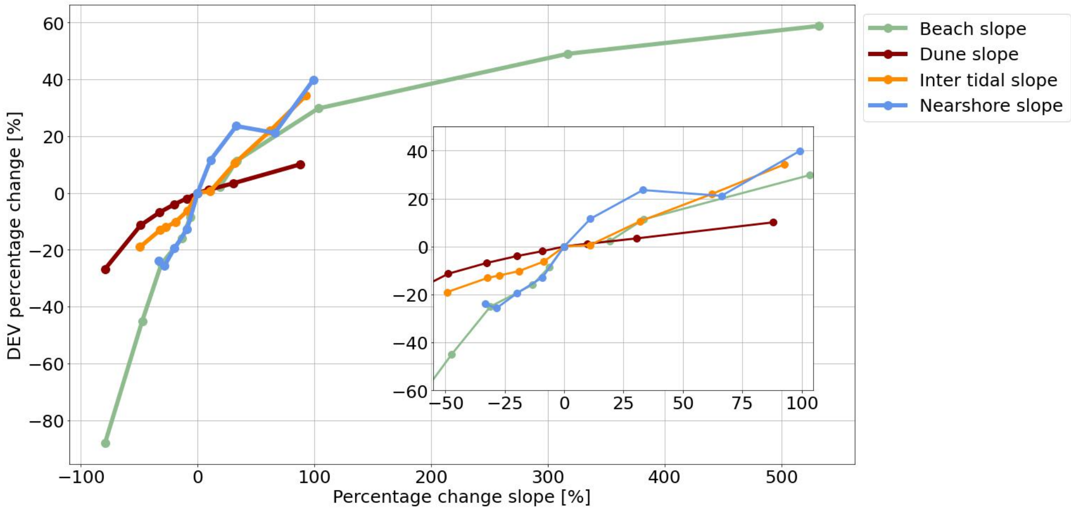


Figure 4.3: Sensitivity of the DEV for different beach profile modifications. Both the change in DEV and slope are indicated in percentage. Varying colors represent different profile sections.

The trends for the inter-tidal and nearshore slope seems more or less linear on the analysed range. The nearshore slope, however, shows a sudden decrease in DEV around a slope change of +60% and a sudden increase around -35%. These sudden changes are not in line with the expected dune erosion behaviour. The dune- and beach slope show a more logarithmic trend. Because of its larger range and high sensitivity this is especially clear for the beach slope.

Analysing the DEV for profiles with modified dune crest height in Figure 4.4, the sensitivity is much smaller. Small changes in DEV are only observed for the first method of dune height modification. This method comes with a change in dune slope and therefore influences the DEV the same as the dune width modification. The profiles that undergo modifications solely in the dune crest height while keeping the dune slope unchanged do not induce any changes in DEV. This indicates that the dune erosion events do not reach the crest of the dune and is therefore not driven by dune crest height.

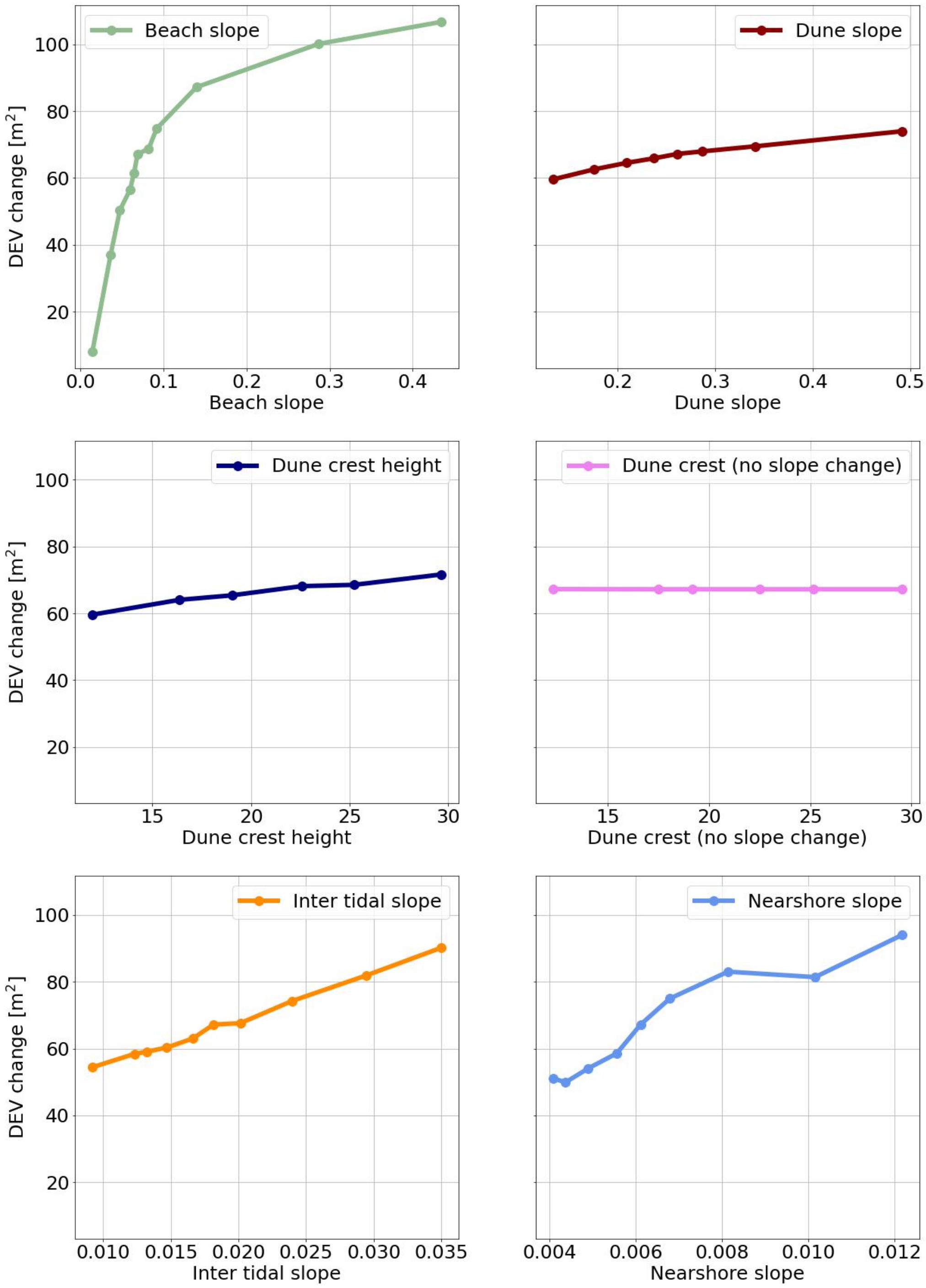


Figure 4.4: All analysed beach parameter (beach, dune, dunecrest, inter-tidal, nearshore) vs the DEV on the same y-scale.

4.2. Surrogate modelling

This section presents the results of the experiments highlighted in chapter 3. These results should give further insight into the performance of several convolutional neural network alternatives. The results will be presented in the corresponding sequence and performance metrics as introduced in the methodology chapter. To provide clarity and additional information, some sections are provided with more detailed analyses.

4.2.1. Exploration

Input data

Difference

Experiments for predicting the absolute elevation points and the difference between elevation points as discussed in Figure 3.2.1 are presented below. The "difference model" is outperforming the "absolute model" for almost every model run. This illustrated with a scatter plot of the MSE_{dune} and $MSE_{profile}$

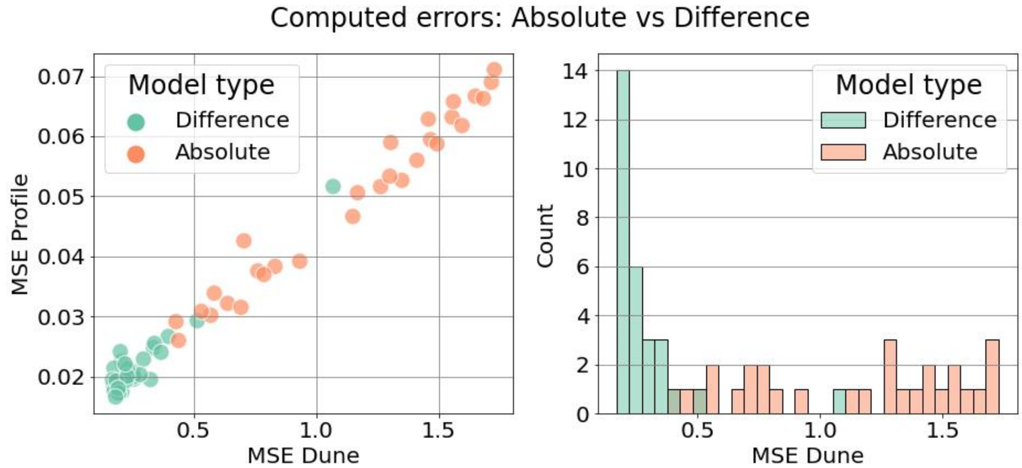


Figure 4.5: Computed MSE for training U-Net on absolute- (red) and difference (green) elevation points.

Grid

Using a U-Net structure with standard parameter settings (Appendix C) and trained on the difference, the optimal input grid for the cross-shore profiles is sought. This is achieved by varying between three parameter as presented in subsection 3.2.1. 4.6 demonstrates the disparity in MSE at the dune, resulting from differences in the cross-shore point spacing. Particularly, two extremes ($\Delta x = 1$ and $\Delta x = 10$ show higher errors. Visually inspection of these results (Appendix I) indicate the grid size of 5 and 10 m are too coarse to correctly interpret the dune shapes.

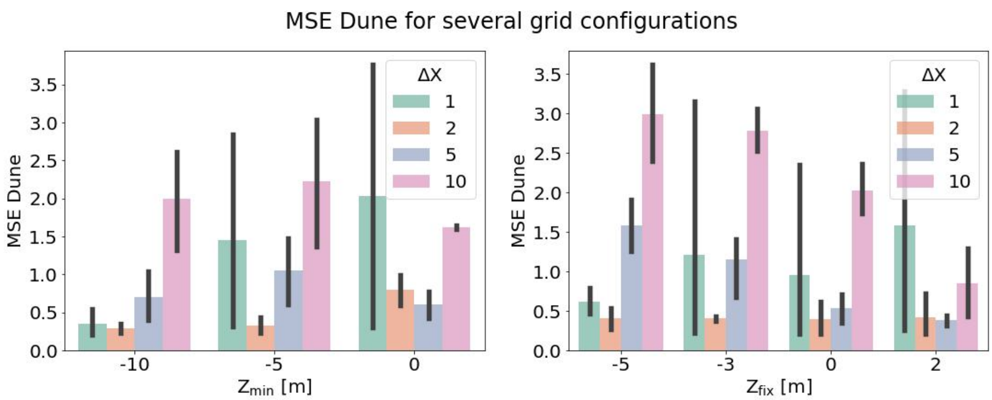


Figure 4.6: Computed MSE at the dune for different grid compositions. The left figure illustrates the results for z_{min}. The left figure illustrates the results for z_{fix}. Different ΔX are indicated by the colors.

Using $\Delta x = 2$, the minimal (Z_{min}) and fixed (Z_{fix}) elevation are inspected with more detail. After increasing Z_{min} above a value of -5 m, the observed errors start to become higher and more variable (Figure 4.7). For the fixed elevation for all profiles, the most promising results occur for an elevation of around MSL.

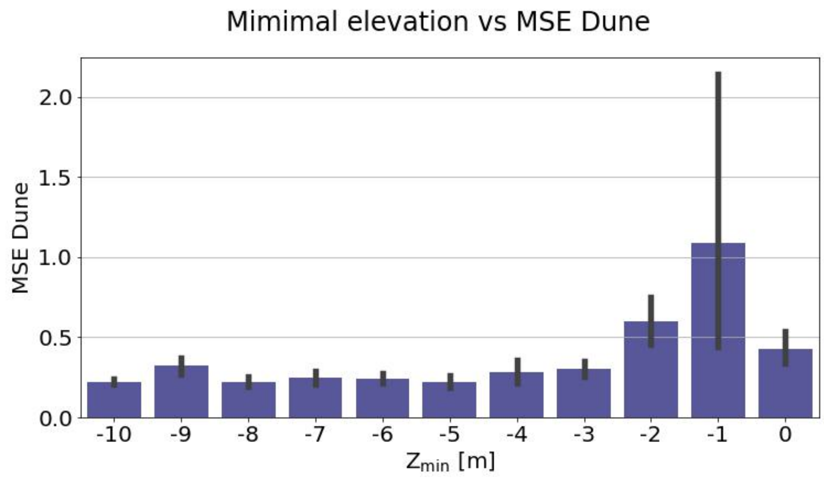


Figure 4.7: MSE at the dune as a result of minimal elevation values in the grid.

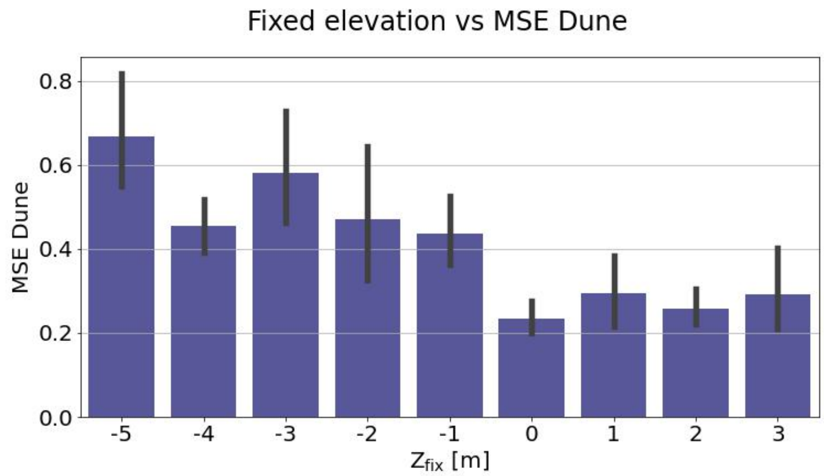


Figure 4.8: MSE at the dune as a result of fixed elevation values of the grid.

Based on these results, a cross-shore grid spacing of 2 m is chosen for the continuation of this analysis. Besides that, the minimal elevation (Z_{min}) is set at -5 m and for the fixed elevation (Z_{fix}) a value of 0 m is used. This representation of the profiles in this grid is shown in Figure 4.9. As depicted, the profile characterized by the gentlest slope serves as the reference for extending the remaining profiles. Moreover, all profiles intersect at the fixed point of z = 0.

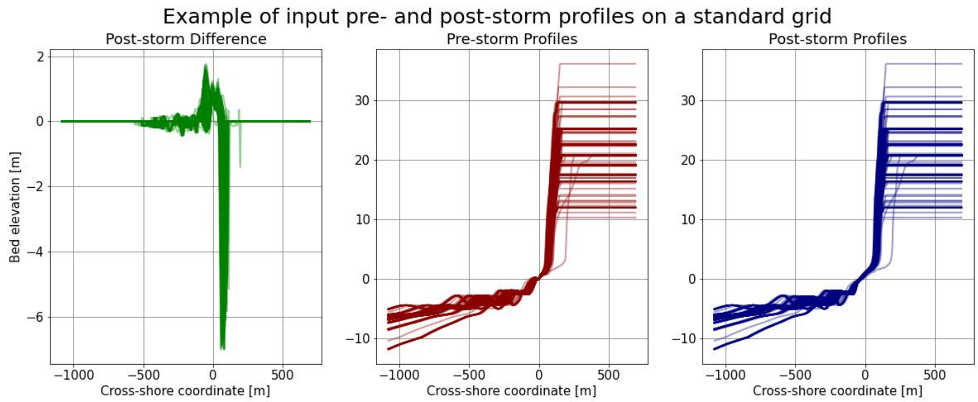


Figure 4.9: Input and outputs profiles in standard grid configuration. The left figure indicates the post-storm difference, middle and right indicate the representative pre- and post-storm profiles.

Network structure

Next, the network structure and hyperparameters in the training process are considered. All results are represented by means of the MSE Dune.

Network depth and width

First results of the optimization study showed that the data at hand (500 profiles) does not require a deep network structure to account for the modelled elevation changes. Figure 4.10 shows that the models perform best for a network depth of 2 and first channel size of 32. Using these results, other hyperparameters have been tuned. In Figure 4.19, the depth and width of the network are revised again.

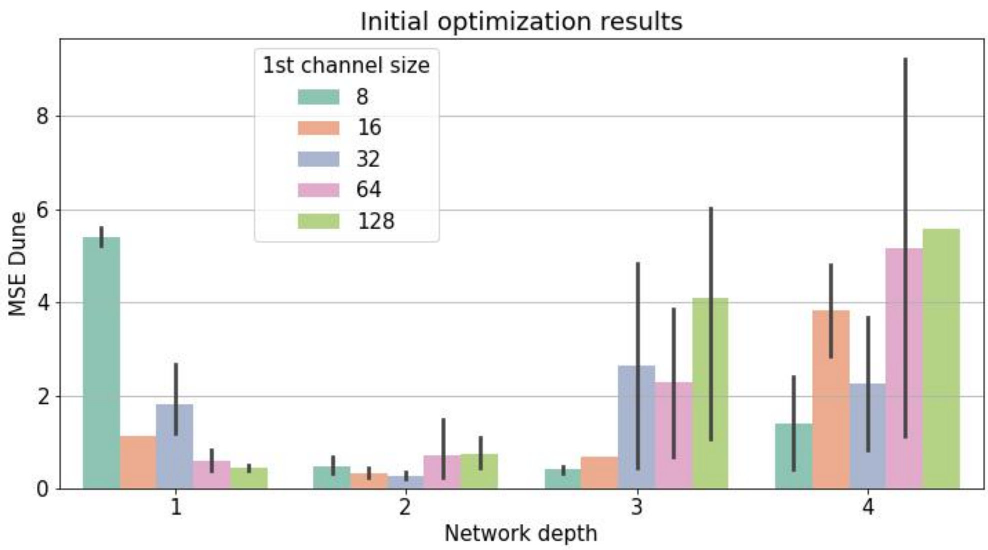


Figure 4.10: Initial results for U-Net depth and width. The x-axis represents an increasing network depth. The varying colors indicate different network depths (first channel size).

Skip connections

This section presents the results for a distinction between two different model types: with and without skip connections. Clearly, the predicted dune shapes are better for U-Net structures with skip-connections. This is found in both the mean square error (Figure 4.11) and the visual inspection (Figure 4.12). Without applying skip connections, the predicted profile shows oscillations at the dune. This does not occur for alternatives that do use skip connections. These results are discussed in subsection 5.2.4.

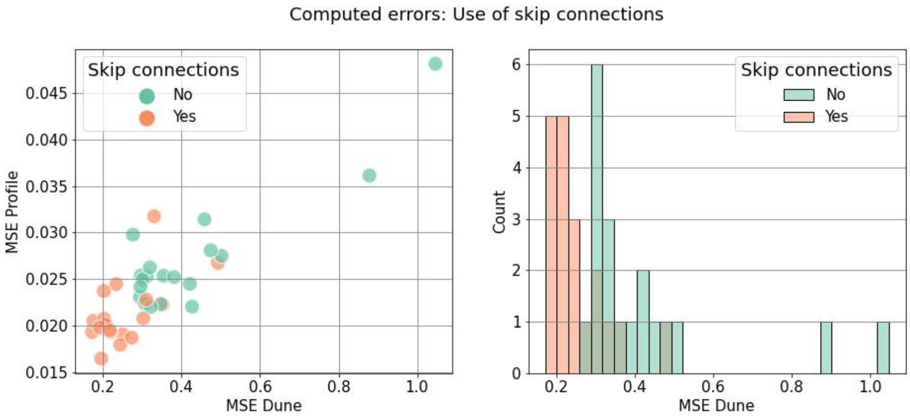


Figure 4.11: MSE at the dune as a results of usage of (no) skip-connection

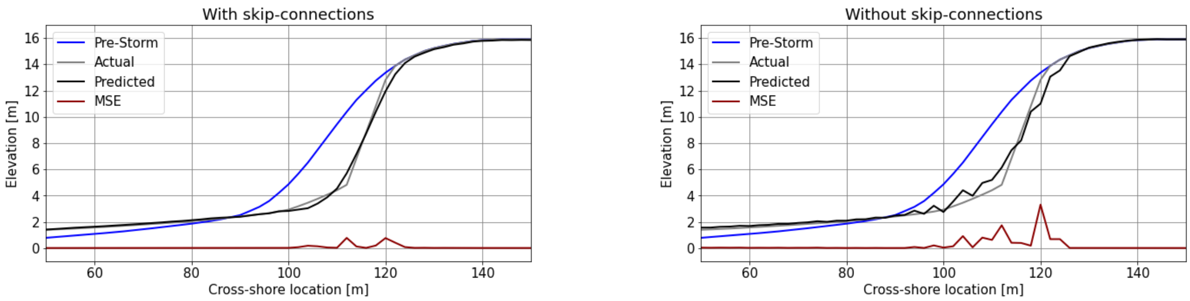


Figure 4.12: Example with and without skip connections of a predicted profile

Other hyperparameters

The results for several hyperparameters (1) kernel size, (2) pooling size, (3) learning rate and (4) batch are illustrated in Figure 4.19. The results for the kernel and pooling size are in line with the results found in other U-Net optimization schemes. The learning rate seems to be optimal at a value of 0.00075. This can also be found when analysing different loss curves (Appendix H). The loss curve for a learning rate of 0.00075, generally, shows the most optimal shape. For the batch size, it is hard to find a particular trend. While the error seems to be constantly increasing for larger batch sizes, this trend is disrupted by the batch size of 10.

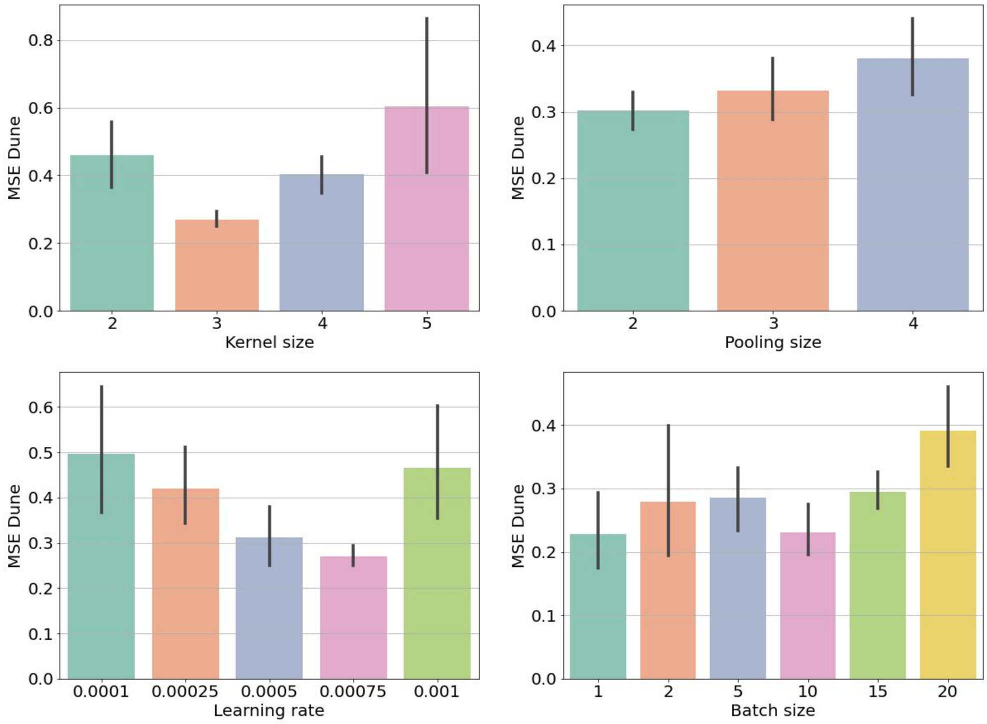


Figure 4.13: MSE at the dune for several hyperparameter alternatives. Kernel size (top left), Pooling size (top right), Learning rate (bottom left) and Batch size (bottom right)

Epoch size

A discussed in Figure 3.2.1, the optimization is carried out with an early stopping tool. This tool uses a patience of 7 epochs for the validation loss to increase. As described in Figure 4.19, the number of epochs that is reached ranges between 5 and 30 epochs and the average epoch that is marked as a checkpoint is 14.

Evaluation: Network depth and width

Since appropriate values for the hyperparameters are obtained and more stable results can be generated, the depth and the width of the neural network can be further analysed. The chosen values for the U-Net version 1 are presented in Table C.2). The results of this analysis are presented in Figure 4.14.

Some interesting trends can be observed in this analysis. In line with the initial results, the network depth of 2, seems like the most suitable network structure. On average, the error for a network depth of 3 and 4 are higher. However, when considering the size of the first channel in the network per different trends can be observed. For shallow networks (network depth = 2), a larger channel size seems beneficial. On the contrary, for a deeper network (network depth = 4), a smaller first channel leads to a smaller error. It seems that, for this dataset, the network only needs a certain amount of complexity (either through network depth or channel size). These results are discussed in subsection 5.2.2.

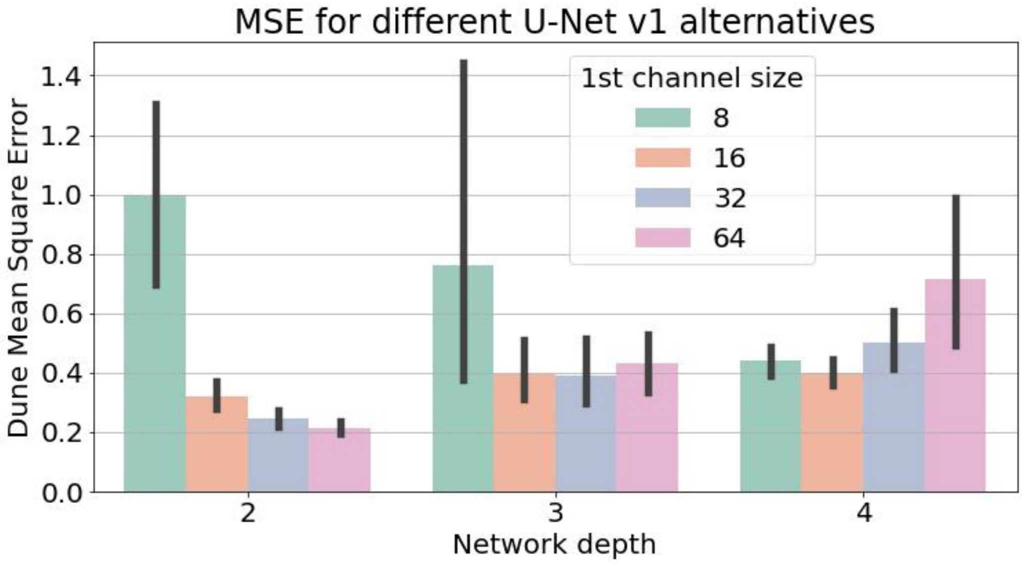


Figure 4.14: Final optimization results for U-Net depth and width in the exploration phase. Varying colors indicate the different channel sizes.

While not used for this research, Appendix K contains the feature maps that are found in the channels of different layers in a 2-layer U-Net structure. It is noticeable that oscillations emerge as the input progresses towards the decoding branch of the neural network structure.

Training algorithm

Loss Function: Targeting

So far, the neural network has been trained on the difference (Equation 3.3) for the full profile. By amplifying the computed error for a certain profile section (dune), it is attempted to target this part of the profile. However, for the actual difference, this amplification already exists naturally. To show the importance/relevance of this targeting principle, the data is normalized such that this natural weight in the difference is reduced.

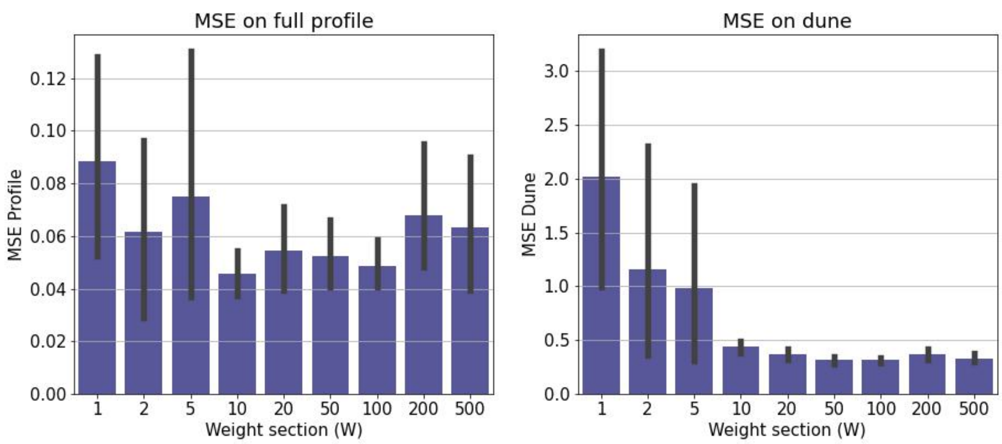


Figure 4.15: Resulting MSE of applying a weight on the MSE at the dune section for the full profile (left) and dune (right)

Figure 4.15 shows the MSE as a result of applying a weight (W) to the duneface of the profile. The computed MSE for the full profile does not differ for different weights. For the dune, however, the MSE reduces significantly when a larger weight is applied. In other words, the amplification of the error on a certain profile section has the expected result.

Surrogate model applicability

To test the applicability of the surrogate model to extract morphological parameters, the predicted DEVs are compared to the target DEVs. This is done for a single training run of U-Net.
Although there will be some natural variability in found in the predictions of U-Net (Appendix H, Figure H.6), the findings presented below are consistently found for the U-Net model presented above.

Dune Erosion Volumes

Plotting the predicted DEV (DEV_{pred}) over the target DEV (DEV_{target}) for each profile in the test dataset (Figure 4.16), the best possible fit for each profile is located at the line $DEV_{pred} = DEV_{target}$ (red line). For the U-Net architecture with a depth of 2 layers and a kernel size of 3, the results of such an analysis are presented in Figure 4.16 (left). As shown, the predictions are centred around the red line but show a distinct horizontal alignment. This horizontal alignment can be vertically separated by means of the height of the dune (Z_{dune}).

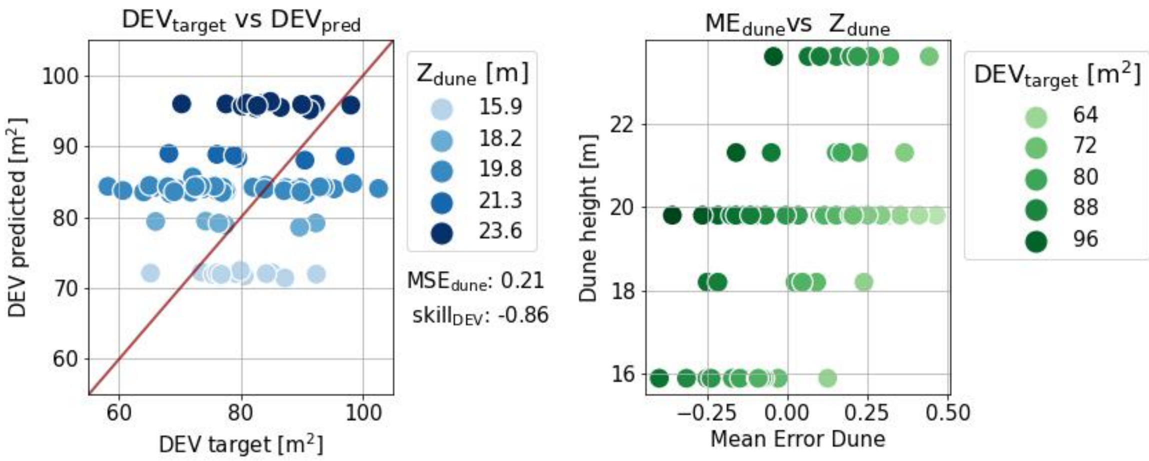


Figure 4.16: Target DEV vs Predicted DEV (left). The red line indicates the situation for which the target DEV is predicted perfectly ($DEV_{pred} = DEV_{target}$). Mean Error at Dune vs Dune height (right).

Note: It is important to stress that the U-Net profile predictions have a slight elevation difference on top of the dune (0.01 m). This is accounted for by initializing an upper threshold for the calculation of the DEV at the dunecrest (Appendix J).

Observing these results, it is suggested that U-Net is predicting one typical post-storm profile and is scaling that profile by means of the height of the dune. Statistically, this makes sense, physically, however, it is not in line with the modelled driving mechanisms of dune erosion. As highlighted in subsection 4.1.3, the dune height is not at all an important profile parameter for the modelled DEV in this dataset. While for this U-Net structure, there seems to be a significant relation between the two. This relation is illustrated in Figure 4.16 (right), showing an increase in predicted DEV for a larger dune height. Besides that, judging from the colouring, large mean errors occur for profiles that are on the boundaries of the variability in the test data. Such that the outliers in the dataset, which suppose to have large and small DEVs, have a larger error on the dune for the prediction. These results suggest that, for the single-profile-based training dataset and network structure, the surrogate model is predicting one profile shape and is scaling this prediction to the inputted pre-storm profile.

