Applying Spatio-Temporal Flood Modelling: Insights from Hydraulics-Based Graph Neural Networks on real world scenarios

MSc Thesis

Submitted in partial fulfilment of the requirements for the degree of
Master of Science in Civil Engineering by

Sergio Bulte

To be defended publicly on Tuesday 14th of May 2024 at 15:45.

Delft University of Technology | Faculty of Civil Engineering and Geosciences

Student number:	5414040		
Thesis committee:	dr.ir.	R. Taormina	TU Delft
	ir.	R. Bentivoglio	TU Delft
	dr.ir.	M. Pregnolato	TU Delft
	ir.	R. Dahm	Deltares
	dr.	L. Carniato	Deltares
	ir.	R. Hutten	Deltares

Preface

This is my final thesis report for the MSc Civil Engineering, track Watermanagement at the Faculty of Civil Engineering at the Delft University of Technology. The thesis was completed in cooperation with Deltares at the department of Catchment and Urban Hydrology.

This thesis marks the completion of my MSc studies, and with that the end of my academic carreer. Therefore, this is a good moment to reflect back on the past 7 years. During these years, I learned an incredible amount of new knowledge, met the most amazing people and had the opportunity to do so at three different universities.

For this project, I want to extend the sincerest attitude to my graduation committee. I thank Riccardo, main supervisor, for sparking my interest and introducing me to the deep learning topic. You were able to give me thought-provoking feedback during our meetings in a positive constructive way whilst also granting me the freedom to adjust the project to my own input. I want to thank Maria for her insightful feedback and her expertise to spin this theoretically heavy subject to a more practical direction. And last, but certainly not least, I want to extend my gratitude to Roberto, without whom this project would not have been possible. Besides the obvious contribution, the model, you were always available for a, as I called it, ‘’quick chat’’. In reality, these were more often than not not so quick. You made plenty of time to answer my questions and I always left from your office high spirited with recharged motivation.

Special thanks as well to my company supervisors at Deltares. I thank Luca for taking the time to have a scientific discussion regarding the topic. Your feedback from the sideline really made me think twice about certain subjects. I thank Rinske for helping me set up the Delft3D models. Without your small crash course I would have been struggling heavily with dissecting the models from the waterboards and creating my datasets. And finally, I want to extend my deepest gratitude to Ruben, my daily company supervisor. You were always able to tie the knots together, move the project in to the right direction and ask interesting questions to make me think more critically about certain aspects. You knew exactly how to properly manage the project and put me in contact with all the people I needed, both from inside and outside of Deltares. Besides the useful feedback I must also say I really enjoyed our weekly meetings.

Finally, I was able to work on this research for so long because of the people around me. I want to thank my roommates, my friends and my classmates for their indirect support. Being around you in my free time made me take my mind of the thesis which gave me refreshing insights and renew energy to continue the work. I also want to thank my family, for helping me throughout for the past 7 years and making everything I do possible. From fieldworks and exchanges abroad during my BSc to my internship in my MSc, it would not have been possible without your support. You have always been my example and I owe everything I accomplish to you. Last but not least I want to thank my girlfriend Amber. Not only for your assistance on this thesis, helping me with your superior scientific skills, but also for supporting me every step throughout the process.

Sergio Bulte Garcia

Applying Spatio-Temporal Flood Modelling: Insights from Hydraulics-Based Graph Neural Networks on real world scenarios

Sergio Bulte Garcia[1, 2], **Roberto Bentivoglio**[1], **Luca Carniato**[2], **Rinske Hutten**[2], **Maria Pregnolato**[3], **Ruben Dahm**[2], **and Riccardo Taormina**[1]

[1]Department of Water Management, Faculty of Civil Engineering and Geosciences, Delft University of Technology, Delft, The Netherlands
[2]Department of Catchment and Urban Hydrology, Unit of Inland Water Systems, Deltares, Delft, The Netherlands
[3]Department of Hydraulic Engineering, Faculty of Civil Engineering and Geosciences, Delft University of Technology, Delft, The Netherlands

