Master Thesis

Convolutional Neural Networks for modelling riverbed level variations

ENVM4010: Master thesis

Mitali Ajay Uke

Delft University of Technology

TU Delft Deltares

Master Thesis

Convolutional Neural Networks for modelling riverbed level variations

by

Mitali Ajay Uke

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday October 18, 2024 at 14:00.

Student number:	5013917	
Project duration:	March 18, 2024 – October 18, 2024	
Thesis committee:	Dr. ir. R. Taormina,	TU Delft (main supervisor)
	Dr. ir. E. Mosselman,	TU Delft & Deltares
	Dr. ir. A. Baar,	TU Delft
	Dr. ir. V. Chavarrias,	Deltares
	R. Bentivoglio,	TU Delft (PhD candidate)

Cover: Image created by AI (Microsoft Bing)

TU Delft

Deltares
Enabling Delta Life

Preface

This thesis report marks the final part of my master's degree, MSc Environmental Engineering, which I began in September 2022. As I reflect on this challenging yet rewarding journey, I am filled with emotions of joy and pride. Throughout this experience, I had the opportunity to learn and grow, tackling new challenges each day. This research was especially meaningful because it allowed me to explore my interests in modelling, while also stepping out of my comfort zone by applying knowledge gained during the crossover module I followed.

I would like to extend my sincere gratitude to my entire thesis committee for their valuable input and encouragement throughout the thesis process. I especially appreciate my supervisors, Riccardo and Victor, for supporting me and finding time to always address my concerns. Other members, Erik, Anne, and Roberto, have been unwavering in their support whenever needed. Last but not least, I would like to express my heartfelt thanks to my dear family and friends who never failed to motivate and provide me with the appreciation I needed. The journey was not easy yet I was able to power through this interesting research and cannot thank everyone enough.

As I conclude this preface, I am happy with the accomplishments so far and super excited about future opportunities. I hope to contribute effectively to the work I become a part of and remain inspired to keep learning always. I take with me values and knowledge that will stay with me lifelong. I am incredibly lucky that I was blessed with the opportunity to learn more.

Mitali Ajay Uke
Delft, October 2024

Abstract

Understanding morphodynamic processes and structures is essential for effective river management and enhancing our knowledge of river systems. River bars can be investigated through theoretical analyses, field measurements, experimental studies, or numerical modelling. While numerical modelling offers accuracy, it is computationally intensive, making multiple simulations or parameter calibrations both time-consuming and impractical. The numerical simulations follow physical laws and equations which make the runtime significantly high making instantaneous results in times of emergencies unfeasible.

Convolutional neural networks (CNNs), a type of data-driven modelling, have been employed to study the physical parameters defined by linear stability analysis. This study utilizes Delft3D simulations to generate diverse datasets, facilitating easier access and variability. A specific type of riverbar pattern, and alternate bars were chosen for simplicity. The CNN model takes initial bed levels as inputs and provides predictions for the next step of bed level or a time series, with velocity included as an additional parameter to assess its influences on the model performance.

The model is able to predict the bar behaviour with R^2 being 0.99. The model can predict bar suppression or migration solely based on the initial bed levels provided. The model performance did not improve with an additional input parameter although this possibility can be explored with other architectures. However, the model currently lacks accuracy in making one-step-ahead predictions, potentially due to boundary issues within the numerical model or the CNN itself. Further optimization and exploration of additional methods are necessary. The integration of physical parameters into the training process may improve prediction accuracy. Strong conclusions cannot be drawn until additional research is conducted.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

Abbreviation	Definition
3D, 2D	3-dimensional, 2-dimensional
AI	Artificial Intelligence
CNN	Convolutional Neural Network
DL	Deep Learning
FCANN	Fully Connected Artificial Neural Network
GNNs	Graph Neural Networks
LSTM	Long-Short Term Memory
ML	Machine Learning
MSE	Mean Squared Error
NAN	Not A Number
PCA	Principal Component Analysis
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
SWE	Shallow Water Equations

1

Introduction

River morphodynamics is the study of interactions between water flow and sediments, focusing on the behaviour of river bedforms across spatial scales, from small ripples to large bars and meanders (Wright and Crosato, 2011). River bars are riverbed structures that form under specific bed and flow conditions. These structures are crucial features of river topography, potentially obstructing water intakes, restricting navigation, and interacting with existing engineering structures such as bridge piers and hydraulic constructions (Crosato and Mosselman, 2020). They also influence various aspects of river engineering, including water and sediment conveyance, bank protection, and fishery management. Effective river training and rehabilitation projects necessitate the proper management of these bars. For instance, the Waal River in the Netherlands has been managed for centuries to facilitate navigation and mitigate ice growth, which in turn regulates the number of bars in the main channel and prevents undesired upstream erosion (Duró et al., 2016). River bars play a significant role in the formation of river meanders and braiding, channel widening, and bifurcations (Redolfi, 2021).

River bars can be investigated through theoretical analyses, field measurements, numerical modelling, and experimental studies. The topography of the river is determined using physical methods, remote sensing techniques, or numerical models. However, physical methods are labour-intensive, and remote sensing techniques are less effective for predicting time series data. Numerical models are increasingly important in river and coastal engineering due to their ability to predict both short-term and long-term river evolution. Applications of numerical modelling include simulating long-term morphodynamics driven by river flow and tides (Guo et al., 2015), assessing the impact of erosion and deposition on river bends with self-formed widths (Eke et al., 2014), and examining the response of river bars to spatial variations in channel width (Gonzalez and Balajewicz, 2018).

Physics-based models discretize the shallow water equations using finite-difference or finite-volume methods to approximate fluid motion. Nevertheless, these models can be computationally expensive and are often unsuitable for inverse problems such as parameter calibration, or running multiple scenarios. The numerical models along with this also require long runtimes making them less feasible for performing multiple simulations. For instance, simulating a numerical model over several decades and tens of kilometres may require hours or days, even for one-dimensional morphodynamic models (Barneveld et al., 2024). While models for short time frames may be faster, they remain less efficient. Furthermore, managing inputs in high-dimensional numerical models during parameter optimization can be challenging, with computational costs increasing with finer grid resolutions. Although numerical methods are accurate, they cannot provide near-instantaneous predictions which are essential for hazard management such as for flooding.

An alternative solution involves data-based models (Reichstein et al., 2019), which learn complex patterns between inputs and outputs without relying on physical equations. These models are trained on data representing the desired parameters by fitting the provided data. These models do not solve complex physical equations leading to faster and more efficient computations. Convolutional neural networks (CNNs), a type of deep learning model, are particularly well-suited for processing and analyzing data through trainable filters and convolutional layers. CNNs can efficiently extract spatial information

from data and provide predictions that are both rapid and comparable to numerical simulations.

1.1. Research Questions

Having discussed the relevance and importance of efficiently studying riverbed variations, the primary objective of this research will be to investigate the potential of CNN models in predicting riverbed variations in bar formations. The formulated main research question for this thesis is as follows:

Main Research Question:

How accurately can convolutional neural networks predict alternate river bar variations compared to the numerical models?

To answer this question, the following sub-questions will be addressed:

1. How accurately can a convolutional neural network predict bed elevations of river bars in one-step ahead predictions?

2. What are the necessary inputs and how do they affect model performance?

3. How accurately can the model estimate the physical parameters of the river bars?

4. What is the magnitude of speed-up provided by the convolutional neural network compared to run time of the numerical model?

1.2. Significance of this research

This research aims to contribute to the field of river dynamics by introducing a new method to predict the behaviour of river bars. Traditional approaches rely on numerical or experimental methods for predicting riverbed level variations and other properties like discharge and water depths. The AI-based approach proposed in this study is expected to deliver faster results and enhance efficiency. This study could significantly advance computing efficiency in riverbed variation studies and provide real-time monitoring. It aims to strengthen the interdisciplinary bridge between computer science and environmental engineering. The primary focus is to explore convolutional neural networks (CNNs) and their architectures to predict specific riverbed forms, particularly river bars.

The model developed in this research will be designed for straight river channels, with the scope of emulating alternate river bar patterns. The main objective is to develop and validate a CNN model that can accurately predict the river bed elevations which will in turn provide information on the physical properties of the bar patterns, such as growth rate and migration speed. Physical parameters can be incorporated into the training process to improve accuracy if necessary. The study utilizes Delft3D simulations to generate training data to avoid the issue of lack of data availability and will provide a diverse training dataset.

The insights gained from this research can potentially be applied to predict other riverbed forms by adapting the methodology developed here, thereby broadening the applications of AI in morphodynamics. In summary, the application of an AI-based approach to riverbed analysis, as explored in this study, has the potential to be highly valuable.

1.3. Thesis organisation

This thesis report contains five main chapters. Chapter 2 elaborates on the theoretical background of alternate bars and provides a thorough introduction to deep learning. It mentions prior studies conducted regarding deep learning in morphodynamics, thereby addressing the research gap.

Chapter 3 describes the methodology used to answer the research questions in this study. It includes running numerical simulations, pre-processing, training, validating the deep learning model, and

evaluating the results. The following Chapter 4 provides the results, comparing the predicted output values with the true output values using various evaluation metrics.

A critical analysis of the results is provided in Chapter 5. This chapter discusses the limitations of this research and potential research ideas for the future. The research is concluded in Chapter 6.

2

Literature review

This chapter reviews the existing literature on river bars and deep learning, with a focus on previous research integrating artificial intelligence into morphodynamics. The interaction between flow conditions, sediment, and the riverbed can lead to various bedforms, ranging from small ripples to larger structures such as dunes, antidunes, and bars (Colombini et al., 1987). These morphological bedforms can be classified into relevant spatial scales: the reach scale, which covers large sections of the river; the cross-section scale, which pertains to transverse variations in the flow field or water depth; and the depth scale, which focuses on vertical variations in flow (Wright and Crosato, 2011). The following section discusses river bars, which are a phenomenon occurring at the reach scale.

2.1. River Bars

River bars are characterized by periodic depositional and scour patterns resulting from sediment-water interactions. These features emerge during low flow conditions and can vary in type, including alternate bars, central bars (mid-channel bars), or multiple bars (braided bars). River bars can be analyzed through theoretical analyses, experimental findings, field observations, and numerical simulations.

According to Crosato and Mosselman, 2020, river bars can be categorized into three main types:

- **Single Bars**: Single bars form due to permanent perturbations or forcing from geometric features or discontinuities in the river channel, such as hydraulic structures, bedrock formations, bends, or irregularities in the bankline. These bars are also known as forced bars. Examples include point bars, which develop inside river bends, and central bars, which form in wider river sections.
- **Periodic Bars**: Periodic bars develop when water flows over loose material under specific hydraulic and sediment mobility conditions. Also referred to as free bars, these can be either steady or migrating downstream. They do not require external forcing for their formation and are commonly found in straight channels.
- **Hybrid Bars**: Hybrid bars result from both morphodynamic instability and external forcing. The forcing induces permanent flow alterations at specific locations, stabilizing the bar's position after some distance. These bars are frequently observed in natural rivers.

For simplicity, this research focuses on alternate bars, a type of periodic bar. Alternate bars are defined as a regular sequence of scour and depositional features located on alternating sides of the river channel (Tubino et al., 1999; Redolfi, 2021). Diagonal fronts separate these bumps and scours, and they arise spontaneously in initially flat-bedded, straight channels (Nelson, 1990). These bars, a type of periodic bar, develop in the presence of initial morphodynamic instability and are common in channelized river systems. They form when the erodible bed is unstable, with the height and wavelength of the bars scaling with the flow depth and channel width, respectively (Lanzoni, 2000).

The number of parallel rows of bars in a river channel is referred to as the bar mode m. Alternate bars have a mode m of one, while multiple bars exhibit a mode greater than two. Figure 2.1 illustrates

the planimetric diagram of alternate bars, with bar width and length shown in the x-y plane and bar height can be imagined in the perpendicular z-direction.