More detailed analyses of the predictions of U-Net for specific profiles are presented in Figure 4.17. For this plot, pre-storm profiles (training data) which have solely been modified for the nearshore slope (top left) are isolated. As indicated in the top right figure, the pre-storm dune shape is identical for all input profiles. Different dune erosion quantities and dune shapes are modelled with XBeach (bottom left). The prediction of

U-Net (bottom right), however, does not show these different dune shapes. Instead, only one dune shape is modelled for all pre-storm profile shapes. This indicates that the current U-Net architecture is unable to correctly interpret pre-storm profile characteristics to predict erosion processes at the dune.

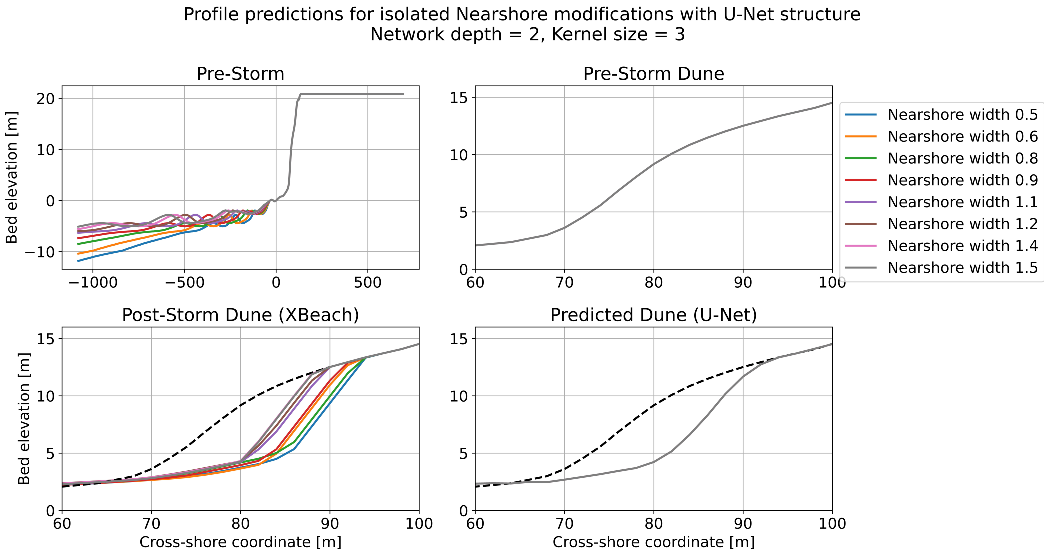


Figure 4.17: Computed and target dune shapes for different nearshore slopes. The top figures show the pre-storm profiles with different limits on the axis. The bottom figures present the target and predicted post-storm dune shapes. The predicted dune shapes by U-Net are exactly the same and overlap.

Similar analyses have been carried out for profiles which were solely modified on beach slope (Appendix I).

Evaluation: Network depth and kernel size

Studying the effect of altering the network structure on DEV predictions, it was discovered that implementing a network depth of 3 layers and a kernel size of 10, the aforementioned issues become less distinguished. The predicted DEV does not rely on the dune height anymore and the different nearshore modifications lead to a different morphological response. U-Net is able to transfer information from deep nearshore areas to the dune (Appendix I, Figure I.2).

When plotting the DEV prediction for several network structures (Figure 4.18), a similar trend is found. Using deeper networks and larger kernel sizes, U-Net is able to interpret the morphological change. The model is no longer predicting scaled post-storm profiles and is able to capture the variations of DEV in the data. The horizontal alignment vanishes for deeper networks and larger kernel sizes. However, the surrogate model is unable to assess the scale of the dune erosion. Thus, an increases of the MSE_{dune} and reduction of $skill-DEV$ is observed. These results are discussed in subsection 5.2.2.

Analysing the results for the training data (Appendix H, Figure H.7), these observation are confirmed. Surrogate models using more layers and increased kernel sizes are able to capture the variability in the data. On the contrary, as indicated by the improved error statistics (MSE_{dune} and $skill_{DEV}$), the scale of the dune erosion is also interpreted correctly.

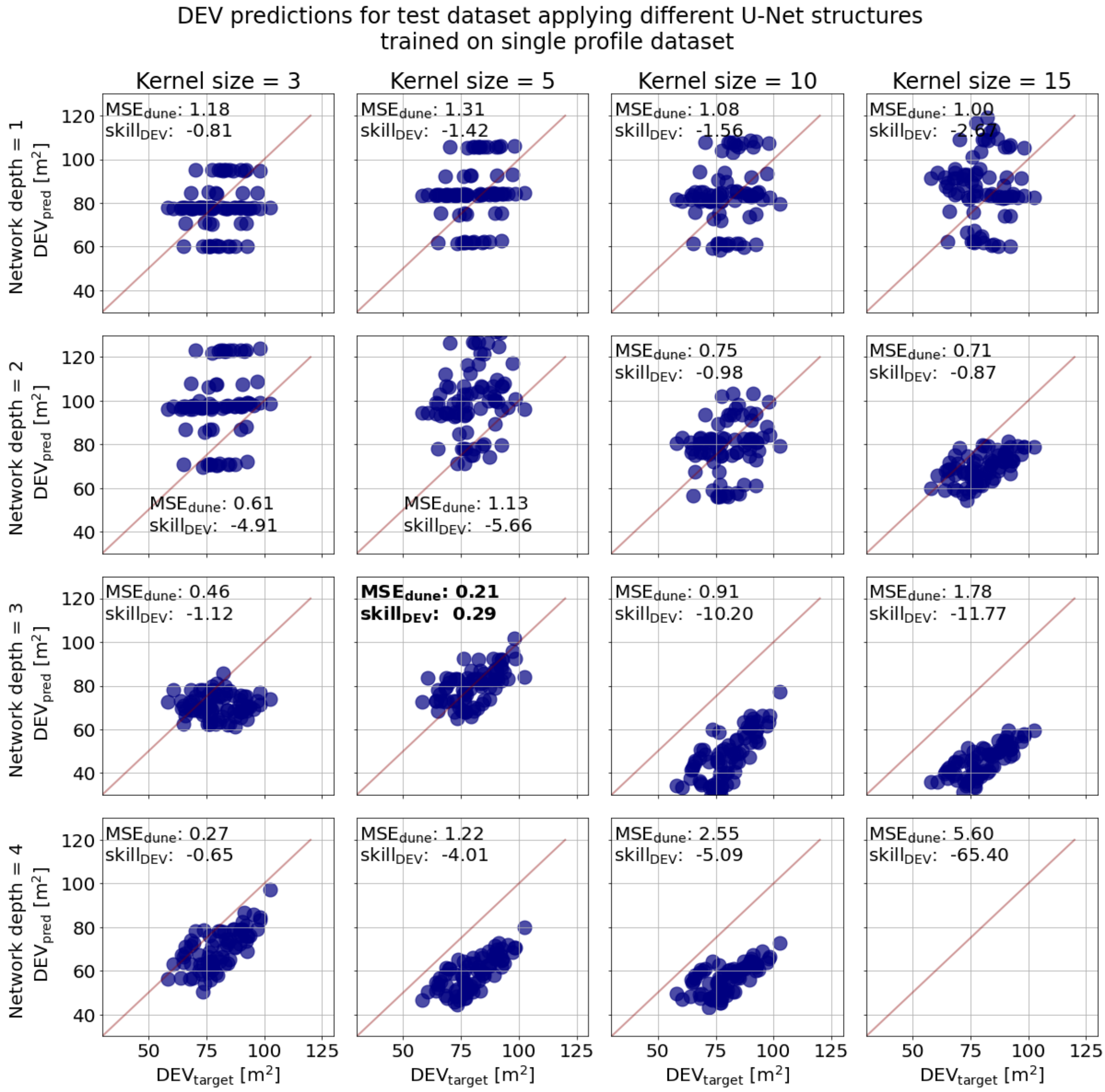


Figure 4.18: Predicted DEVs vs target DEVs for different U-Net structures trained on a single-profile-based dataset. The best-performing network structure is highlighted in bold.

Data quantity

As addressed in Figure 3.2.1, the surrogate model will be tested for several quantities of profiles. A subset of the training data is taken based on a certain percentage. The results for this analysis are presented in Figure 4.19. This graph indicates that using 100 to 200 profiles is enough for convergence of the surrogate model towards a comparable performance as with the full dataset. This is about 20 - 40% of the original dataset.

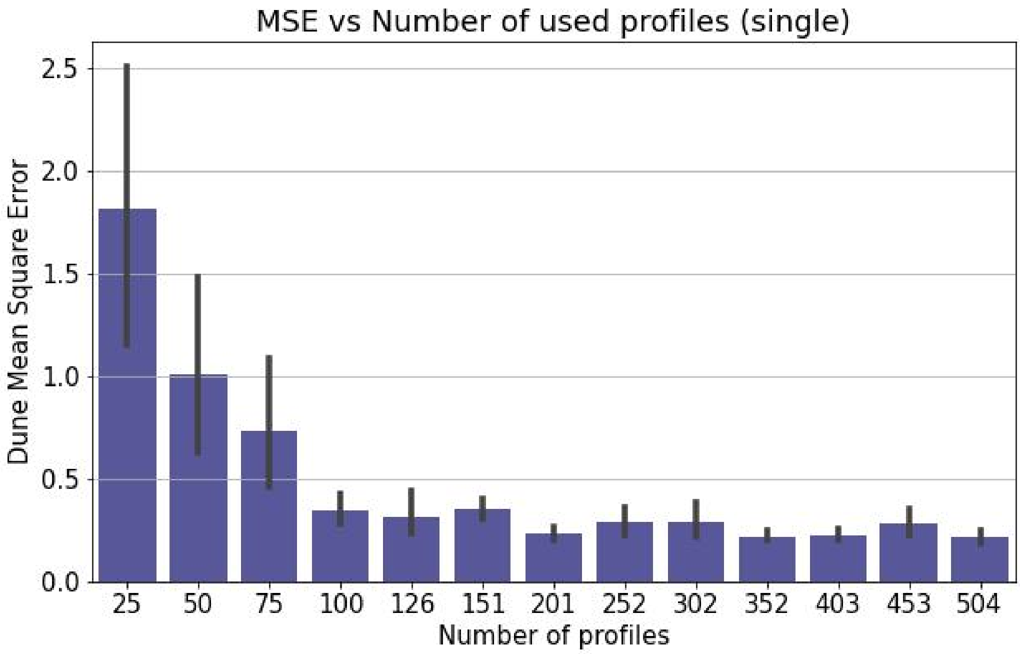


Figure 4.19: Amount of profiles in train dataset and obtained MSE error

4.2.2. Upscaling

This section describes the results that were found in the upscaling phase of this project. To be able to judge the performance of the model in a realistic situation and varying input space, a test dataset with actual profiles along the Holland Coast is used (Appendix F).
To set up a realistic training dataset and varying input space, the original profile of the dataset is expanded from 1 to 4 profiles. These 4 profiles (a single profile from the exploration phase and 3 additional ones) along the Holland coast are illustrated in section D.2. These profiles are again modified with the same modifications as described in subsection 3.1.1. Resulting in a new set of 2000 profiles, with a mean DEV of 49.6 m^2 and DEV standard deviation of 18.0 m^2 . The specifics of this dataset are illustrated in Figure 4.20.

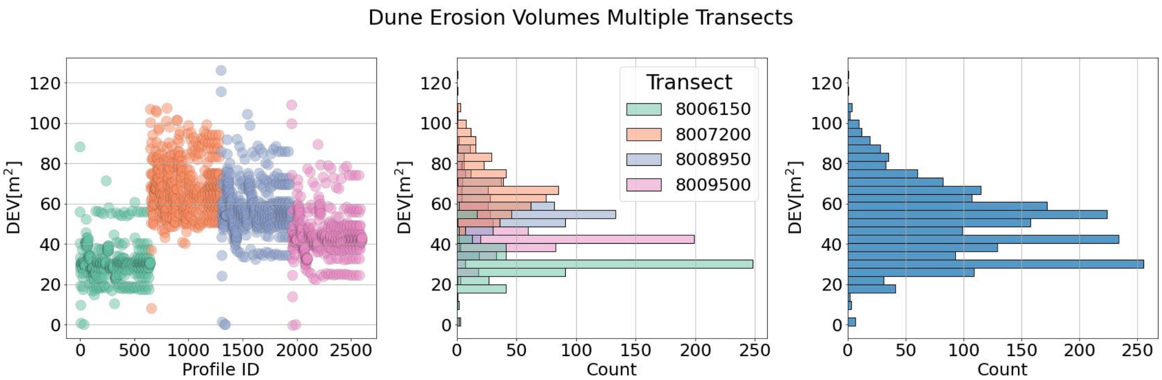


Figure 4.20: The DEV calculated for the enlarged dataset. Each color represents a different original profile.

As presented in Appendix E, overfitting issues arise when 2000 profiles are used in the training dataset. To mitigate these overfitting concerns, the sampling density of the modification factors is reduced. Consequently, an updated set of modification factors is generated, resulting in a dataset comprising 404 input profiles. Further details regarding this dataset are presented in Figure 4.21.

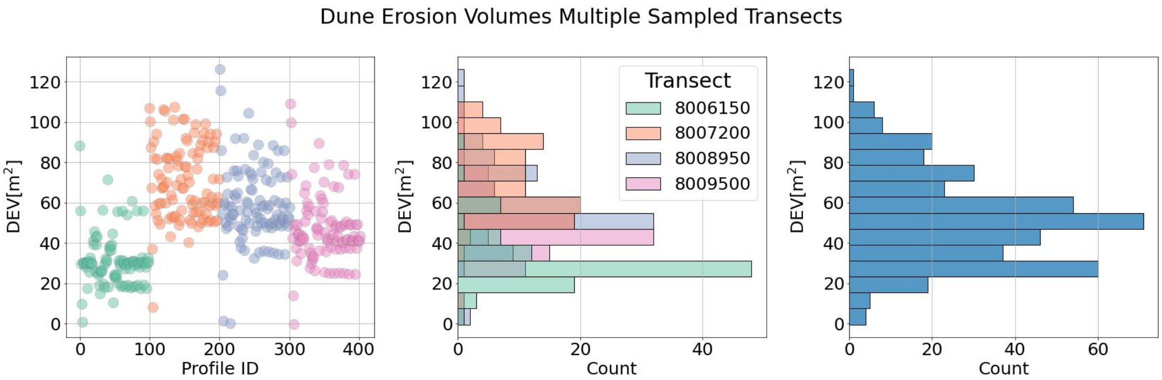


Figure 4.21: The DEV calculated for the enlarged dataset. Each color represents a different original profile.

Single- vs Multi-profile-based training dataset

This section presents the performance of the surrogate model when trained on the single- or multi-profile-based dataset for various network depths, widths (first channel size) and kernel sizes. Both these analyses will be carried out based on the MSE_{dune} and the $skill_{DEV}$ to indicate both the performance and applicability of the U-Net structure.

Network depth and network width

Figure 4.22 shows the MSE on the dune for the single- and multi-profile-based training dataset. For the dataset with a single profile, a network depth of 3 or 4 layers yield the best results. For a network depth of 2, a reduction in errors for increasing channel sizes can be observed. However, for deep networks and large channel sizes, this trend disappears and the performance for larger channels decreases. For a network depth of 4, the performance of the surrogate model blows up for large channels. This also goes for the dataset with multiple profiles. Although, the MSE for different network depths seems to be more stable, this blow-up already occurs at a network depth of 3. The best results is obtained for a network depth of 4 and channel size of 32.

Figure 4.23 illustrates the skill of the U-Net in predicting DEVs (a skill of 1 indicates the best performance). The y-axis for the skill is ranging from -1 to 1, models with a skill below -1 do not show any significant skill. It can be seen that the skill improves for deeper networks and even some network architectures incidentally reach a skill above 0. While showing a similar trend, the skill for the multiple profile dataset does frequently reach above 0. Especially deep networks with smaller channel sizes perform relatively favourable. With this analysis the predictive capability between the two different training datasets becomes quite clear. The higher variability in the training data becomes more applicable for this realistic test data.

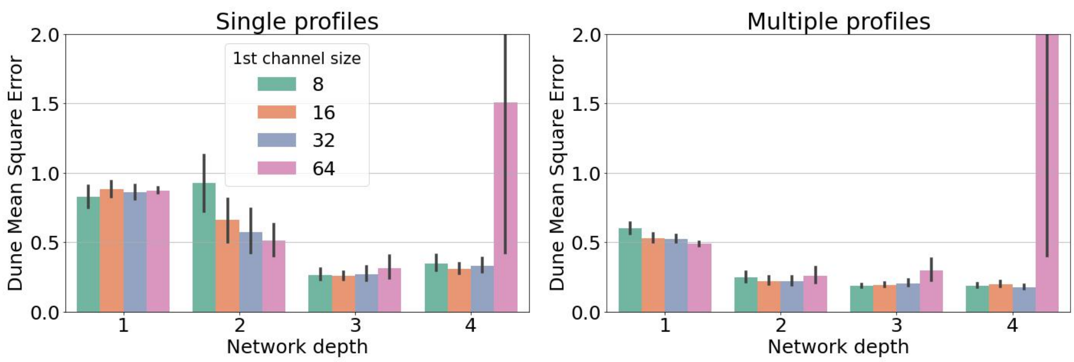


Figure 4.22: Depth and width of U-Net architecture vs MSE_{dune} for a single- and multi-profile-based training dataset.

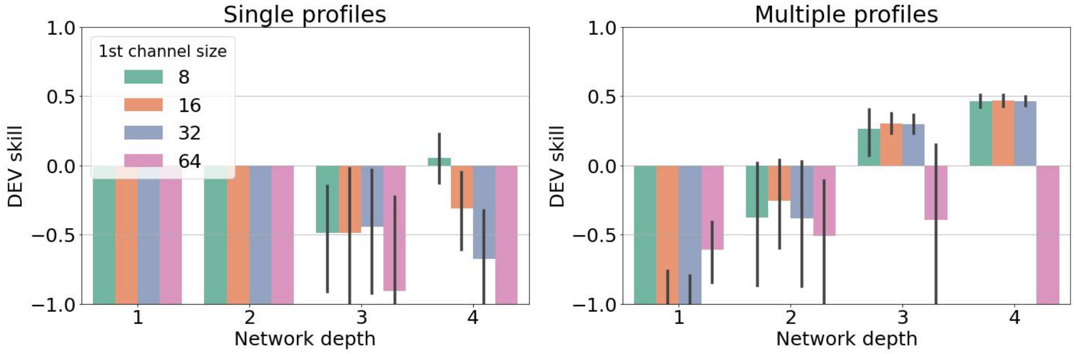


Figure 4.23: Depth and width of U-Net architecture vs $skill_{DEV}$ for a single- and multi-profile-based training dataset.

Network depth and kernel size

As addressed in Appendix E, kernel size can be an important parameter for the interpretation of profile shapes and resulting dune erosion. While this was not found for the previous test data, the results for realistic test case do show significant trends.

All in all, the dataset with multiple profiles yields better results compared to the single-profile-based dataset. The obtained error (Figure 4.24) is smaller and skill (Figure 4.25) is higher for the multi-profile dataset. Studying the effect of different kernel sizes, a larger kernel size yields positive results. For both training datasets, the larger kernel size of 10 outperforms the kernel size of 3. This difference is especially pronounced for shallow networks of 2 layers. When examining the skill for deeper networks (3 or 4 layers), it is shown that larger kernels are also beneficial for deeper networks, but the gain is less pronounced. Interestingly, the same skill level is achieved for a small kernel size in the 4-layer model as for a large kernel size in 3-layer model.

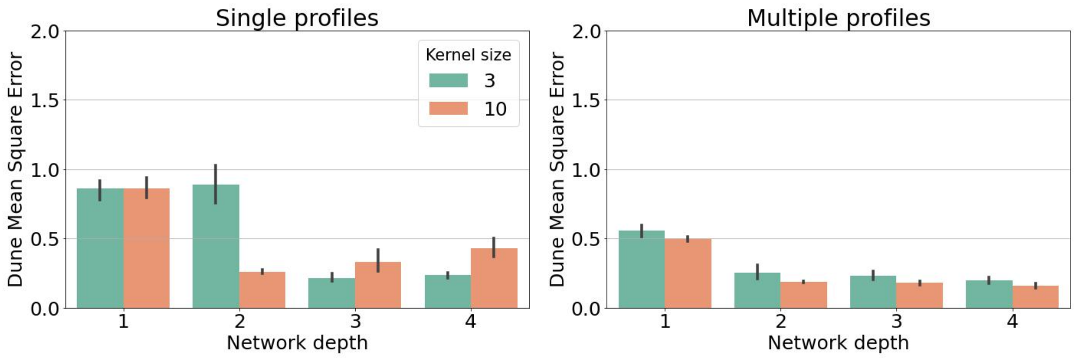


Figure 4.24: Depth and width of U-Net architecture vs MSE_{dune} for a single- and multi-profile-based training dataset. The presented results are carried out for a constant first channel size of 32.

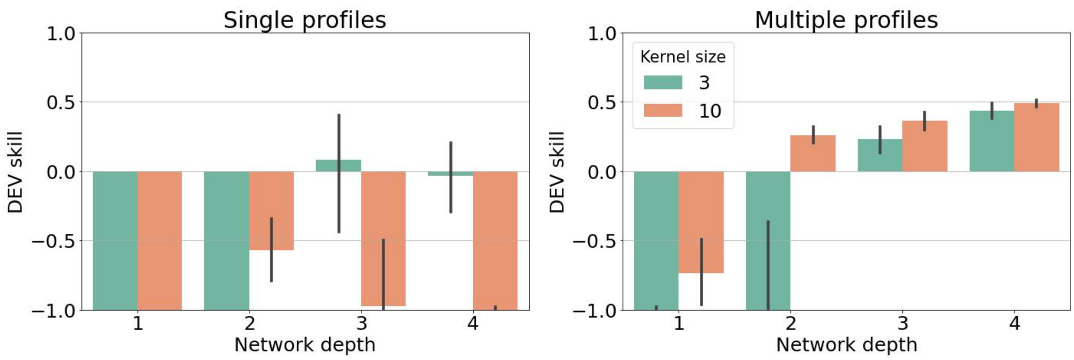


Figure 4.25: Depth and kernel size of U-Net architecture vs $skill_{DEV}$ for a single- and multi-profile-based training dataset. The presented results are carried out for a constant first channel size of 32.

U-Net structures

Given a training dataset with multiple profiles improves the model accuracy and skill, more detailed analyses can be carried out. This is initiated with similar analyses as presented earlier in Figure 4.18, but for the multi-profile training dataset and realistic test data. Subsequently, a visual inspection is carried out on the predicted post-storm profile shapes as a result of the induced network changes.

Figure 4.26 presents the predicted DEV vs the target DEV for several network structures. In general, as U-Net structures transition from shallow to deep, the quality of the predictions is increased. Especially for small kernel sizes this behaviour is quite clear, while for larger kernel size the trend is less pronounced. Besides that, the increase in kernel size is most beneficial for shallow networks.

The bottom right plot shows a complete collapse of the predictive capability of the surrogate model. Using a network depth of 4 and kernel size of 10, the input dimension is significantly reduced, resulting in inadequate predictive capability. The interpretation of these results is discussed in subsection 5.2.2.

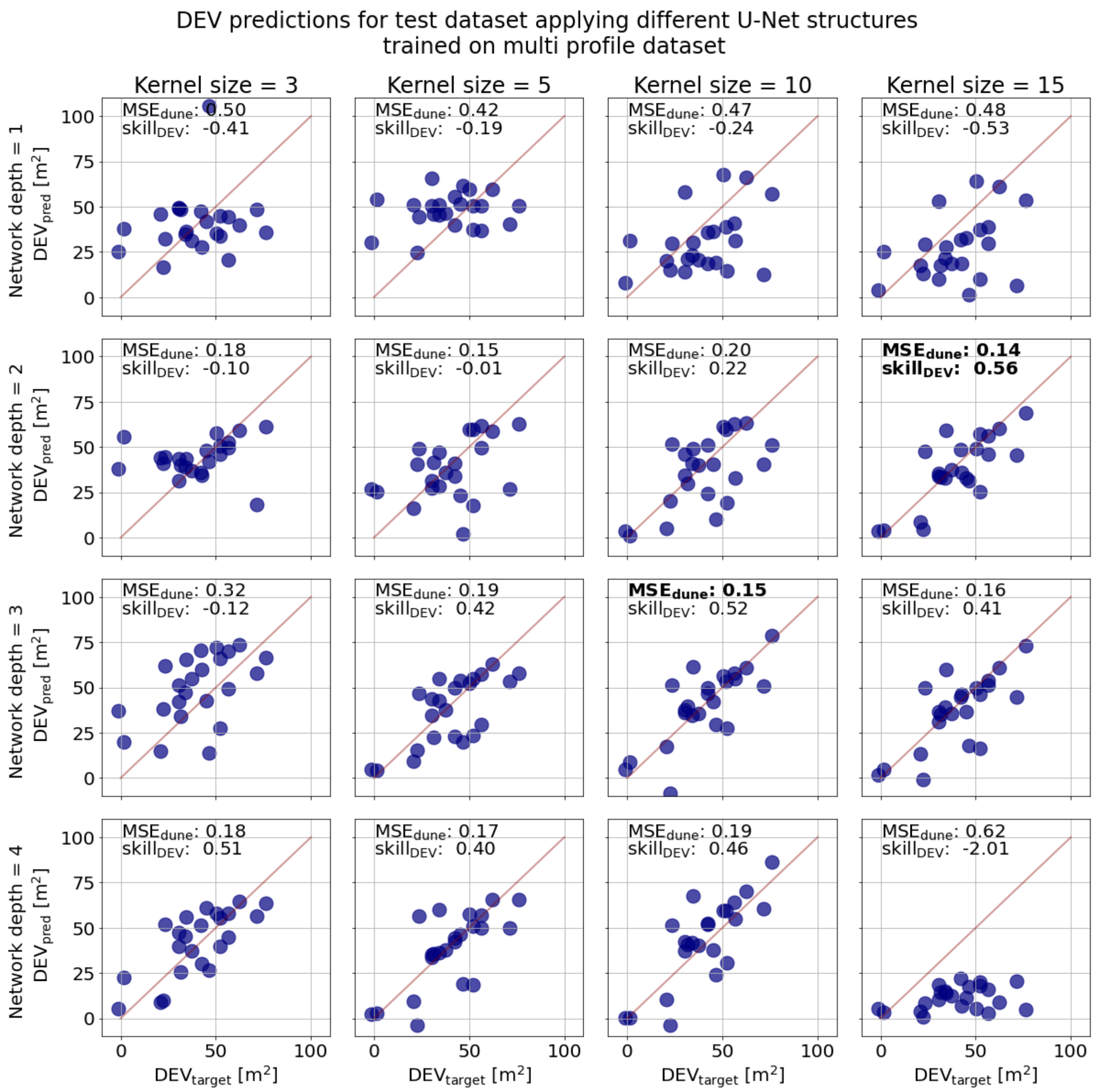


Figure 4.26: Predicted DEV vs the target DEV for several network structures. Increasing network depth over the y-axis and increaing kernel size over the x-axis. The MSE_{dune} and $skill_{DEV}$ are printed in each figure and the best performing structure are printed in bold.

Network depth

In Figure 4.27, the effects of increasing the network depth is illustrated for a single profile. In this case, the deepening of the U-Net architecture leads to a reduction of the error at the dune. From left to right, the post-storm shape profile is simulated more accurately.

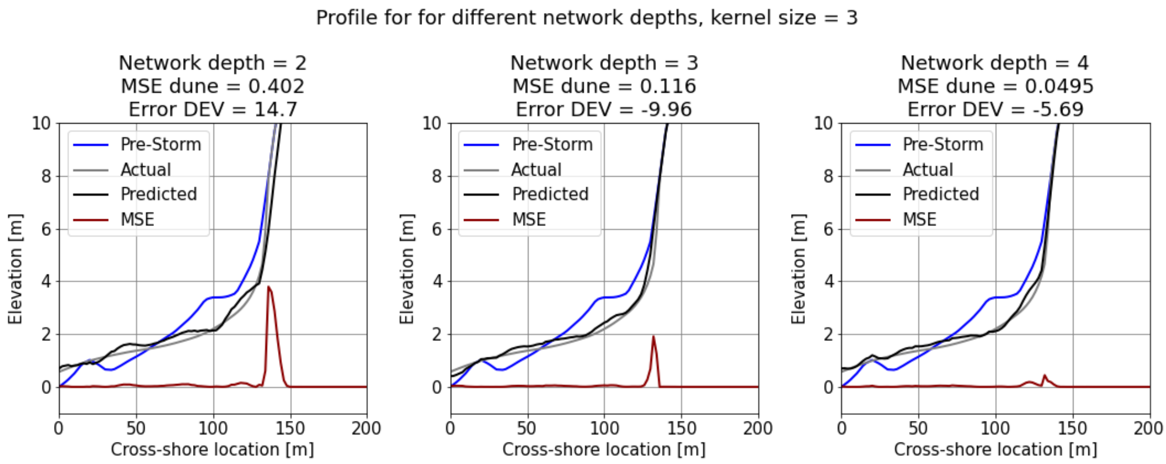


Figure 4.27: Predicted and target differences and dune shapes for profile in test data

For a large kernel size (Figure 4.28), the error also reduces over the course of deeper networks. However, examining the shape of the dune, the alignment with the actual post-storm profile is worse for a deep network. The prediction shows a small distortion at the eroded duneface. This distortion has the shape of the pre-storm profile.

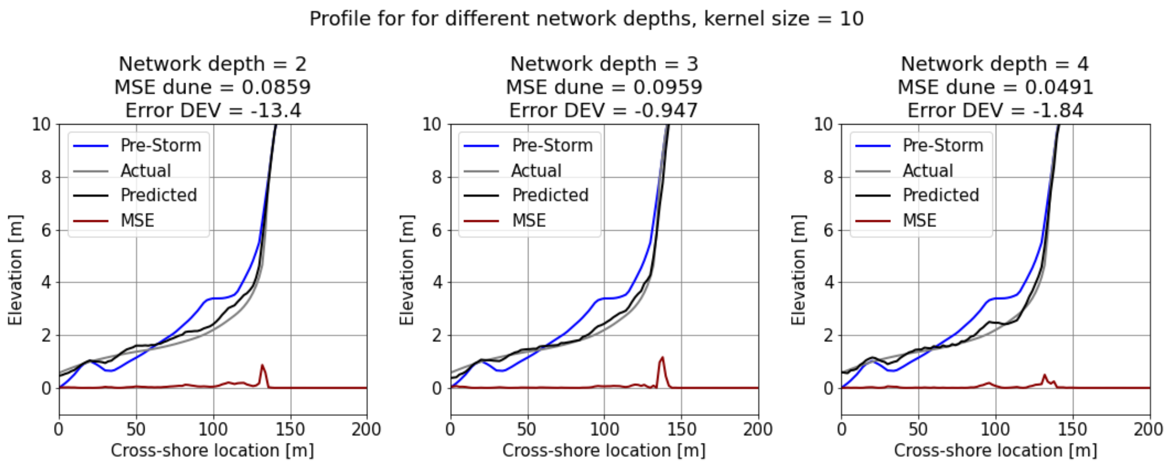


Figure 4.28: Predicted and target differences and dune shapes for profile in test data

Kernel size

For this analysis, the results of two kernel sizes are presented. Figure 4.29 shows that an expansion of the kernel size results to a better fit of the post-storm profile. For a shallow and wide network (network depth = 2, channel size = 64), the predicted profile becomes smoother for larger kernel sizes. Note that the error that occurs for the calculation of the DEV does not represent the quality of the shape. For the left plot, the shape of dune overestimates the amount of erosion for higher elevations, but underestimates for lower elevations on the dune. Because these volumes cancel each other out, this is not found in the DEV error statistic.

The opposite trend is observed when a deep network (network depth = 4, channel size = 32) is applied (Figure 4.30). While the goodness of the fit increases, while the shape seems to follow the shape of the input-profile. The distinct post-storm profile shape as computed by XBeach is not present.