Correspondence: Sergio Bulte Garcia (Sergio.Bulte@deltares.nl)

Abstract. Understanding the propagation of a flood is crucial for effective emergency response measures. While traditional numerical models provide reliable flood simulations, their high computational costs pose significant limitations during emergencies. Deep learning models have recently demonstrated significant potential in accelerating hydrological calculations while preserving high accuracy. Although various deep learning flood models have been developed, many are limited to specific case studies or neglect the dynamic propagation of flood waves, significantly constraining their application during emergencies. To address this, Bentivoglio et al. (2023) proposes the use of a physics-based surrogate model for spatio-temporal flood modelling; the shallow water equation graph neural network (SWE-GNN). The model demonstrated promising results on small virtual landscapes, showcasing strong generalizability to unseen breach locations and domains, while achieving significant computational speed ups. In this research, we will assess the real-world applicability of the SWE-GNN for time-sensitive situations. We selected two dike rings in the Netherlands as our case study areas. The model underwent training and testing within the same domain to evaluate its application during a crisis. We assess performance using statistical metrics and practical evaluations, including direct and indirect damage models. The SWE-GNN model is able to correctly predict the spatio-temporal evolution of floods for unseen breach locations. The mean average errors in time are of 0.027 m and 0.029 m for water depth and of 0.007 m^2/s and 0.006 m^2/s on units discharge. The resulting flood maps prove viable for practical applicability, presenting good results for both direct as indirect damage assessment. Additionally, the SWE-GNN demonstrates a speedup of roughly 7 to 8 times for the test case areas compared to a traditional numerical model. In this project, we affirm that the SWE-GNN represents a promising innovation for a new approach to time-sensitive flood modeling, providing a reliable alternative to numerical models in situations with time constraints.

1 Introduction

Floods are among some of the most devastating natural disasters, causing huge losses to life and vast property damages yearly across the globe (Jonkman and Vrijling, 2008). According to the World Resources Institute (Ward et al., 2020), by 2030 the number of people negatively affected by floods will be roughly 150 million a year worldwide. Whilst fatalities as a result of floods have been on the decline, the economic impact of floods have been increasing over the past decades. Floods already represent the costliest natural disasters globally and it is estimated that the financial damages will surpass the 100 billion euro per year by the end of this century (Serre and Heinzlef, 2018). Flood protection measures such as dikes are used to protect inland areas from flood events. Nevertheless, there are instances where these measures can fail, which results in a flood (Apel et al., 2009). In those cases, it is pivotal to predict and understand flood behaviour for effective flood management and damage mitigation. This crucial understanding applies to flood preparedness, as well as time-sensitive emergency

applications (Henonin et al., 2013).

Accurate flood models play a crucial role in the risk assessment, early warning, and preparedness for flood events. Numerical models offer reliable results in their flood simulations from dike breaches. The models employ computational algorithms and physical equations to replicate the interactions between water, dike structures, and surrounding terrain. The flood propagation is simulated by considering hydraulic processes such as breach development and flow dynamics. Among various modelling techniques the hydrodynamic 2D models are widely favored (Teng et al., 2017; Bates, 2022). These models offer the spatial and temporal evolution of floods by solving the Shallow Water Equations (SWE) (Vreugdenhil, 1994), by employing finite-volume or finite-difference methods (Alcrudo and Garcia-Navarro, 1993). Numerical models can take topographic elements, dike characteristics, and initial water levels as inputs to compute the propagation of the flood with its water depths, flow velocity, and other relevant parameters for different flood stages (Anees et al., 2016).