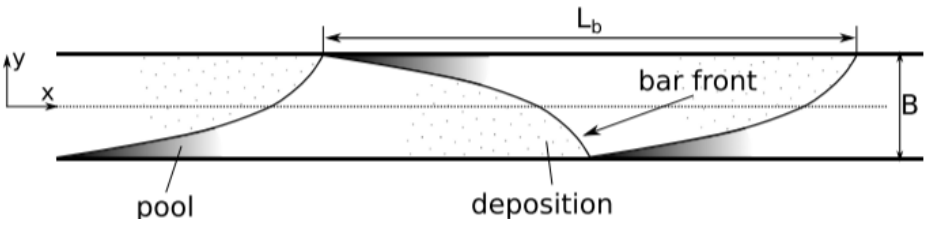

Figure 2.1: Schematic representation of alternate bars in a channel. L_b is bar length and B is channel width. (Siviglia et al., 2013)

The behaviour of water and sediments can be explained using shallow water equations in combination with Exner's equations. This combines 2-D mass conservation for fluids and momentum conservation for flow in a channel with a cohesionless bottom along with the sediment balance equation. The governing equations are defined over a Cartesian reference system *(x,y,z)* where *x-y* is the horizontal plane and *z* is vertical.

$$\frac{\partial H}{\partial t} + \frac{\partial}{\partial x}(q_x + q_{sx}) + \frac{\partial}{\partial y}(q_y + q_{sy}) = 0, \tag{2.1}$$

$$\frac{\partial q_x}{\partial t} + \frac{\partial}{\partial x}\left(\frac{q_x^2}{D} + \frac{1}{2}gH^2 - gH\eta\right) + \frac{\partial}{\partial y}\left(\frac{q_x q_y}{D}\right) + gH\frac{\partial n}{\partial x} + gDS_{fx} = 0, \tag{2.2}$$

$$\frac{\partial q_y}{\partial t} + \frac{\partial}{\partial x}\left(\frac{q_x q_y}{D}\right) + \frac{\partial}{\partial y}\left(\frac{q_y^2}{D} + \frac{1}{2}gH^2 - gH\eta\right) + gH\frac{\partial n}{\partial y} + gDS_{fy} = 0, \tag{2.3}$$

$$\frac{\partial \eta}{\partial t} + \frac{\partial q_{sx}}{\partial x} + \frac{\partial q_{sy}}{\partial y} = 0. \tag{2.4}$$

where *H* is the water surface elevation, η is the bed elevation,
g is gravitational acceleration, *D* is the water depth,
q_x, q_y are x-y components of $\vec{q}$ which is flow discharge per unit width,
q_{sx}, q_{sy} are x-y components of $\vec{q_s}$ which is bed-load per unit width divided by constant porosity term $(1 - \lambda_p)$ and
$\vec{S_f}$ is dimensionless friction term denoted by x-y component S_{fx} and S_{fy}.

Equation 2.1 is the mass conservation equation also known as the continuity equation. It ensures that mass in the considered system is conserved over time. Equation 2.2 and 2.3 implies that momentum of water is conserved in x and y directions respectively. Equation 2.4 governs the sediment transport; also known as Exner's Equation. According to Exner's equation for sediment transport, sedimentation occurs where flow decelerates while erosion occurs where flow accelerates. All the four equations together account for the conservation of both liquid and solid phases in the system.

2.1.1. Analytical solutions

Mathematical theories of river bars rely on depth-averaged and turbulence-averaged equations. Analytical solutions to these equations simplify the more complex numerical models, as the latter cannot be solved analytically. This simplification involves linearization and assumes that perturbations are harmonic functions. According to Redolfi, 2021, two- and three-dimensional mathematical models have been used to explore various effects on alternate bars, including sediment heterogeneity, flow variability, interactions with forced bars, suspended loads, and vegetation. The initial instability and fully developed stability of alternate bars are studied using both linear and nonlinear analyses (Nelson, 1990). The theories are outlined as follows:

Linear Stability Analysis

Several dimensionless parameters that represent channel properties can be defined as follows:

$$\beta = \frac{B}{2D_0}, \quad \theta = \frac{SD_0}{\Delta D_s}, \quad d_s = \frac{D_s}{D_0}, \quad \nu = \frac{\pi B}{L_b} \tag{2.5}$$

where β is aspect ratio, θ is Shields stress, d_s is relative roughness and ν is the wavenumber. B is channel width, S is bottom slope, Δ is relative submerged sediment density, D_s is sediment diameter and D_0 is reach-averaged flow depth

Linear stability analysis offers a criterion for understanding the marginal stability conditions of small-amplitude perturbations over a flatbed with a uniform slope. These conditions are influenced by the channel width-to-depth ratio. This analysis establishes the relationship between bar wavelength, migration speed, and growth or damping in both time and space. Bar wavelength is the distance between two bar scours while migration speed is the speed by which given bar amplitude moves over time. The bars may be subjected to growth in time or suppressing over time which means bars can either experience amplification of the bars or reduction of the amplitude of the bars. The perturbations in the channel are assumed to be infinitesimal, which justifies the linearization approximation. Perturbations are smaller than the flow depth. Alternate bars do not form below a threshold width-to-depth ratio, which depends on characteristics such as relative roughness and the Shields parameter. A neutral curve is identified, representing conditions where bars are expected neither to grow nor decay (Lanzoni, 2000). This theory does not provide actual values for bar amplitude, which is determined using nonlinear approaches.

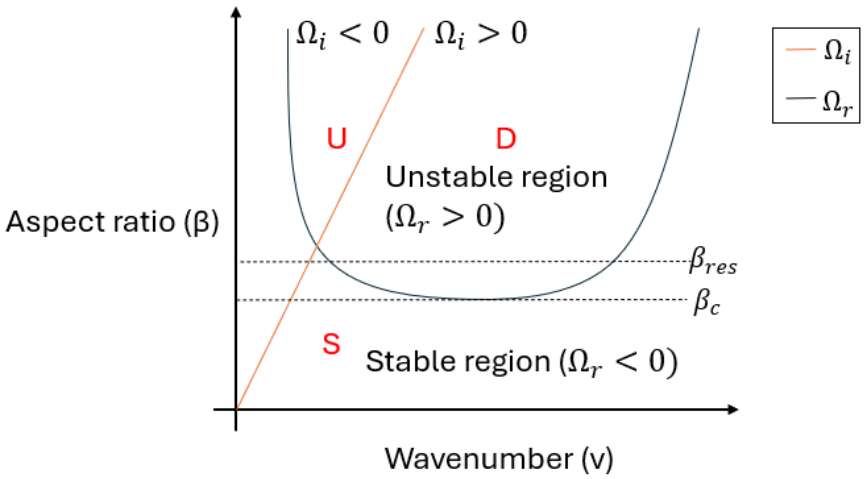

Figure 2.2: Neural stability and migration curves in the (Wavenumber (ν), Aspect ratio (β)) plane. Ω_r is the neutral curve for stability and Ω_i is the neutral curve for migration.

Neutral stability curves determine whether the bars will migrate or be suppressed. The migration curve determines upstream or downstream migration. The region 'S' in Figure 2.2 represents stable region. 'U' and 'D' are in unstable regions with bars migrating upstream and downstream respectively.

Weakly Nonlinear and Fully Nonlinear Theories

Weakly nonlinear and fully nonlinear theories provide alternative perspectives on bar behaviour. Weakly nonlinear approaches share similar assumptions with linear theory, but they accommodate perturbations that are small but not infinitesimally small. These theories offer insights into the growth of perturbations toward finite amplitude (Colombini and Tubino, 1991).

2.1.2. Numerical Modelling

Quasi-steady flow models,which incorporate system nonlinearity, can predict bar wavelengths and their growth more effectively than linear approaches (Lesser et al., 2004). These models provide a more

comprehensive understanding of bar dynamics by accounting for significant nonlinear effects. Numerical modelling involves converting continuous partial differential equations into discrete difference equations to obtain point values. This approach accounts for the complete nonlinearity of the governing equations and allows for the study of morphodynamic processes with spatially varying features and non-uniform sediments. Numerical models can simulate the evolution of bars from their inception to fully developed conditions. Common numerical methods include finite difference, finite element, and finite volume approaches (Wright and Crosato, 2011). These discretized equations are transformed into a set of algebraic equations solvable by computers. Examples of numerical models used in this field include HSTAR, SSIIM, FaSTMECH, and Delft3D (Hu et al., 2023).

Delft3D

Delft3D is the numerical simulation software utilized in this research. Developed by Deltares, Delft3D is an open-source software suite that encompasses several modules for various aspects of hydrodynamic, morphodynamic, wave, water quality, and particle modelling research. Each module can be operated independently or in combination with others, all within a unified user interface. The software suite includes utilities for generating curvilinear grids (RGFGRID), creating grid-oriented data (QUICKIN), and visualizing and animating simulation results (Delft3D-QUICKPLOT).

The key modules in Delft3D are:

- Delft3D-FLOW: This module is used for multidimensional simulations of non-steady flow and transport phenomena under tidal and meteorological forcing. It supports both two-dimensional (2DH, depth-averaged) and three-dimensional (3D) flow and transport simulations. Delft3D-FLOW operates on rectilinear, curvilinear, and spherical grids and is the primary module used in this research. It is applicable to simulating river flows, deep lakes, reservoirs, salt intrusion, and other applications. The module solves the nonlinear shallow water equations, derived from the three-dimensional Navier-Stokes equations for incompressible surface flow, using a finite difference method. In Delft3D-FLOW, water levels are defined at the center of discretized cells, while velocity components are perpendicular to the grid cell edges.
- Delft3D-MOR: For morphodynamic computations.
- Delft3D-Water Quality: For water quality computations.
- Delft3D-WAQ PART: For particle modelling.
- Delft3D-ECO: For ecological modelling.

Delft3D-FLOW will be the focus of this research due to its extensive capabilities in simulating various hydrodynamic and morphodynamic conditions (Deltares, 2024). Numerical simulations as mentioned in Chapter 1, can be resource-intensive and fail to provide instantaneous results. These models heavily rely on complying with the physical laws of nature and require information about boundary conditions and system parameters. This information may not be available for certain considered locations. Moreover, anthropogenic factors influencing riverbed patterns which are not taken into account in the physical equations such as human interventions may be neglected in these models. This leads to the consideration of a data-based approach which has been discussed in the next section.

2.2. Deep Learning

Machine learning, a branch of Artificial Intelligence (AI), involves learning from data to generalize and make predictions on unseen data. It is broadly categorized into three main types: supervised learning, unsupervised learning, and reinforcement learning.

Supervised Learning: This approach involves training algorithms on labeled data, where the correct output is known. Common techniques include regression and classification.

Unsupervised Learning: In this approach, algorithms are trained on unlabeled data, and the goal is to uncover hidden patterns or structures. Techniques include clustering and generative models.

Deep learning, a subset of machine learning, is employed in this research. Implementing deep learning models requires understanding several key components and processes. For a detailed description of these components, see Prince, 2024.

The deep learning process involves several critical steps:

1. Defining Loss Functions: The loss function quantifies the difference between the model's predictions and the ground truth. The model's objective is to minimize this loss during training.
2. Fitting Models: Convolutional Neural Networks (CNNs), used in this research, are characterized by their local connections and shared weights. The model architecture includes convolutional and pooling layers, which extract features from the data, and activation functions such as ReLU, sigmoid, and tanh, which introduce non-linearity and enhance the model's ability to capture complex patterns (Sharma et al., 2020).
3. Model Training: The network parameters are adjusted through training. This involves computing gradients of the loss function and iteratively updating the model's weights and biases using the backpropagation algorithm. The goal is to minimize the loss over many iterations (epochs).
4. Hyperparameter Tuning: Hyperparameters, such as learning rate and batch size, affect model performance but are not updated during the training process. Instead, they must be selected through experimentation and trials.

Understanding these elements is crucial for effectively implementing and optimizing deep learning models. In this research, Convolutional Neural networks (CNNs), a subset of deep learning, have been considered. CNNs consist of a series of convolution and pooling layers to understand features in the given image. Parameters like padding, stride and kernel filters are the primary parameters determining the size of the output feature maps. CNNs have been chosen because of their ability to analyse visual data and learn the hierarchical characteristics of the visual data. Furthermore, they are effective when trained on large training datasets and can work with inputs of varying sizes. This is possible because CNNs possess the property of translational invariance meaning that they can recognize the same pattern regardless of the location it is placed in the input. Lastly, CNNs can generalize of unseen datasets (Krichen, 2023).

2.3. Artificial Intelligence and Morphodynamics

Understanding hydrodynamic parameters, such as river runoff, water levels, and velocity, along with morphodynamic parameters like sediment composition and riverbed topography, is crucial in the fields of water resources, river dynamics, and sediment transport.

The application of AI in water resources has gained significant importance due to the abundance of data made available by recent advancements in computational processing capacity. Convolutional Neural Networks (CNNs), in particular, have shown the ability to process multidimensional data effectively, making them successful in predicting water levels, forecasting floods, and performing other tasks. For example, water levels in lakes can be predicted using a deep learning model like CNN-LSTM (Goldstein and Coco, 2014) and be applied to forecasting rainfall-runoff (Van et al., 2020). Kabir et al., 2020 used a CNN model to estimate water depths during floods. It showed that the CNN model was able to perform the task accurately by capturing the flooded cells. This shows applications of CNN in real-time modelling/forcasting. However, the application of CNNs in morphodynamics remains limited, mainly focusing on sediment load modelling and classification. Few of these studies are - Hosseiny et al., 2023;Bhattacharya et al., 2007 and Ermilov et al., 2023.