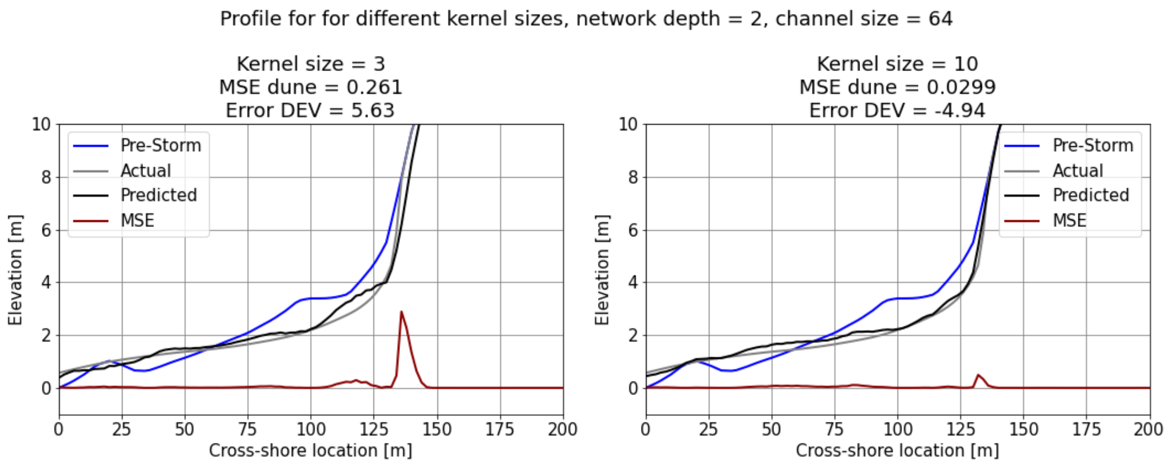


Figure 4.29: Predicted and target differences and dune shapes for profile in test data

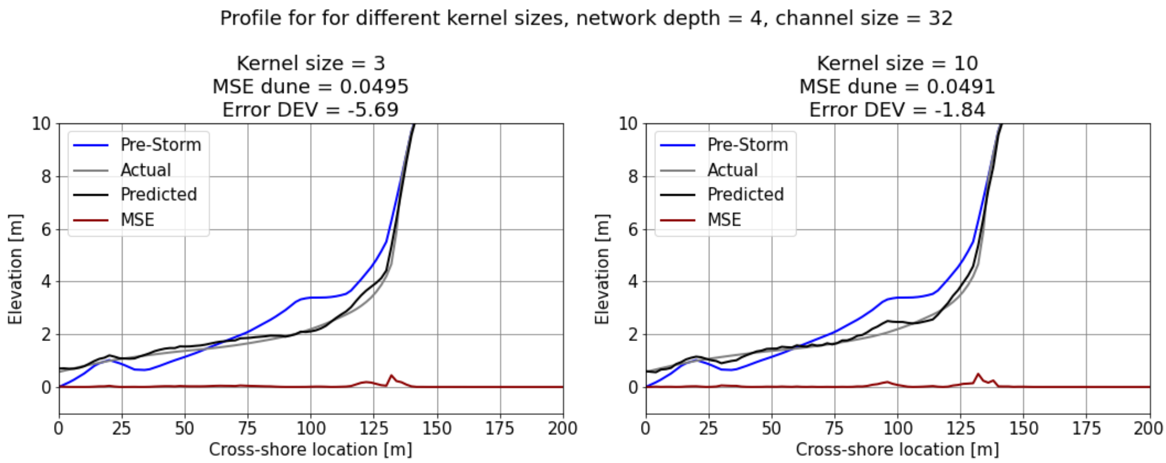


Figure 4.30: Predicted and target differences and dune shapes for profile in test data

Channel size

Altering the channel size does not lead to a significant change to the general appearance of the predicted post-storm profile. In general, the influence of the channel size depends on the depth of network. As shown in Figure 4.31, the error at the duneface can be reduced for shallow networks (network depth = 2, kernel size = 10). While for deeper networks (Figure 4.32), the error on the duneface increases for larger channel sizes.

The common shape of the post-storm profile prediction does not change too much and follows the shape of the pre-storm profile. Next to that, for large channel sizes the predictive capability of U-Net vanishes. The results blow up and the predicted pre-storm profile hardly differs from the pre-storm profile. For all predictions in a deep network and large kernel size, the predicted post-storm profile shape has similar features as the pre-storm profile shape.

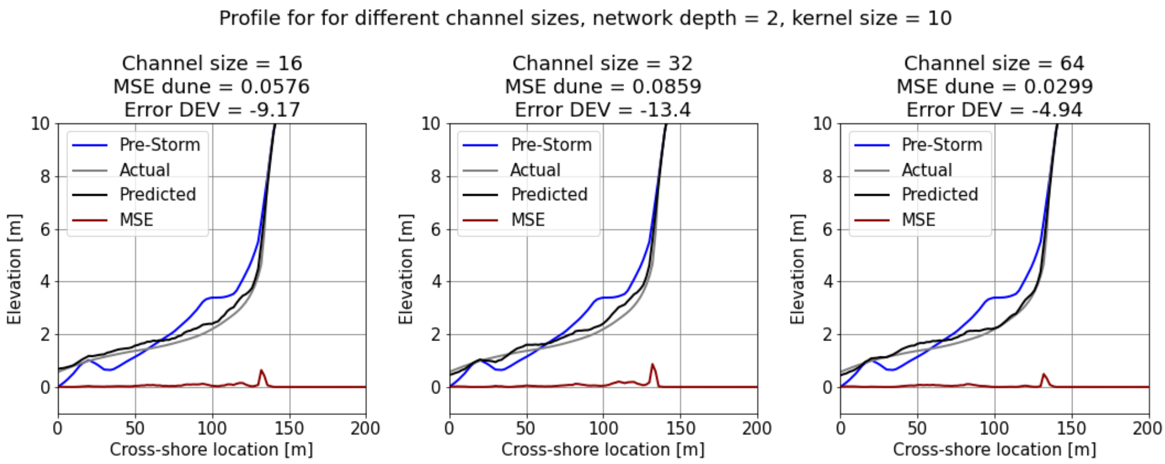


Figure 4.31: Predicted and target differences and dune shapes for profile in test data

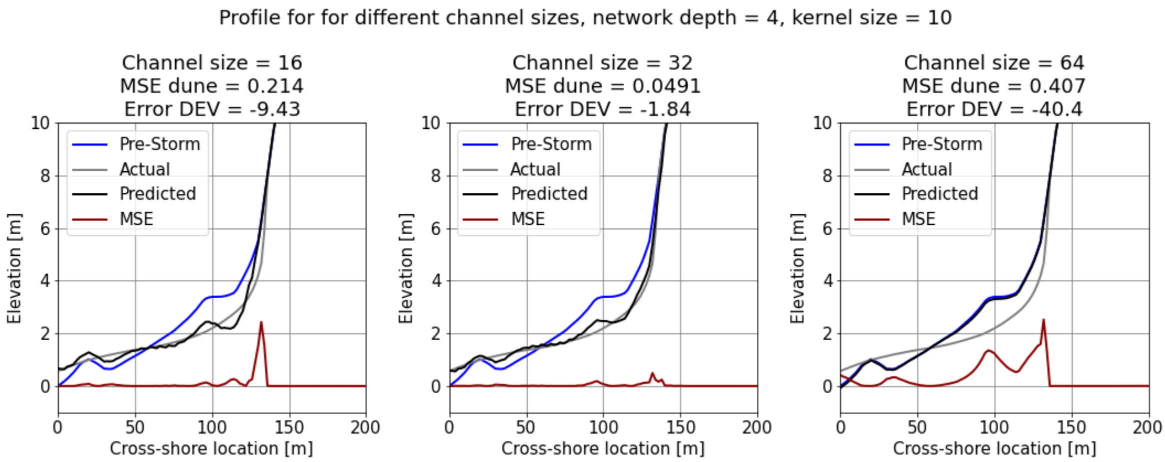


Figure 4.32: Predicted and target differences and dune shapes for profile in test data

Note: These results are an example of how the U-Net architecture affects the predicted profiles. Visual inspection of other profiles led to similar overall trends, but each has its complications concerning the influence of the pre-storm profile shape.

Model performance for selected network structure

To assess the performance of U-Net in modelling dune erosion processes, a specific U-Net structure is selected. Based on findings from the previous section, a surrogate model using a U-Net structure with a **depth of 3 layers, first channel size of 32 and kernel size of 10** is set up. Initially, the capability of U-Net to capture dune erosion processes is assessed. Subsequently, some specific profiles are highlighted in more detail.

Dune erosion processes

Judging from Figure 4.33, U-Net is able to mimic the positive relationship between the beach slope of the input profiles and the resulting dune erosion. Although errors occur for individual profiles, the trend between the target and prediction shows similarities. In fact, both the XBeach predictions and U-Net predictions show an increase in DEV for steeper beaches. This relation is also highlighted in the parameter sensitivity analyses (subsection 4.1.3). It is remarkable to observe that the relationship between beach slope and DEV seems to be more pronounced for the U-Net prediction than for the XBeach target. This is confirmed by the computed pearson coefficients (ρ).

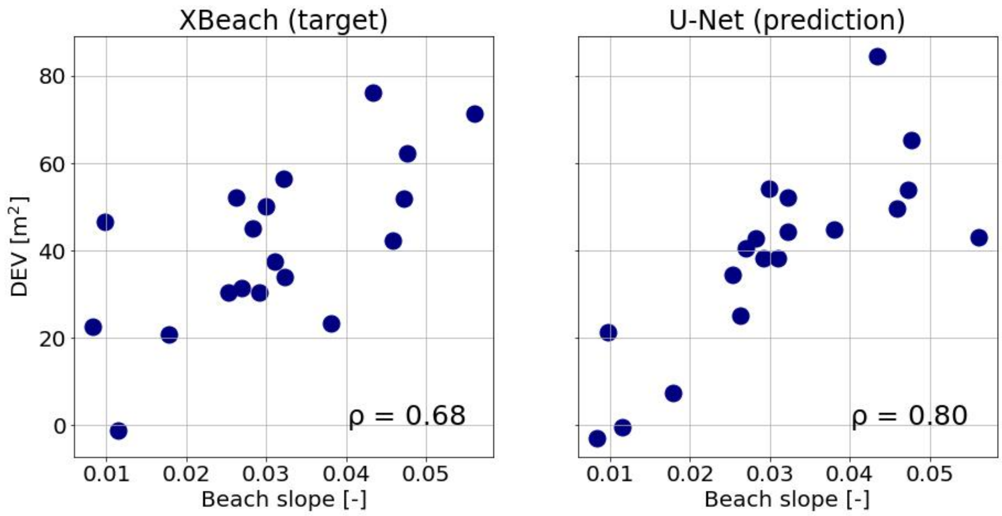


Figure 4.33: The target DEV using XBeach (left) for different beach slopes and predicted DEV using a surrogate model (U-Net) for different beach slopes. The strength of the relationship between the beach slope and DEV is expressed by the pearson correlation coefficient (ρ)

Predicted profiles
The results for all profiles in the test dataset are highlighted in Appendix I. Some predictions worth mentioning are presented in this section. Briefly, a relatively good profile fit is followed by two poor predictions (over- and underprediction).

For the good prediction (Figure 4.34), the post-storm difference is approached relatively accurate. Although the prediction is oscillating a little, it has a similar shape and amplitude as the actual difference. Although, unlike the training data, the dune area has multiple features behind the eroded area, the shape of the post-storm profile is predicted accurately.

A profile for which an underprediction of the DEV was found is illustrated in Figure 4.35. This can already be seen in the illustration of the difference. The prediction is constantly lower than the actual values. Analysing the actual elevation plots, it can be seen that U-Net falsely projects the pre-storm shape of the dune on the post-storm profile. In this case, this is a small elevation difference at the end of the erosion areas. This small alternation on the foredune leads to an overestimation of the sediment that remains on the dune after the storm. In Figure 4.36, an overprediction of the DEV is presented. The impact of the storm is overpredicted and the dune shape is affected too severely. This results in a large MSE at the dune and a mismatch of the actual and predicted profile shapes. These results are discussed in subsection 5.2.4.

Good fit

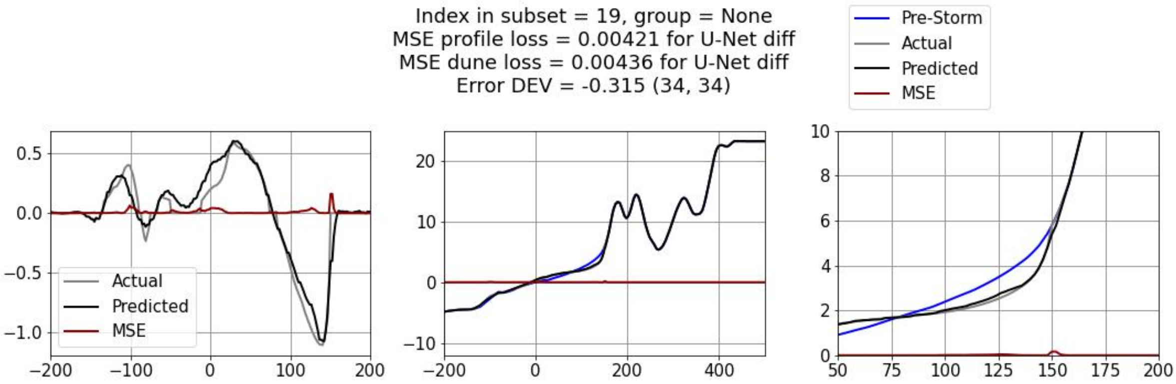


Figure 4.34: Good prediction of DEV: Predicted and target differences and dune shapes for profile in test data

Poor fit, underprediction

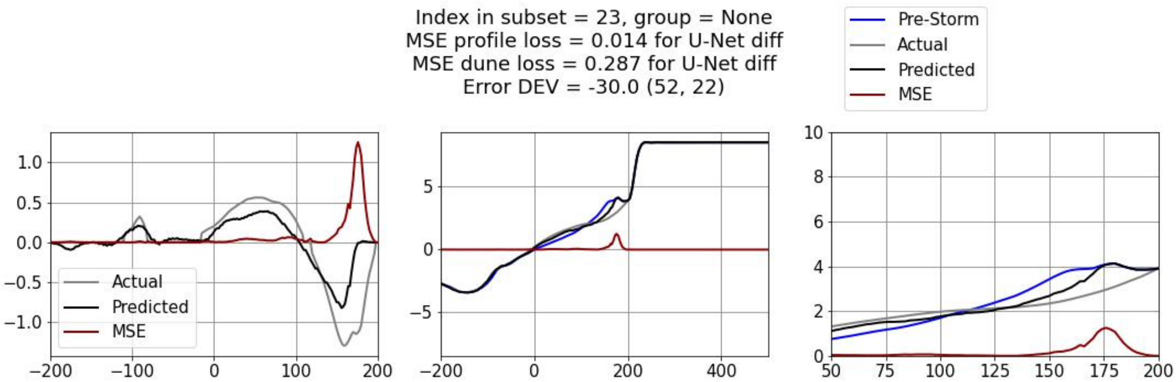


Figure 4.35: Underprediction of DEV: Predicted and target differences and dune shapes for profile in test data

Poor fit, overprediction

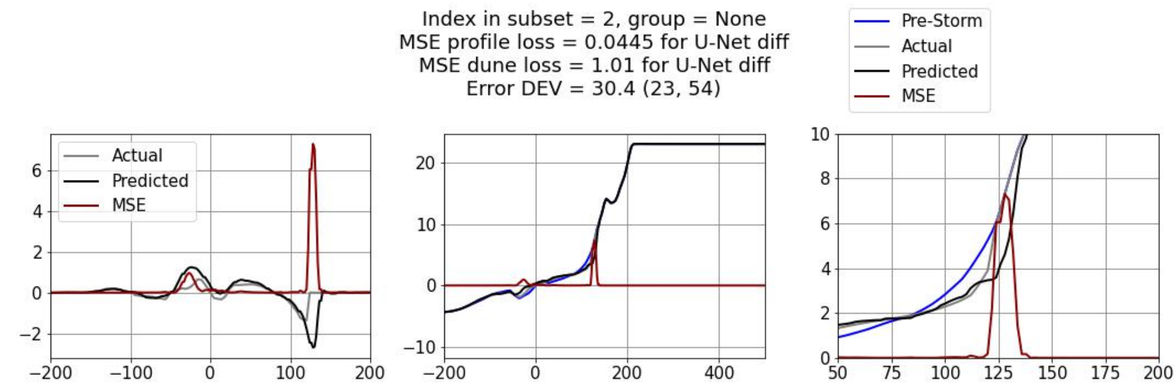


Figure 4.36: Overprediction of DEV: Predicted and target differences and dune shapes for profile in test data

5 Discussion

5.1. XBeach

By isolating the modifications to the original profile and analysing modelled DEV, the expected erosion response was recorded. In general, steep slopes in submerged profile sections lead to larger erosion quantities due to decreased dissipation of wave heights. This effect was recorded on a large range for the beach section. Within this large range, a decay in the impact of the storm for gentle slopes can be found. This trend was also found in literature (subsection 2.1.2).

All modelled post-storm profiles in the simplified dataset remain in the collision regime. Therefore, dune erosion does not reach above the dunecrest and the modifications of the dunecrest height do not influence the DEV. The induced modifications yield an appropriate variability of DEV to train the neural network. Note that the dunecrest height is a large signal in the input data (Figure D.2), but that this hardly influences the modelled DEV. For this project, no issues were raised, but for future research this should be treated with care to prevent fitting of the model to input without a change of signal in the output.

As presented in Figure 3.8, the endpoints method for slope calculation can result in an oversimplification of the calculated slope for a profile section. While giving a suitable indication of the slope this should be taken into account when assessing the results. Especially for the duneface, due to its non-linear trend, significantly higher values could be observed than indicated by the endpoints method.

Throughout this research, the storm conditions remained unchanged. These conditions are assumed stationary over the full duration of the storm. No variation in tide and build-up of the surge was modelled. This highly simplified situation would never occur at the actual Holland coast system. Using realistic storm shapes would result in different temporal evolution of the dune erosion pattern and the modelled post-storm difference would likely be less drastic compared to stationary conditions. As for the essence of this research, predicting post-storm profile shapes, realistic storm shapes (results as long as a certain signal is present) would not influence the presented results.

5.2. Surrogate modelling

5.2.1. Scale of coastal erosion processes

In the exploration phase of this research it became clear that the scale of the input (bed elevation) and the modelled signal (sediment transport) are from different orders of magnitude. While the bed elevation reaches from -30 m to 20 m for several full input profiles, the signal of the observed erosion at the dune is on average about 5 m. Therefore, detecting this signal poses a challenge for a U-Net architecture and it needs assistance to identify these dune erosion processes. This is achieved through several techniques.

The first technique issued in this report is the standardization of the cross-shore grid. This grid should solely include the areas of interest of the modeller. This can be obtained by inducing a minimal elevation (z_{min}) that limits the considered range of the input. Next to that, the input profiles can be fixed on a certain elevation (z_{fix}). This isolates the location of the erosion processes at a certain point in the 1D input and helps U-Net to interpret the modelled changes.

Next to standardization of the cross-shore grid, the input data can also be pre-processed. As shown in Figure 4.5, training U-Net to predict the difference between the pre- and post-storm profiles instead of the absolute post-storm elevation is beneficial for the performance of the surrogate model. Through this technique, the signal of erosion processes becomes more detectable for U-Net. Due to the large differences in dynamic areas of the profile, a natural weight is present at these profiles section. The mean square error computed for this area is significantly higher and is recognized by U-Net.

This weight can also be forced upon the input by applying a mask over the input. This, however, only shows an increase in performance when this weight is not present in the original input. Therefore, the gain in performance for the difference model is little. On the contrary, for normalized elevation data, the natural weight in the difference is not present. This facilitates the application of forced weights on the input and enables improvement of the model performance in the areas of interest.

5.2.2. U-Net structure complexity

In general, the complexity of the U-Net structure should align with the complexity captured in the training data. As indicated by the results, the complexity of the U-Net structure can be derived from the resolution of the input, network depth, network width and kernel size. The complexity captured in the data relies on the alongshore variability that is captured in the training data. These differ between the exploration- (low complexity) and upscaling phase (high complexity). This section discusses the results for the network structure. subsection 5.2.3 issues the results with respect to the applied data.

The resolution of the input determines the number of datapoints present in the 1D input. For pre- and post-storm profiles this is accounted for by minimum elevation (z_{min}) and the spacing (Δx) in the input grid. For the spacing, the obtained dune shapes for coarse grids (Δx = 10 and Δx = 5) were simply too crude to correctly extract the erosion processes. When applying a fine grid size of Δx = 1 (grid size used in XBeach), the results show overfitting issues. Especially when smaller elevation ranges (higher z_{min}), U-Net post-storm profile predictions start showing oscillations.

Eventually, a grid spacing of 2 m and minimum elevation of -5 m was employed to standardize the data. This grid is not re-evaluated for new U-Net alternatives in later stages of the research. Using a larger variability of input profiles in the training data, might require a more flexible grid standardization. For example, cross-shore profiles that do not comply with the standard Holland Coast profile outlook would require more detail. This should be taken into account when proceeding with the U-Net architecture.

Exploration

Based on results presented in Figure 4.10 and Figure 4.14 it can be stated that the network depth is a crucial hyperparameter for the performance of U-Net. Especially to reduce the MSE on the dune section of the profile for the exploration training and test dataset, a relatively shallow network is favourable. The complexity captured in this training/test dataset does not require deep U-Net architecture. The combination of the data and network structure does lead to a simplification of the training process and physically undesired profile predictions Figure 4.16.

Another parameter to tune the complexity of the model is the width (channel size) of the U-Net structure. Figure 4.14 illustrates that this feature shows contrasting trends for different network depths. Whereas more channels seem favourable for shallow networks, it seems disadvantageous for deeper channels. The channel size could be interpreted as an extra stimulus for the extracted complexity. This is beneficial for shallow networks, but not needed for deeper architectures. It is important to stress that the network depth and width both influence the amount of features extracted from the data and should be balanced out according to the format of the inputted data.

Exploration: Shallow U-Net structure

In the exploration phase, when studying the applicability of the U-Net profile predictions by extracting the DEV, the statistical interpretation of U-Net and lack of physical knowledge becomes clear. Judging from Figure 4.16, the predicted DEV and mean errors on the dune can be categorized based on dune crest height. However, as presented in subsection 4.1.3, the dune erosion processes do not rely on the height of the dune. These results suggest that U-Net is incorrectly interpreting the single-profile training and test data and not linking the input data to the modelled erosion processes.

Instead, a 2-layer U-Net model is completely relying on statistics when finding relationships between the input and output. For a single profile dataset and a shallow network depth U-Net is able to predict one single profile and scale that to the supplied input profiles. Through a shallow network, the U-Net structure is unable to simulate dune erosion processes and attempts to find the statistically optimal solution. Due to the similar shape of the post-storm dune, this shape is approached relatively good.

Exploration: Deep and increased complexity U-Net structure

As discussed before, shallow U-Net architectures are good at interpreting local shape changes of the dune, but lack the capability to link these local changes to pre-storm profile characteristics. This issue raises questions about the adjustments that could be made to the U-Net architecture and improves the capabilities to simulate coastal processes. As presented in the results (Figure 4.18), this can be achieved by increasing the network depth and kernel size.

Increasing the network depth allows for the extraction of high-level features. Especially in deeper parts of the CNN, the original spatial structure is abandoned and features are able to interact. For example, in case of a pre-storm profile, for the features found nearshore to be able to interact with the erosion processes at the dune a sufficient network depth and kernel size should be provided. In case of a shallow network (network depth = 2 and kernel size = 3), this capability of CNN is not provided (Figure 4.17).

Increasing the kernel size is particularly beneficial for shallow networks. As explained in subsection 2.2.2, a convolutional layer extracts several features from the input with a certain range of pixels (kernel size). Throughout the depth of the network, extracted features start to interact. With fewer layers, the receptive field of each layer remains limited even when using larger kernel sizes. This helps in preserving finer spatial details and allows for interactions between those. For deeper network structures this changes. Large kernels result in an increase in the receptive field and potential loss of fine-grained spatial details.

Reflecting on these insights for the profile prediction for one single profile, trends found in the optimization results (Figure 4.2.2 and Figure 4.2.2) are confirmed. Deeper network and large kernel sizes can reduce the computed error for post-storm profile prediction. Together, however, they lead to incorrect post-storm profile shapes. Whereas, the kernel size results in an improvement in performance for shallow networks, this effect is smaller for deeper networks. The kernel size should therefore be treated with great care.

Upscaling

The issues raised during the exploration are confirmed during the upscaling phase of this research. The ability to correctly interpret dune erosion processes in pre- and post-storm profiles comes in for more complex U-Net structures. This claim has been confirmed through analyses on the training data during the exploration phase, as illustrated in Appendix H (Figure H.7). The obtained skill and accuracy further validate this claim for the realistic test data.

As observed in Figure 4.26, increasing the complexity of the U-Net structure, characterized by the network depth, network width, and kernel size improves the performance of the surrogate model. Shallow networks benefit more from larger kernel sizes than channel sizes than deep networks.

The improvement of the performance is limited to a certain complexity. As discussed before, adding too many parameters in the U-Net architecture leads to diminished prediction capability. U-Net starts overfitting and oscillations occur at the predicted profiles. Besides that, for deep networks with very large kernels (bottom right in Figure 4.26), the surrogate model prediction takes the original shape of the pre-storm profile. This confirms the discussion points presented in the previous section on large kernel sizes for shallow and deep network structures.

5.2.3. Training and test data

Exploration

For the exploration phase of this research, the single-profile-based test and training dataset have shown their potential to explore the possibilities for post-storm profile predictions. Insights were gained about the driving mechanisms of dune erosion. Subsequently, these insights were used to evaluate the performance of U-Net and its ability to replicate coastal processes.

The understanding gained about the influence of U-Net structure on the simulating dune erosion processes (subsection 5.2.2) does not correspond with the results gained for the test case in the exploration phase. When testing on a dataset which is based on a single profile, shallow networks are favourable. Projecting these results on the training data, U-Net is able to mimic coastal processes better for larger network depth and kernel sizes, but the bias on the test dataset predictions is increased.

Due to the lack of alongshore variability in profiles, the process captured in the training dataset are 1D. Therefore, the optimal post-storm profile is a single profile scaled by means of the pre-storm input profile. Adjusting the network architecture allows for the interpretation of the training data by U-Net, but does not show improved performance for the test data. This can be explained by the fact that the test data in the exploration phase does not capture the same variability in the input space as present in the training dataset. The accuracy for the test dataset is better represented by a shallow network, but for upscaling purposes, this test data and performance metric do not give an adequate performance indication.

Upscaling

In the upscaling phases of this research, an enlarged training dataset based on four different profiles is set up. This introduces a cross-shore diversity and a larger range of erosion events. Next to that, the performance of the surrogate model is tested on a dataset of actual Holland Coast profiles and two performance indicators (MSE_{dune} and $skill_{DEV}$).

Using these training and test cases, the performance of the surrogate model showed significant improvement compared to the single-profile-based training dataset. The potential of deeper networks and increased kernel size are also reflected upon the test data and have a positive influence on the U-Net predictions.

To prevent overfitting and reduce the size of the training dataset, the originally obtained multi-profile training data was sampled on a smaller density to reduce the size of the dataset (Appendix E). This was accomplished by excluding input profiles that were generated using specific modification factors. Hence, the resulting sampled dataset consisted of fewer modifications per original profile. This method of sampling could be replaced by clustering techniques. Techniques, such as the Maximum Dissimilarity Analysis (MDA) and k-means, can be used to reduce the size of an original dataset by generating a representative subset. This subset consists of profiles, which embody a cluster of profiles within the original dataset. This way, the size of the dataset is reduced while keeping the variability of the input space intact. Using this technique, a smaller training dataset could be used to capture the same (or greater) variability.

The difference between the exploration- and upscaling phase is illustrated in Figure 5.1. The applied test data directly influences the required network complexity of U-Net.

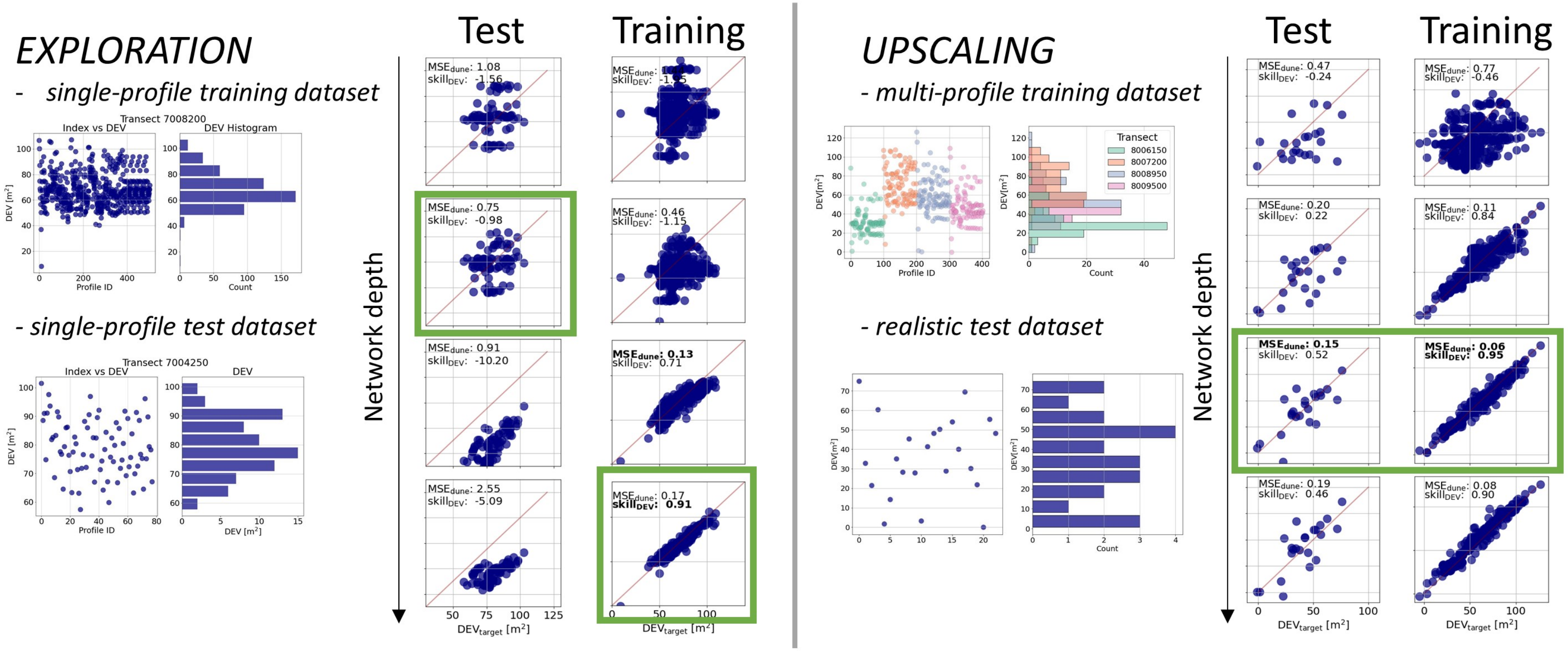


Figure 5.1: Overview of the required network depth of U-Net for the exploration and upscaling phase. In the exploration phase, the test data is an incorrect representation of the training data. Resulting in the requirement of shallow network depths and a lack of predictive capability. In the upscaling phase, a better understanding of the training data by U-Net is reflected upon the test data. This confirms the need for a deeper network structure to capture coastal processes with U-Net. The favourable network depth for both the training and test data is indicated by a green square.

5.2.4. Profile shape prediction

Findings in the exploration phase revealed the feasibility of predicting profile shapes using U-Net architectures. While this does not directly measure the predictive skill of the network, it demonstrates the potential to predict post-storm profile shapes based on the pre-storm profiles. The correlation between the elevation datapoints and high input resolution that come with pre-storm profile input, is captured correctly by the convolutional neural network. Using skip-connections proves advantageous in preventing oscillations and reducing errors on the dune. In general, post-storm profile predictions are of better quality when high-resolution information from the encoder pathway is transferred to low-resolution information from the decoder pathway. This preserves the original spatial information captured in the input data. However, it is worth noting that apart from the removal of skip-connections, other CNN architectures were not subjected to testing in this study.

While the overall shape of a post-storm profile is captured correctly, the severity of the dune erosion is not. Using the single-profile training dataset and shallow networks, the pre-storm profile shapes cannot be linked to the dune erosion processes. Suggestions such as increasing the network depth and kernel size to overcome this issue are addressed in subsection 5.2.2.

It was found that U-Net has trouble overcoming spatial alterations at the location of erosion processes. The spatial alterations at the dune are projected on the pre-storm profile. For deep network structures and large kernel sizes, U-Net is able to overcome these alterations on a larger scale, but not at a smaller scale. When using a large kernel, the receptive field of each convolutional layer increases, potentially leading to a loss of fine-grained spatial details. The model may become less sensitive to local patterns and fail to capture small, intricate features in the input. That is why these shortcomings are less distinct when applying smaller kernels.

Besides that, the issue could be overcome by adding profiles with these kinds of alternations at the foredune will likely increase the understanding of U-Net. These results will be of great interest when other dune erosion regimes are considered to see if U-Net is able to make correct interpretations for large changes in the dune geometry.

5.2.5. Performance metric

Reflecting on the results of the exploration phase of this research, it becomes evident that relying solely on the mean square error of the dune profile may not suffice as the sole indicator of the model's performance, particularly when considering the upscaling objectives of the surrogate model. This aspect was previously acknowledged in the methodology, and the insights gained in the preceding section further reinforce this consideration. As a result, emphasis is placed on the skill of predicting DEVs during the upscaling phase of this research.