However, the computational complexity of numerical models restricts their real-time applicability during emergencies, and pre-ran simulations from a database are unlikely to suit situation specific boundary conditions (Henonin et al., 2013; Nayak et al., 2005). To address this issue, several approaches have been developed to accelerate the solution of these equations. One approach focuses on using high-performance computing and parallelization techniques to enhance computational efficiency (Hu et al., 2022; Petaccia et al., 2016). This method offers good results but proves to be computationally heavy and restricted by numerical constraints. Another way of increasing the computational speed is to use a simplified form of the SWE. The local inertia form was, for example, used in several research and offered some speed ups at the cost of accuracy (Bates et al., 2010; Almeida et al., 2012). However, these methods lacked stability at low friction scenarios and presented a general limitation of predicting wave propagation at relatively high Froude numbers. This is a common constraint for flood models where locations with higher Froude numbers can lead to model instability (Neal et al., 2012). Several other approximation methods have proven to also only be valid for domains with low spatial and temporal gradients. These limitations need to be taken into account when utilizing such techniques (Costabile et al., 2017).

The use of data-driven alternatives offers a good solution for time sensitive applications (Bentivoglio et al., 2022). As of recent, neural networks (NNs) are an upcoming method for deep learning flood modelling. NNs demonstrate powerful capabilities in approximating strong non-linear correlations, in addition to the ability to automatically discover the representations required, such as those necessary for classification in raw data (LeCun et al., 2015). In many instances NN-based models have shown to outperform other machine learning methods for flood modelling in terms of speed and accuracy (Wang et al., 2020; Zhao et al., 2020).
Generally, the studies of NN-based models can be divided into categories based on their architecture. The simplest method for deep learning flood modelling remains a multilayer perceptron (MLP). Within MLP architectures we differentiate between fully connected and encoder-decoder (ED) MLPs. ED MLPs only consider certain latent representations of the input data to represent the output (Taormina and Galelli, 2018). MLPs are widely favored for deep learning flood modelling but do produce less coherent results due to their lack of inductive bias. Convolutional neural networks (CNNs) are also adopted for spatial flood predictions. These models do contain an inductive bias and therefore are found to widely outperform MLPs (Bentivoglio et al., 2022). Finally, recurrent neural networks (RNNs) are used in temporal modelling and sequential data analysis.
In Guo et al. (2021), maximum flood water depths are predicted using image-to-image modelling techniques. Their CNN was able to produce promising results with significant speed ups after training. Zhou et al. (2022) explores the performance of a 1-D CNN with a spatial reduction and reconstruction method to simulate the spatio-temporal variation of flood inundation of a set of locations. In the last few years, RNNs have also proven to offer reliable and computationally fast modelling results for different flood processes. For example, in multi-step-ahead flood forecasting, where RNNs demonstrated superior performance than alternative models (Chang et al., 2014; Chen et al., 2013). However, one constraint of the previously mentioned models is the inability to perform well over unseen topographies.
This issue, where a model struggles to generalize to unseen locations, is explored in several studies with the aim of potentially applying data-driven models on domains beyond their initial training data. In terms of flood modelling, Guo et al. (2022) proposed a CNN-based model that can be reused on different catchments with different topography. To process the variable sizes and shapes of the catchments, the study divided the domains into patches. Löwe et al. (2021) developed a CNN for maximum flood prediction. The model is trained on raster data from a city and later tested on unseen raster data from the same city. Both of these flood models achieved good accuracy levels and showed promising generalizability capabilities, but lacked the dynamic behaviour of the flood in space and time.