Riverbed topography, a three-dimensional feature varying over time, was previously difficult to study due to the scarcity of data, as measurements were traditionally taken using wading or shipboard sonar equipment (these methods are time-consuming and costly), which may explain the limited number of studies in this subject. However, recent advancements in remote sensing have significantly improved the quality and availability of riverbed measurements. Kiani-Oshtorjani and Ancey, 2023 trained a CNN model to infer riverbed topography from velocity fields. This requires huge datasets to be made available which is unfeasible and expensive. Therefore, an alternative way of obtaining training datasets is needed. So, The possibility of emulation of numerical models was explored in some studies. de Melo et al., 2022 emulated estuarine morphology using CNNs based on results from numerical models; while Liu et al., 2024 conducted a case study in the middle reaches of the Yangtze River using four datasets of varying data abundance. They evaluated the performance of three different deep-learning models for this case study.They noticed the need to find a model which is able to generalize to river sections

and explore other machine learning methods than FCANNs, RNNs and LSTMs.

Therefore, a detailed study of specific riverbed formations using deep learning models, based either on numerical models or data, has not yet been thoroughly explored. A method to train a model to predict river bed elevations using a synthetic dataset is an option to avoid the lack of data availability issues and perform simulations quicker and more efficiently. This research aims to address this gap.

3

Methodology

This chapter outlines the research methodology employed to develop and validate a convolutional neural network (CNN) model for predicting alternate river bar patterns, focusing on two main predictions: one-step-ahead prediction and time series forecasting. Achieving strong predictive performance in a deep learning model depends on factors such as the number of simulations, the diversity of the data, and its complexity. To efficiently obtain a wide variety of data, Delft3D simulation results are utilized to train the CNN model. The aim is to check whether the deep learning model predicts results that align with the linear stability analysis in simulating the initial development of alternate bars. These simulation outputs are pre-processed before being fed into the CNN. Once the model is trained, its performance is evaluated using unseen datasets. Key physical parameters, including growth rate and migration speed, are calculated to assess the model's accuracy, alongside evaluation metrics that quantify the error between the ground truth and the predicted outcomes. The detailed methodology is provided in the following sections.

3.1. Numerical model setup

The input for the deep learning model is generated using Delft3D -FLOW to provide the desired variability in the training dataset and to circumvent the challenges associated with acquiring relevant real-world data. The results from the numerical model are utilized as ground truth (actual values) to establish a benchmark for comparison with the deep learning model. Synthetic data is produced for a straight channel measuring 1000 meters in length and 30 meters in width, with non-erodible banks. The grid size used in the simulations is 10 meters by 3 meters. Simulations are conducted over a period of 10.5 days with 12-hour time steps, allowing the examination of short-term bar evolutions. The spatial and temporal resolution of these simulations is consistent across all generated datasets. Uniform flow is assumed for the channel and is specified in the model's boundary conditions.

To investigate the behaviour of bars with specific wavelengths, bed topography with predefined perturbations of these wavelengths is created. The numerical setup is influenced by the work of Siviglia et al., 2013. The focus is on alternate bars that develop due to the inherent stability of incompressible flow in a straight channel, given a specific discharge sediment flux and grain size. Two-dimensional perturbations with small amplitudes are superimposed on a flatbed along a uniform longitudinal slope to replicate scenarios consistent with linear theories. The initial amplitude of these harmonic perturbations is assumed to be 5% of the reference water depth. The perturbations in the riverbed are generated using the following equation:

$$\eta(x, y) = A \cos\left(\frac{\pi y}{B}\right) \cos\left(\frac{2\pi x}{L_b}\right) \tag{3.1}$$

where,
η is the Bed level perturbation
A is the amplitude of the bars,
L_b is the longitudinal bar length.

The aspect ratio (β) is kept constant across all simulations. The investigation of bar behaviour across varying aspect ratios is beyond the scope of this thesis. The aspect ratio utilized in the numerical model is 30/(2×1), which equals 15 [-] (Equation 2.5). The wavenumbers (ν) considered in the simulations fall into two categories: the stable high-wavenumber region, where bars are suppressed, and the unstable high-wavenumber region, where bars migrate downstream. Dataset 1 includes alternate bars with wavelengths from the stable region on the (λ, β) curve, whereas Dataset 2 contains migrating alternate bars. The datasets have been differentiated based on their output types with bars suppressing in Dataset 1 and migrating in Dataset 2. An example from the simulation results for Dataset 1 and Dataset 2 has been shown in Figures 3.1 and 3.2 respectively.

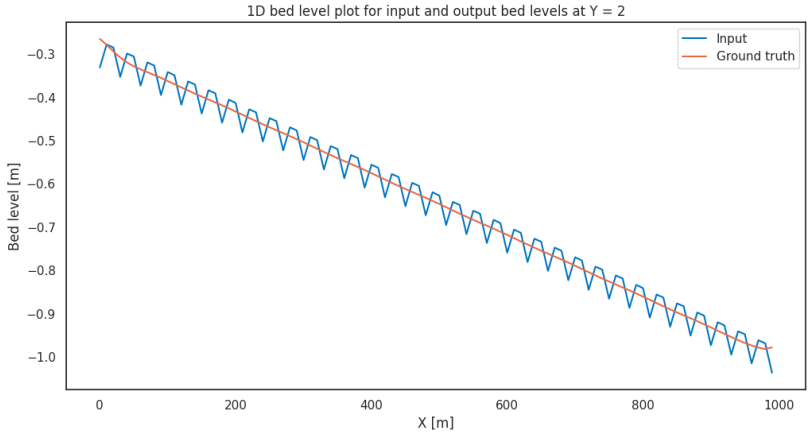

Figure 3.1: 1D plot of bed levels from Dataset 1 at timestep 1 (as input) and timestep 2 (as ground truth).

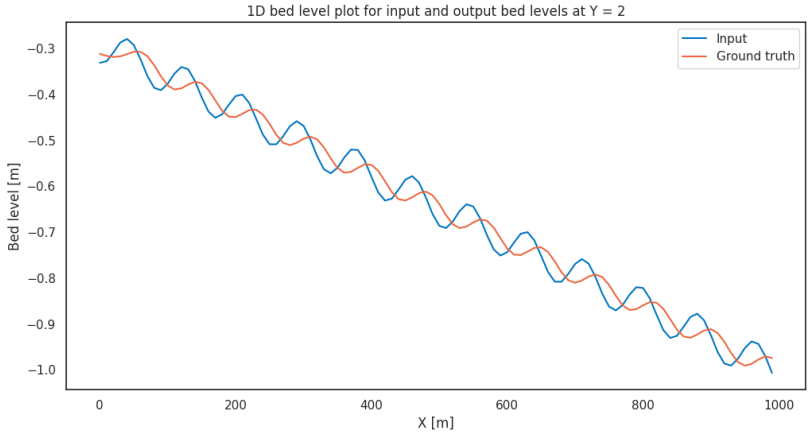

Figure 3.2: 1D plot of bed levels from Dataset 2 at timestep 1 (as input) and timestep 2 (as ground truth).

Upstream migrating bars and positive growth rates are not addressed in this study. The input parameters used to generate the bathymetry files in Delft3D are detailed in Table 3.1. For each simulation run, the initial bed elevation changes are analyzed to compute the growth rate and migration speed of the bars.

Table 3.1: Input parameters for generating bathymetry files

Parameter	Value
Bed Width B (m)	30
Reach averaged flow depth D_0 [m]	1
Aspect ratio β	15
Wavelength Dataset 1 L_b [m]	15- 40
Wavelength Dataset 2 L_b [m]	45- 250
Initial amplitude A_0 [m]	5% of flow depth
Number of simulations in Dataset 1	60
Number of simulations in Dataset 2	60

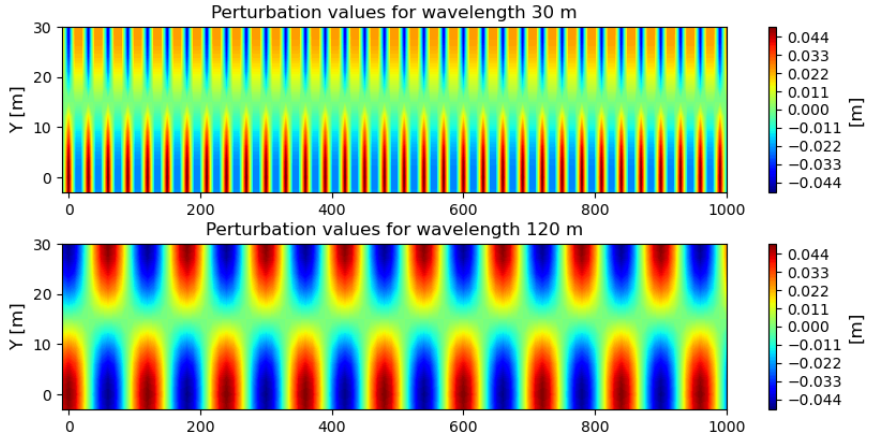

Figure 3.3: Bed perturbation values of one sample each from Dataset 1 and 2. These values are added further to uniform bed with slope.

Provided in Figure 3.3 is an example of the generated perturbation values to add to the bathymetry grid as input to the Delft3D model. The example includes one sample each of Dataset 1 and Dataset 2. All other parameters in the numerical model are constant over all simulations. The map files from the outputs are saved for further processing. The files include all the parameters over the defined time intervals inside the map file made of .dat and .def files. A summary of important parameters and input files in the Delft3D simulations are given in Table 3.2.

3.2. Data pre-processing

Preprocessing of the numerical simulation results was required to advance to the subsequent stages of the research. The outputs from the Delft3D simulations for both datasets were in formats incompatible with Python, necessitating their handling in MATLAB using Delft3D-Quickplot functions. The desired parameters—namely bed level at water level points, water depth, and depth-averaged velocity—were extracted from the output files and exported. The exported files include grid x and y values alongside data for each timestep. All outputs are saved as time series over the (12,102) grid. The inputs and outputs for the deep learning model, both for training and testing, consist of tensors with shapes corresponding to the number of inputs/outputs and grid size.

During data import, NAN values were identified at the boundaries of the domain and removed to prevent issues with backpropagation errors and loss function breakdown, resulting in an adjusted input and output size of (10,100). The bed elevation values range from -0.30 to -1 meters, and the depth-averaged velocity values range from 0.9 to 1 meters per second. The inputs for the deep learning model include the initial (timestep 1) bed elevation and velocity, while the outputs can be a single bed

Table 3.2: Overview of Input files for Delft3D simulations

Input file	Parameter	Value	Unit
Master Definition file (.mdf)	Simulation Time	10.5	days
	Dt	12.09	hours
	Rhow	1000	kg/m^3
	Tempw	1.5	°C
Grid file (.grd)	Coordinate System	Cartesian	-
	Domain size	101 x 11	cells
Boundary conditions (.bct)	Type	Uniform	-
	Interpolation	Linear	-
	Boundary condition 1: total discharge	30	m^3/s
	Boundary condition 2: water elevation	-3.56E-03	m
Open boundaries (.bnd)	Upstream	total discharge time-series	-
	Downstream	water level time-series	-
Bathymetry file (.dep)	Changed per simulation	-	-
Morphology characteristics (.mor)	Bed Update	True	-
	Sediment type	bedload	-
	Soil specific density	2.65E+03	kg/m^3
	Sediment diameter	1.00E-03	m
Output files (.dat and .def)	Timeseries outputs considered	bed level at water level points	m
		depth averaged velocity	m/s
		water depth	m

elevation grid or a time series. Although additional parameters could be incorporated into both inputs and outputs, this was not explored in the current research.

To prevent bias and ensure equal treatment of all coefficients during training, the inputs and outputs were normalized. Normalization brings values to a common scale, enhancing the model's performance. A min-max scaler was employed for this purpose, as it is straightforward to implement. The formula for the min-max scaler is given by:

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{3.2}$$

This formula represents x-values in the dataset and similar can be done for y-values.