Nevertheless, these two metrics fail to provide a comprehensive indication of the quality of the post-storm profile shape. Visual inspections are required to provide insight into the actual shape of the post-storm profile. However, visual inspections can not serve as a quantifiable means to optimize modelling results. Therefore, the inclusion of additional metrics is necessary to accurately evaluate the shape of the post-storm profile in optimization schemes.

The root mean squared transport error (RMSTE) developed by Bosboom (2019) to evaluate sediment transport processes could be integrated into the validation metric to correctly interpret the fit of predicted profiles. This error metric is defined as the root-mean-square of the optimal transport field. The RMSTE takes into account both the quantity of misplaced sediment and the distance over which this sediment needs to be transported. This sensitivity of the RMSTE enables a more comprehensive evaluation of the model's performance in capturing the spatial characteristics of sediment transport. This could be a valuable metric to capture the shape features of the predicted profile.

Besides that, additional morphological parameters could be extracted from the post-storm profile prediction. Currently, the analyses on the skill of the surrogate model are solely carried out on the DEV. This establishes simplicity and allows for comparative studies, but could be supplemented with other metrics such as the post-storm location of the dune toe, location of dune crest and slope of the duneface. This would provide supplementary support to strengthen the assessment of the model's performance. These metrics could also be evaluated based on the same skill indicator (Murphy, 1988), as showcased by Gharagozlou et al. (2022). Since the DEV is a volume representing a certain area, these additional metrics could help to specify the actual location of predicted elevation points. This way, the shape of the post-storm dune is captured in more detail.

5.3. U-Net evaluation

As presented in Figure 4.33, U-Net is able to correctly simulate the relationship between the beach slope and DEV. For steep slopes, large DEV are modelled. This is in agreement with the results presented in subsection 4.1.3. While the trend is harder to detect for this data, a positive relationship can be observed. This trend is more pronounced for U-Net predictions than for XBeach predictions.

It should be stressed that the other hyperparameters (pooling size, learning rate, activation function, etc.) used for U-Net and the multi-profile dataset are similar to the results obtained in Figure 4.2.1 and have not been re-evaluated. Tuning your neural network structure and adjusting the training dataset should be carried out in parallel. Therefore, the currently used hyperparameters should be re-evaluated in further stages of this research.

Lastly, it is important to emphasize that U-Net was originally designed for image segmentation tasks. Hence, the generated output is an encoded and decoded representation of the input. Consequently, the predicted post-storm profiles may exhibit similar characteristics to the input pre-storm profiles. This characteristic is spotted for deep network structures and large kernel sizes (Figure 4.32. It is important to emphasize that this underlying principle of U-Net can be overcome by selecting a suitable network architecture. Furthermore, incorporating greater diversity in the input space of the training data would result in improved performance on outliers within the data.

6
Conclusion

This chapter attempts to answer the research questions proposed in section 1.3 to reach the main objective of ***enabling fast prediction of actual post-storm sandy profiles along the Holland Coast using neural networks and XBeach.***

1. What response in dune erosion volumes is found in the post-storm profiles as a result of slope changes of sandy pre-storm profiles using a simplified dataset and XBeach?

Using a dataset of simplified profiles, a large variety of DEVs is captured in the corresponding post-storm profiles. In general, a steeper slope of a certain profile section leads to a larger impact on the dune. For the considered simplified set of cross-shore profiles, the beach and nearshore slope show the highest sensitivity to the modelled DEV. The dune response remains in the collision regime and does not reach the top of the dune. Therefore, the height of the dunecrest of the pre-storm profiles has a low impact on the storm response at the dune.

2. What performance metrics can be used to evaluate surrogate modelling using neural networks for post-storm profile shape prediction?

The exploration phase of this research showed that the MSE can not be the only performance metric in evaluating surrogate models. Additional metrics such as the skill on extracted variables are therefore incorporated in this research. These metrics should always be provided with visual inspections of surrogate model predictions. As the surrogate model tends to rely on statistics instead of physics, a miss match between the modeller's and model's understanding can only be overcome if predictions are visually inspected.

However, in the next stages of this research, additional performance metrics, such as the root-mean-square transport error, should be introduced. Next to that, the skill metric should also be applied to other morphological parameters that define the post-storm profile. This allows for a better representation of the actual shape of the predicted post-storm profile.

3. **To what extent are pre-processing tools and neural networks able to make post-storm profile shape predictions for a simplified dataset?**

Notably, using a simplified dataset, convolutional neural network and a U-Net architecture, shapes of post-storm profiles can be reproduced. The quality of the prediction strongly relies on pre-processing tools, the neural network structure and the training/test data.

For a neural network to process, the pre- and post-storm profiles should be presented in a fixed grid with constant spacing. The active section of the profile should be highlighted by means of a minimum value and a fixed elevation. This active profile section can also be accentuated by training the neural network on the post-storm difference or applying a weight to this profile section.

It can be concluded that the U-Net architecture is a suitable neural network to predict post-storm profile shapes. The performance of U-Net and introduced errors predominantly rely on the network depth, network width and kernel size. In general, the complexity of the U-Net structure should simultaneously match the complexity found in the data and allow for enough depth to facilitate interactions between different profile sections. Detailed analyses of the network depth and kernel size showed the effect of different U-Net structures on the interpretation of coastal processes by the neural network.

The exploration phase showed the importance of an appropriate test dataset. While being suitable for the exploration of several U-Net alternatives, it lacked the capability to judge the interpretation of coastal processes by U-Net. Eventually, a more detailed analysis of the interpretation of specific profiles in the training data by U-Net turned out to be crucial to understand the interpretation of coastal processes by U-Net.

4. Can the neural network structure, obtained in the exploration phase, be scaled-up to predict post-storm profile shapes of actual Holland Coast profile shapes?

Applying the obtained U-Net structure for a realistic training and test dataset allowed for the prediction of actual post-storm profiles along the Holland Coast. Results on the network structure and model performance showed similar trends as was found in the exploration phase. All in all, a U-Net-based surrogate model succeeds in describing the positive relationship between the beach slope and dune erosion volumes.

The multi-profile training dataset outperformed the single-profile training dataset for all network structures. It can be concluded that including more alongshore variability in the training data is beneficial for the surrogate model performance. Using additional performance metrics, these trends could be quantified. An average skill of 0.51 for predicting DEVs of 21 pre-storm profiles at the Holland Coast for stationary storm conditions was achieved.

The results in the upscaling phase confirmed the insights gained in the exploration phase. More complexity in the U-Net structure allows for a better understanding of dune erosion processes by U-Net. Again, the network depth and kernel size proved to be crucial components of the U-Net structure. Using a shallow network, more predictive capability can be obtained by increasing the kernel size. The effect of large kernels vanishes for deep networks.

Concluding from the visual inspections of the predicted post-storm profiles, U-Net is an appropriate tool to describe cross-shore profile changes. However, having an encoding/decoding structure, predictions made with U-Net originate from the original input structure. This becomes clear when features in the dynamic areas are present in the test data but absent in the training dataset.

7 Recommendations

This chapter describes recommendations arising from this study. This includes practical implications of the acquired knowledge of this research and possible further research.

Practical implications

In its current state, the U-Net model for post-storm profile predictions does not have a significant practical implication. While this research showed many possibilities and shortcomings of a U-Net surrogate modelling strategy, the obtained skill and consistency do not yet allow for sufficient confidence in the model predictions. Using this model for quick assessments and preliminary studies, it needs improvement in several aspects.

This application of U-Net would be a suitable showcase for data-driven modelling in hydraulic engineering. As a result of the simplicity captured in the input and output data, this research presents a case that is understandable for every hydraulic engineer and mathematician. This simplicity allows for the trace down of interaction between U-Net structure and morphological storm response. This promotes understanding of U-Net's interpretation of spatial data.

Hyperparameters

When proceeding with the results obtained in this research, it is important to stress that the hyperparameters should be constantly re-evaluated. In this research, hyperparameters such as the pooling size, learning rate and batch size have had a fixed value throughout the continuation and improvement of the model. This fixed value originates from the first model version. For future research, these hyperparameters should be evaluated in parallel to other modifications of the model. Next to that, the considered range for the hyperparameters should not be too narrow. The kernel size is a good example of such a parameter. Initially, for the first optimization scheme, a range up to a size of 5 was considered. While in later stages with new training and test dataset, a kernel size of 10 turned out to be beneficial.

Training data

Currently, the model is trained on a multi-profile dataset of 404 profiles originating from a set of four profiles along the Holland coast. This showed improved results compared to a single-profile dataset. The training data could be upscaled to a realistic scenario for which a larger variety of the Holland coast is captured. Including more variety in the input space can be achieved through several techniques.

Using more than four profiles as the base of your training dataset. This would be a more suitable representation of the Holland Coast test dataset. Through this technique, there would be the possibility to include more alongshore variability and a wider range of input shapes. This way, the cross-shore profiles on the Holland coast are better represented.

Increasing the number of base profiles can come with a reduction of the amount of applied modifications. To account for enough variability in the input space, selecting the base profiles should be done carefully. Several clustering techniques could be applied to make sure the Holland Coast is adequately represented. The methods presented by Athanasiou et al. (2021) would be a suitable reference point for this analysis.

Note: Evaluation of the U-Net architecture would be required for a new training dataset.

Performance metric

For future research, performance metric should be revised. The currently applied MSE and skill give a good indication of the model performance. However, applying visual inspections, it was found that the shapes of the predicted post-storm profile are not correctly represented by these metric.

This issue is addressed by Bosboom and taken care of by introducing the root-mean-square transport error (RMSTE). This metric accounts for the optimal transport process and is able to deal with spatial characteristics of sediment transport. For future research, it is advised to include this metric to judge the performance of the surrogate model.

Currently, the skill of the surrogate model is solely calculated for the dune erosion volume. In next steps, other morphological indicators could be used to give a more comprehensive indication of the shape representation. For example. calculating the skill on beach width reduction, dune toe- or dune crest displacement could turn out the be a valuable metric.

For this research, these metrics are only used in the post-training performance judgement. However, these could also be used in the actual training of the U-Net structure. This method could be similar to physics-informed neural networks (PINNs). For PINNs, the loss function is supplied with an additional variable and this variable supplies additional information which is used during training of the model. In this case, the morphological parameter extracted from the predicted profiles, such as the DEV, can be added to the loss function. Since it is expected that this variable will be of different importance than the elevation point, the computed error is multiplied with a weight (W). This could be integrated into a loss function (Equation 7.1).

$$E = MSE_{elevation} + MSE_{DEV} = \frac{1}{m}\left(\sum(y_r - \hat{y}_r)^2 + W\left(\frac{1}{m}\sum(DEV - D\hat{E}V)^2\right)\right) \tag{7.1}$$

Storm conditions

Currently the surrogate model is trained for a single stationary storm and only takes pre-storm profile input. To improve the resilience and broaden the applicability of the model, the incorporation of storm conditions would be favourable. When including storm conditions, the model could be used for a wider variety of storm scenarios.

A major point of attention for including storm conditions in a U-Net architecture, is the difference in the reference frame of storms compared to the 1D profile data. This could be accounted for by "injecting" the storm conditions at the deepest layer of the network's architecture. Deeper into the network, the spatial reference frame of the input has been reduced the most. Therefore, it makes sense to introduce the storm conditions in the deepest part of the neural network. This was attempted by Löwe et al. (2021) for urban pluvial flood events. Showing that introduction of weather conditions is possible for 3D flood maps.

Other erosion/restoration regimes

The current model is trained on profiles and a single storm that remain in the collision regime. This is typical for the Holland coast and prevailing storm conditions in this area. However, looking at other sandy coasts, a wider range of erosion regimes is observed. Overwash and inundation regimes, for example, introduce drastic morphological changes at the dune. It would be interesting to analyse if U-Net is capable to represent these regimes.

Next to dune erosion, dune restoration is also a widely observed sediment transport process. Generally, after storm erosion, dunes restore to their pre-storm equilibrium situation. Restoration processes take place on a significantly larger timescale than erosion processes. In theory, U-Net should also be able to predict the restoration shape changes. However, the variability in the output space will likely be larger due to the longer duration of this process and the absence of a distinct post-restoration profile shape.

Bibliography

A, D. A., Anand, N., Lubloy, E., & Arulraj, G. P. (2021). Deep learning based thermal crack detection on structural concrete exposed to elevated temperature. *Advances in Structural Engineering*, *24*, 1896–1909. https://doi.org/10.1177/1369433220986637

Amiri, M., Brooks, R., & Rivaz, H. (2020). Fine-tuning u-net for ultrasound image segmentation: Different layers, different outcomes. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, *67*, 2510–2518. https://doi.org/10.1109/TUFFC.2020.3015081

Arora, A. (2020). U-net: A pytorch implementation in 60 lines of code. https://amaarora.github.io/2020/09/13/unet.html

Athanasiou, P., Dongeren, A. V., Giardino, A., Vousdoukas, M., Antolinez, J. A. A., & Ranasinghe, R. (2022). Estimating dune erosion at the regional scale using a meta-model based on neural networks. https://doi.org/10.5194/nhess-2022-106

Athanasiou, P., van Dongeren, A., Giardino, A., Vousdoukas, M., Antolinez, J. A., & Ranasinghe, R. (2021). A clustering approach for predicting dune morphodynamic response to storms using typological coastal profiles: A case study at the dutch coast. *Frontiers in Marine Science*, *8*. https://doi.org/10.3389/fmars.2021.747754

Athanasiou, P., van Dongeren, A., Giardino, A., Vousdoukas, M., Gaytan-Aguilar, S., & Ranasinghe, R. (2019). Global distribution of nearshore slopes with implications for coastal retreat. *Earth System Science Data*, *11*, 1515–1529. https://doi.org/10.5194/essd-11-1515-2019

Baumhoer, C. A., Dietz, A. J., Kneisel, C., & Kuenzer, C. (2019). Automated extraction of antarctic glacier and ice shelf fronts from sentinel-1 imagery using deep learning. *Remote Sensing*, *11*. https://doi.org/10.3390/RS11212529

Beuzen, T., Harley, M. D., Splinter, K. D., & Turner, I. L. (2019). Controls of variability in berm and dune storm erosion. *Journal of Geophysical Research: Earth Surface*, *124*, 2647–2665. https://doi.org/10.1029/2019JF005184

Beuzen, T., & Splinter, K. (2020). Machine learning and coastal processes. *Sandy Beach Morphodynamics*, 689–710. https://doi.org/10.1016/B978-0-08-102927-5.00028-X

Bomers, A. (2021). Predicting outflow hydrographs of potential dike breaches in a bifurcating river system using narx neural networks. *Hydrology*, *8*, 87. https://doi.org/10.3390/HYDROLOGY8020087

Bosboom, J. (2019). Quantifying the quality of coastal morphological predictions. https://doi.org/10.4233/uuid:e4dc2dfc-6c9c-4849-8aa9-befa3001e2a3

Bosboom, J., & Stive, M. J. F. (2021). *Coastal dynamics*. https://doi.org/10.5074/T.2021.001

Castelle, B., & Harley, M. D. (2020). *Extreme events: Impact and recovery*. https://hal.archives-ouvertes.fr/hal-03044275

Chondros, M., Tsoukala, V., Metallinos, A., Papadimitriou, A., & Memos, C. (2021). A coastal flood early-warning system based on offshore sea state forecasts and artificial neural networks. *Journal of Marine Science and Engineering 2021, Vol. 9, Page 1272*, *9*, 1272. https://doi.org/10.3390/JMSE9111272

Daly, C., Roelvink, D., Dongeren, A. V., de Vries, J. V. T., & McCall, R. (2011). Short wave breaking effects on low frequency waves. *Coastal Engineering Proceedings*, *1*, 20. https://doi.org/10.9753/icce.v32.waves.20

Eichinger, M., Heinlein, A., & Klawonn, A. (2022). Surrogate convolutional neural network models for steady computational fluid dynamics simulations. *ISBN*, *56*, 235–255. https://doi.org/10.1553/ETNA_VOL56S235

Gharagozlou, A., Anderson, D. L., Gorski, J. F., & Dietrich, J. C. (2022). Emulator for eroded beach and dune profiles due to storms. *Journal of Geophysical Research: Earth Surface*, *127*. https://doi.org/10.1029/2022JF006620

Hallermeier, R. J. (1981). *A profile zonation for seasonal sand beaches from wave climate*.

Hashemi, M. R., Ghadampour, Z., & Neill, S. P. (2010). Using an artificial neural network to model seasonal changes in beach profiles. *Ocean Engineering*, *37*, 1345–1356. https://doi.org/10.1016/J.OCEANENG.2010.07.004

Hebb, D. O. (O. (1949). *The organization of behavior a neuropsychological theory*. L. Erlbaum Associates. https://books.google.com/books/about/The_Organization_of_Behavior.html?hl=nl&id=ddB4AgAAQBAJ

Jia, G., Taflanidis, A. A., Nadal-Caraballo, N. C., Melby, J. A., Kennedy, A. B., & Smith, J. M. (2016). Surrogate modeling forăpeak or time-dependent storm surge predictionăover an extended coastal region using an existing database of synthetic storms. *Natural Hazards*, *81*, 909–938. https://doi.org/10.1007/S11069-015-2111-1

Kabiri-Samani, A. R., Aghaee-Tarazjani, J., Borghei, S. M., & Jeng, D. S. (2011). Application of neural networks and fuzzy logic models to long-shore sediment transport. *Applied Soft Computing*, *11*, 2880–2887. https://doi.org/10.1016/J.ASOC.2010.11.021

Karunarathna, H., Pender, D., Ranasinghe, R., Short, A. D., & Reeve, D. E. (2014). The effects of storm clustering on beach profile variability. *Marine Geology*, *348*, 103–112. https://doi.org/10.1016/J.MARGEO.2013.12.007

Kim, S. W., Melby, J. A., Nadal-Caraballo, N. C., & Ratcliff, J. (2015). A time-dependent surrogate model for storm surge prediction based on an artificial neural network using high-fidelity synthetic hurricane modeling. *Natural Hazards*, *76*, 565–585. https://doi.org/10.1007/S11069-014-1508-6/FIGURES/9

Lecun, Y., Bottou, L. E., Bengio, Y., & Abstract|, P. H. (1998). *Gradient-based learning applied to document recognition*.

López, I., Aragonés, L., Villacampa, Y., & Serra, J. C. (2017). Neural network for determining the characteristic points of the bars. *Ocean Engineering*, *136*, 141–151. https://doi.org/10.1016/J.OCEANENG.2017.03.033

Löwe, R., Böhm, J., Jensen, D. G., Leandro, J., & Rasmussen, S. H. (2021). U-flood topographic deep learning for predicting urban pluvial flood water depth. *Journal of Hydrology*, *603*. https://doi.org/10.1016/j.jhydrol.2021.126898

Luijendijk, A., Hagenaars, G., Ranasinghe, R., Baart, F., Donchyts, G., & Aarninkhof, S. (2018). The state of the world's beaches. *Scientific Reports*, *8*. https://doi.org/10.1038/s41598-018-24630-6

machine-learning.paperspace.com. (2020). Activation function. https://machine-learning.paperspace.com/wiki/activation-function

Maretto, R. V., Fonseca, L. M., Jacobs, N., Körting, T. S., Bendini, H. N., & Parente, L. L. (2021). Spatio-temporal deep learning approach to map deforestation in amazon rainforest. *IEEE Geoscience and Remote Sensing Letters*, *18*, 771–775. https://doi.org/10.1109/LGRS.2020.2986407

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics 1943 5:4*, *5*, 115–133. https://doi.org/10.1007/BF02478259

Minsky, M., & Papert, S. A. (1969). *Perceptrons*. https://mitpress.mit.edu/9780262630221/perceptrons/

Murphy, A. (1988). Skill scores based on the mean square error and their relationships to the correlation coefficient.

Nourani, V., Hakimzadeh, H., Amini, A. B., Lab, S. A. F., & Amini, H. H. A. B. (2012). Implementation of artificial neural network technique in the simulation of dam breach hydrograph. *Journal of Hydroinformatics*, *14*, 478–496. https://doi.org/10.2166/HYDRO.2011.114

Nyuytiymbiy, K. (2020). Parameters and hyperparameters in machine learning and deep learning. https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Prakash, N., Manconi, A., & Loew, S. (2020). Mapping landslides on eo data: Performance of deep learning models vs. traditional machine learning models. *Remote Sensing*, *12*. https://doi.org/10.3390/RS12030346

Razavi, S., Tolson, B. A., & Burn, D. H. (2012). Review of surrogate modeling in water resources. *Water Resources Research*, *48*, 7401. https://doi.org/10.1029/2011WR011527

Rijkswaterstaat. (2021). *Kustlijnkaarten 2022*.

Roelvink, D., Reniers, A., Dongeren, A. V., Vries, J. V. T. D., Lescinski, J., & Report, R. M. (2010). *Xbeach model description and manual version 6 xbeach model description and manual*.

Roelvink, D., Reniers, A., van Dongeren, A., van Thiel de Vries, J., McCall, R., & Lescinski, J. (2009). Modelling storm impacts on beaches, dunes and barrier islands. *Coastal Engineering*, *56*, 1133–1152. https://doi.org/10.1016/j.coastaleng.2009.08.006

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. http://arxiv.org/abs/1505.04597

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*, 386–408. https://doi.org/10.1037/H0042519

Ruessink, B. G., & Jeuken, M. C. (2002). Dunefoot dynamics along the dutch coast. *Earth Surface Processes and Landforms*, *27*, 1043–1056. https://doi.org/10.1002/ESP.391

Sallenger, A. H. (2000). *Storm impact scale for barrier islands* (3).

Santos, V. M., Wahl, T., W.Long, J., Passeri, D. L., & Plant, N. G. (2019). Combining numerical and statistical models to predict storm-induced dune erosion. https://doi.org/10.1029/2019JF005016

Serda, M., Becker, F. G., Cleary, M., Team, R. M., Holtermann, H., The, D., Agenda, N., Science, P., Sk, S. K., Hinnebusch, R., A, R. H., Rabinovich, I., Olmert, Y., Uld, D. Q. G. L. Q., Ri, W. K. H. U., Lq, V., Frxqwu, W. K. H., Zklfk, E., Edvhg, L. V., . . .,. (2022). The development of a fast-computing emulator to predict dike failure during severe flood events (G. Balint, B. Antala, C. Carty, J.-M. A. Mabieme, I. B. Amar, & A. Kaplanova, Eds.). *Uniwersytet lski*, *7*, 343–354. https://doi.org/10.2/JQUERY.MIN.JS

Sun, S., Bhirangi, R., & Vogler, N. (2021). Lecture 16: Building blocks of deep learning. https://sailinglab.github.io/pgm-spring-2019/notes/lecture-16/

Tijssen, A., & Diermanse, F. (2010). *Storm surge duration and storm duration at hoek van holland sbw-belastingen*.

van Gent, M. R., van den Boogaard, H. F., Pozueta, B., & Medina, J. R. (2007). Neural network modelling of wave overtopping at coastal structures. *Coastal Engineering*, *54*, 586–593. https://doi.org/10.1016/J.COASTALENG.2006.12.001

van IJzendoorn, C. (2021). Jarkus analysis toolbox (jat). https://jarkus-analysis-toolbox.readthedocs.io/en/latest/#

van Rijn, L. C. (2009). Prediction of dune erosion due to storms. *Coastal Engineering*, *56*, 441–457. https://doi.org/10.1016/J.COASTALENG.2008.10.006

van Thiel de Vries, J. S., van Gent, M. R., Walstra, D. J., & Reniers, A. J. (2008). Analysis of dune erosion processes in large-scale flume experiments. *Coastal Engineering*, *55*, 1028–1040. https://doi.org/10.1016/J.COASTALENG.2008.04.004

van Thiel-De Vries, J. (2009). *Dune erosion during storm surges*.

Vellinga, P. (1982). Beach and dune erosion during storm surges. *Coastal Engineering*, *6*, 361–387. https://doi.org/10.1016/0378-3839(82)90007-2

Verhaeghe, H., Rouck, J. D., & van der Meer, J. (2008). Combined classifierquantifier model: A 2-phases neural model for prediction of wave overtopping at coastal structures. *Coastal Engineering*, *55*, 357–374. https://doi.org/10.1016/J.COASTALENG.2007.12.002

Werbos, P. (1994). *The roots of backpropagation: From ordered derivatives to neural networks ... - paul john werbos - google boeken*. https://books.google.nl/books?hl=nl&lr=&id=WdR3OOM2gBwC&oi=fnd&pg=PA1&dq=paul+werbos+1974&ots=M6ow2XP1-0&sig=oEKOBHEqw69uho6k1EfqAu_vZv0#v=onepage&q=paul20werbos201974&f=false

Widrow, B., & Hoff, M. E. (1962). Associative storage and retrieval of digital information in networks of adaptive neurons. *Biological Prototypes and Synthetic Systems*, 160–160. https://doi.org/10.1007/978-1-4684-1716-6_25

Yang, X., Li, X., Ye, Y., Lau, R. Y., Zhang, X., & Huang, X. (2019). Road detection and centerline extraction via deep recurrent convolutional neural network u-net. *IEEE Transactions on Geoscience and Remote Sensing*, *57*, 7209–7220. https://doi.org/10.1109/TGRS.2019.2912301

Zijlema, M. (2011). *Cie4340 computational modelling of flow and transport*. http://fluidmechanics.tudelft.nl

A
XBeach set-up

	XBeach parameter	Definition	Value
Physical processes	wavemodel	determines what kind of waves are included in model	surfbeat
Grid parameters	nx	number of grid point in x	1085
	ny	number of grid point in y	0
	vardx	option of variable grid size	1 (True)
	alfa	angle of grid	0.0
	thetamin	lower directional limit	-90
	thetamax	upper directional limit	-90
	dtheta	directional resolution	180
Model time parameters	CFL	courant number	0.9
Wave boundary conditions	wbctype	file type wave boundary conditions	jonstable
Wave-spectrum boundary conditions	random	random generator	0 (off)
Flow boundary conditions	epsi	weighted factor between steady flow and particle velocity	-1.0
	tidetype	interpretation of the tide	velocity
Wave breaking parameters	break	option breaker model	roelvink_daly
	gamma	breaking parameter	0.46
	gamma2	end of breaking	0.34
	alpha	wave dissipation coefficient	1.38
Flow parameters	bedfriction	bed friction model	manning
	bedfriccoef	bed friction coefficient	0.02
Sediment transport parameters	form	waveshape model	vanthiel
	waveform	equilibrium sediment concentration formulation	vanthiel_vanrijn
Morphology parameters	morfac	morphological acceleration factor	5.0
	morstart	time start of morphological updates	6000
	wetslp	critical avalanching slope underwater	0.15

Table A.1: XBeach parameters with definition obtained from XBeach manual (Roelvink et al., 2009))

B

U-Net python code

```python
# -*- coding: utf-8 -*-
"""
Created on Thu Nov 24 10:38:47 2022

@author: asselt
"""

import numpy as np # this module is useful to work with numerical arrays
import torch
import torchvision
from torch import nn
import torch.nn.functional as F


class Block(nn.Module):
    def __init__(self, in_ch, out_ch):
        super().__init__()
        self.conv1 = nn.Conv1d(in_ch, out_ch, 3)
        self.relu  = nn.ReLU()
        self.conv2 = nn.Conv1d(out_ch, out_ch, 3)

    def forward(self, x):
        x = self.conv1(x)
        x = self.relu(x)
        x = self.conv2(x)
        return x


class Encoder(nn.Module):
    def __init__(self, chs):
        super().__init__()
        self.enc_blocks = nn.ModuleList([Block(chs[i], chs[i+1]) for i in range(len(chs)-1)])
        self.pool       = nn.MaxPool1d(2)

    def forward(self, x):
        ftrs = []
        for block in self.enc_blocks:
            x = block(x)
            ftrs.append(x)
            x = self.pool(x)
        return ftrs


class Decoder(nn.Module):
    def __init__(self, chs):
        super().__init__()
        self.chs         = chs
        self.upconvs    = nn.ModuleList([nn.ConvTranspose1d(chs[i], chs[i+1], 3, 2) for i in range(len(chs)-1)])
        self.dec_blocks = nn.ModuleList([Block(chs[i], chs[i+1]) for i in range(len(chs)-1)])

    def forward(self, x, encoder_features):
        for i in range(len(self.chs)-1):
            x        = self.upconvs[i](x)
            enc_ftrs = self.crop(encoder_features[i], x)
            x        = torch.cat([x, enc_ftrs], dim=1)
            x        = self.dec_blocks[i](x)

        return x
```

```python

    def crop(self, enc_ftrs, x): #To get encoding feature maps into the right size
        _, H, W = x.shape
        enc_ftrs   = torchvision.transforms.CenterCrop([H, W])(enc_ftrs)
        return enc_ftrs



class UNet(nn.Module):
    def __init__(self, enc_chs=(1, 16, 32,64,128,256), dec_chs=(256, 128, 64, 32, 16), num_class=1,
         retain_dim=True, out_sz=[290]):
        super().__init__()
        self.encoder     = Encoder(enc_chs)
        self.decoder     = Decoder(dec_chs)
        self.head        = nn.Conv1d(dec_chs[-1], num_class, 1)
        self.retain_dim  = retain_dim
        self.out_sz      = out_sz


    def forward(self, x):
        enc_ftrs = self.encoder(x)
        out      = self.decoder(enc_ftrs[-1], enc_ftrs[::-1][1:])
        out      = self.head(out)
        if self.retain_dim:
            out = F.interpolate(out, self.out_sz)
        return out
```

C U-Net structures

C.1. Initial U-Net settings

Parameter	Value
Model type	Diff.
Network depth	2
Channel size	32
Kernel size	3
Pooling size	2
Learning rate	0.0001
Batch size	15
Validation percentage	0.2
Optimizer	Adam
Loss function	MSE
Activation function	ReLu

Table C.1: Initial U-Net settings

C.2. U-Net version 1

Parameter	Value
Model type	Diff.
Network depth	-
Channel size	-
Kernel size	3
Pooling size	2
Learning rate	0.0075
Batch size	15
Validation percentage	0.2
Optimizer	Adam
Loss function	MSE
Activation function	ReLu

Table C.2: U-Net v1 settings

D
Training datasets

D.1. Single-profile-based training dataset

D.1.1. Dataset base

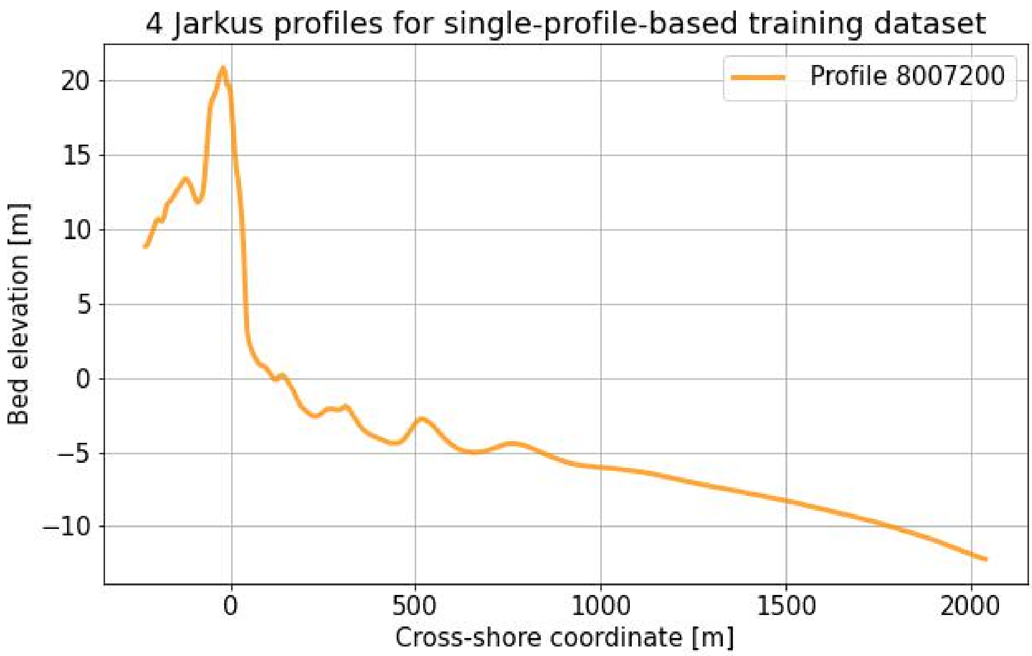


Figure D.1: The profile which is used to set-up the single-profile-based training dataset.

D.1.2. Modification factors

Profile parameter	Modification factors	Second modification factor
Beach width	0.1, 0.2, 0.5, **0.75, 0.9**, 1, **1.1, 1.25**, 1.5, 2, 5	0.5, 0.75, 0.9, 1.1, 1.25, 1.5
Inter-tidal area width	0.5, 0.6, **0.75, 0.9**, 1, **1.1, 1.25**, 1.4, 1.5, 2	0.5, 0.75, 0.9, 1.1, 1.25, 1.5
Nearshore width	0.5, 0.6, **0.75, 0.9**, 1, **1.1, 1.25**, 1.4, 1.5, 2	0.5, 0.75, 0.9, 1.1, 1.25, 1.5
Dune width	0.5, **0.75, 0.9**, 1, **1.1, 1.25**, 1.5, 2, 5	0.5, 0.75, 0.9, 1.1, 1.25, 1.5
Dune height	0.5, **0.75, 0.9**, 1, **1.1, 1.25**, 1.5	0.5, 0.75, 0.9, 1.1, 1.25, 1.5
Dunecrest height	0.5, **0.75, 0.9**, 1, **1.1, 1.25**, 1.5	

Table D.1: Profile parameters and imposed modification factors. The bold values in the modification factor columns represent the profiles which undergo a second modification.