However, the spatio-temporal evolution of the flood is crucial for the authorities during emergencies in order to ensure effective flood response measures. For this reason, Bentivoglio et al. (2023) proposes a hydraulic-based Graph Neural Network (SWE-GNN). This model significantly reduces the hydrodynamic computation time, ensures high prediction accuracy, and overcomes the generalizability

issue whilst providing the spatio-temporal behaviour of a flood. The graph neural network (GNN) generalizes a CNN to an irregular domain, which in this case is a graph that represents the mathematical domain of the model. Each node in the graph represents a finite volume cell which has corresponding hydraulic variables. The fluxes of water between these cells follow an explicit numerical discretization of the SWE from which the GNN learns how to propagate water based on the hydraulic gradient between cells. The SWE-GNN proves to be valuable, opening the door to a new approach for replacing numerical models with a deep learning counterpart. The results presented in the paper demonstrate the promising potential of GNNs for flood simulations. This model, however, has only been applied to virtual domains that are squared and relatively small. To fully explore the capabilities of the SWE-GNN, this study will apply the model on a real world case study. With a computational speed up of over two orders of magnitude and an increased generalizability, the integration of this model could be beneficial for practical use during emergencies.

In this paper, we apply the SWE-GNN for flood simulation on two real world case studies: the dike rings 43 and 49, located in the Netherlands. The performance of the model is assessed through statistical metrics, as well as evaluated for practical application for the first 48 hours of a calamity. We examine the trade-off between the increased computational speed and data quality and analyze the performance on impact metrics. If functional, this model could be a promising development for deep learning flood modelling. Offering practitioners a tool for fast and reliable flood results which can be useful during hot phases of emergencies or for speeding up probabilistic calculations.

2 Methodology

This section briefly introduces the topic of graph neural networks before presenting the SWE-GNN and the numerical model used to simulate flooding. Subsequently it presents the metrics used to assess the performance of the SWE-GNN, the theoretical framework to assess the corresponding flood damages, and finally the employed damage models.

2.1 Graph Neural Networks

Graph neural networks are certain deep learning models that can be applied to graph structured domains. The mathematical structures of these graphs consist of nodes v_i connected to each other by edges e_{ij}. Where the indexes of nodes represent the i-th node and the indexes of edges represent the nodes it connects.

Nodes can have certain assigned properties, called node features. The features for a node can be expressed by a vector as $\mathbf{x}_i \in \mathbb{R}^N$, where N represents the number of features. Additionally, edges can also contain edge features. These can be represented by a vector as $\mathbf{e}_{ij} \in \mathbb{R}^D$ where D denotes the number of features.

If two nodes are connected by an edge, they are able to pass messages to each other. The information in these messages depends on the node features of the two adjacent nodes and the edge features of the edge connecting them. A message between node v_i and v_j can be computed as

$$\mathbf{m}_{ij} = \psi(\mathbf{x}_i, \mathbf{x}_j, \mathbf{e}_{ij}) \tag{1}$$

When a node receives a message from its neighbor, it can update its own node features with the message information as follows

$$\mathbf{h}_i = \sigma\left(\mathbf{x}_i, \bigoplus_{v_j \in \mathcal{N}_i} \mathbf{m}_{ij}\right) \tag{2}$$

In this equation, $\mathbf{h}_i$ denotes the hidden state of node v_i. The hidden state is the updated representation of the node after the message passing. $\sigma(\cdot)$ is a function that is employed to induce non-linearity. $\oplus(\cdot)$ represents a function which obtains all the messages from neighboring nodes. If this process is executed once, information can only pass between nodes neighboring each other. However, the architecture of graph neural networks allows for a number of L layers to be stacked. Each of these layers represent a repetition of the message passing process, updating the hidden states of all nodes for each layer every time. This results in the propagation of information between nodes further away from each other. Specifically nodes that are separated by L edges (Sanchez-Lengeling et al., 2021).

In the context of this research, nodes represent the mathematical grid cells of our case study areas. The associated node features will be discussed in the next section.

2.2 SWE-GNN

Bentivoglio et al. (2023) proposed the use of the Shallow Water Equations Graph Neural Network (SWE-GNN) for modelling floods. This deep learning model combines graph neural networks (GNN) with the finite volume method to solve the shallow water equations. In this model, the hydrological fluxes are learned instead of being calculated. This is done during the training phase, where it requires numerical flooding simulations as data for training and comparing. The advantages of employing a GNN over alternative neural networks lie in for example the propagation rule of the network. This allows the propagation of water to happen in a coherent manner, which is not the case in other deep learning models. Additionally, GNNs can perform on irregular domains like graphs whilst proving to produce reliable results for fluid dynamics as well as for partial differential equations (Horie and Mitsume (2022)).