3.3. Deep Learning model

The input to the CNN model included initial bed elevations (and depth-averaged velocity) and the output is either bed elevations at the next time step or whole timeseries. The model architecture was defined by adding convolution layers and pooling layers. Advanced architectures like U-net or ResNet could not be used due to the small grid of the considered simulations. Mean squared error (MSE) was used as loss function in the training process. The Adam optimizer with a learning rate of 0.001 was used in this process. Table 3.3 summarizes the hyperparameters in the model training process. The available data was split into training, validation and testing datasets. Training data was 80 % and validation and testing were 10 % each. This ensures enough data for training and avoids overfitting. The model architecture chosen in this research was obtained by trying simple CNN architectures. The architecture used in the research is shown as follows:

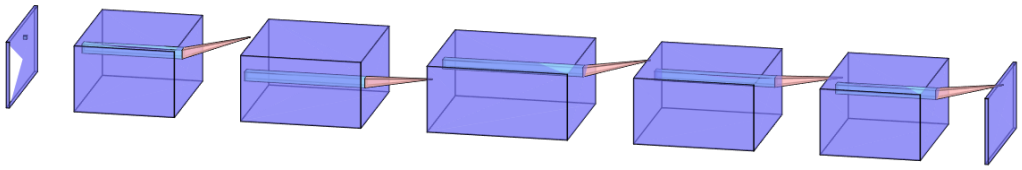

Figure 3.4: CNN model architecture, with 6 convolutional blocks with ReLU activations and Batch Normalization

The model takes an input of size (10,100) which goes through convolution layers returning the output of size (10,100). In the case of two input parameters, the input channels in the model are set to two and the output remains the same. Detailed summary of the model architecture can be found in Chapter 7.

Parameter	Value
Kernel size	3x3
Strides	1
Padding	1
Activation function	ReLU
Optimization function	Adam
Drop out rate	0.0
Number of epochs	100
Learning rate	0.001
Batch size	8
Trainable parameters	333,697 (for timeseries - 4,318,452)

Table 3.3: Summary of the DL Model hyperparameters

3.4. Post-processing

The CNN model until the previous step made use of the training data and validation data provided. Once the model is trained, it must be tested on the dataset it has not previously seen. This way, the model performance in terms of overfitting to the training data and the ability to generalize on similar datasets will be better known. The results of the CNN model are denormalized by multiplying with the standard deviation and adding the mean of the training dataset. The bed levels contain the slope component in the values which was detrended to effectively assess bar amplitude changes. The detrended 1D plots for a simulation from each of the datasets is given below:

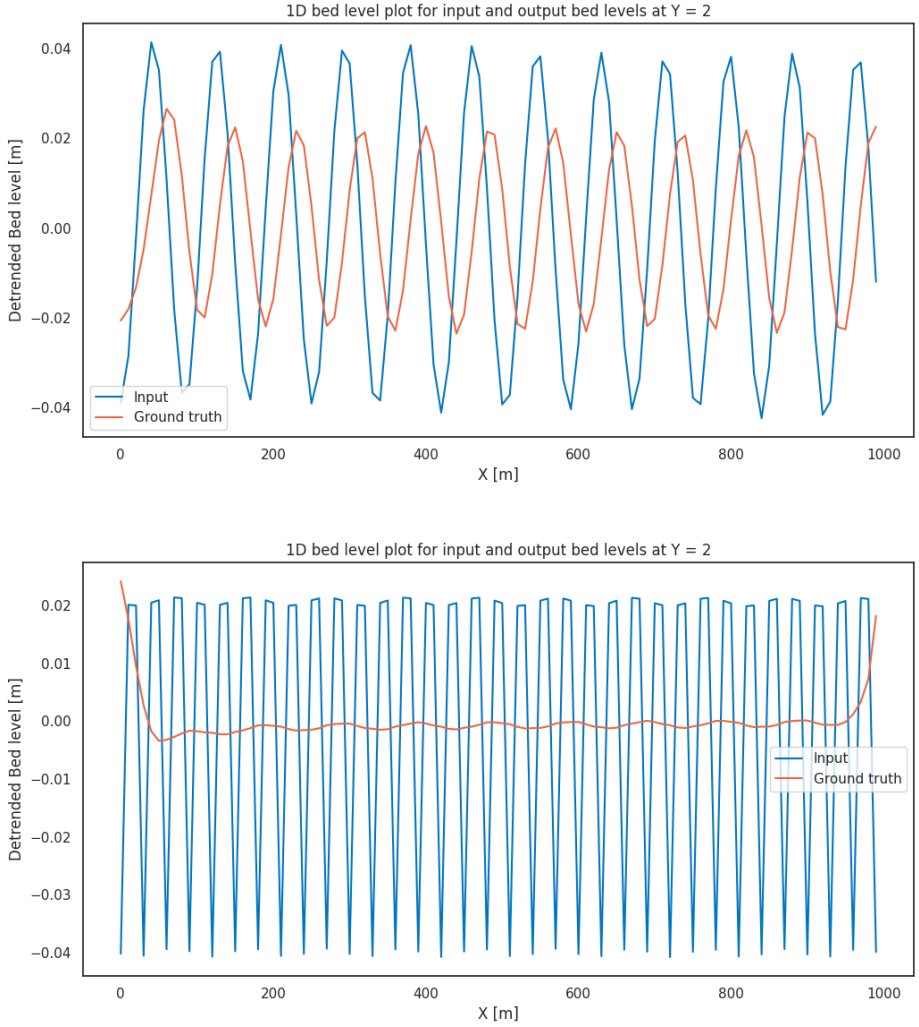

Figure 3.5: top: Detrended bed level values at timestep 1 and 2 from Dataset 1. bottom: Detrended bed level values at timestep 1 and 2 from Dataset 2

The outputs of the CNN model are compared with the ground truth (outputs of the numerical model) to assess the quality of the results. The results include a comparison between predictions of CNN model with ground truth visually and with relative error over the grid. The metrics used to compute overall differences are Coefficient of Determination (R^2) and Root Mean Squared Error (RMSE). Both the evaluation metrics assess the model results by comparing the results of the CNN model with the ground truth using different formulas. These are described as follows:

- **Coefficient of Determination (R^2)**: R^2 measures the goodness of fit. It statistically measures of shift of the dependent variable from the independent variable in a regression model. Values of R^2 range from 0 to 1. 0 indicates poor fit while 1 indicates perfect fit. R^2 itself cannot be sufficient to assess model performance but gives useful information about the variance of the predictions with respect to the ground truth.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{3.3}$$

  where SS_{res} is the residual sum of squares value, SS_{tot} is the total sum of squares
- **Root Mean Squared Error (RMSE)**: RMSE computes the square root of the mean squared error. It is the same unit as the data and provides a quantitative measure of model performance. RMSE

is a better evaluation metric as it is less sensitive to outliers than MSE.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{3.4}$$

The time taken for training and predictions is logged for each model and simulation, respectively. In addition to these metrics, two physical parameters—growth rate (m/h) and migration speed (m/h)—are computed for both the ground truth and the predictions. These parameters are essential for interpreting the results and understanding the behavior of alternate bars.

The growth rate of the alternate bars is calculated by determining the amplitude of the wave in the bed elevation after removing the slope trend. The growth rate is derived from the amplitude relation ($A \propto e^{\omega t}$) as described by linear stability analysis. Migration speed is calculated by identifying the first peaks in both the input and output data. The difference in the x-values of these peaks provides the distance the bar has moved, which is then divided by the elapsed time to determine the speed. The formulas for growth rate (Ω)) and migration speed (ω) are applied as follows:

$$\Omega = \frac{1}{t}\log\frac{A_{output}}{A_{input}} \tag{3.5}$$

$$\omega = \frac{X_{peak,output} - X_{peak,input}}{t} \tag{3.6}$$

4

Results and Interpretation

This chapter presents the key findings of the research that was conducted. The model architecture was the most cumbersome part of running the whole training process. The model was executed for different cases to understand its potential. The results in this chapter will be provided in two sections. First, model implementation on dataset 1 and dataset 2. Second, an additional input parameter in the model namely depth-averaged velocity.

The results mentioned in the following sections show the capability to predict next-step bed levels solely based on initial bed level input. Many alternative architectures were implemented to find the one mentioned below. U-net was used with alterations to the architecture by reducing convolution and pooling functions to take into account small grid images in this research. The possibility of adding residual connections in the simple CNN model was explored as well. The number of input simulations also impacted the quality of the results. The results of these trials has been given in Chapter 7.

4.1. Model implementation on Dataset 1 and Dataset 2

In this case, the model was trained and tested on datasets 1 and 2 separately and together. The model performance of all the runs was computed and can be seen in Table 4.1. Input provided to the model is the initial bed level and output is bed level at the next timestep. The bed level values over the y-axis have influence of slope in the output. Hence, the bed level in 1D was first detrended and then growth rate and migration speed were computed. The model was trained within 442.6 seconds on average.

Dataset	Average R^2 [-]	Average RMSE [m]	Training time (s)	Average model run time (s)
dataset 1	-	1.09E-05	220.29	0.01674
dataset 2	-	0.0128	575.02	0.01472
dataset 1 and 2	0.9894	0.0044	422.73	0.0148

Table 4.1: Performance metrics for Datasets 1 and 2

The complete results of analysis on dataset 1 and 2 done together are shown below. To do so, two simulations, one from dataset 1 and other from dataset 2 are considered from the testing data. For individual analysis of datasets 1 and 2, relative error plots are mentioned. Figures 4.1 (from Dataset 1) and 4.3 (from Dataset 2) show a contour plot of the input along with ground truth and prediction of the CNN model. The last plot in the figure shows the difference between the ground truth and predicted values over the grid. Figures 4.2 (from Dataset 1) and 4.4 (from Dataset 2) show a 1D plot over the second grid row of the considered grid. The plot shows input bed level, actual bed level and predicted bed level. The model is not very accurate even though RMSE seems to be in ranges of centimetres. The errors are spread throughout the grid but maximum at the downstream end (right of the grid). The R^2 value suggests that the results are closer to the ground truth.

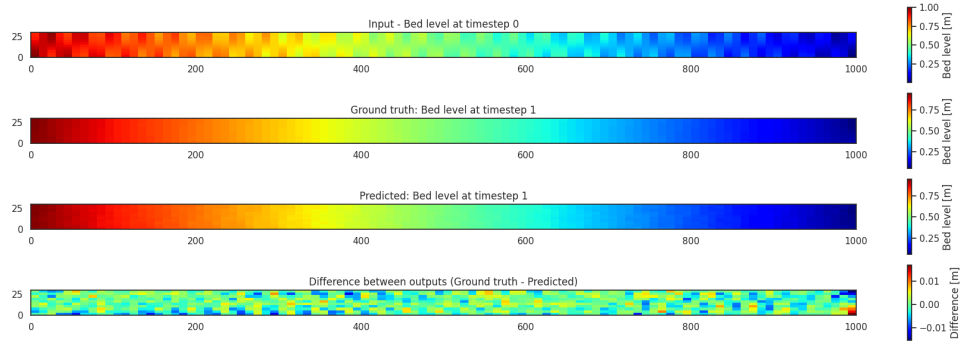

Figure 4.1: Bed level of input with ground truth and predicted bed level values. Wavelength of bars considered = 37.4 m (dataset 1).

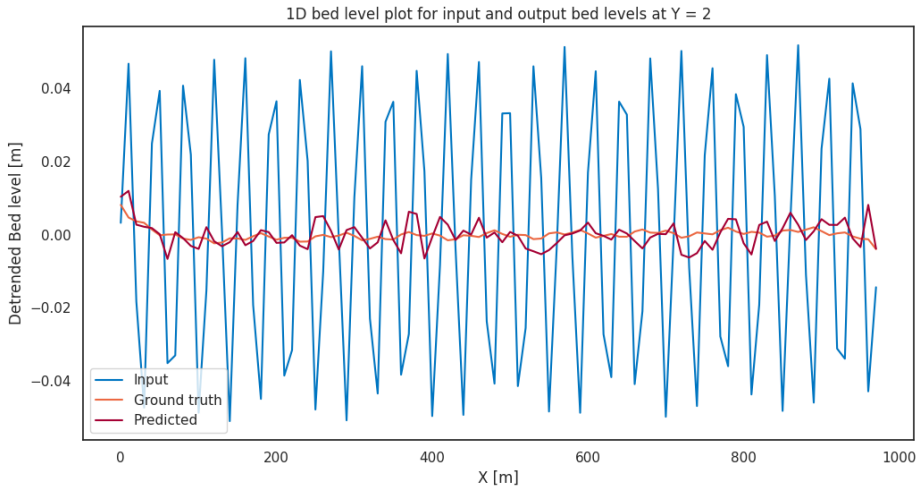

Figure 4.2: Bed level of input with ground truth and predicted bed level values over a defined y value. Wavelength of bars considered = 37.4 m (dataset 1).