D.1.3. Dataset in standard cross-shore grid

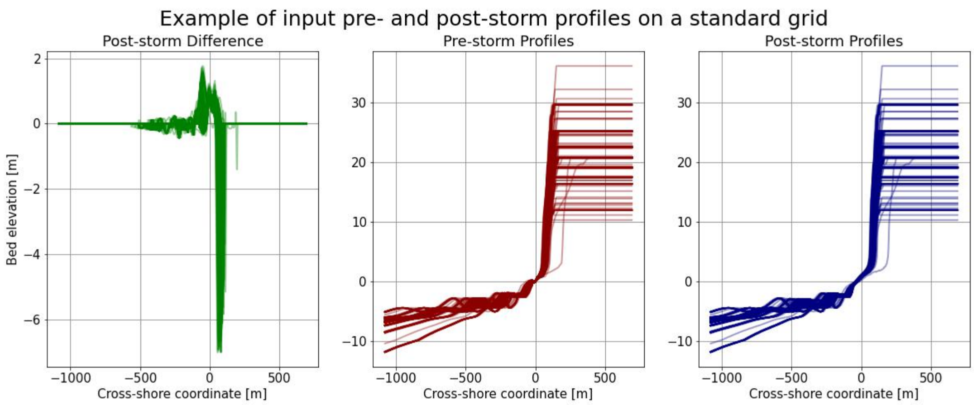


Figure D.2: Input and outputs profiles in standard grid configuration. The left figure indicates the post-storm difference, middle and right indicate the representative pre- and post-storm profiles.

D.1.4. DEV variability in dataset

D.1.5. Comparison training dataset and Holland Coast

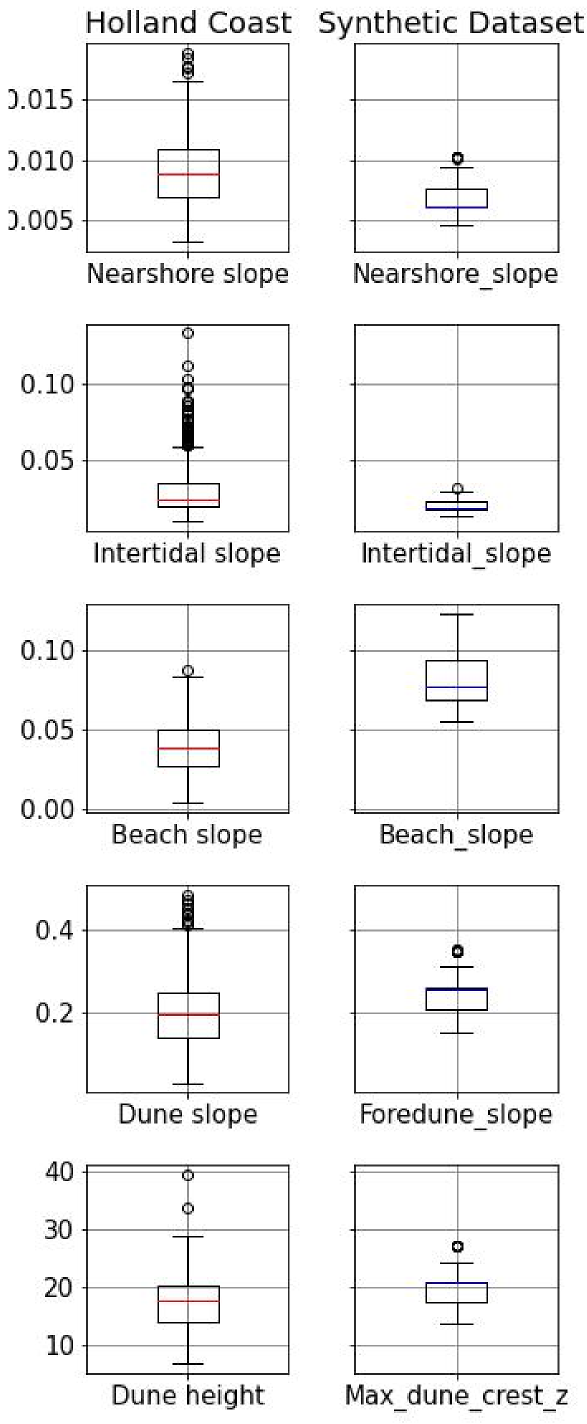


Figure D.3: Boxplots for coastal parameter at the Holland Coast and Synthetic Training dataset.

D.2. Multi-profile-based training dataset

D.2.1. Dataset base

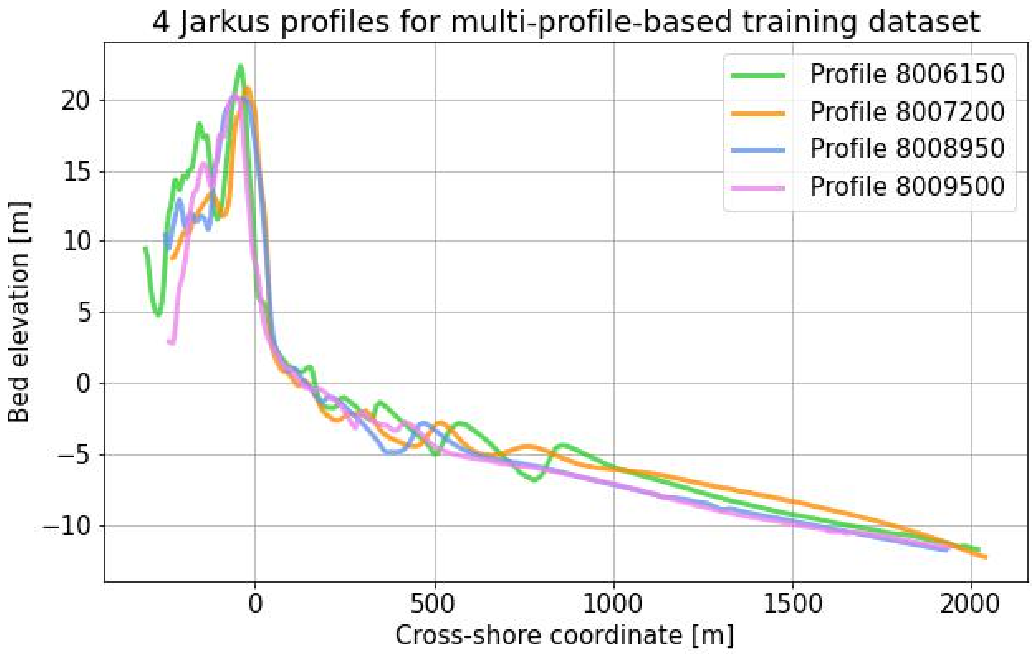


Figure D.4: The 4 profiles which are used to set-up the multi profile dataset.

D.2.2. Modification factors

Profile parameter	Modification factors	Second modification factor
Beach width	0.1, 0.2, 0.5, **0.75**, 1, **1.25**, 1.5, 2, 5	0.5, 0.75, 0.9, 1.1, 1.25, 1.5
Inter-tidal area width	0.5, 0.6, **0.75**, 1, **1.25**, 1.4, 1.5, 2	0.5, 0.9, 1.1, 1.5
Nearshore width	0.5, 0.6, **0.75**, 1, **1.25**, 1.4, 1.5, 2	0.5, 0.9, 1.1, 1.5
Dune width	0.5, **0.75**, 1, **1.25**, 1.5, 2, 5	0.5, 0.9, 1.1, 1.5
Dune height	0.5, **0.75**, 1, **1.25**, 1.5	0.5, 0.9, 1.1, 1.5
Dunecrest height	0.5, **0.75**, 1, **1.25**, 1.5	0.5, 0.9, 1.1, 1.5

Table D.2: Profile parameters and imposed modification factors

D.2.3. Dataset in standard cross-shore grid

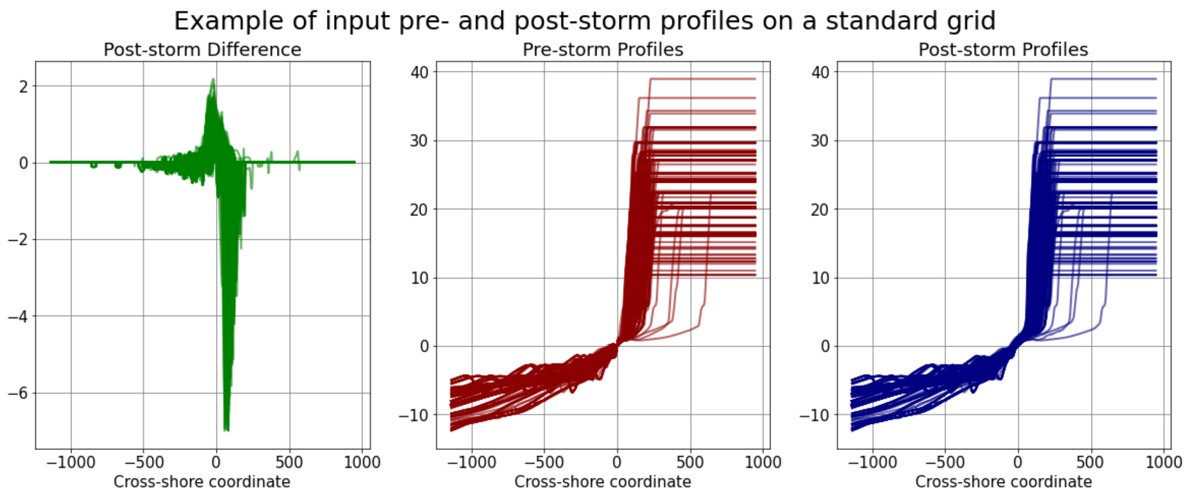


Figure D.5: Input and outputs profiles in standard grid configuration. The left figure indicates the post-storm difference, middle and right indicate the representative pre- and post-storm profiles.

D.2.4. DEV variability in dataset

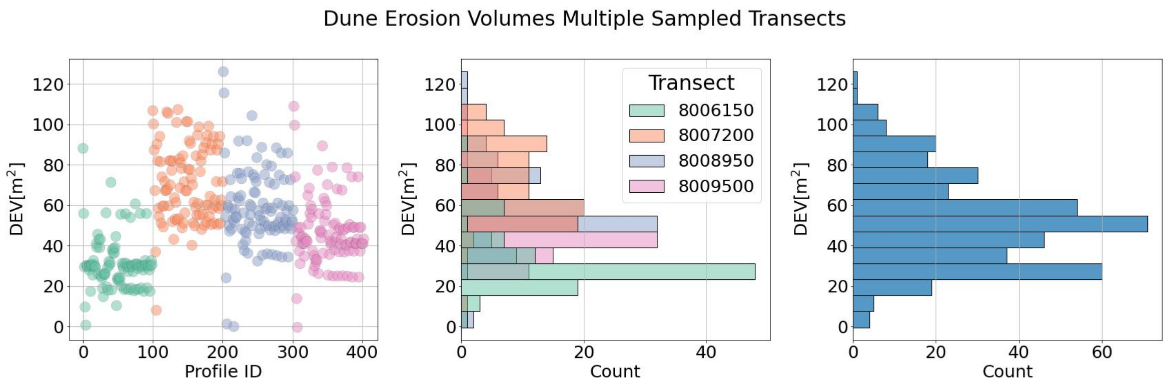


Figure D.6: The variability of DEV captured in the dataset. The DEV per profile id (left) and a histogram plot of the DEV data (right)

E

Data quantity multi-profile dataset

Network structure

Initially, the previously obtained U-Net structure is evaluated for both training datasets (single- and multiple original profiles). When testing on the exploration single-profile test dataset, the results presented in Figure E.1 look different from Figure 4.14. Whereas the results for a training dataset with a single profile showed a significantly better performance for a network depth of 2, this seems to be less distinct for a multi profile training dataset. U-Net can predict with a similar accuracy for all network depths. However, the the computed MSE error for all network depths remains above the previously obtained lowest MSE for the single profile dataset (red line).

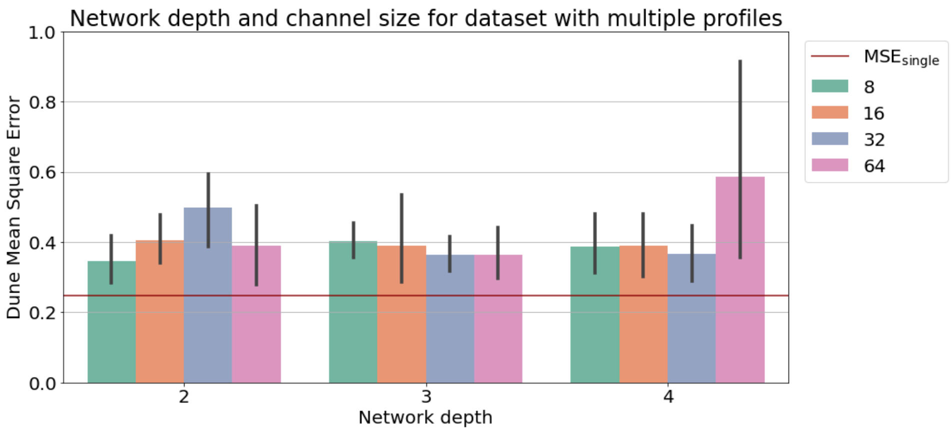


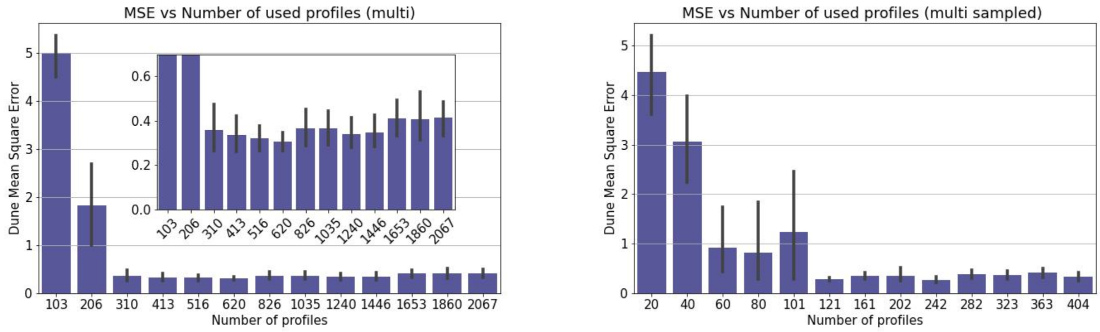
Figure E.1: Network depth and width for dataset with multiple profiles

This lower accuracy might be caused by (1) overfitting and (2) less appropriate test data. These issues are addressed is next sections.

Sampling density

When carrying out the same data quantity analyses for the new dataset, the MSE error converges around a stable value around 200 - 300 profiles. After 500 - 600 profiles the error actually starts increasing again and overfitting issues start to arise Figure E.2a.

To get rid of these overfitting issues, the sampling density of the modification factors is reduced. The updated modification factors compile a dataset of 404 and can be found in section D.2. To clarify, this dataset still consists of four original profiles, but is sampled on different set of modification. The plot below shows the updated plot for the sampled data. Still, the model shows convergence towards a certain error but does not show an increase for larger profile quantities (Figure E.2b).



(a) Amount of profiles in train dataset and obtained MSE error

(b) Amount of profiles in sampled train dataset and obtained MSE error

Figure E.2: Data quantity analyses of multi-profile training data with and without sampling

F

Test datasets

F.1. Single-profile-based test dataset

F.1.1. Dataset base

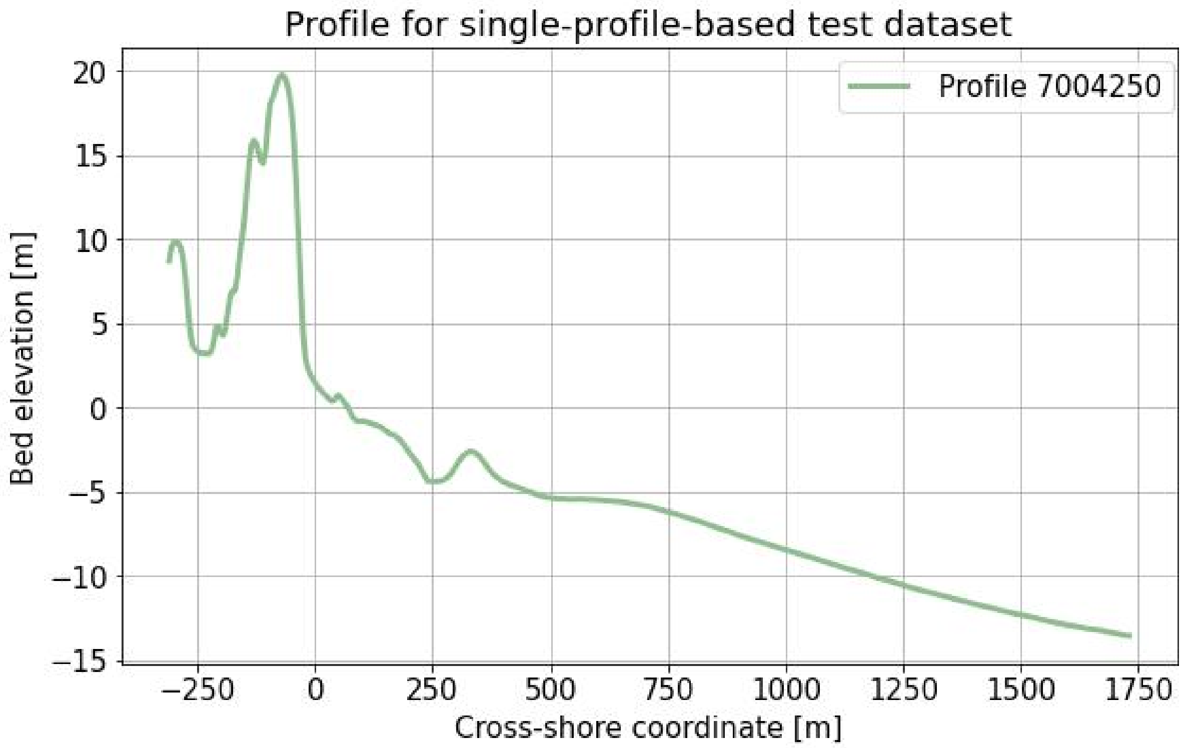


Figure F.1: Original profile for test dataset in exploration phase

F.1.2. Modification factors

Profile parameter	**Modification factors**	**Second modification factor**
Beach width	**0.75**, **0.9**, **1**, **1.1**, **1.25**	0.75, 1.25
Inter-tidal area width	**0.75**, **0.9**, **1.1**, **1.25**	0.75, 1.25
Nearshore width	**0.75**, **0.9**, **1.1**, **1.25**	0.75, 1.25
Dune width	**0.75**, **0.9**, **1.1**, **1.25**	-
Dune height	**0.75**, **0.9**, **1.1**, **1.25**	0.75, 1.25

Table F.1: Profile parameters and imposed modification factors

F.1.3. Dataset in standard cross-shore grid

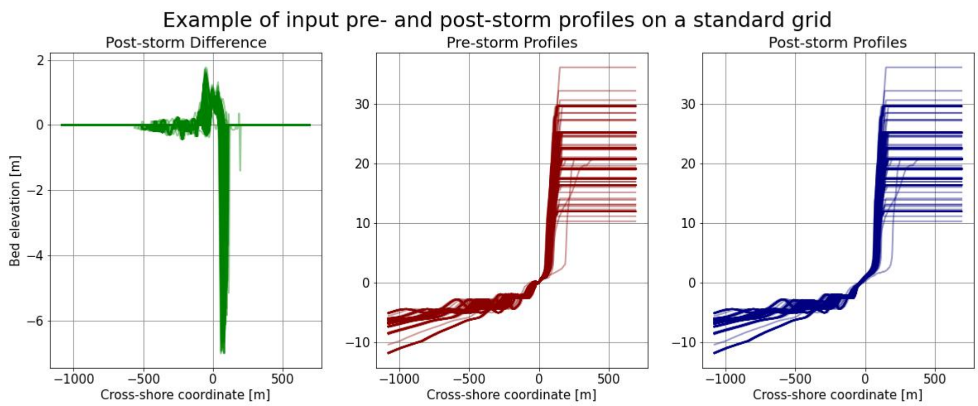


Figure F.2: Input and outputs profiles in standard grid configuration. The left figure indicates the post-storm difference, middle and right indicate the representative pre- and post-storm profiles.

F.1.4. DEV variability in dataset

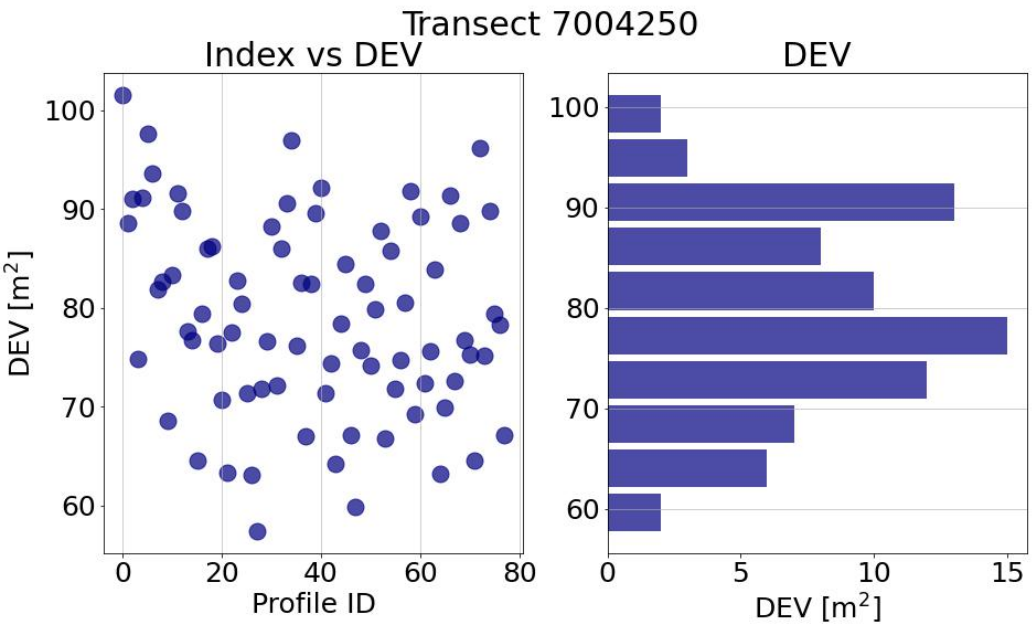


Figure F.3: The variability of DEV captured in the exploration test dataset. The DEV per profile id (left) and a histogram plot of the DEV data (right)

F.2. Holland coast test transects

F.2.1. Dataset base

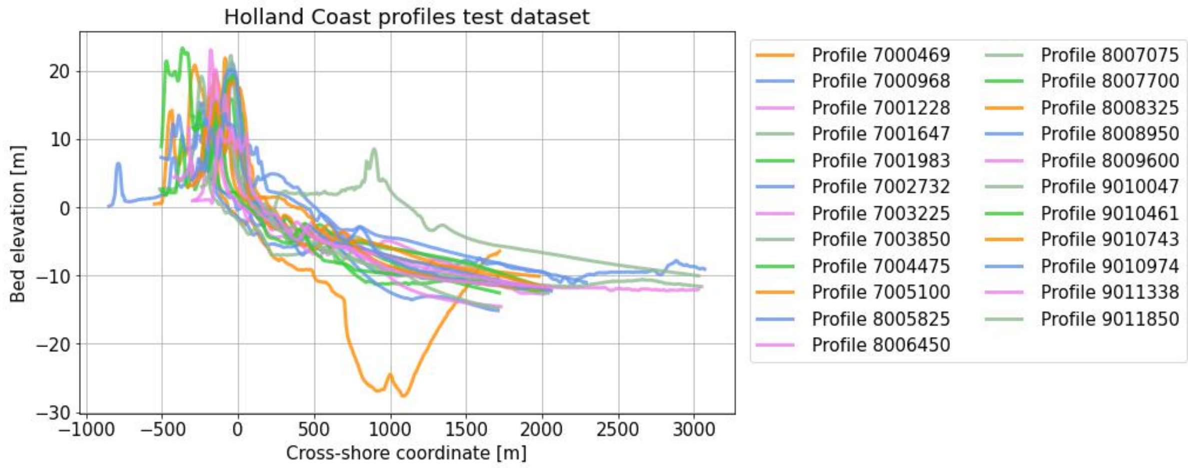


Figure F.4: Profiles in Holland Coast test dataset

F.2.2. Dataset in standard cross-shore grid

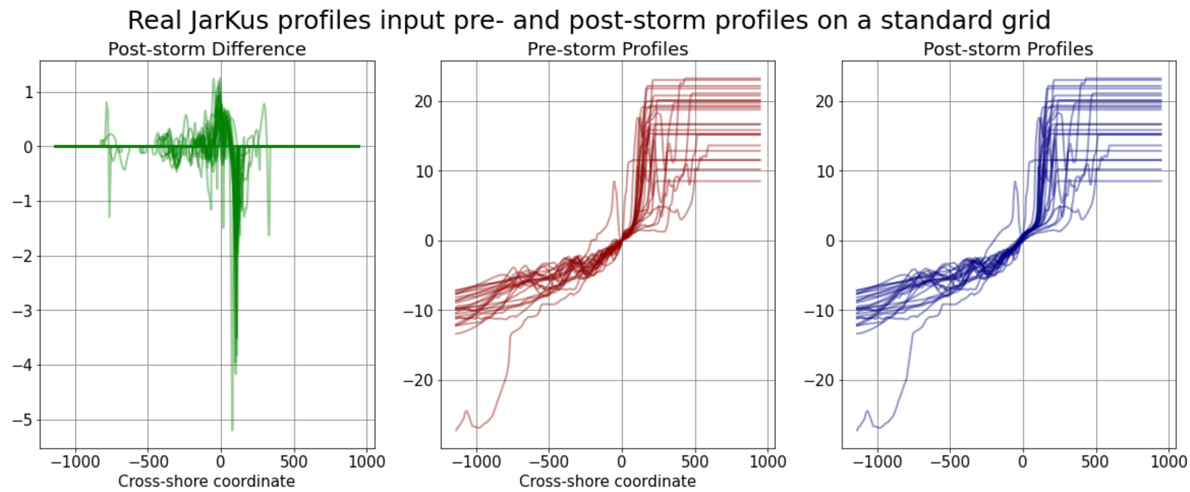


Figure F.5: The test dataset with actual profiles along the Holland Coast in the U-Net grid.

F.2.3. DEV variability in dataset

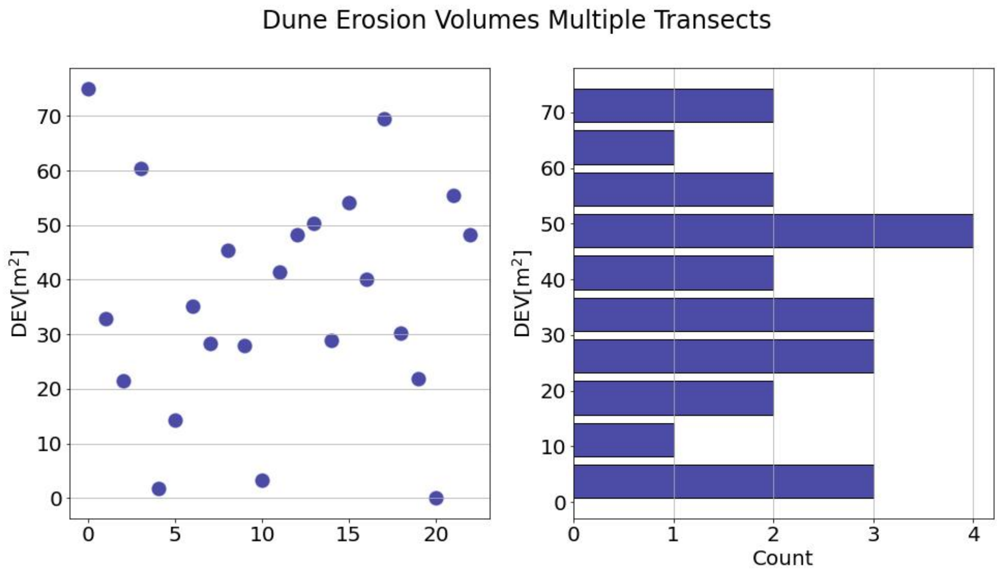
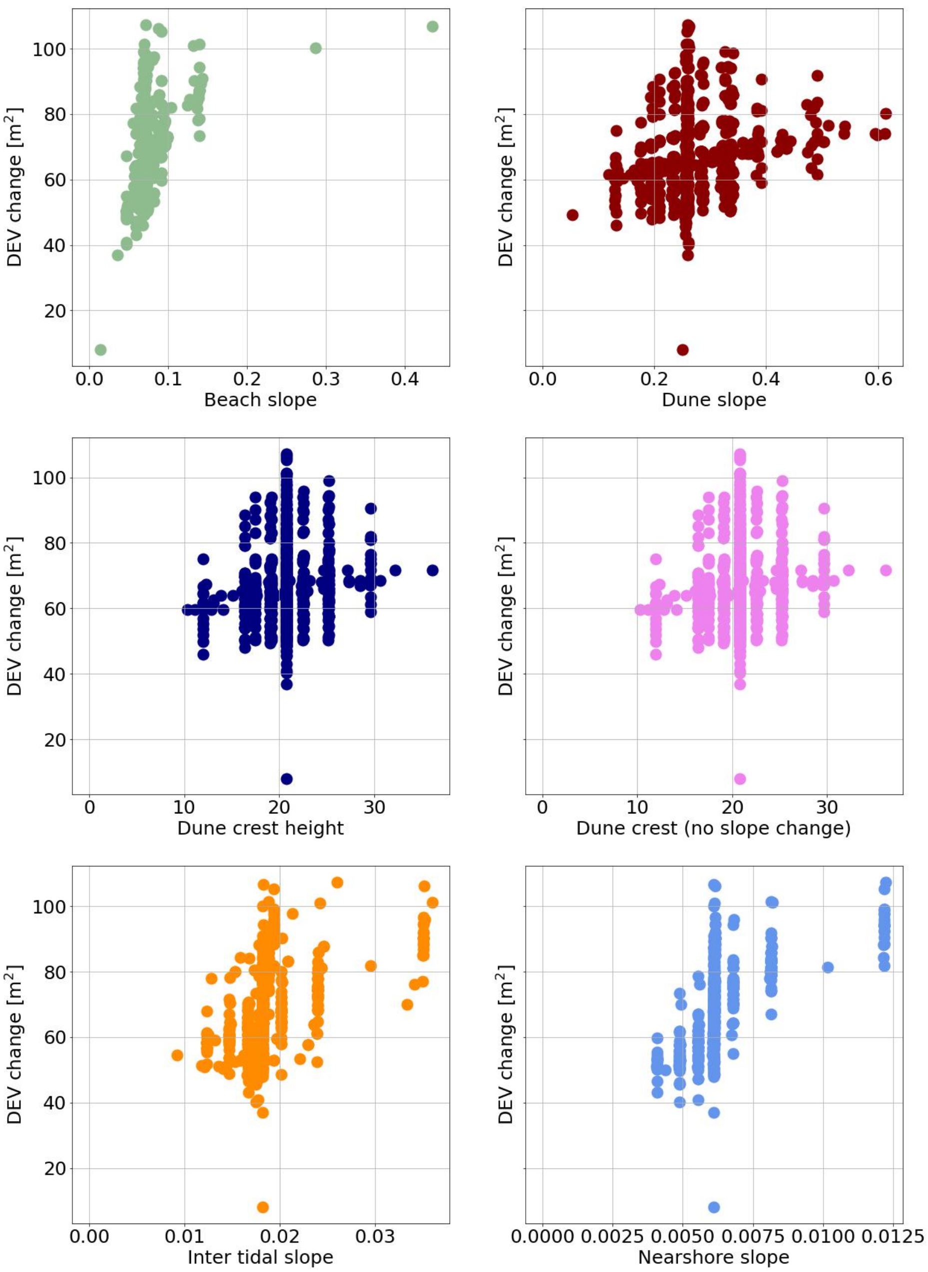