The SWE-GNN model incorporates both static and dynamic

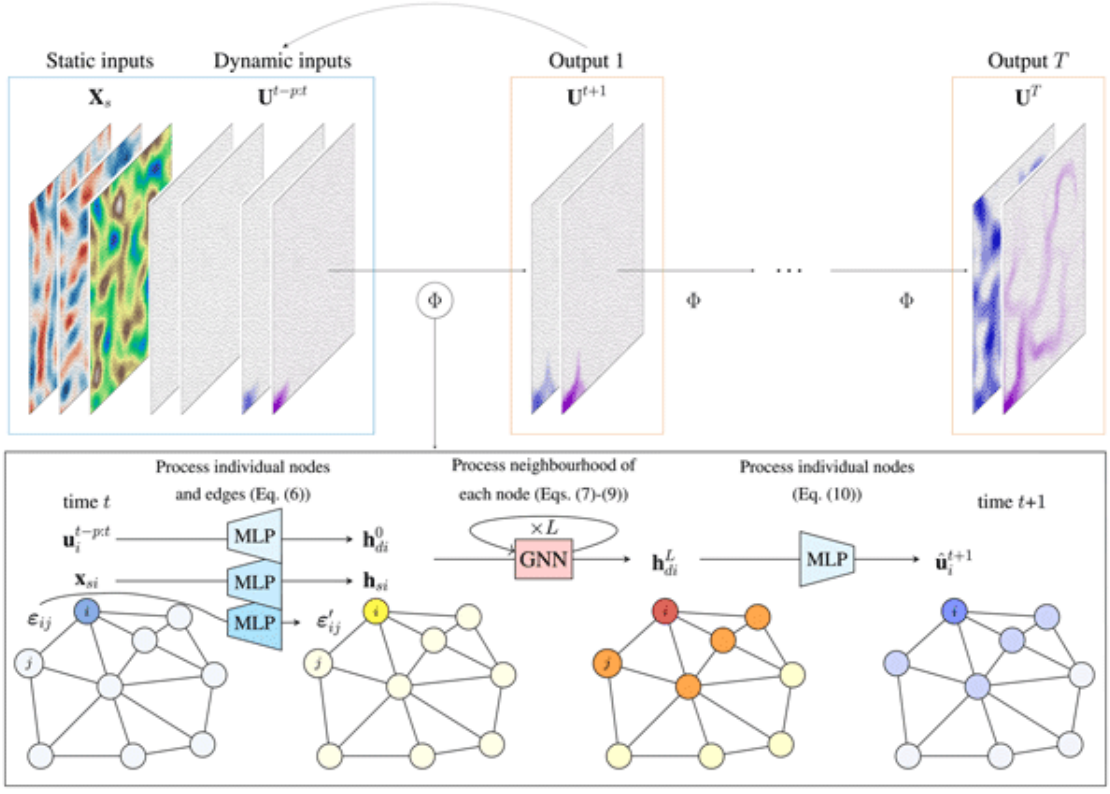


Figure 1. Overview of the architecture structure of the SWE-GNN as presented in (Bentivoglio et al. (2023)). The model ϕ takes both the static and dynamic features at time t (blue box) and produces a calculation on their evolution in time (orange box). The model uses its own predictions as input for the subsequent calculations to auto-repressively determine the spatio-temporal evolution of the flood wave. The encoder-processor-decoder structure of the model is shown in the black box. Both the static and the dynamic node inputs (first node mesh) are encoded into higher-dimensional embeddings (yellow nodes) using three separate multi-layer perceptrons. These are then used as input for the L layers of the GNN. Subsequently, the output of the GNN (red and orange nodes) is decoded via a shared multi-layer perceptron and summed to the hydraulic variables at time t. This results in the final prediction of the variables at time t+1 (blue nodes).