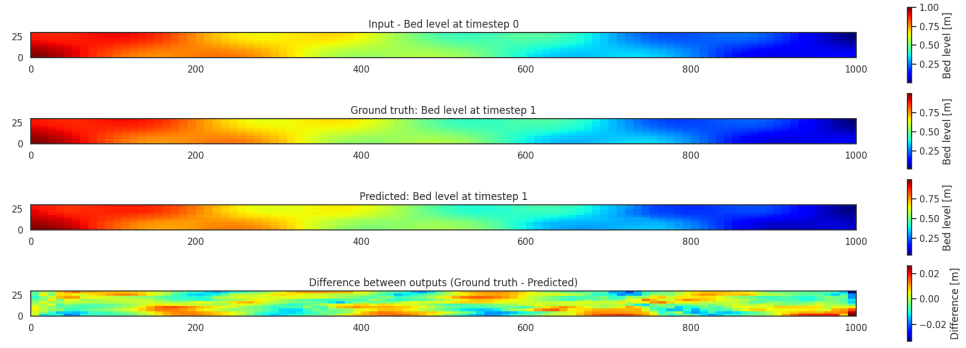

Figure 4.3: Bed level of input with ground truth and predicted bed level values. Wavelength of bars considered = 250 m (dataset 1).

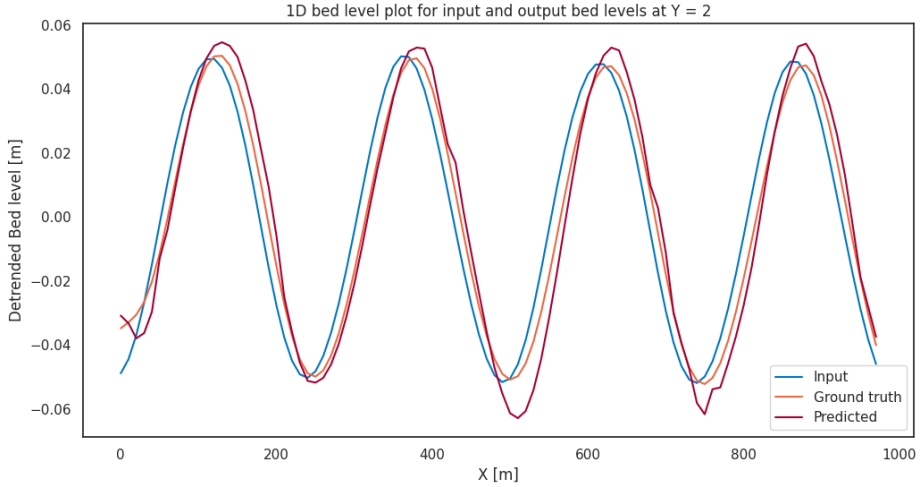

Figure 4.4: Bed level of input with ground truth and predicted bed level values over a defined y value. Wavelength of bars considered = 250 m (dataset 1).

If the relative error between truth bed levels values and predicted bed values is considered, the error is mostly saturated in the boundaries of the grid. The results can be found in Figure 4.5. This phenomenon was also seen in models trained on solely Dataset 1 or Dataset 2. One of the reasons could be errors from the numerical simulation due to the implementation of finite differences or boundaries were not properly defined while running simulations. This could also be caused due to bias of deep learning model trying to overfit the data at boundaries for all simulations.

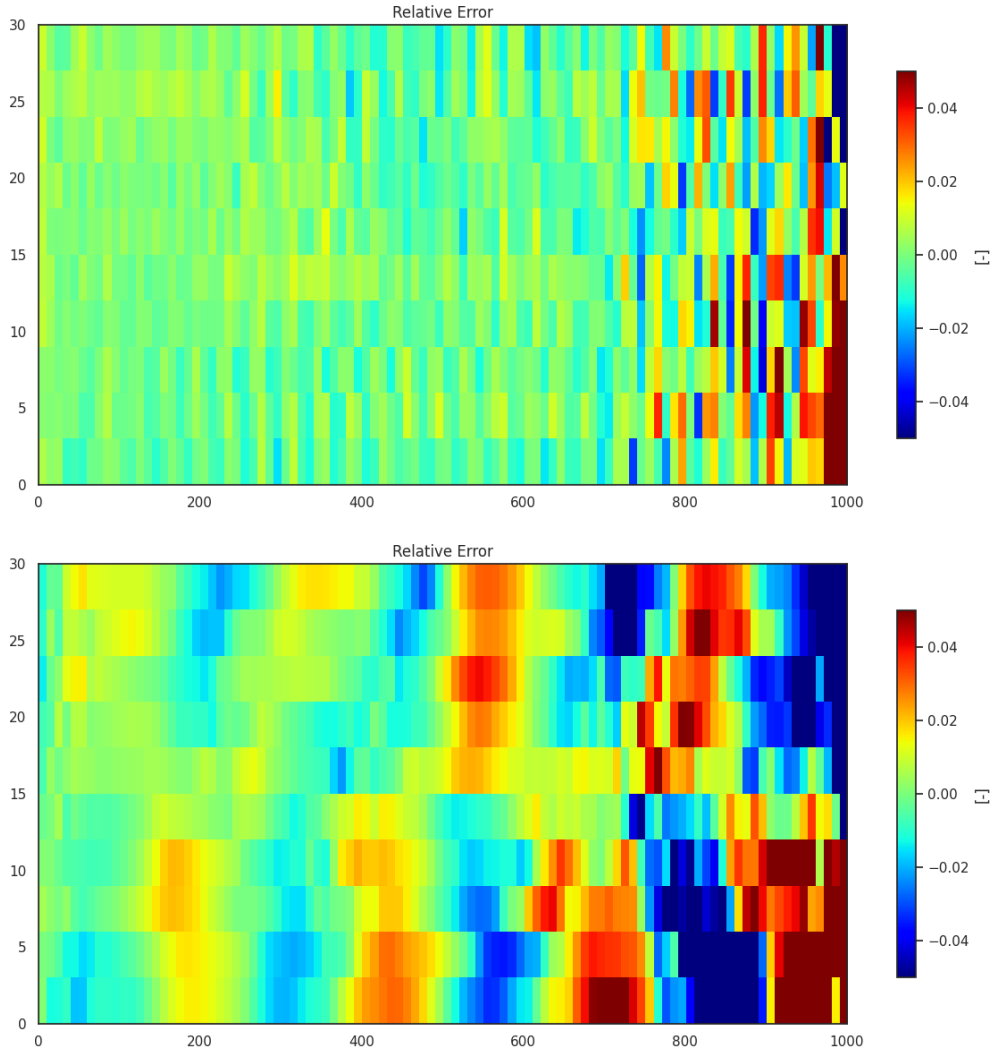

Figure 4.5: top: Relative error in prediction for wavelength 37.4 m from Dataset 1. bottom: Relative error in prediction for wavelength 250.0 m from Dataset 2

If the physical parameters, growth rate and migration speed are considered, the model is able to learn the growth-suppression pattern by predicting growth rate values in the vicinity of the actual growth rate values. The model predicts migration speeds for lower wavenumbers (Dataset 2) accurately however fails to predict them for higher wavenumbers. The results for the growth rate and the migration speed of the bars are given in Figure 4.6.

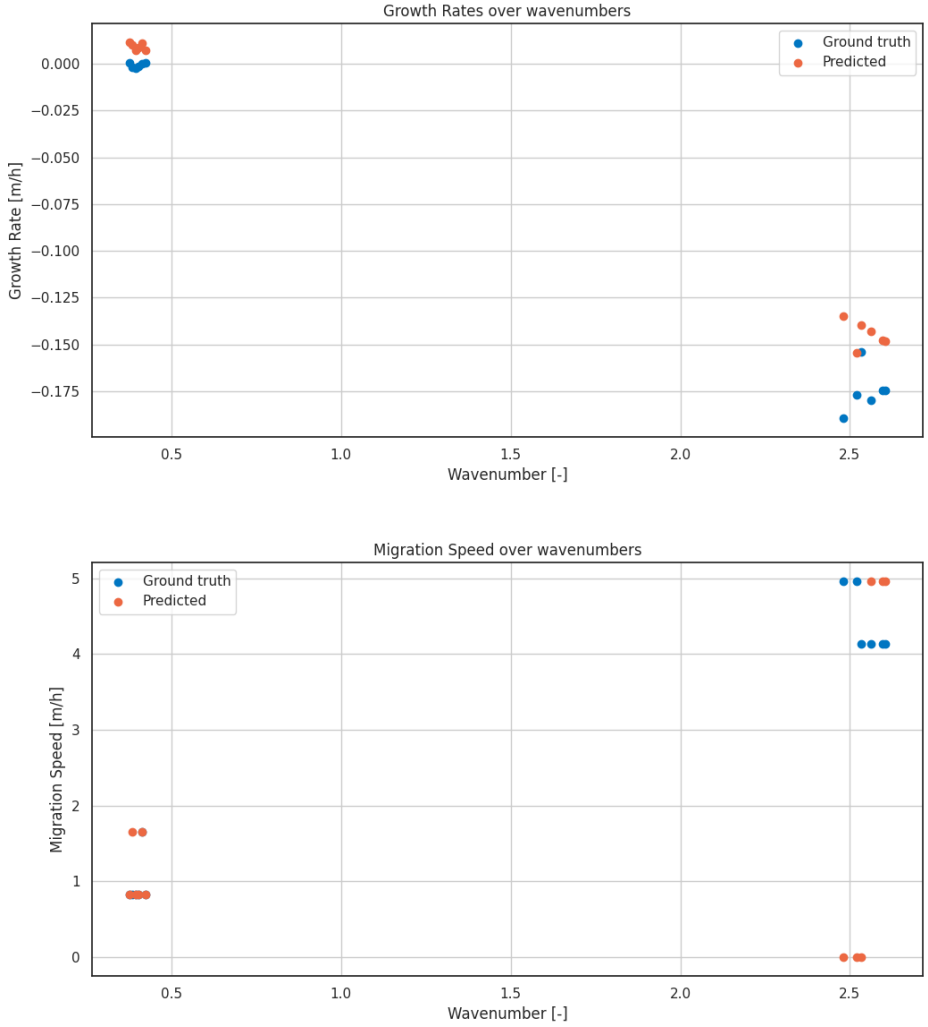

Figure 4.6: Growth rate and migration speed values computed for ground truth and predicted bed levels.

4.2. Modelling with additional input parameter

Additional parameters in the model may provide a higher chance of extracting useful features and finding hidden patterns in the data which were not possibly seen by only one input. Depth-averaged velocity was chosen as the additional parameter since it plays a major role in shaping the river bed level. It could help couple with the hydrodynamic model. Water depth could be another parameter that can be used. The model performance with the model trained on an additional parameter is given in Table 4.2. It can be noticed from the results that the model performance has not been influenced much by the addition of an extra input parameter. This is not expected however possible since the model might try to overfit or make the prediction more complex than needed. This could also be because of the model's incapability to deal with lesser data like in this case. The results of bed levels are smoother in the middle and get extreme towards the boundaries.

Dataset	Average R^2 [-]	Average RMSE [m]	Training time (s)	Average model run time (s)
dataset 1 and 2	0.9883	0.0099	432.53	0.0160

Table 4.2: Performance metrics for Datasets 1 and 2

The results of the 2D grid for a wavelength of 36.2 m are presented below in Figure 4.7. The physical parameters are computed on the boundary using first peaks and it can be seen in the 1D plot

that the model performs poorly mostly at the boundaries. This could explain the worse performance of this model. The growth rate and migration speed plots are presented in Figure 4.8

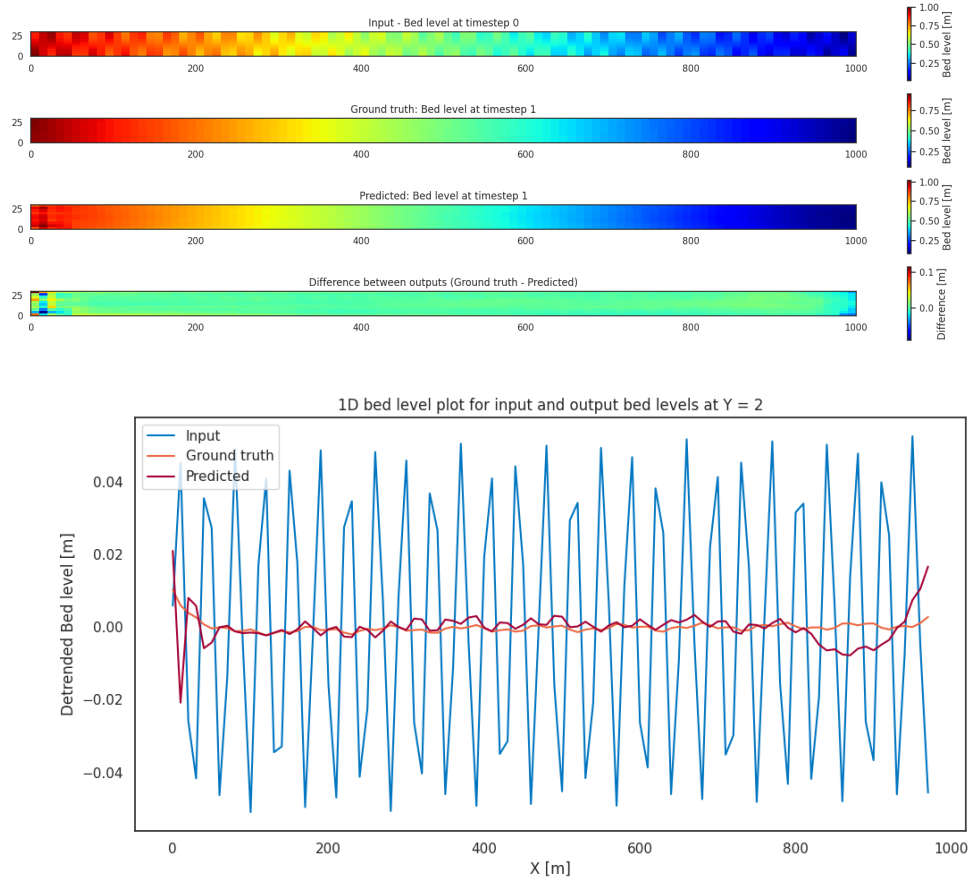

Figure 4.7: Top: 2D grid plot of bed levels input along with ground truth and predicted values. Bottom: 1D representation of the bed level values for wavelength = 36.2 m