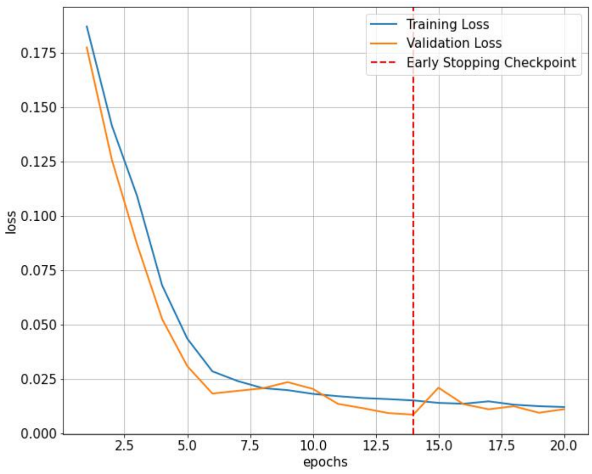
Figure F.6: The variability of DEV captured in the dataset. The DEV per profile id (left) and a histogram plot of the DEV data (right)

G
Parameter sensitivity analysis

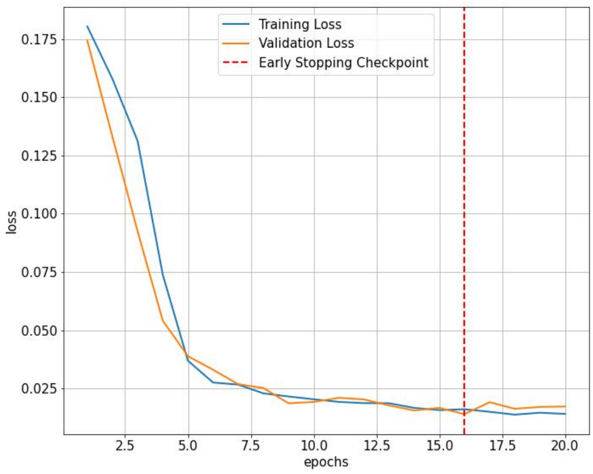


H
Hyperparameters U-Net

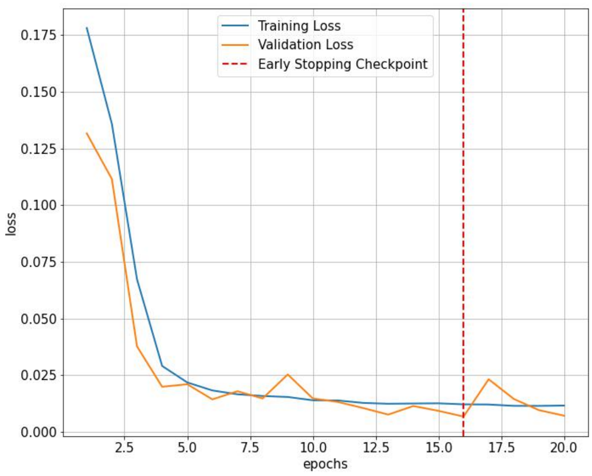
H.1. Learning rate



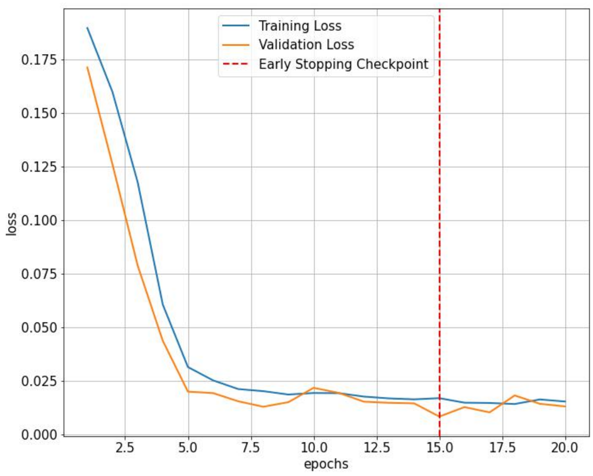
(a) Learning curve for LR = 0.1



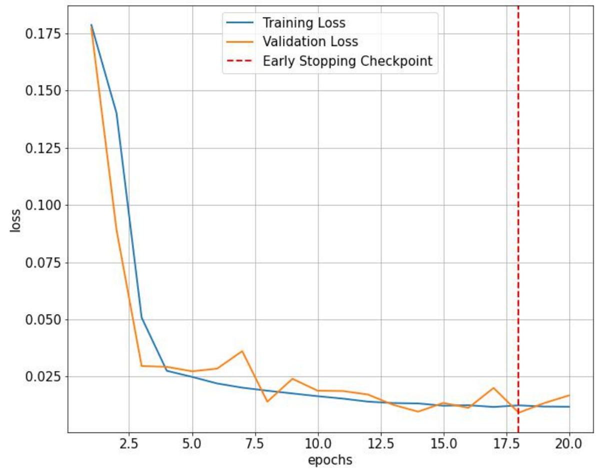
(b) Learning curve for LR = 0.01



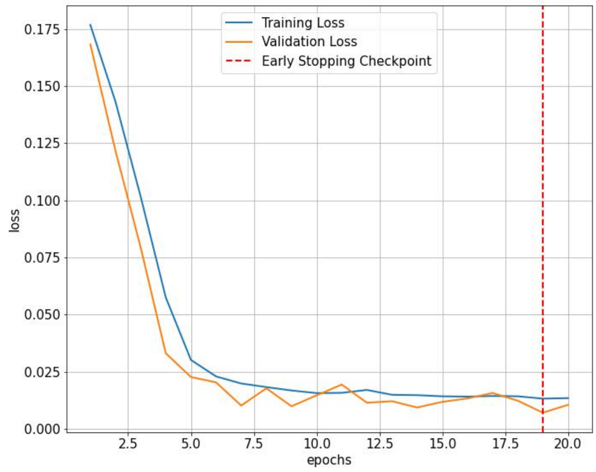
(a) Learning curve for LR = 0.001



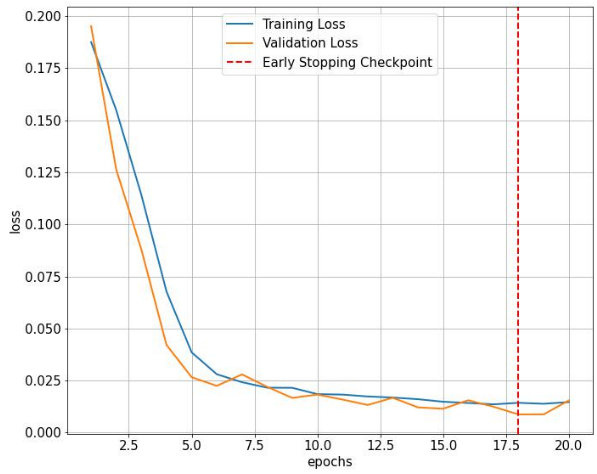
(b) Learning curve for LR = 0.00075



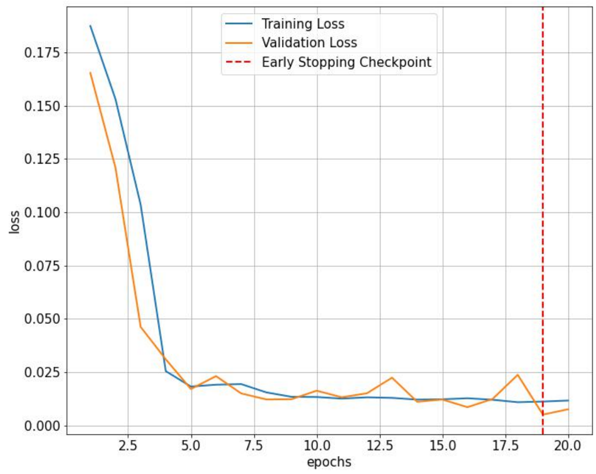
(a) Learning curve for LR = 0.0005



(b) Learning curve for LR = 0.00025



(a) Learning curve for LR = 0.0001



(b) Learning curve for LR = 0.00001

H.2. Epoch size

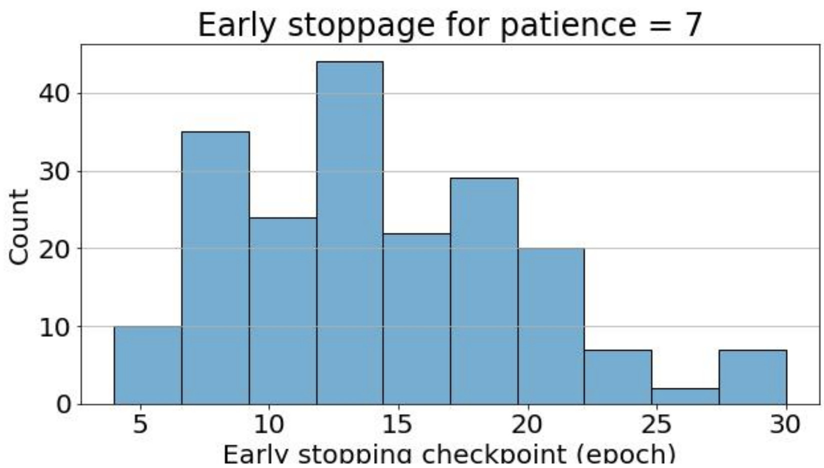


Figure H.5: Number of epochs before early stopping is induced

H.3. Variability in U-Net predictions

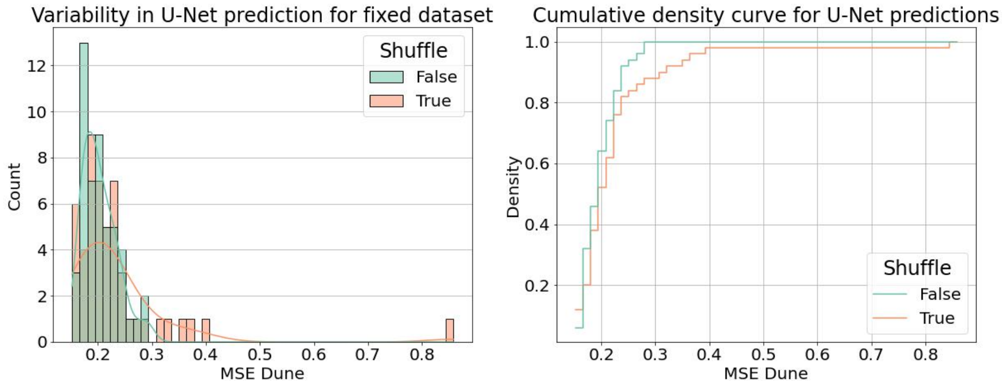


Figure H.6: Variability in U-Net predictions with a fixed U-Net structure.

H.4. Network depth and Kernel size for training dataset

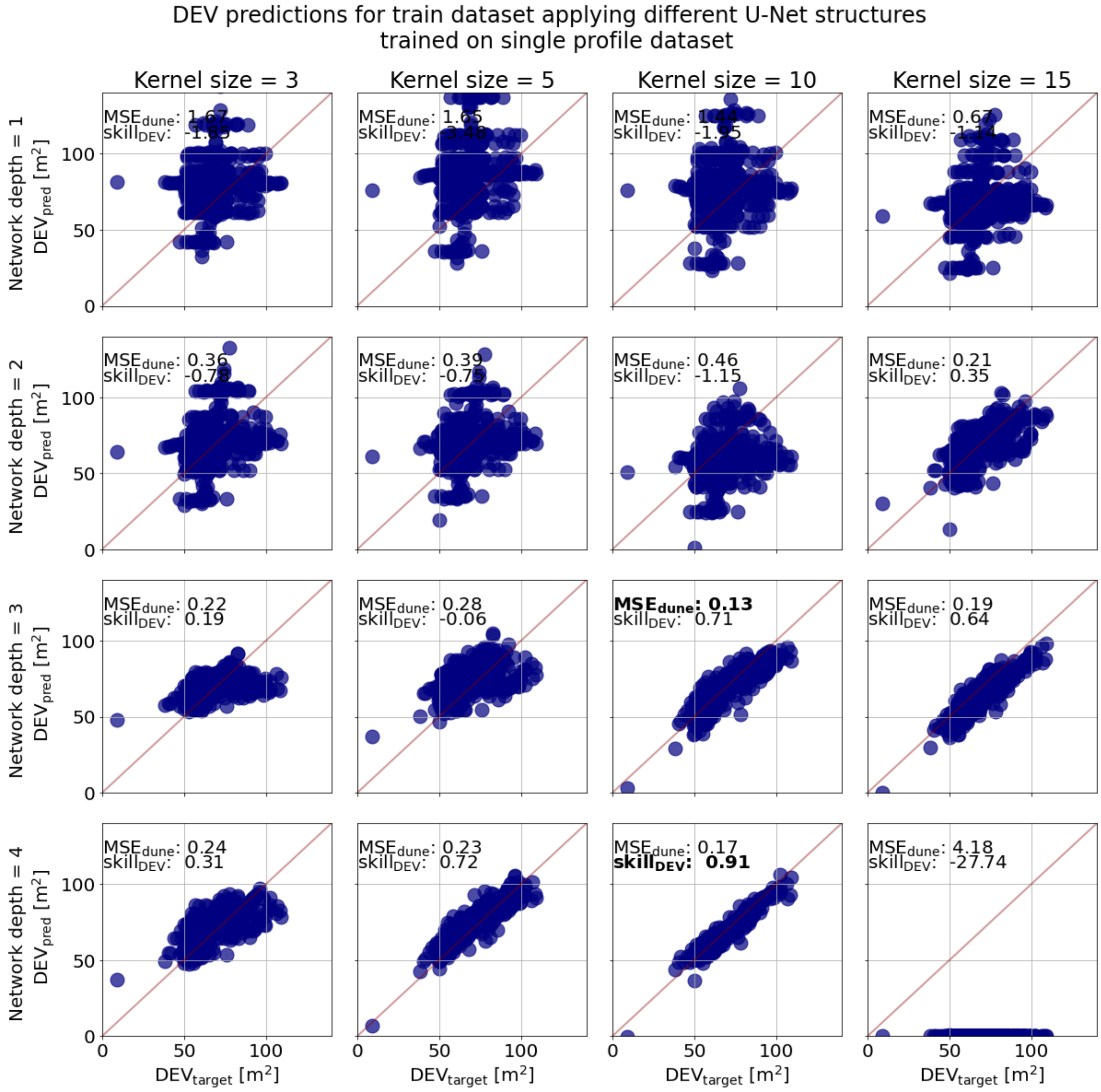


Figure H.7: Caption

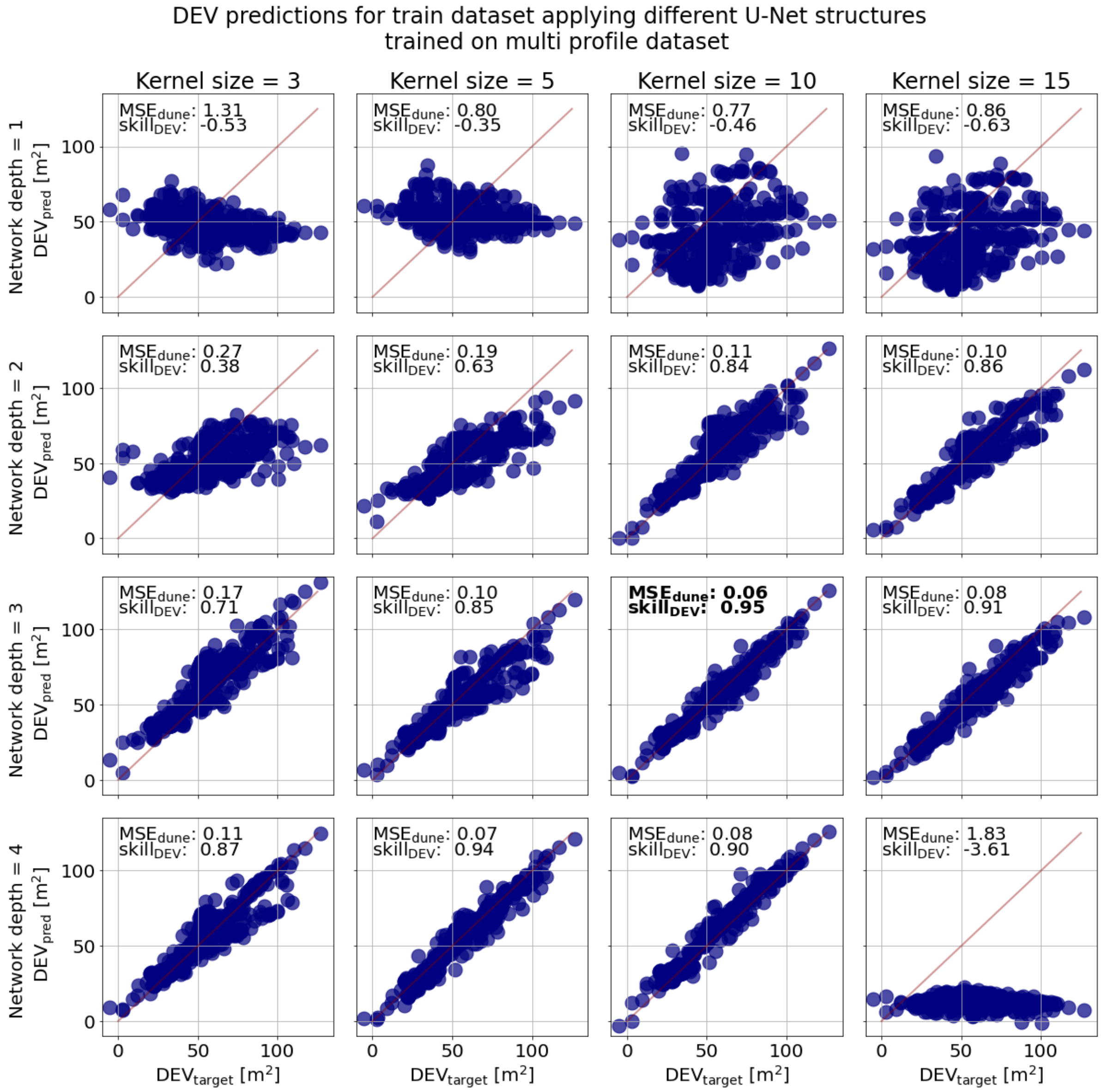


Figure H.8: Caption

I

Profile predictions U-Net

Dune shape with different x-grid spacing

See next page

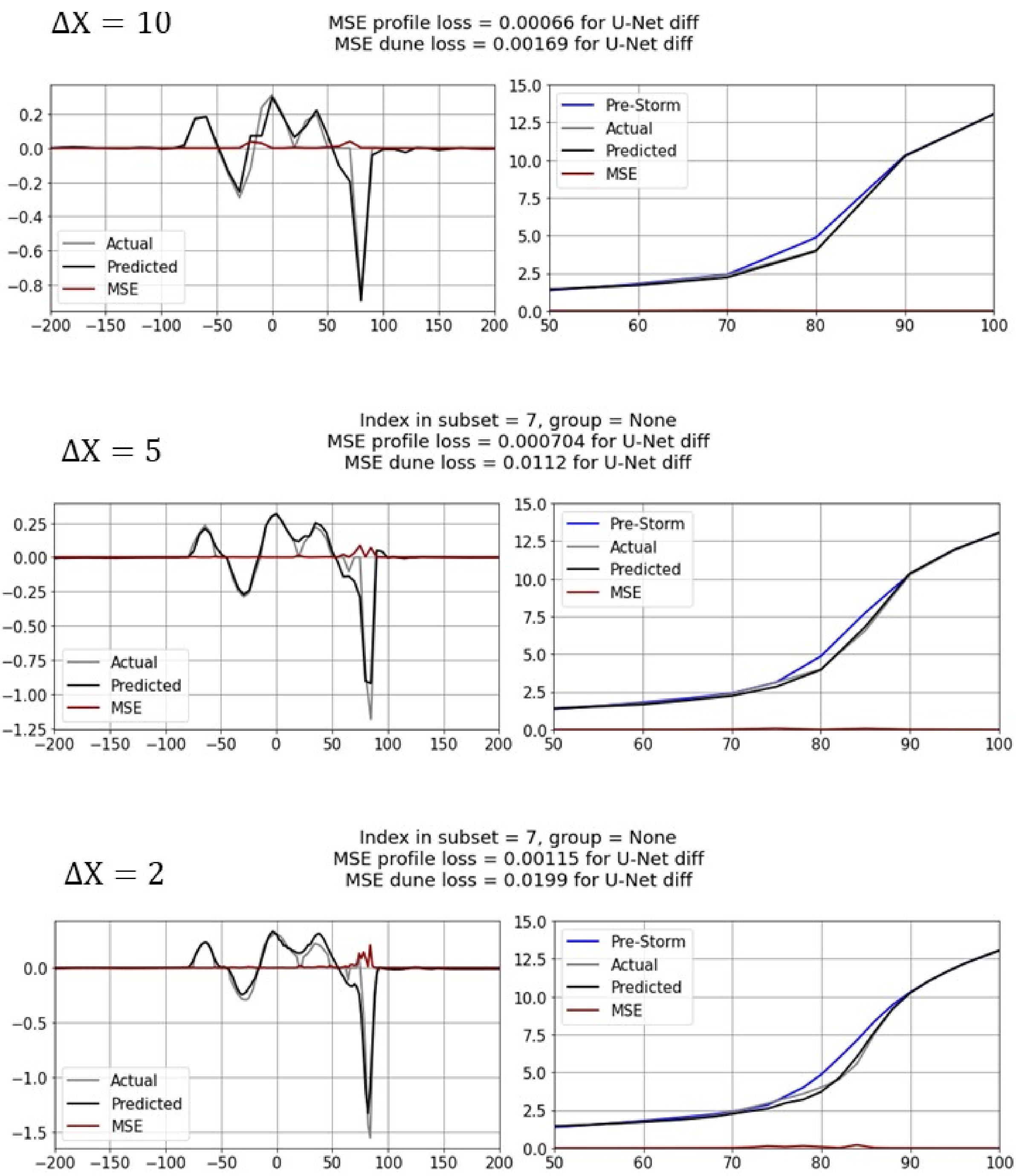


Figure I.1: Dune shape for different Δx

Nearshore modifications, dataset single profile, Network depth = 3, Kernel size = 10

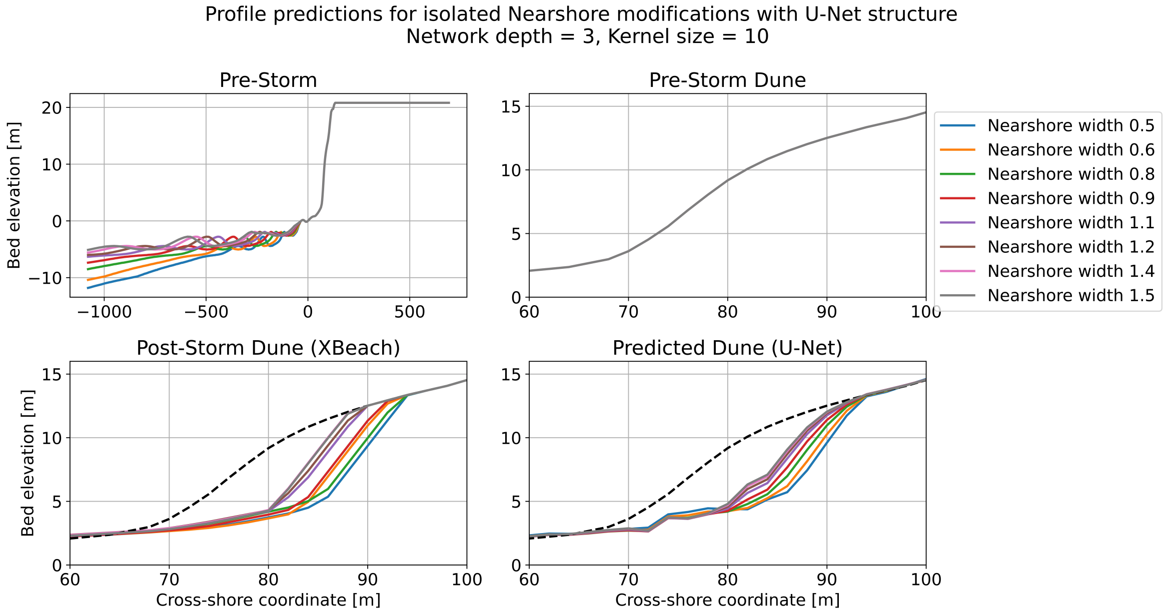


Figure I.2: Actual and target dune shapes for different nearshore slopes for increased kernel size and network depth

Beach modifications, dataset single profile, Network depth = 2, Kernel size = 3

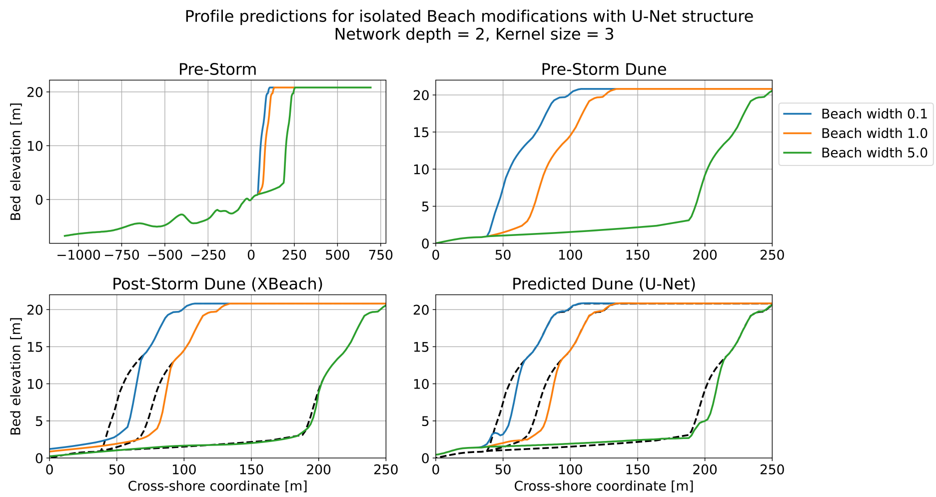


Figure I.3: Actual and target dune shapes for different beach slopes for a shallow U-Net structure

Beach modifications, dataset single profile, Network depth = 3, Kernel size = 10

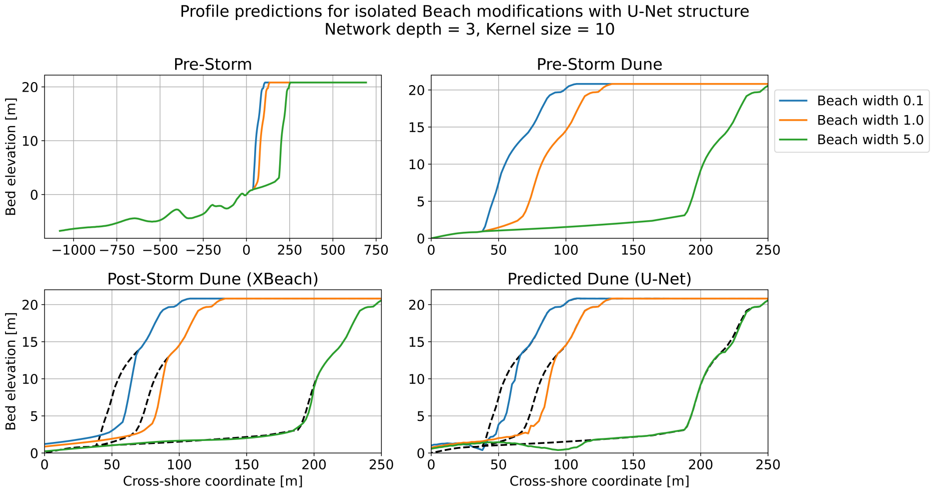


Figure I.4: Actual and target dune shapes for different beach slopes for for increased kernel size and network depth

Double penalty effect

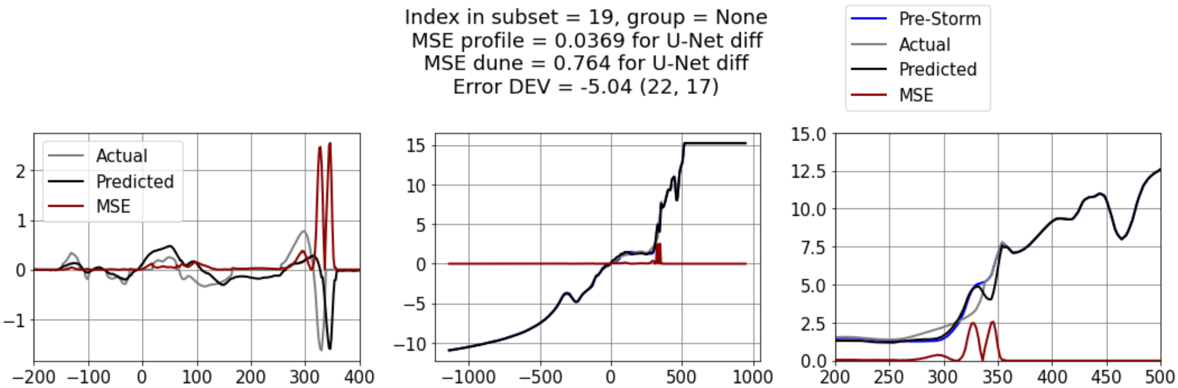


Figure I.5: Predicted and target differences and dune shapes for profile in test data showing the double penalty effect that comes with the MSE.