features as input. The static features represent the topography of the domain, while the dynamic features capture the hydraulic variables at time t. The model output consists of the hydraulic variables at time t+1 as follows

$$\hat{\mathbf{U}}^{t+1} = \mathbf{U}^{t} + \Phi(\mathbf{X}_s, \mathbf{U}^{t-p:t}, \boldsymbol{\mathcal{E}}), \tag{3}$$

The output $\hat{\mathbf{U}}^{t+1}$ are the predicted hydraulic variables at time t+1, $\mathbf{U}^{t}$ corresponds to the hydraulic variables at time t, Φ is the GNN model that autoregressively calculates the hydraulic variables for the time step, $\mathbf{X}_s$ consists of the static node features, $\mathbf{U}^{t-p:t}$ are the dynamic node features and $\boldsymbol{\mathcal{E}}$ are the edge features that present the geometry of the mesh. The complete structure is illustrated in detail in Figure 1. More in depth information on the functioning of the SWE-GNN can be found in Appendix A or in the original paper; Bentivoglio et al. (2023).

For this research, some modifications were made to the SWE-GNN to accommodate variable-sized domains instead of squared ones. The SWE-GNN version employed for conducting this research, required the first time step of testing simulations to be simulated by a numerical solver, in this case Delft3D-FM (Deltares, 2024). This resulting first step of the flood evolution is then used as input for the SWE-GNN.

2.3 Numerical model

The numerical model used for this research is the high fidelity numerical solver Delft3D FM Suite 1D2D from Deltares, hereafter referred to as Delft3D. The model is widely adopted and applied both worldwide as well as nationally for a variety of riverine, coastal and estuarine hydrodynamic applications. Dike breaches can accurately be simulated by incorporating aspects such as breach development and river water levels. Additionally, the model is able to simulate a flood originating from a point source, either with a fixed discharge or a defined hydrograph. This model employs an implicit scheme on grids and variable time steps to solve the Shallow Water Equations (Deltares, 2024).

Delft3D can be operated by making use of an external, python based, programming software. This enables users to continuously model multiple simulations with changing boundary conditions by utilizing a batch file.

2.4 Metrics

To assess the statistical performance of the SWE-GNN, we consider various metrics. Firstly, in order to measure the spatial error of the flood distribution we calculate the critical success index (CSI) as

$$\text{CSI} = \frac{TP}{TP + FP + FN} \tag{4}$$

TP refers to true positives and entails all the pixels where the SWE-GNN correctly predicted flood. FP stands for false positives, which is the number of cells where the model wrongly predicted flood. FN is false negatives, which refers to the cells where the model did not recognize flooding. In the CSI we leave out the true negatives, as this would lead to an overestimation of the performance since large parts of the domain are not flooded. This is especially apparent in the first few time steps of the simulation at the start of the dike breach. This value is computed for every time step of a certain simulation and gives insight into the spatial accuracy of detecting a flood or no flood per pixel, given a certain threshold. In this study, we will employ flood thresholds of $0.05\mathrm{m}$ and $0.3\mathrm{m}$ to differentiate between wet and dry pixels, since they are widely recognized as a good thresholds for indicating the beginning of flood and for safe driving in regular civilian cars (Pregnolato et al., 2017).

Besides measuring the accuracy of predicting a certain class, we also measure the error of the data simulated. For this, we employ the multi-step-ahead Mean Average Error (MAE) for the hydraulic variables water depth and flow velocity, which is defined as

$$\text{MAE} = \frac{1}{H} \sum_{\tau=1}^{H} \|\hat{u}^{\tau} - u^{\tau}\|_1 \tag{5}$$

Where H refers to the prediction horizon which depends on the total simulation time and the temporal resolution. $\hat{u}^{\tau}$ represents the predicted value of the hydraulic variable at time τ and u^{τ} represents the true value of the hydraulic variable at time τ.