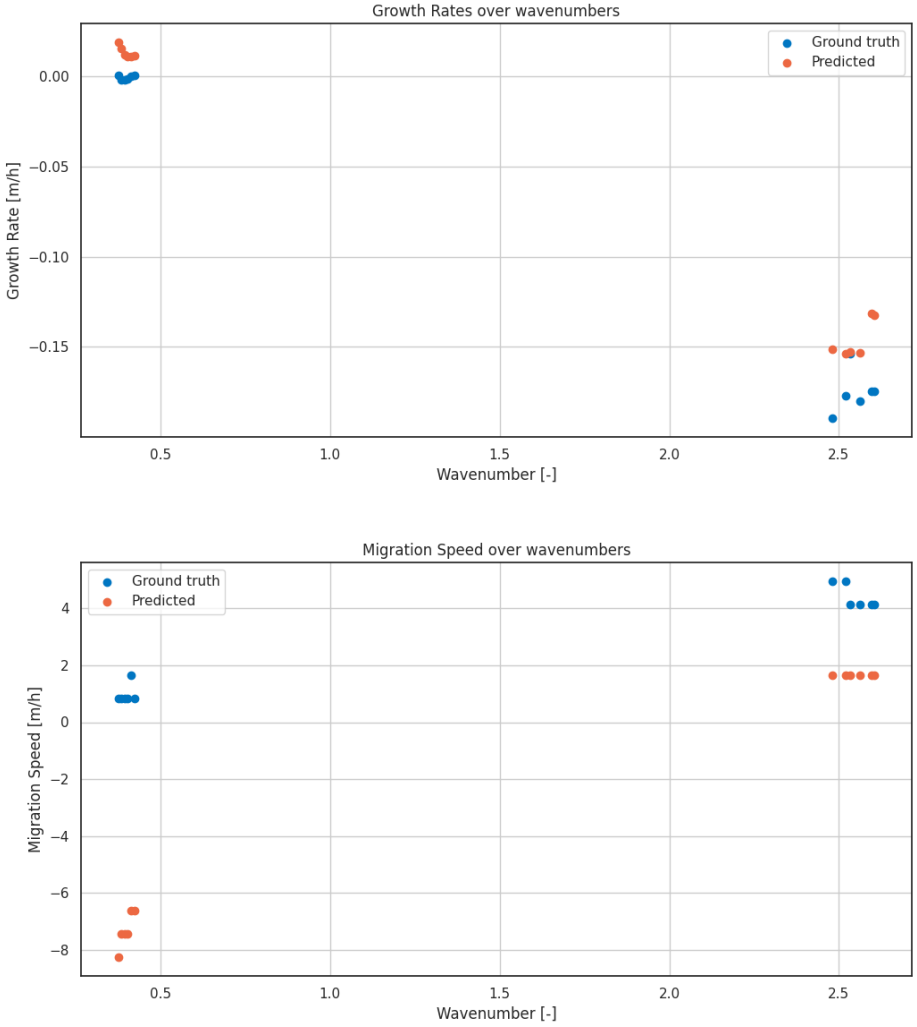

Figure 4.8: Growth rate and migration values

The growth rate values have not significantly changed however migration speed could not be well predicted. This could also be a result of the boundary predictions of the model. As a conclusion of this experiment, adding an extra parameter could not improve the model performance.

An experiment was conducted using the timeseries of 10.5 days to compare the runtime of the CNN model compared to the runtime of a numerical model. The architecture mentioned in the methodology was used by changing the number of output channels to be 21. The results would require extra optimisation hence are not mentioned. An estimate of runtime can however be made. The model on average predicted the timeseries in 0.04 seconds. The average runtime for numerical simulations was 5.56 minutes. The model was able to provide 8000x speedup in the simulation time. This can be significant once the model is accurate.

5
Discussion

The CNN model performed well in predicting the patterns by learning migration-suppression characteristics. The results of the CNN model show less variance from the ground truth values which is suggested from the R^2 value. The value is high also due to lesser difference between the inputs and outputs.

The number of simulations for CNN model plays a crucial role in the model performance and increasing the number of simulations improved the model performance. The number of simulations was doubled from 30 to 60 for this task. The model seems to perform better in predicting growth rate more consistently than migration speed. One potential reason could be the process of finding peaks in the bed levels. In order to compute the migration speed, distance between peaks is considered but the first peak could be further than the one desired due to the set threshold. Ideally the migration speed of simulations in dataset 1 would be difficult to compute the migration because of the complete suppression which explains the huge differences in the migration speeds for high wavenumbers between the predicted and true migration speeds.

Absolute error between the predictions and ground truth is well spread within the grid while the relative error is prominently visible in the downstream of the considered domain. The CNN model would have been influenced by the boundaries more if the NAN values had been replaced by 0.

The model needs to be optimised and new methods to improve the given task can be thought of. The model is not accurate yet and is implemented in the simplest models and simplest scenarios.

5.1. Limitations of the research

The CNN model employed in this research is subject to biases due to limited variability in aspect ratios, boundary conditions, and other parameters defined by physical laws. Consequently, the model performs well only for scenarios similar to its training data and is not suitable for unseen data scenarios. The input data used in this research did not account for higher bar wavelengths, which means the domain of upstream migration remains unexplored. To gain a comprehensive understanding of linear stability, all relevant domains should be included. Model architecture and hyperparameters are critical to the model's performance; thus, detailed optimization of these aspects is essential.

The grid used in this research was relatively small, precluding the use of well-established architectures such as ResNet or U-Net. This issue could be addressed by refining the grid during data preprocessing or by adjusting the model's depth by reducing the number of convolutional and pooling layers. Exploring potential CNN architectures could help identify the most suitable models for this purpose.

The CNN model's predictions are constrained by the provided Cartesian grid. Morphodynamic data, however, can be available in both Cartesian and curvilinear grids, which cannot be directly used in a CNN model without preprocessing. Additionally, the training time for CNN-based models increases exponentially with image resolution (Song et al., 2023). One potential solution is to use vectorized images

instead of raster images. This can be achieved by employing Graph Neural Networks (GNNs), which use input data as graph structures with nodes and edges, accommodating unstructured data types as well (J. Zhou et al., 2020). Consequently, the requirement for a fixed-grid input is mitigated.

Deep learning models involve billions of parameters, but this complexity often comes at the cost of explainability. The results of these models are solely data-driven, creating a dependency on data for robustness and convergence. Incorporating physical parameters into the loss function during training could provide a foundation in physics for the deep learning model. Physics-Informed Neural Networks (PINNs), on the other hand, integrate physical laws through automatic differentiation, bypassing the need for discretization. PINNs have shown potential for solving both forward and inverse problems, as demonstrated by Cedillo et al., 2022. PINNs have been effective in predicting water surface profiles for steady-state open channels with varying flow types and bed profiles. The advantage of PINNs is their ability to incorporate various boundary and initial conditions, generalizing well to unseen scenarios with limited training data. Further exploration of PINNs in predicting riverbed topography is warranted.

Numerical models are known for their computational expense. Although the offline phase of model testing is relatively quick and less costly, training a deep learning model requires substantial computational resources, such as GPUs, which also has environmental implications. Therefore, the consequences of any chosen method must be carefully considered. Training deep learning models can be time-consuming, and ethical considerations, including sustainability, are crucial in engineering practices. van A, 2021 defines Sustainable AI as ”developing AI that sustains environmental resources for current and future generations, supports economic models for societies, and upholds societal values fundamental to a given society.”

5.2. Further Research

The model in this research is trained exclusively on a specific aspect ratio and a limited range of wavelength values, without considering the variability of other parameters from the physical equations. To address this limitation, a global sensitivity analysis could be employed. X. Zhou et al., 2008 describe this approach, which quantifies the impact of varying input parameters simultaneously on the model's outputs. By identifying the most significant input parameters through this analysis, a more comprehensive training dataset can be generated. This will enhance the robustness of the deep learning model, enabling it to adapt to various riverbed characteristics. The model trained on this enriched dataset can then be evaluated through a case study to further assess its generalizability. Alternatively, Principal Component Analysis (PCA) can be utilized to identify and emphasize the significant features influencing riverbed patterns, thereby reducing data complexity. PCA can also be integrated into the CNN model architecture to extract complex patterns that the model may not have captured otherwise. A comparative study of PCA implementations is provided by Amir Mehrabinezhad and Sharifi, 2024.

Consideration should also be given to developing different deep learning models for various morphodynamic formations. This research did not explore the range of bars formed under different conditions. A system of specialized deep-learning models could be developed for distinct tasks. These models would provide predictions when physical characteristics, such as scale and initial conditions, align with the system's constraints. For example, a model designed for low aspect ratios with no forcing and straight channels could predict bar formations accordingly.

Increasing the number of outputs in the model to include additional important parameters, such as other hydrodynamic parameters or turbulence quantities, could enhance the model's utility. Incorporating hydrodynamic inputs, such as velocity, into the deep learning model would help establish a connection between hydrodynamics and morphodynamics.

Long-term predictions pose challenges, particularly when models are trained on data with limited temporal scope. Extrapolation of results is a common limitation in deep learning models; however, this issue can be mitigated by employing sequences of variable lengths for training, applying curriculum learning (Wang et al., 2022), or utilizing hybrid models that combine deep learning with regression

techniques. For instance, He, 2023 demonstrated that combining Long Short-Term Memory (LSTM) models with the ARIMA method outperformed traditional approaches.

6

Conclusion

The aim of this research was to investigate the potential of CNN models in predicting river bed variations. **How accurately can a convolutional neural network predict bed elevations of river bars in one-step-ahead predictions?**
The current model requires further optimization in its architecture and methods to enhance its performance. At present, it is accurately predicting the bed patterns, nor is it capable of forecasting the time series of bed variations. This indicates that the diagnostic step has not been satisfactorily achieved. To address this, the model should be refined, and additional data improvements should be considered.

How does an additional input parameter affect the model's performance?
The inclusion of an additional parameter yielded performance comparable to a model with a single input parameter. The model's performance was more sensitive to boundary values, which could affect the calculation of physical parameters.

How accurately can the model estimate the physical parameters of linear stability analysis?
The model demonstrates better accuracy in predicting the growth rate compared to the migration speed. Incorporating the physical parameters into the loss function during training may enhance the model's performance by providing equal weight to these parameters in its predictions.

What is the magnitude of speed-up provided by the convolutional neural network compared to the runtime of the numerical model? A trial indicated that the CNN model can predict the entire time series in approximately 0.1 seconds, whereas the numerical model requires an average of 5.3 minutes for the same simulations. However, this speed advantage is not yet fully validated due to the model's current accuracy limitations.

In conclusion, this report explores the potential of convolutional neural networks (CNNs) for predicting short-term variations in river bed elevations. The current model requires further refinement and analysis to enhance its performance. Revisiting the data and exploring additional research avenues could significantly contribute to the success of this study and help draw more definitive conclusions.