U-Net (ND3, CS32, KS10), multiple profiles in training data, actual profiles in test data

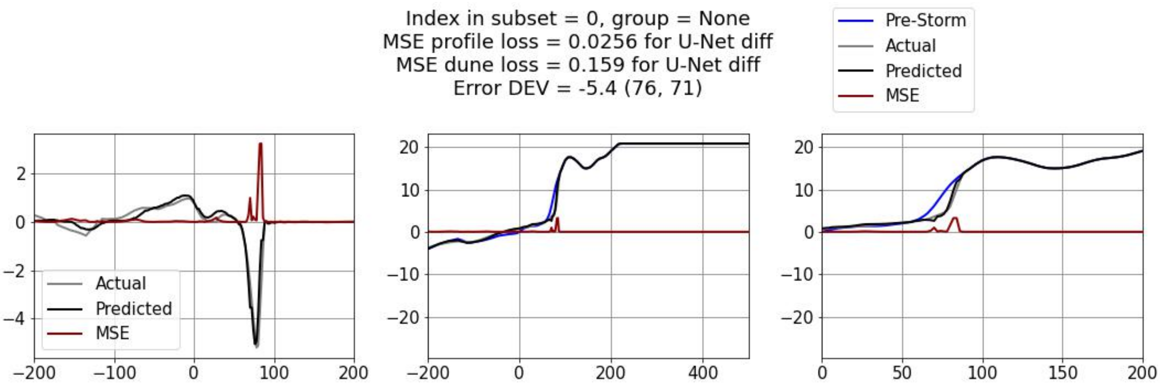


Figure I.6: Predicted and target differences and dune shapes for profile in test data

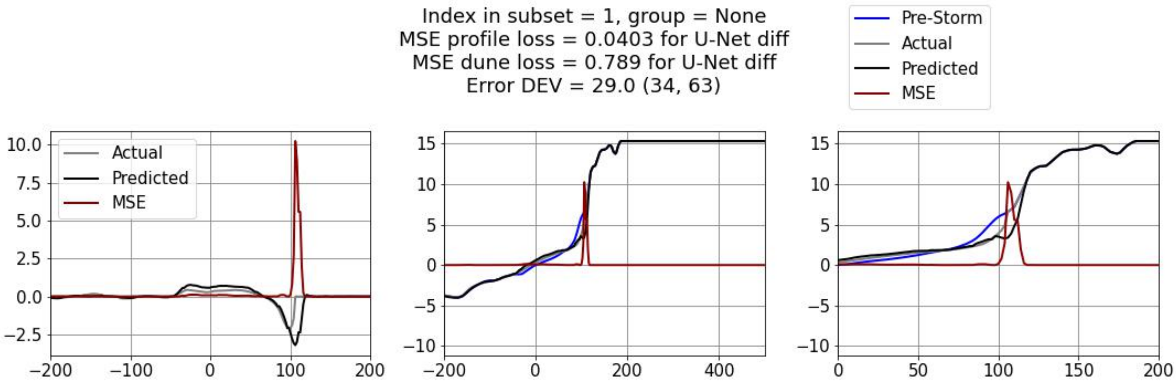


Figure I.7: Predicted and target differences and dune shapes for profile in test data

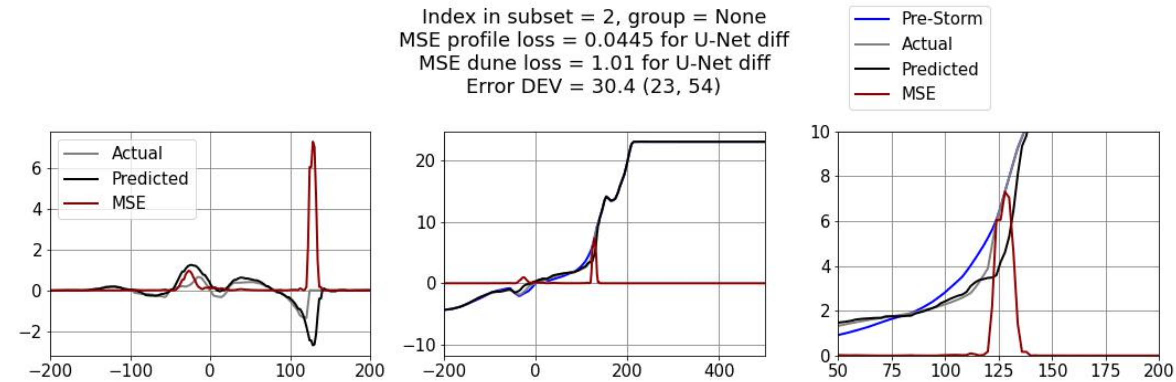


Figure I.8: Predicted and target differences and dune shapes for profile in test data

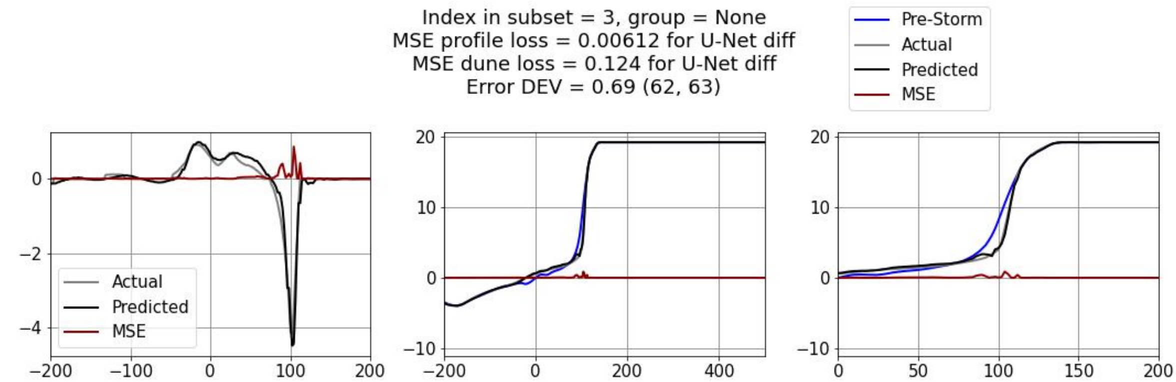


Figure I.9: Predicted and target differences and dune shapes for profile in test data

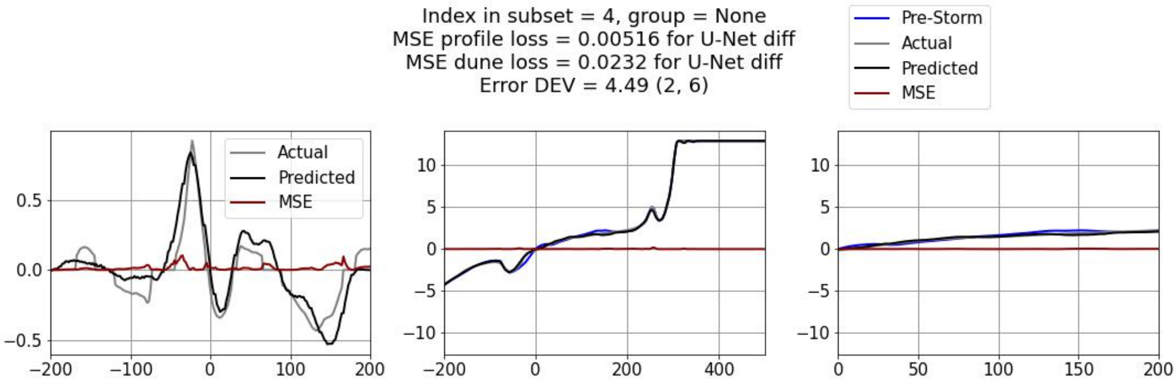


Figure I.10: Predicted and target differences and dune shapes for profile in test data

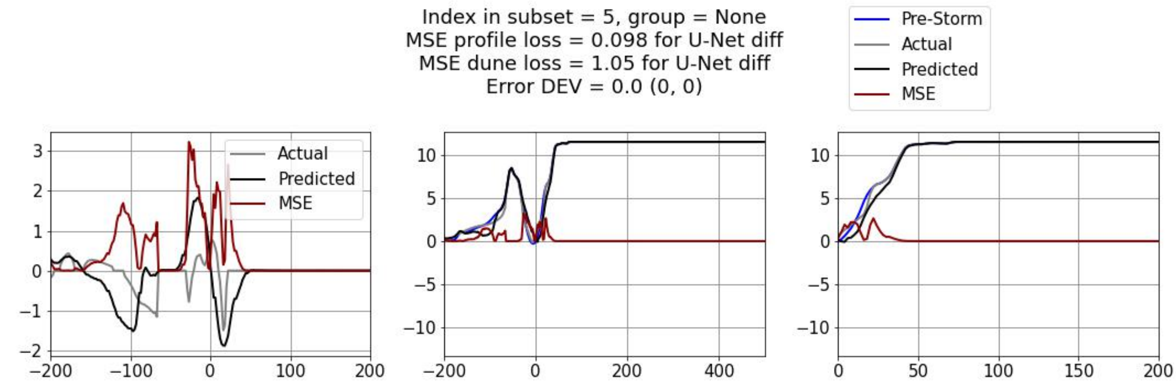


Figure I.11: Predicted and target differences and dune shapes for profile in test data

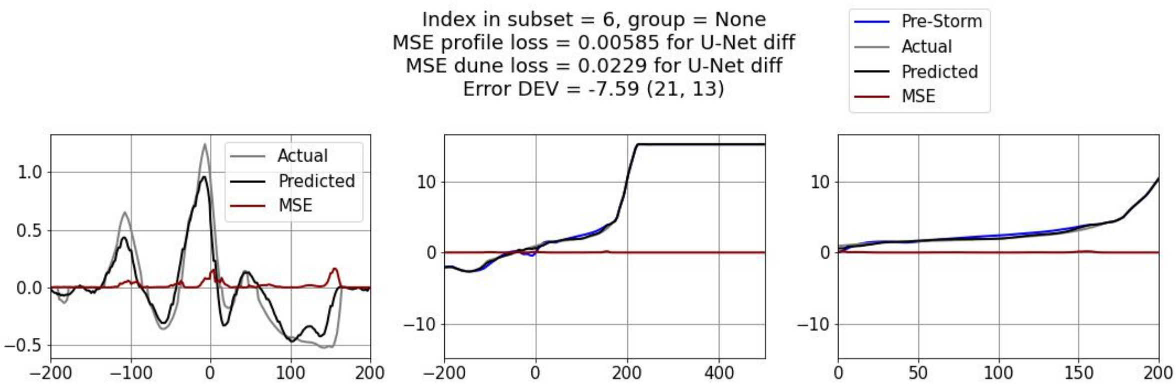


Figure I.12: Predicted and target differences and dune shapes for profile in test data

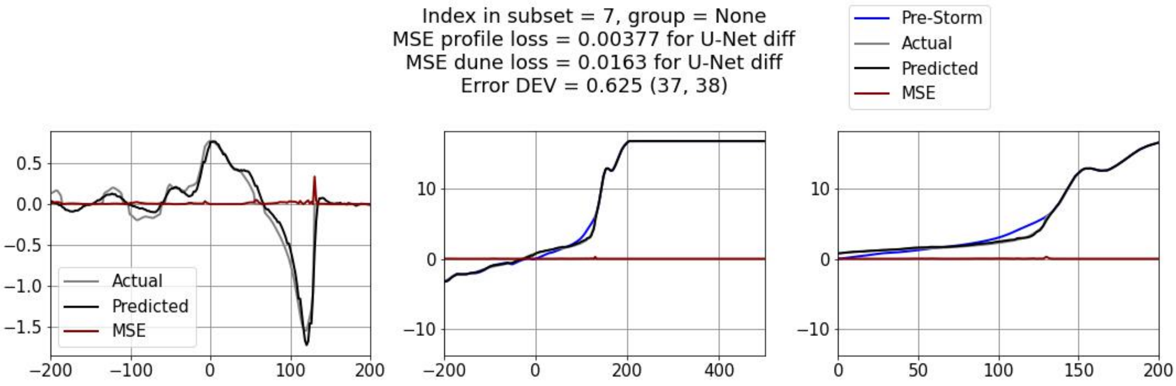


Figure I.13: Predicted and target differences and dune shapes for profile in test data

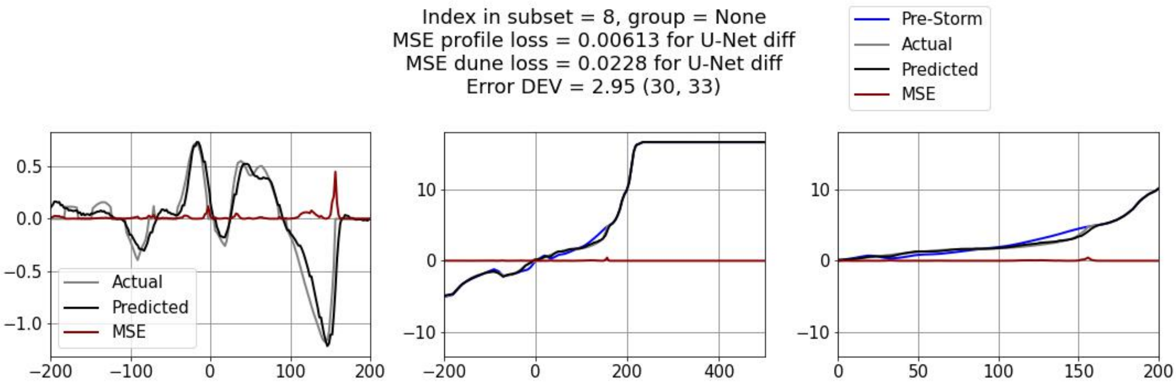


Figure I.14: Predicted and target differences and dune shapes for profile in test data

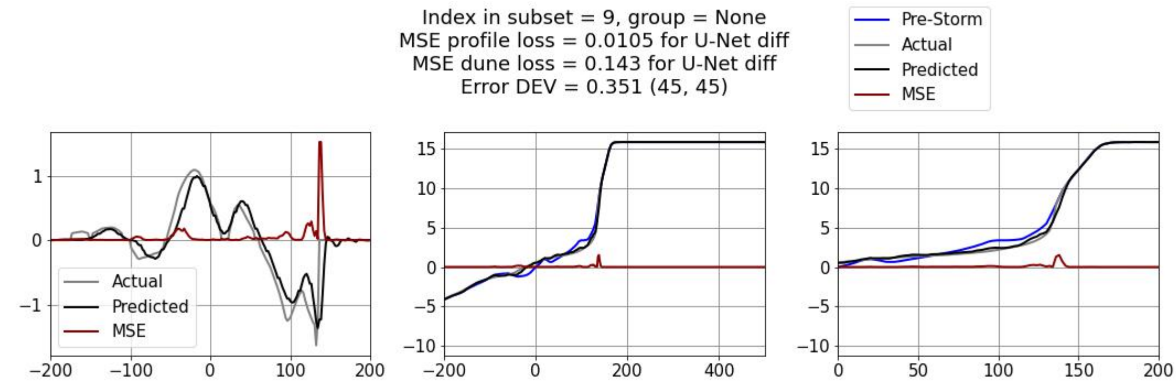


Figure I.15: Predicted and target differences and dune shapes for profile in test data

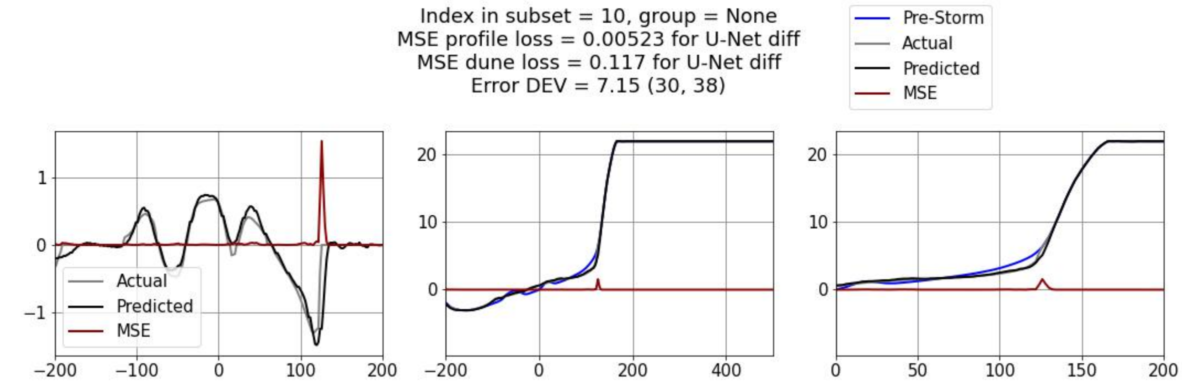


Figure I.16: Predicted and target differences and dune shapes for profile in test data

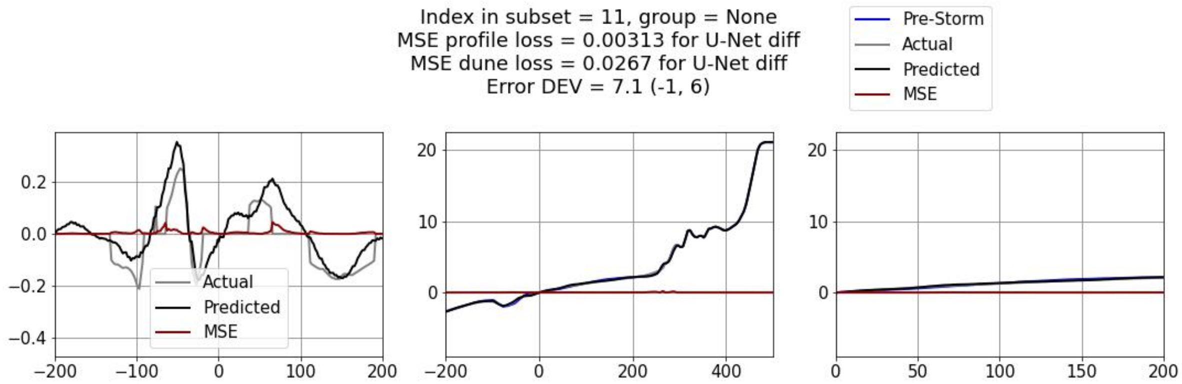


Figure I.17: Predicted and target differences and dune shapes for profile in test data

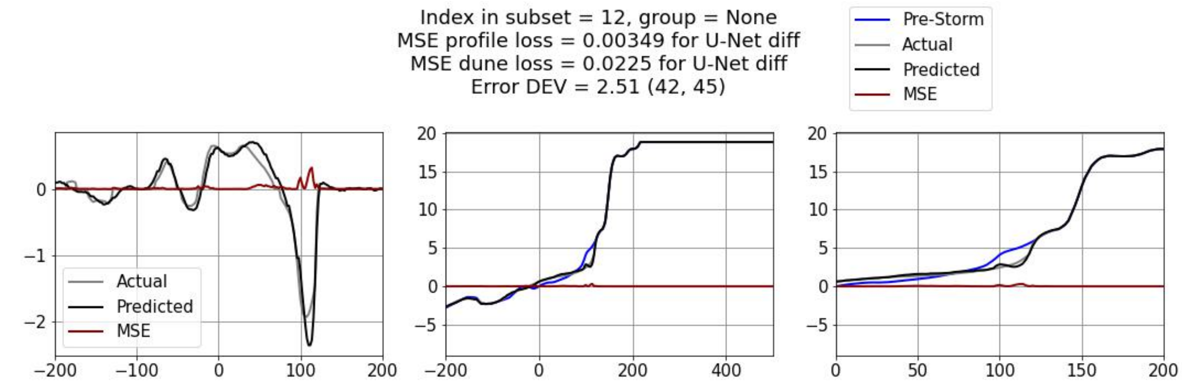


Figure I.18: Predicted and target differences and dune shapes for profile in test data

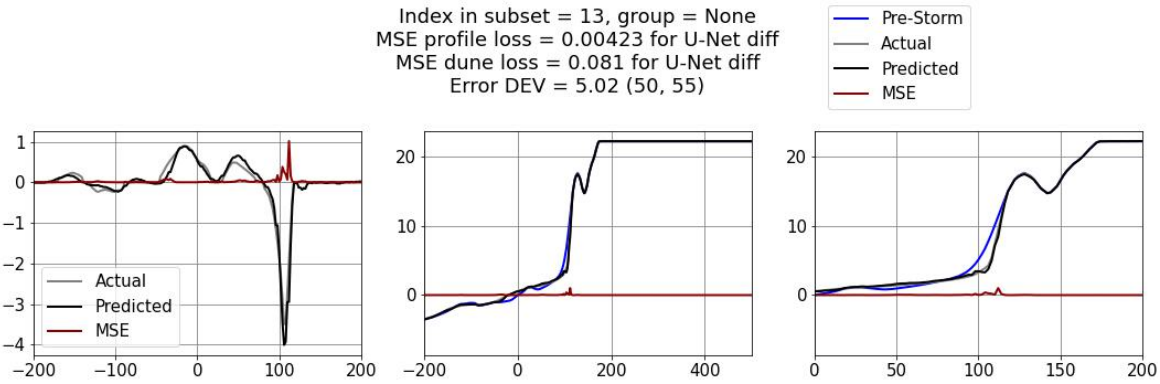


Figure I.19: Predicted and target differences and dune shapes for profile in test data

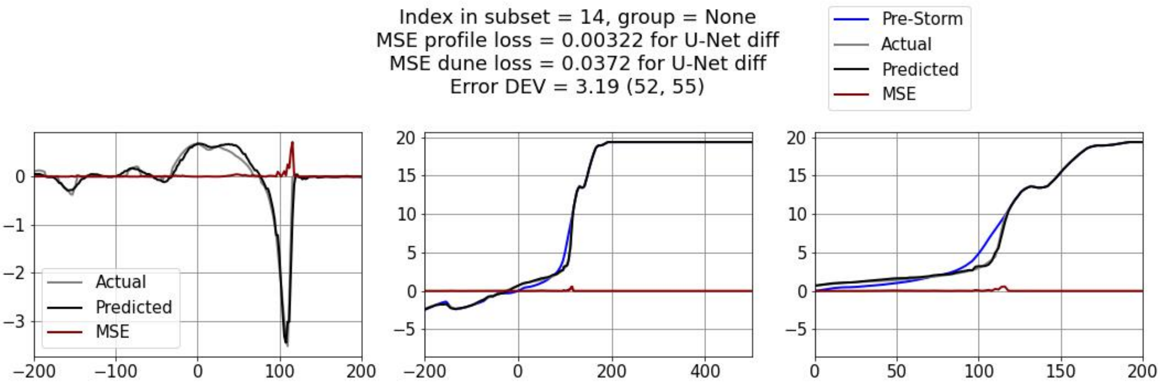


Figure I.20: Predicted and target differences and dune shapes for profile in test data

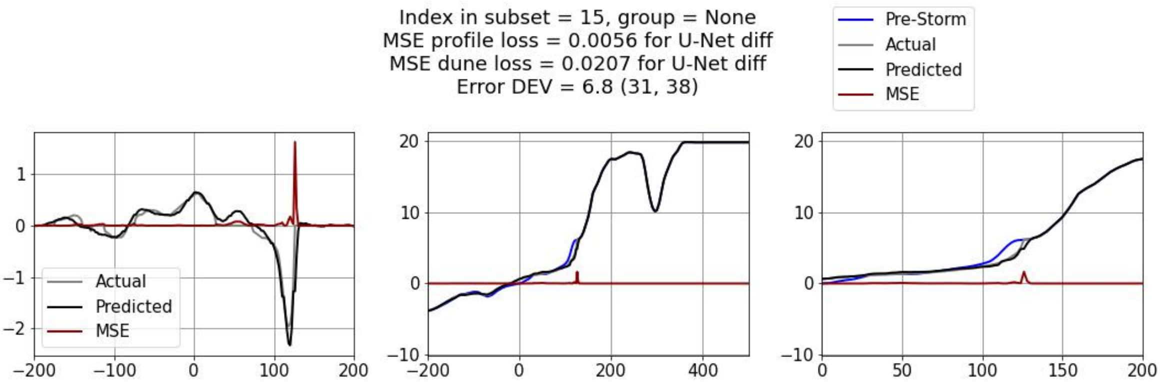


Figure I.21: Predicted and target differences and dune shapes for profile in test data

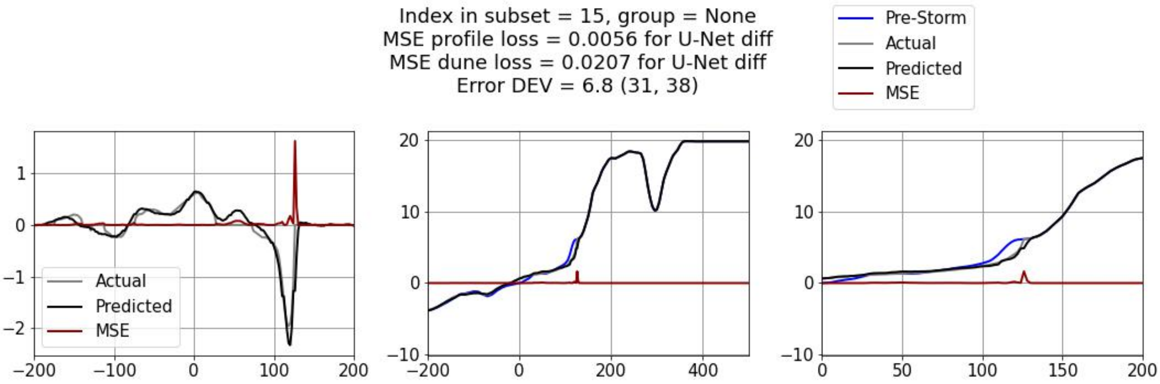


Figure I.22: Predicted and target differences and dune shapes for profile in test data

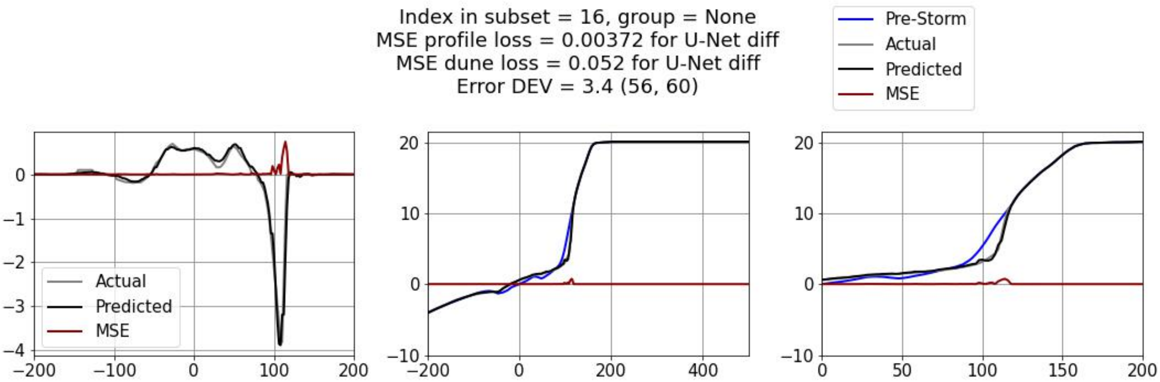


Figure I.23: Predicted and target differences and dune shapes for profile in test data

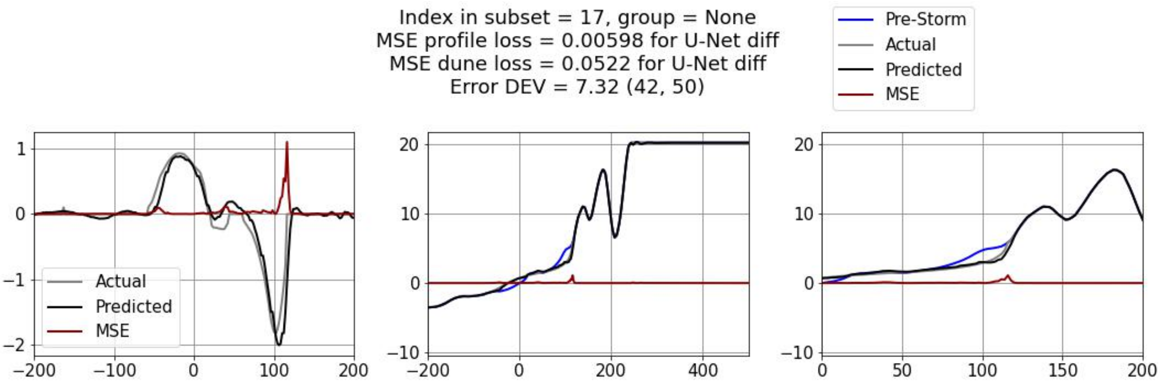


Figure I.24: Predicted and target differences and dune shapes for profile in test data

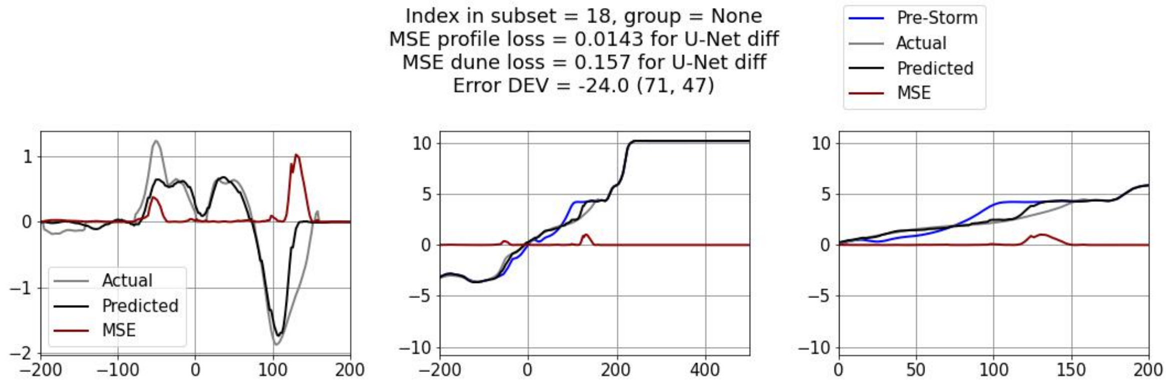


Figure I.25: Predicted and target differences and dune shapes for profile in test data

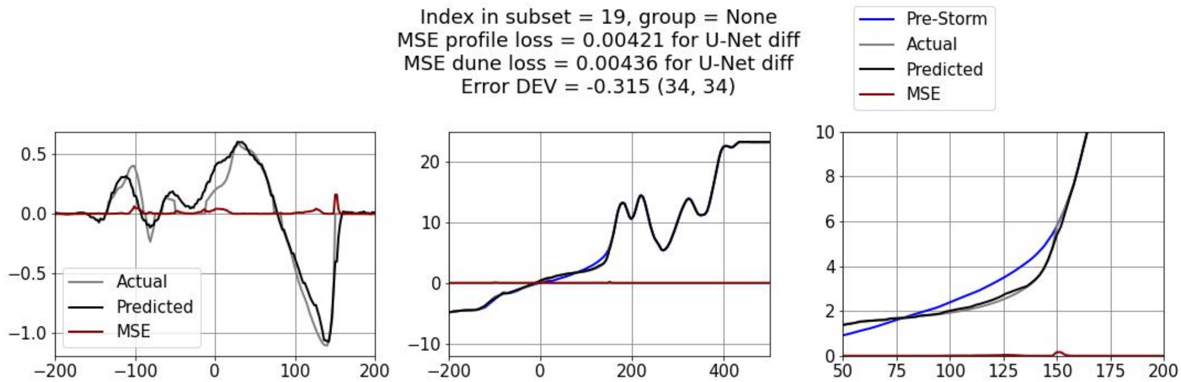


Figure I.26: Predicted and target differences and dune shapes for profile in test data

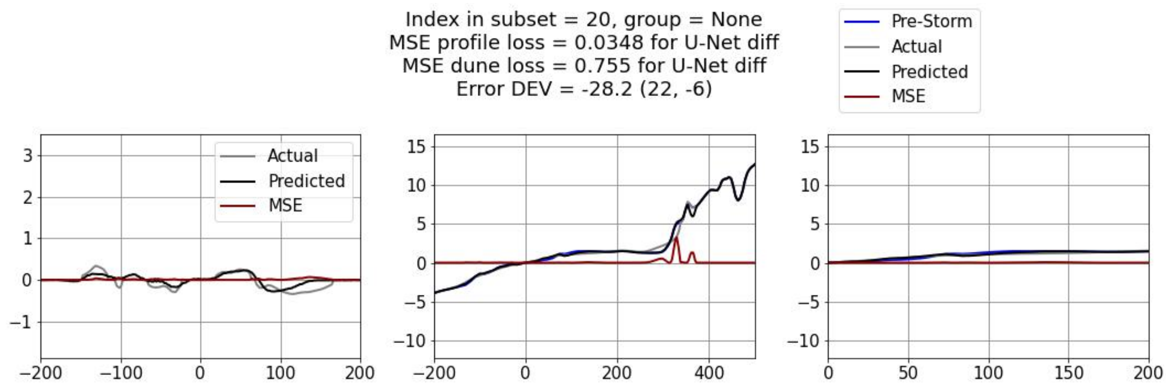


Figure I.27: Predicted and target differences and dune shapes for profile in test data

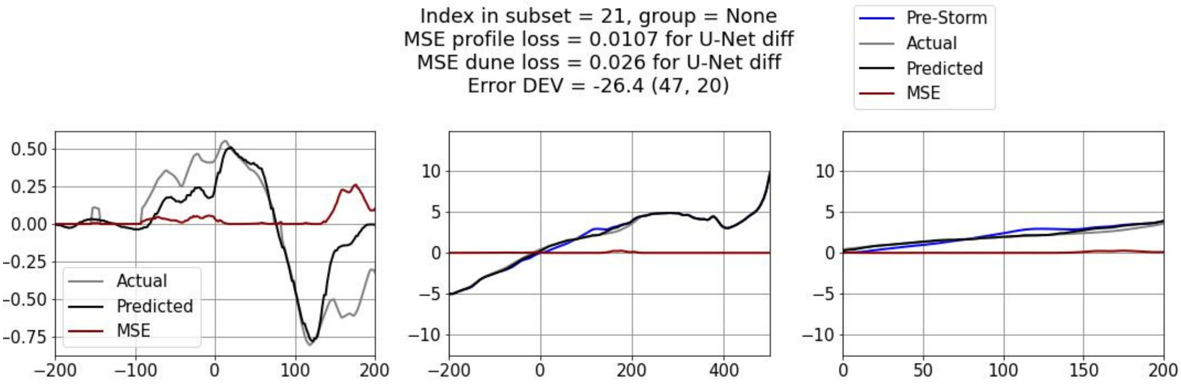


Figure I.28: Predicted and target differences and dune shapes for profile in test data

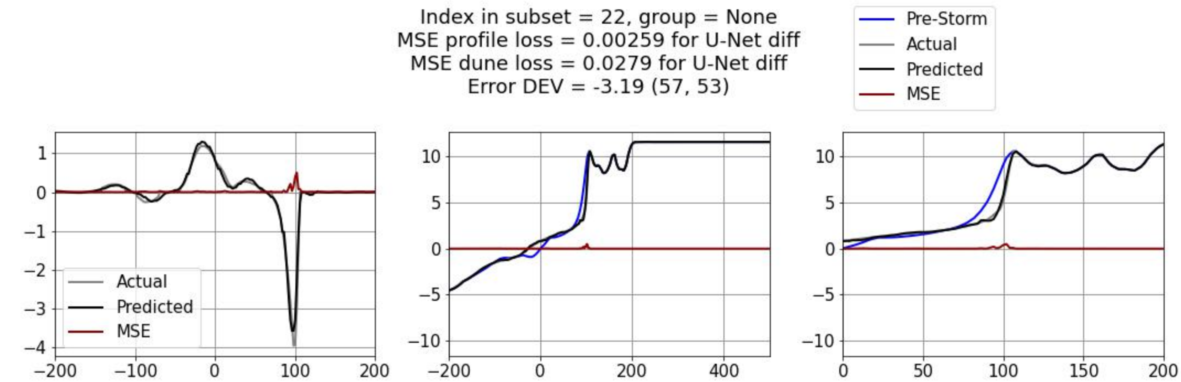


Figure I.29: Predicted and target differences and dune shapes for profile in test data

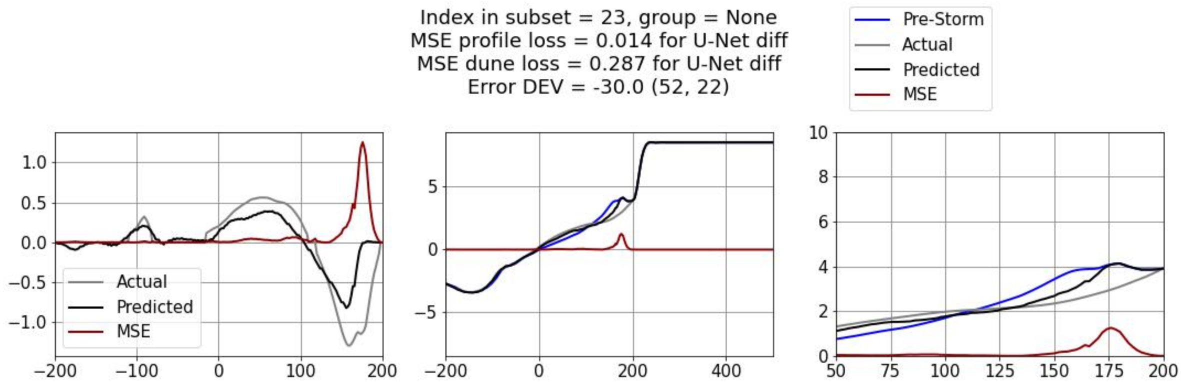


Figure I.30: Predicted and target differences and dune shapes for profile in test data