2.5 Flood damages

In the context of urban flood events, the most interesting aspect for policy makers is the inflicted damage. Flood damage refers to all the destruction caused by a flood event. It comprises all the harmful effects on human life, damages to their belongings, damages on infrastructure, on the ecosystem, and finally also on the effect on the strength of the affected economy. In this research the categorization of damage will be made based on direct and indirect damages.

Direct damages refer to physical destruction caused by the floodwater during the flood event. This entails damages to buildings, infrastructure, and the loss of lives. These damages are generally more straightforward in their estimation and effects (Kok et al., 2004; Penning-Rowsell et al., 2005). Indirect damages refer to secondary losses that stem from the direct damage but are not directly caused by the inundation of water. These include, for example, the disruption of services and the limitation of accessibility due to flooded networks. Indirect damages may not be immediately apparent but can have a long lasting effect on communities and can affect areas larger than those actually inundated by the flood (Nicholls et al., 2014; Rose and Liao, 2005; Koks et al., 2015).
Urban areas heavily rely on infrastructure networks, which are commonly regarded as the fundamental framework supporting proper functioning of communities. Among these networks, transportation systems like roads play a crucial role in ensuring safe and strong communities (Rodrigue, 2020). During floods, roads may experience both direct and indirect damage from the water. Directly, flooding can cause destruction to the road infrastructure itself (e.g. pavement, surface). However, the importance of roads extends beyond direct damages, as they play a crucial role in indirect consequences as well. When a road becomes inaccessible during a flood, it hinders the mobility and connectivity to specific areas, thereby disrupting emergency response efforts in those regions (Yu et al., 2020).

Damages can be expressed in tangible or in intangible terms. Tangible terms refer to measurable and physical losses. These entail, for instance, monetary losses or fatalities due to a hazard. In contrast, intangibles encompass effects that might not be directly observable such as emotional distress or societal disruption. These effects are harder to quantify but can be crucial in understanding the full effect of a calamity (Messner and Meyer, 2007; Thieken et al., 2009).

2.6 Damage models

To evaluate the practical applicability of the SWE-GNN in real case studies, two damage models will be used. The damage models will be SSM2017-v4.0 (Schade Slachtoffer Module) and Ra2Ce (Risk Assessment and Adaptation for Critical infrastructurE), for calculating direct- and indirect damages respectively (Rijkswaterstaat, 2023; Deltares, 2023).

2.6.1 Schade Slachtoffer Module

The SSM2017-v4 is a damage tool created by Rijkswaterstaat, the Dutch national agency responsible for the management and maintenance of the main infrastructure facilities in the Netherlands, including roads and waterways. The tool takes as minimal input the flood depths, but can also take flood velocity into account (Slager and Wagenaar, 2017). For the purpose of this research both are considered in order to not underestimate the damages. The module employs the 'Standard Method damage and casualties'. This method consists of all the possible steps to calculate the damages of a certain flood on a dike ring. This method can take into account conditions such as the location of breach, timing of breach, breach growth, maximum water levels, number of residents, infrastructure vulnerability, and other relevant factors to calculate the total effect of a hazard. More in depth information on the method is provided in Kok et al. (2004). The calculation is based on the unit-loss methodology per category, which relates flood quantities to damage at unit level. Examples of some categories can be agriculture, dwellings, infrastructure. Their corresponding unit is then m^2. These damages can be calculated using the following relation

$$S = \sum_{i=1}^{m} \mathrm{s}_i \sum_{j=1}^{n} \mathrm{f}_{ij}(\mathrm{d}_j)\mathrm{n}_{ij} \tag{6}$$

This equation, which is visualized in Figure 2, calculates the total loss S by summing up the total damage for categories m and all grid cells n. The damage per grid cell is a multiplication of the damage fraction $\mathrm{f}_{ij}(\mathrm{d}_j)$ and the elements exposed n_{ij} multiplied by the value of risk of the different damage categories s_i. The damage fraction is obtained through a damage function (Wagenaar et al., 2019).