References

Amir Mehrabinezhad, M. T., & Sharifi, A. (2024). A comparative study to examine principal component analysis and kernel principal component analysis-based weighting layer for convolutional neural networks. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, *12*(1), 2379526. https://doi.org/10.1080/21681163.2024.2379526

Barneveld, H. J., Mosselman, E., Chavarrías, V., & Hoitink, A. J. (2024). Accuracy assessment of numerical morphological models based on reduced saint-venant equations. *Water Resources Research*, *60*. https://doi.org/10.1029/2023WR035052

Bhattacharya, B., Price, R. K., & Solomatine, D. P. (2007). Machine learning approach to modeling sediment transport. *Journal of Hydraulic Engineering*, *133*(4), 440–450. https://doi.org/10.1061/(ASCE)0733-9429(2007)133:4(440)

Cedillo, S., Núñez, A. G., Sánchez-Cordero, E., Timbe, L., Samaniego, E., & Alvarado, A. (2022). Physics-informed neural network water surface predictability for 1d steady-state open channel cases with different flow types and complex bed profile shapes. *Advanced Modeling and Simulation in Engineering Sciences*, *9*. https://doi.org/10.1186/s40323-022-00226-8

Colombini, M., Seminara, G., & Tubino, M. (1987). Finite-amplitude alternate bars.

Colombini, M., & Tubino, M. (1991). Finite amplitude free-bars: A full nonlinear spectral solution. *Journal of Fluid Mechanics*.

Crosato, A., & Mosselman, E. (2020). An integrated review of river bars for engineering, management and transdisciplinary research. *Water*, *12*, 596. https://doi.org/10.3390/w12020596

Deltares. (2024). *Delft3d 3d/2d modelling suite for integral water solutions hydro-morphodynamics*.

de Melo, W. W., Pinho, J. L., & Iglesias, I. (2022). Emulating the estuarine morphology evolution using a deep convolutional neural network emulator based on hydrodynamic results of a numerical model. *Journal of Hydroinformatics*, *24*. https://doi.org/10.2166/hydro.2022.068

Duró, G., Crosato, A., & Tassi, P. (2016). Numerical study on river bar response to spatial variations of channel width. *Advances in Water Resources*, *93*, 21–38. https://doi.org/10.1016/j.advwatres.2015.10.003

Eke, E., Parker, G., & Shimizu, Y. (2014). Numerical modeling of erosional and depositional bank processes in migrating river bends with self-formed width: Morphody- namics of bar push and bank pull. *J. Geophys Res Earth Surf*.

Ermilov, A. A., Benkő, G., & Baranya, S. (2023). Automated riverbed composition analysis using deep learning on underwater images. *Earth Surface Dynamics*, *11*(6), 1061–1095. https://doi.org/10.5194/esurf-11-1061-2023

Goldstein, E. B., & Coco, G. (2014). A machine learning approach for the prediction of settling velocity. https://doi.org/10.1002/2013WR015116.

Gonzalez, F. J., & Balajewicz, M. (2018). Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems. http://arxiv.org/abs/1808.01346

Guo, L., van der Wegen, M., Roelvink, D. (, Wang, Z. B., & He, Q. (2015). Long-term, process-based morphodynamic modeling of a fluvio-deltaic system, part i: The role of river discharge. *Continental Shelf Research*, *109*, 95–111. https://doi.org/https://doi.org/10.1016/j.csr.2015.09.002

He, C. (2023). A hybrid model based on multi-lstm and arima for time series forcasting. *2023 8th International Conference on Intelligent Computing and Signal Processing (ICSP)*, 612–616. https://doi.org/10.1109/ICSP58490.2023.10248909

Hosseiny, H., Masteller, C. C., Dale, J. E., & Phillips, C. B. (2023). Development of a machine learning model for river bed load. *Earth Surface Dynamics*, *11*(4), 681–693. https://doi.org/10.5194/esurf-11-681-2023

Hu, Y., Yang, H., Zhou, H., & Lv, Q. (2023). A review of numerical modelling of morphodynamics in braided rivers: Mechanisms, insights and challenges. *Water*.

Kabir, S., Patidar, S., Xia, X., Liang, Q., Neal, J., & Pender, G. (2020). A deep convolutional neural network model for rapid prediction of fluvial flood inundation. *Journal of Hydrology*, *590*, 125481. https://doi.org/https://doi.org/10.1016/j.jhydrol.2020.125481

Kiani-Oshtorjani, M., & Ancey, C. (2023). Neural network to infer bed topography from velocity field: U-net architecture on huge experimental data. https://doi.org/10.3390/w15234055

Krichen, M. (2023). Convolutional neural networks: A survey. *MDPI: Computers*.

Lanzoni, S. (2000). Experiments on bar formation in a straight flume 1. uniform sediment. *Water Resources Research*, *36*, 3337–3349. https://doi.org/10.1029/2000WR900160

Lesser, G. R., Roelvink, J. A., van Kester, J. A., & Stelling, G. S. (2004). Development and validation of a three-dimensional morphological model. *Coastal Engineering*, *51*, 883–915. https://doi.org/10.1016/j.coastaleng.2004.07.014

Liu, Y., Yang, Z., Li, W., & Li, M. (2024). Applying machine learning methods to river topography prediction under different data abundances. https://doi.org/10.22541/au.164873886.65750516/v1

Nelson, J. M. (1990). The initial instability and finite-amplitude stability of alternate bars in straight channels.

Prince, S. J. D. (2024). *Understanding deep learning*. MIT Press. http://udlbook.com.

Redolfi, M. (2021). Free alternate bars in rivers: Key physical mechanisms and simple formation criterion. *Water Resources Research*, *57*. https://doi.org/10.1029/2021WR030617

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., & Prabhat. (2019). Deep learning and process understanding for data-driven earth system science. *Nature*, *566*, 195–204. https://doi.org/10.1038/s41586-019-0912-1

Sharma, S., Sharma, S., & Athaiya, A. (2020). activation functions in neural networks. http://www.ijeast.com

Siviglia, A., Stecca, G., Vanzo, D., Zolezzi, G., Toro, E. F., & Tubino, M. (2013). Numerical modelling of two-dimensional morphodynamics with applications to river bars and bifurcations. *Advances in Water Resources*, *52*, 243–260. https://doi.org/10.1016/j.advwatres.2012.11.010

Song, Y., Shen, C., & Liu, X. (2023). A surrogate model for shallow water equations solvers with deep learning. *Journal of Hydraulic Engineering*, *149*. https://doi.org/10.1061/jhend8.hyeng-13190

Tubino, M., Repetto, R., & Zolezzi, G. (1999). Free bars in rivers. *Journal of Hydraulic Research*, *37*, 759–775. https://doi.org/10.1080/00221689909498510

Van, S. P., Le, H. M., Thanh, D. V., Dang, T. D., Loc, H. H., & Anh, D. T. (2020). Deep learning convolutional neural network in rainfall–runoff modelling. *Journal of Hydroinformatics*, *22*(3), 541–561. https://doi.org/10.2166/hydro.2020.095

van A, W. (2021). Sustainable ai: Ai for sustainability and the sustainability of ai. *Springer Link*.

Wang, X., Chen, Y., & Zhu, W. (2022). A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(9), 4555–4576. https://doi.org/10.1109/TPAMI.2021.3069908

Wright, N., & Crosato, A. (2011). The hydrodynamics and morphodynamics of rivers. http://www.elsevier.com/locate/permissionusematerial

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, *1*, 57–81. https://doi.org/https://doi.org/10.1016/j.aiopen.2021.01.001

Zhou, X., Lin, H., & Lin, H. (2008). Global sensitivity analysis. In S. Shekhar & H. Xiong (Eds.), *Encyclopedia of gis* (pp. 408–409). Springer US. https://doi.org/10.1007/978-0-387-35973-1_538

7

Appendix

7.1. Model architecture

The entire code used in this research has been published in the following Github repository: Code Repository. The CNN model summary to understand the model architecture used in this research is as follows:

Layer (type)	**Output Shape**	**Param #**
Conv2d-1	[-1, 32, 10, 100]	608
BatchNorm2d-2	[-1, 32, 10, 100]	64
ReLU-3	[-1, 32, 10, 100]	0
Conv2d-4	[-1, 64, 10, 100]	18,496
BatchNorm2d-5	[-1, 64, 10, 100]	128
ReLU-6	[-1, 64, 10, 100]	0
Conv2d-7	[-1, 128, 10, 100]	73,856
BatchNorm2d-8	[-1, 128, 10, 100]	256
ReLU-9	[-1, 128, 10, 100]	0
Conv2d-10	[-1, 128, 10, 100]	147,584
BatchNorm2d-11	[-1, 128, 10, 100]	256
ReLU-12	[-1, 128, 10, 100]	0
Conv2d-13	[-1, 64, 10, 100]	73,792
BatchNorm2d-14	[-1, 64, 10, 100]	128
ReLU-15	[-1, 64, 10, 100]	0
Conv2d-16	[-1, 32, 10, 100]	18,464
BatchNorm2d-17	[-1, 32, 10, 100]	64
ReLU-18	[-1, 32, 10, 100]	0
Conv2d-19	[-1, 1, 10, 100]	289
Total params:	333,985	
Trainable params:	333,985	
Non-trainable params:	0	
Input size (MB):	0.01	
Forward/backward pass size (MB):	10.26	
Params size (MB):	1.27	
Estimated Total Size (MB):	11.54	

Table 7.1: Layer summary of the model architecture.

7.2. Results of non-functioning models

7.2.1. Using less numerical simulations

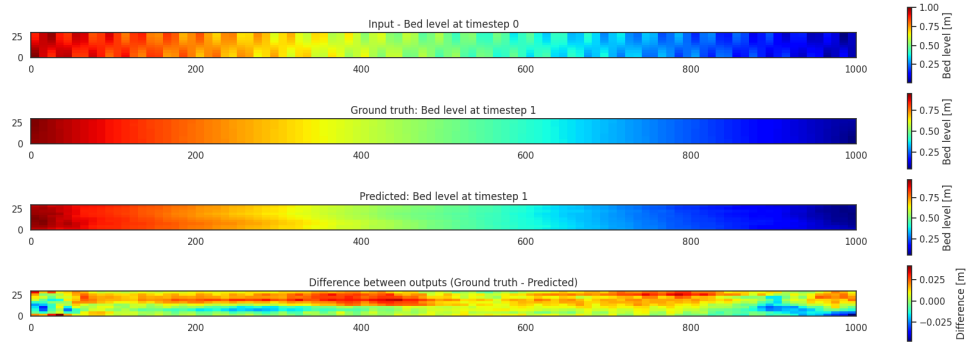

Figure 7.1: Bed level of input with ground truth and predicted bed level values. Wavelength of bars considered = 36.3 m (dataset 1).

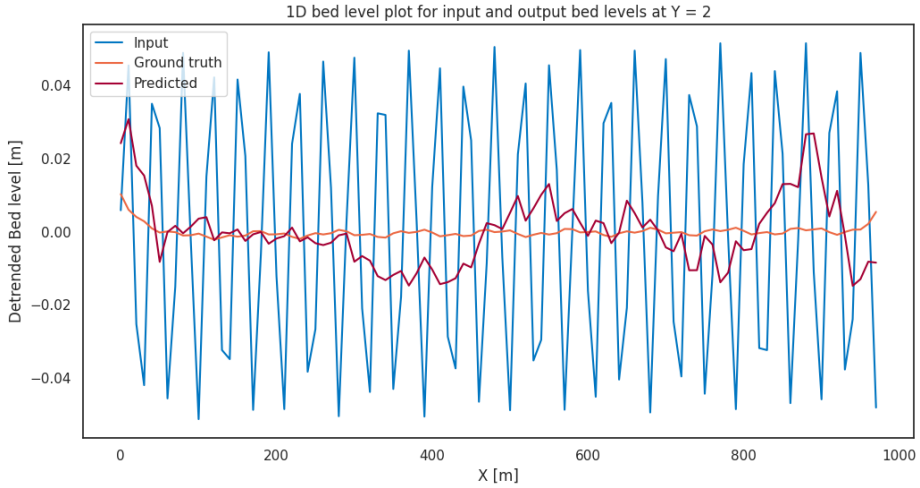

Figure 7.2: Bed level of input with ground truth and predicted bed level values over a defined y value. Wavelength of bars considered = 36.3 m (dataset 1).

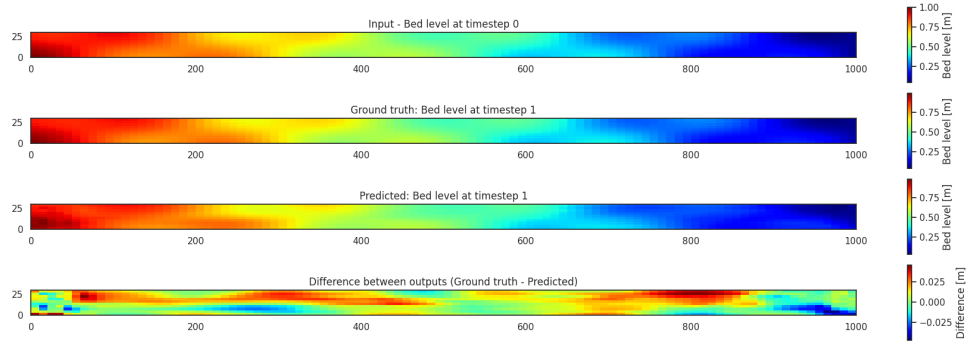

Figure 7.3: Bed level of input with ground truth and predicted bed level values. Wavelength of bars considered = 233.4 m (dataset 1).