J
Calculation of DEV

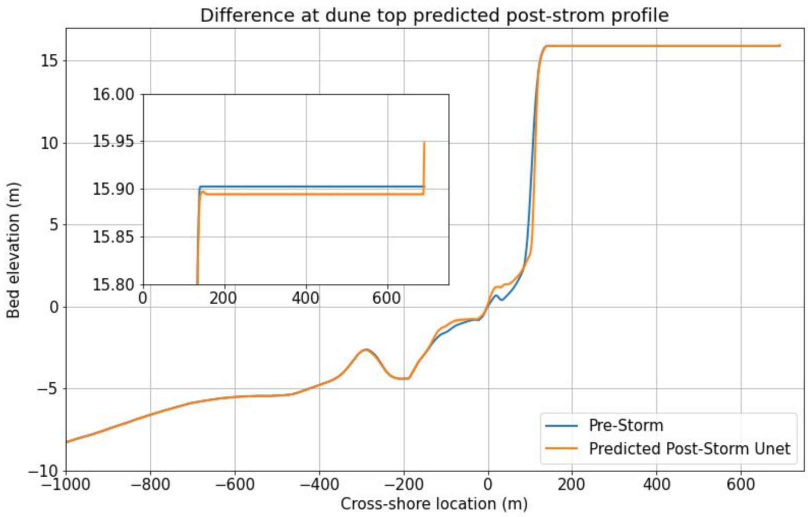


Figure J.1: Difference at the dune top

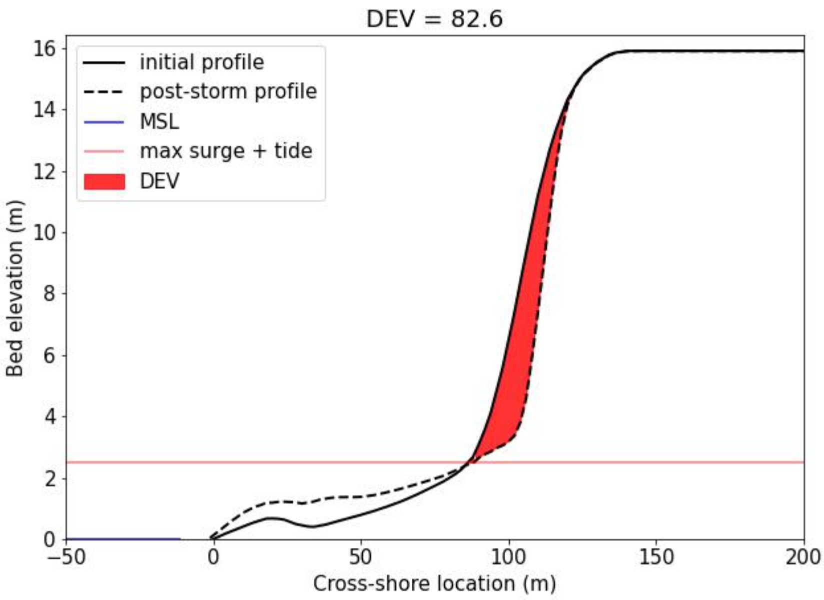


Figure J.2: DEV with regular method

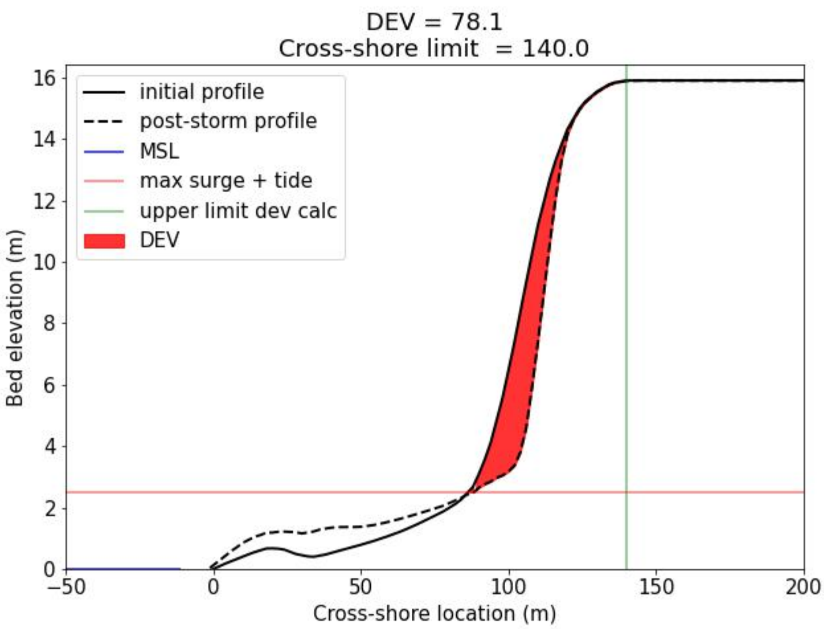
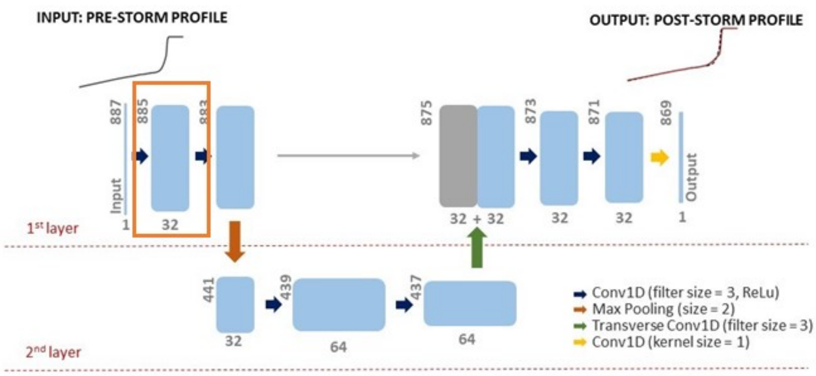


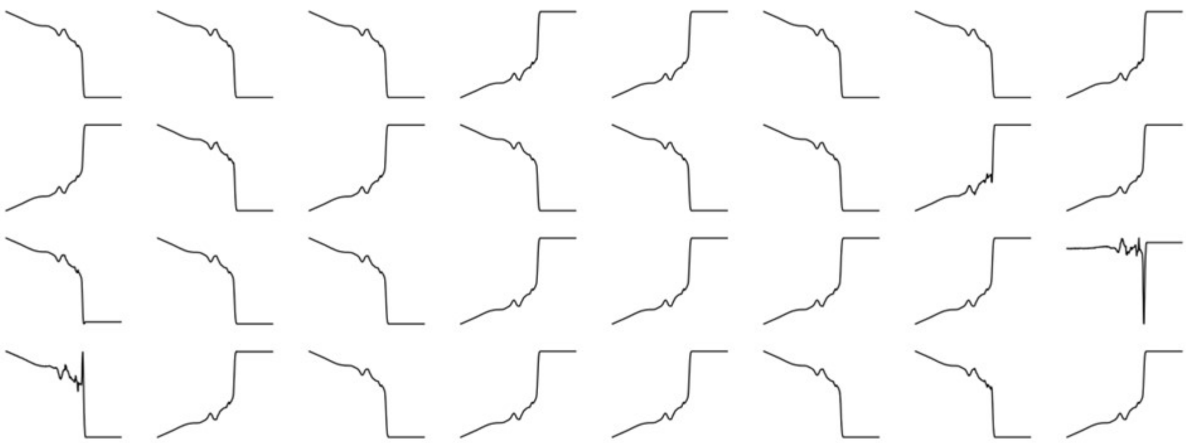
Figure J.3: DEV with cross-shore boundary

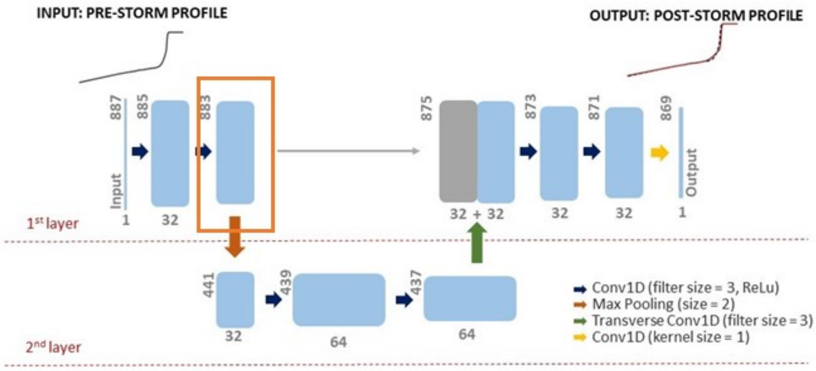
K

Feature maps U-Net

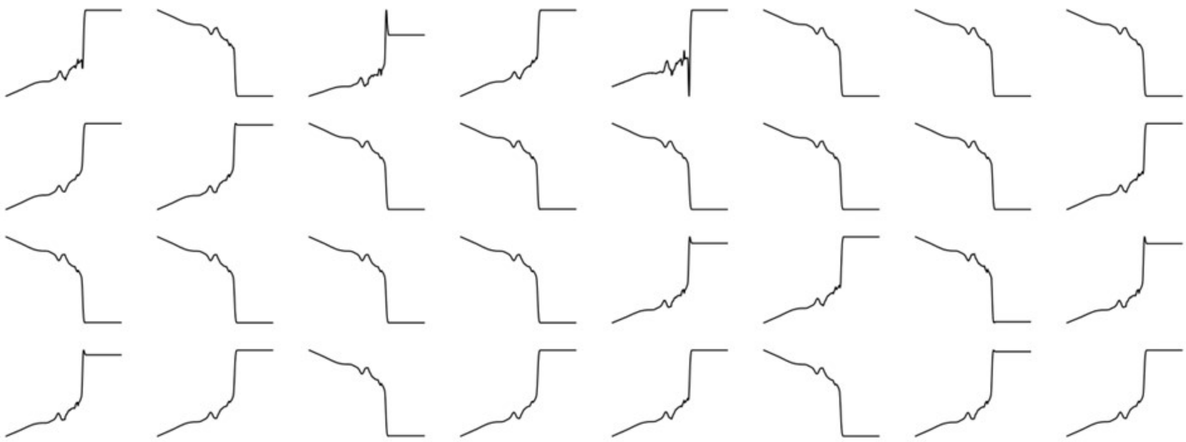


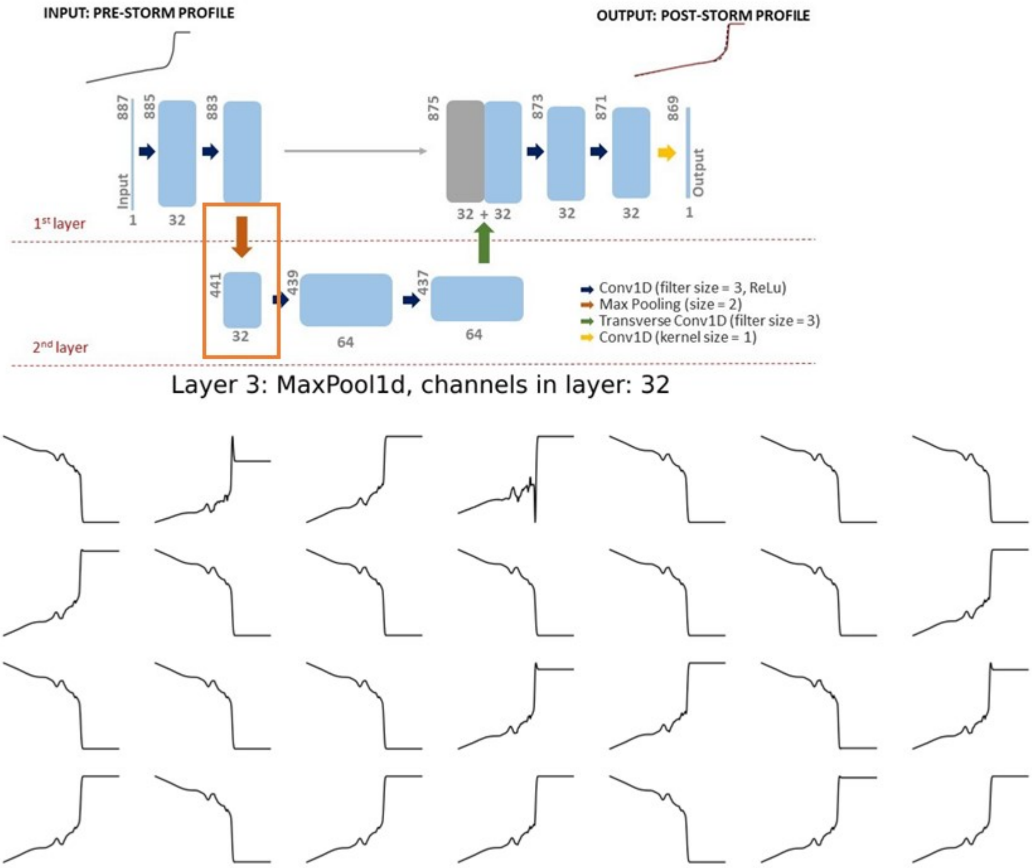
Layer 1: Conv1d, channels in layer: 32

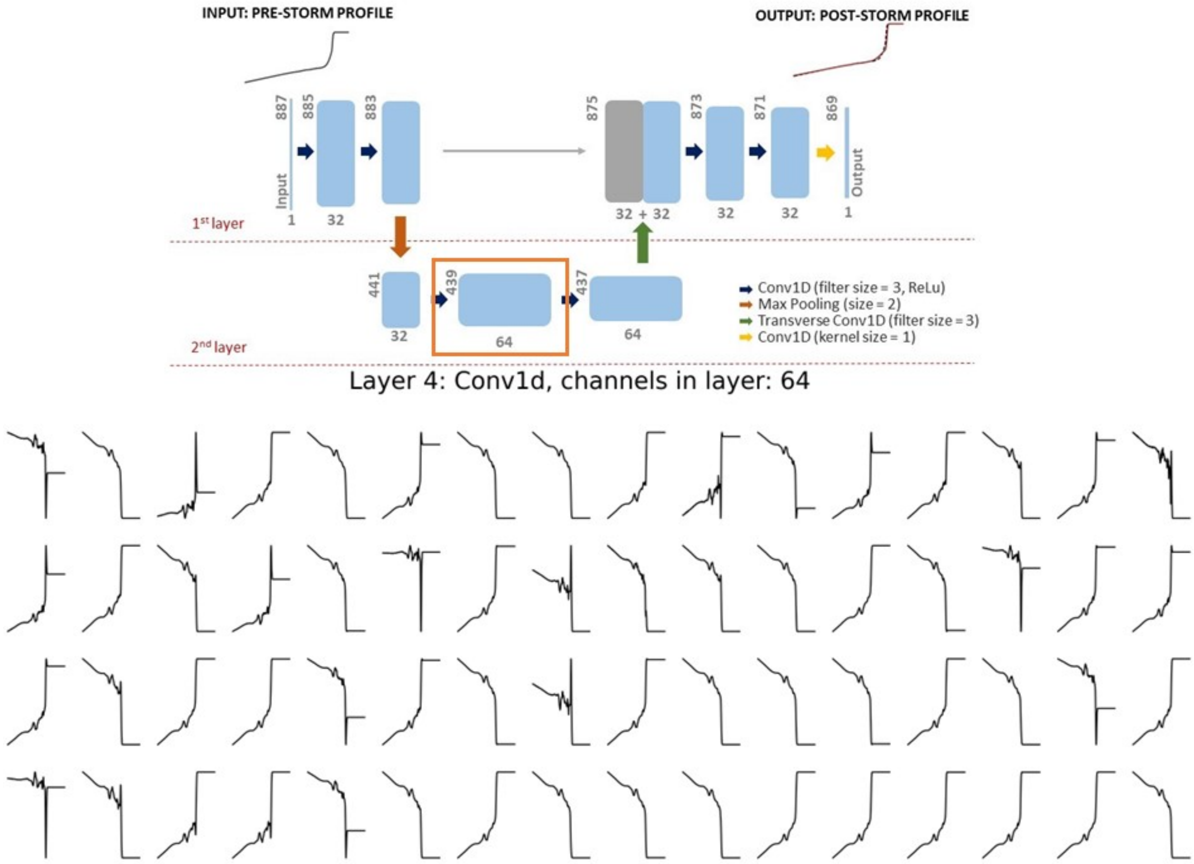


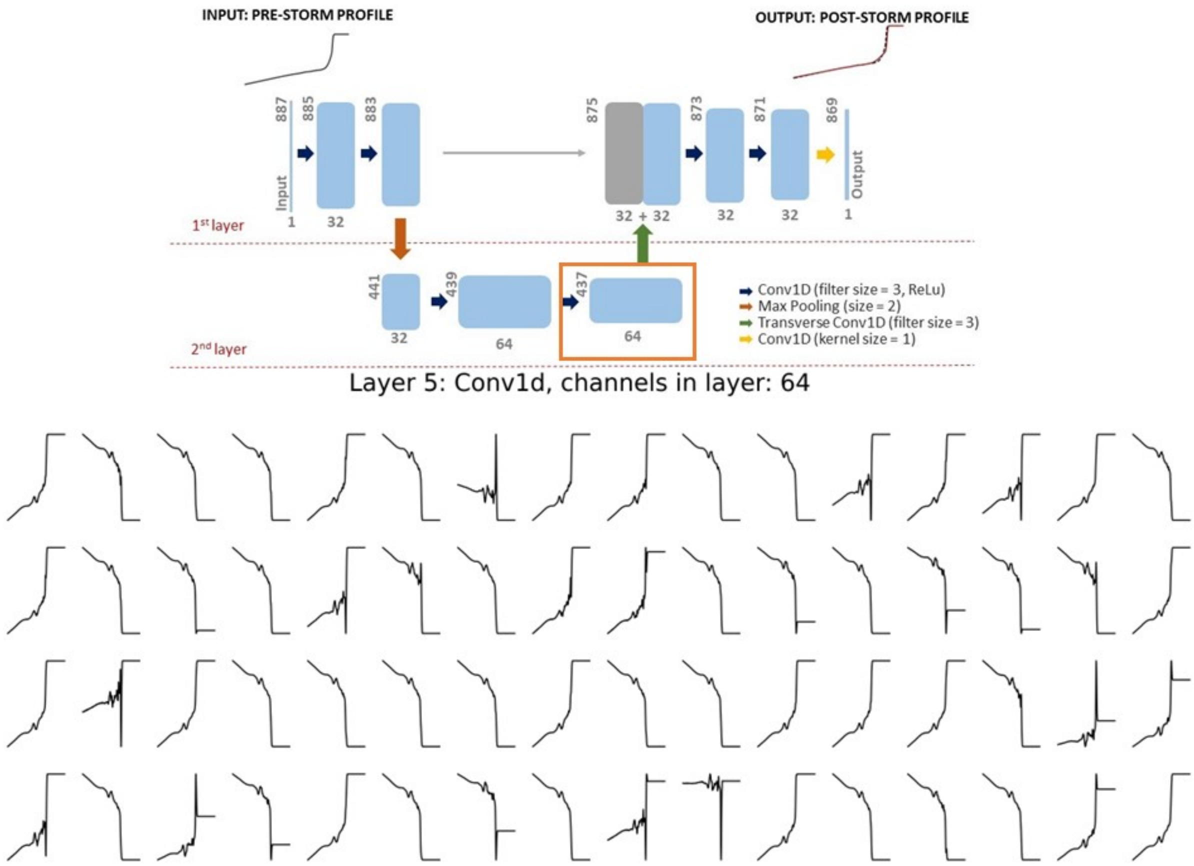


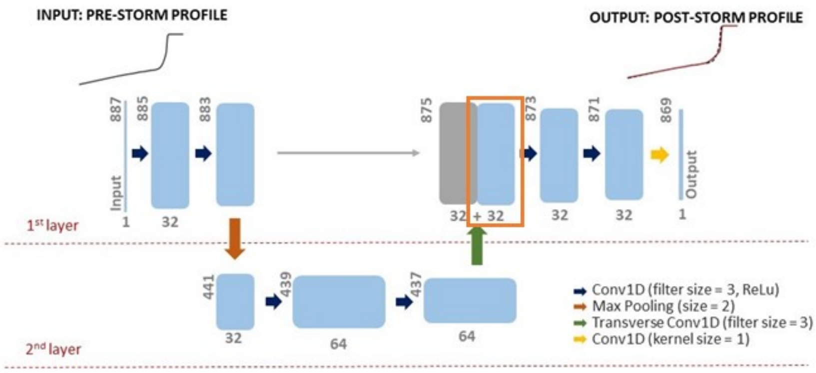
Layer 2: Conv1d, channels in layer: 32

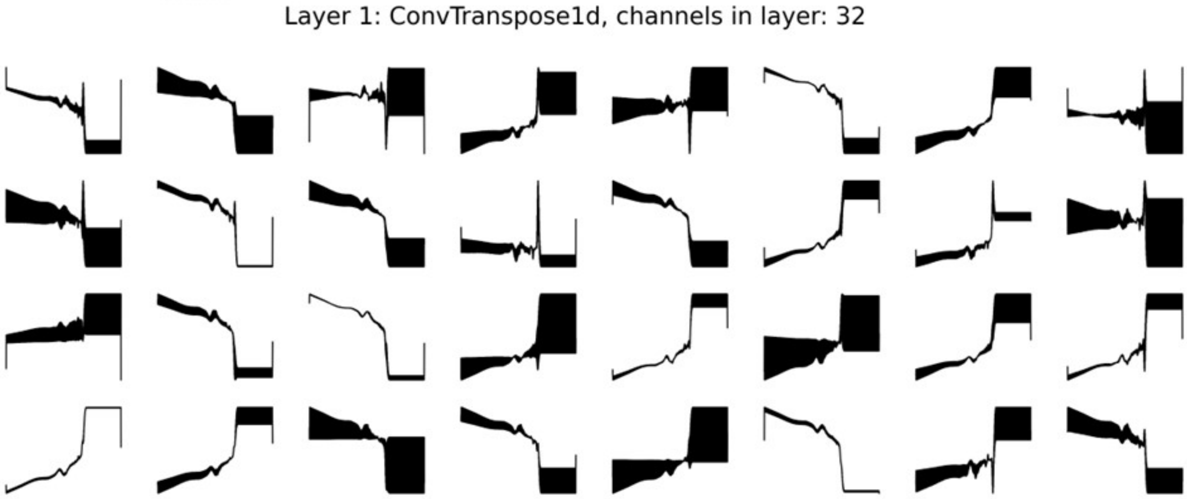


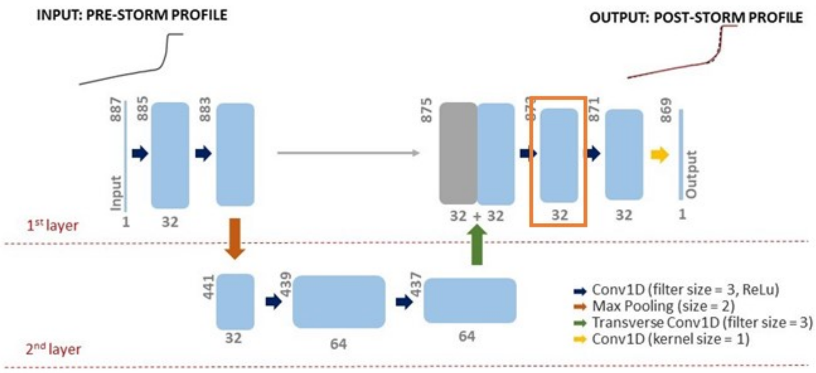

INPUT: PRE-STORM PROFILE
OUTPUT: POST-STORM PROFILE
887
885
883
875
873
871
869
Input
Output
1
32
32 + 32
32
32
1
1st layer
441
439
437
32
64
64
2nd layer
Conv1D (filter size = 3, ReLu)
Max Pooling (size = 2)
Transverse Conv1D (filter size = 3)
Conv1D (kernel size = 1)
Layer 3: MaxPool1d, channels in layer: 32

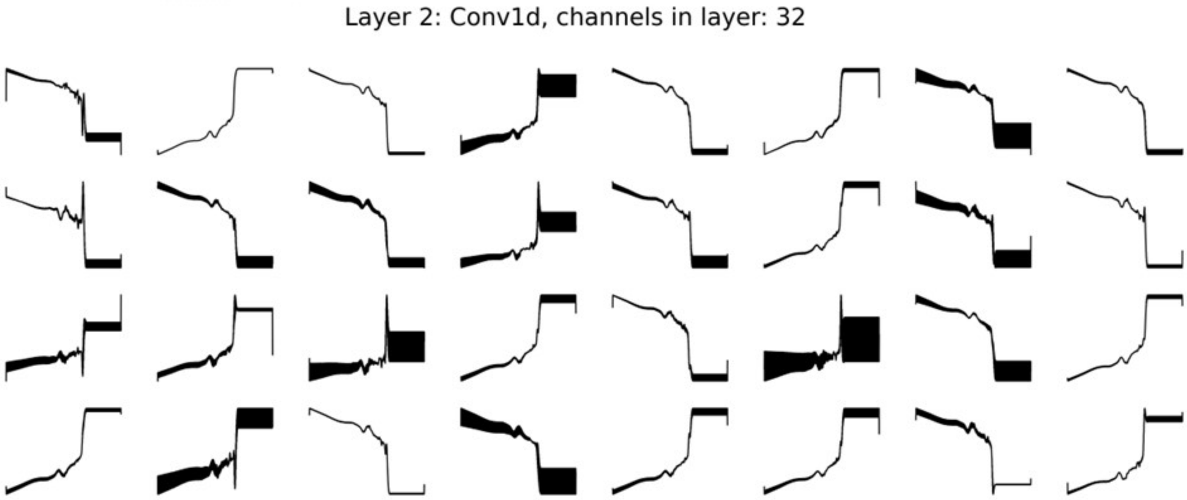

INPUT: PRE-STORM PROFILE
OUTPUT: POST-STORM PROFILE
887
885
883
Input
1
32
875
873
871
869
Output
32 + 32
32
32
1
1st layer
441
439
437
32
64
64
2nd layer
Conv1D (filter size = 3, ReLu)
Max Pooling (size = 2)
Transverse Conv1D (filter size = 3)
Conv1D (kernel size = 1)
Layer 4: Conv1d, channels in layer: 64

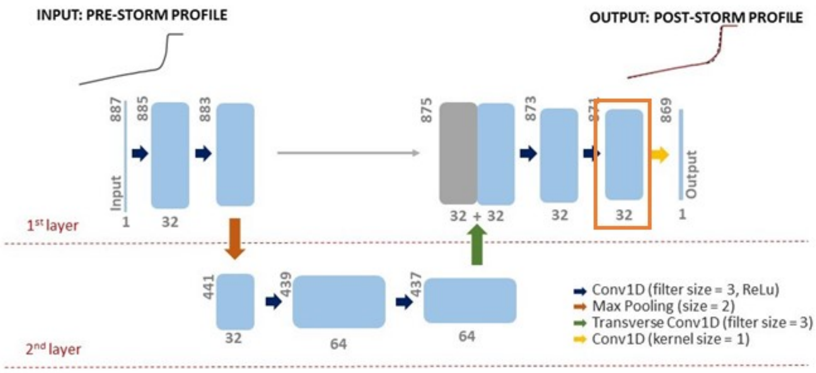

INPUT: PRE-STORM PROFILE
OUTPUT: POST-STORM PROFILE
887
885
883
875
873
871
869
Input
Output
1
32
32 + 32
32
32
1
1st layer
441
439
437
32
64
64
2nd layer
Conv1D (filter size = 3, ReLu)
Max Pooling (size = 2)
Transverse Conv1D (filter size = 3)
Conv1D (kernel size = 1)
Layer 5: Conv1d, channels in layer: 64


INPUT: PRE-STORM PROFILE
OUTPUT: POST-STORM PROFILE
887
885
883
875
873
871
869
Input
Output
1
32
32 + 32
32
32
1
1st layer
441
439
437
32
64
64
2nd layer
Conv1D (filter size = 3, ReLu)
Max Pooling (size = 2)
Transverse Conv1D (filter size = 3)
Conv1D (kernel size = 1)

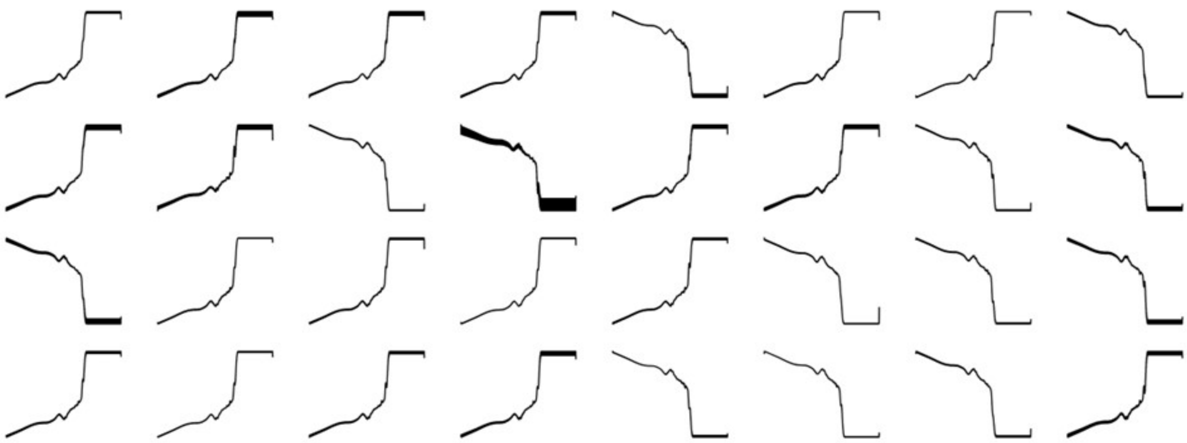

Layer 1: ConvTranspose1d, channels in layer: 32

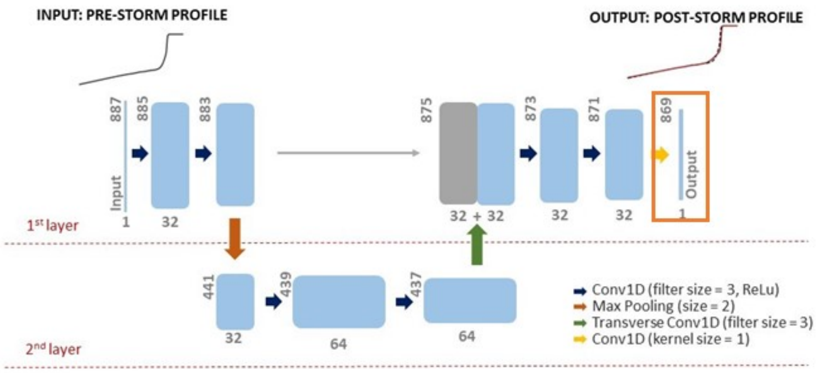

INPUT: PRE-STORM PROFILE
OUTPUT: POST-STORM PROFILE
887
885
883
875
871
869
Input
Output
1
32
32
32 + 32
32
32
1
1st layer
441
439
437
32
64
64
2nd layer
Conv1D (filter size = 3, ReLu)
Max Pooling (size = 2)
Transverse Conv1D (filter size = 3)
Conv1D (kernel size = 1)

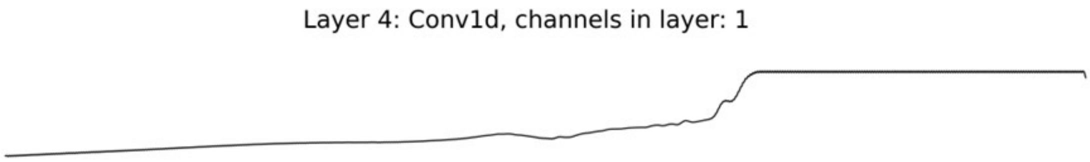

Layer 2: Conv1d, channels in layer: 32



Layer 3: Conv1d, channels in layer: 32




INPUT: PRE-STORM PROFILE
OUTPUT: POST-STORM PROFILE
Input
Output
887
885
883
875
873
871
869
1
32
32 + 32
1st layer
2nd layer
441
439
437
64
Conv1D (filter size = 3, ReLu)
Max Pooling (size = 2)
Transverse Conv1D (filter size = 3)
Conv1D (kernel size = 1)


Layer 4: Conv1d, channels in layer: 1