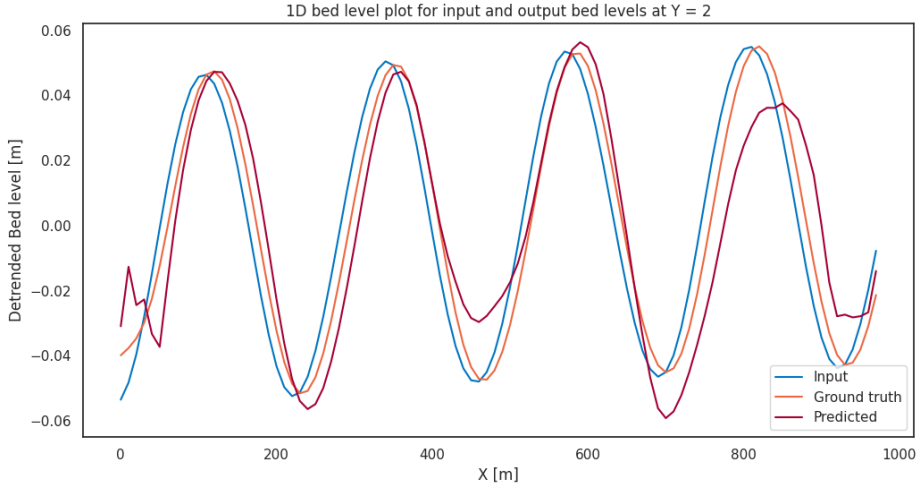

Figure 7.4: Bed level of input with ground truth and predicted bed level values over a defined y value. Wavelength of bars considered = 233.4 m (dataset 1).

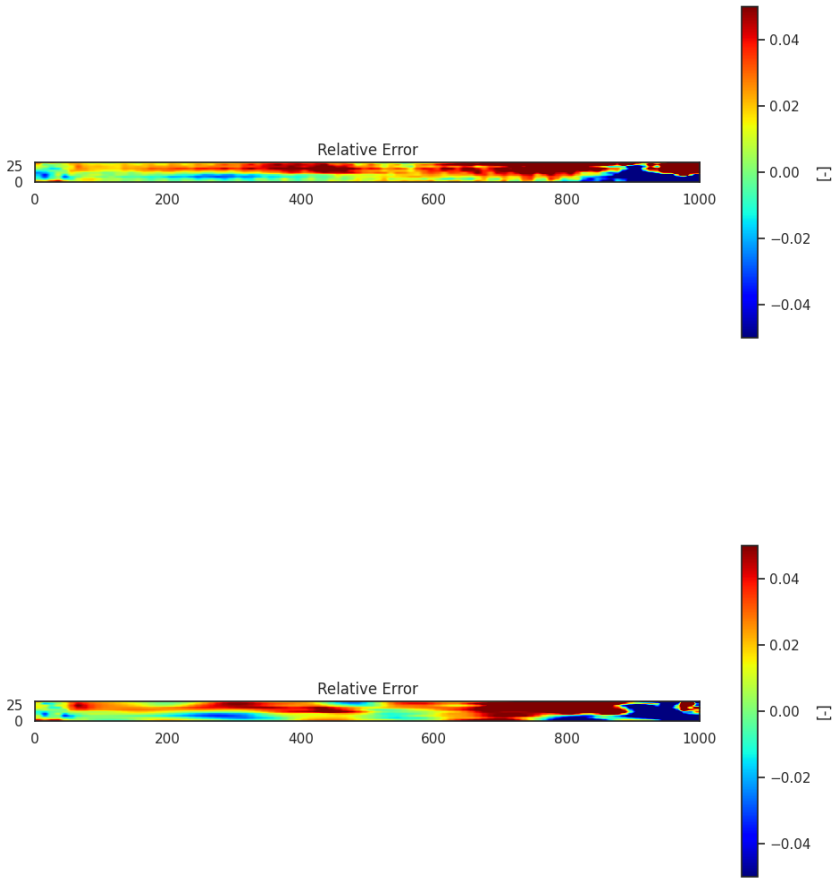

Figure 7.5: top: Relative error in prediction for wavelength 36.3 m from Dataset 1. bottom: Relative error in prediction for wavelength 233.4 m from Dataset 2

7.2.2. U-net model

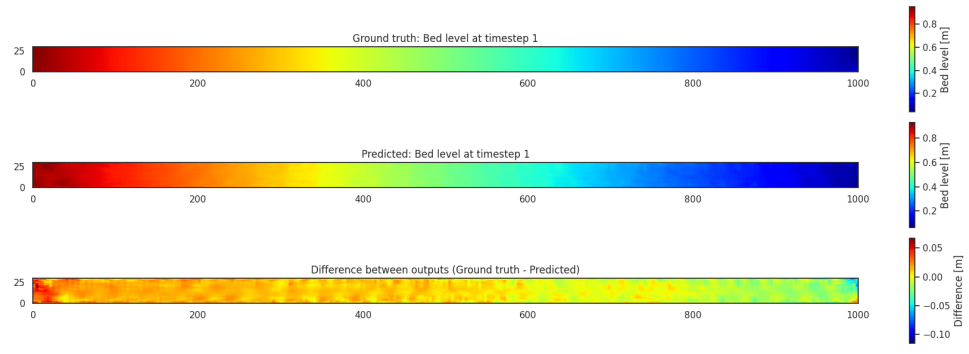

Figure 7.6: Bed level of input with ground truth and predicted bed level values. Wavelength of bars considered = 37.2 m (dataset 1).

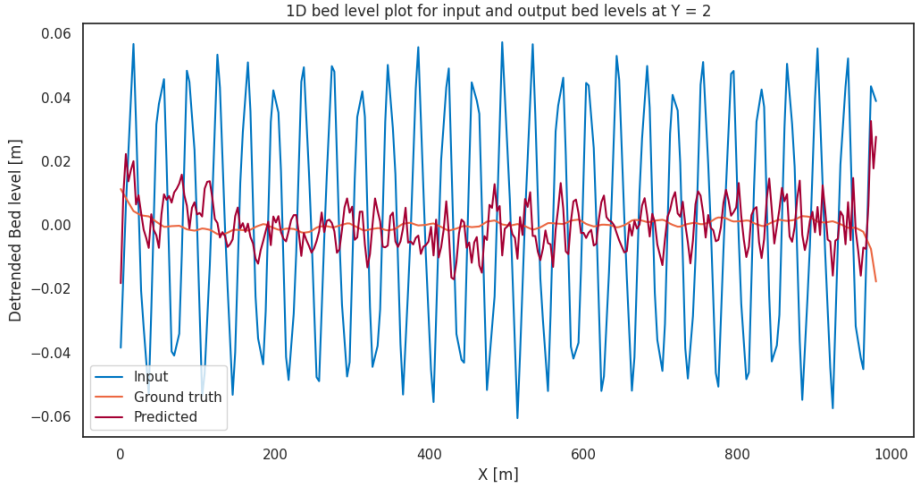

Figure 7.7: Bed level of input with ground truth and predicted bed level values over a defined y value. Wavelength of bars considered = 37.2 m (dataset 1).

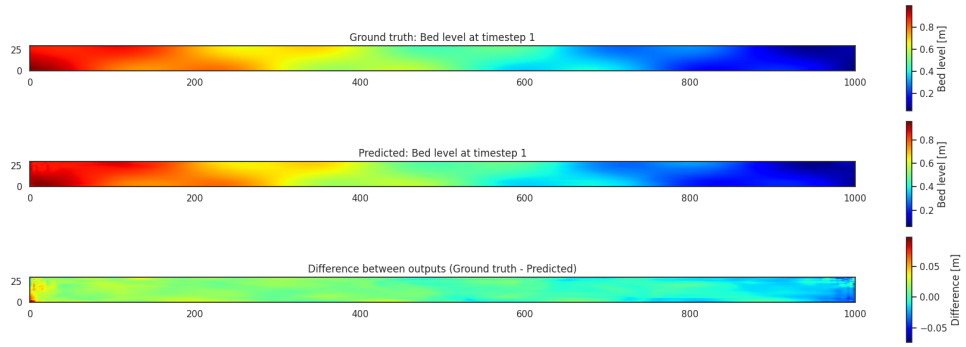

Figure 7.8: Bed level of input with ground truth and predicted bed level values. Wavelength of bars considered = 227.9 m (dataset 1).

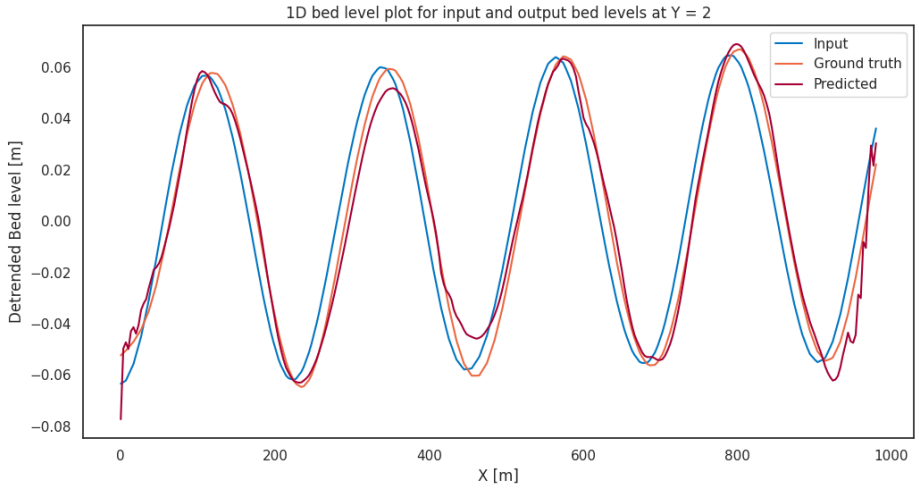

Figure 7.9: Bed level of input with ground truth and predicted bed level values over a defined y value. Wavelength of bars considered = 227.9 m (dataset 1).

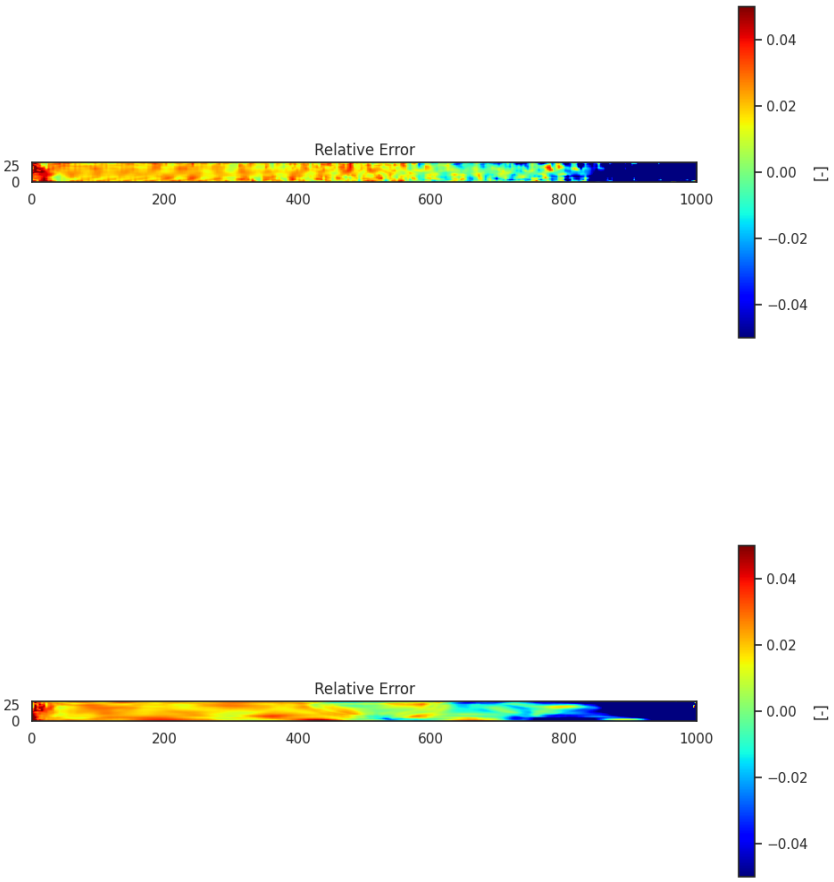

Figure 7.10: top: Relative error in prediction for wavelength 37.2 m from Dataset 1. bottom: Relative error in prediction for wavelength 227.9 m from Dataset 2