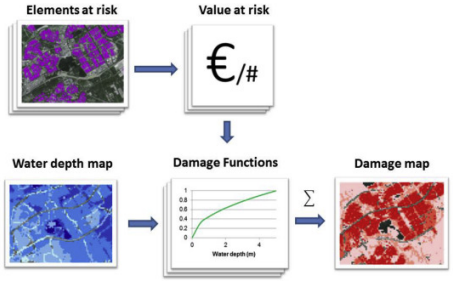


Figure 2. Visualization of the flood damage calculation procedure (Wagenaar et al., 2019)

Every category of damaged good has its own damage function. For example; for low frequency flood areas the damage function for vehicles is displayed by Figure 3.

For the cost of direct damages on buildings like houses the 'replacement cost' is used. This means the monetary cost to restore or rebuild a building to its original state in the original location. Multiple different housing categories are distinguished, from apartments to mansions and the like. For damage to assets like capital goods and vehicles the 'replacement cost' is used, this depends on the current market value of the lost asset. Damage costs produced by the SSM are before taxes.

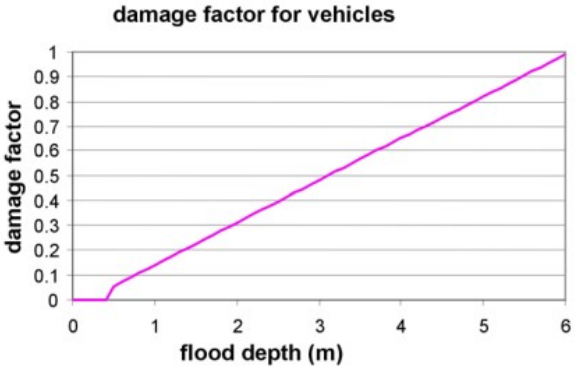


Figure 3. Damage factor per flood depth for vehicles in a low frequency flood area as presented in Kok et al. (2004)

The SSM uses Delft-FIAT as calculation core. FIAT (Fast and Flexible Flood Impact Assessment Tool) is a Python based tool developed by Deltares and widely adopted for flood damage calculations (Slager et al., 2016). For the direct damage assessment both the flooding results of the Delft3D and SWE-GNN runs were analyzed per breach location test case for comparison. In this evaluation, we considered factors such as damage costs, fatalities, and the number of individuals affected.

2.6.2 Risk Assessment and Adaptation for Critical InfrastructurE

For the indirect damages related to road networks a Ra2Ce model was set up on the case study areas. Ra2Ce takes road networks as vector line elements and overlays it with a flood map raster containing water depths. The exact location of the road elements are obtained from the Rijkswaterstaat database in addition with road information like road type and average speeds. The tool is then able to analyze the impact of the hazard on the transportation network. The results can provide insights into network bottlenecks, fully or partially closed off road segments and isolated locations due to the hazard. The executed analysis for this project serves to assess whether certain locations can still access the main part of a transportation network during a disruption. Ra2Ce differentiates between two reasons of isolation: a location can be isolated because of disruption in a nearby link or because of link disruptions in areas further away. For example, a house could be disrupted in using the road network due to flooding on the road right in front of the house or because neighborhood

roads are flooded, preventing departure from the area (Bles et al., 2023).

3 Experimental setup

This section introduces the project approach, the study areas and discusses the dataset creation used for this project. Finally, the training set up to create the SWE-GNN model is explained.

3.1 Project approach

The first step of this research is to set up a workflow. This process is visualized in Figure 4.

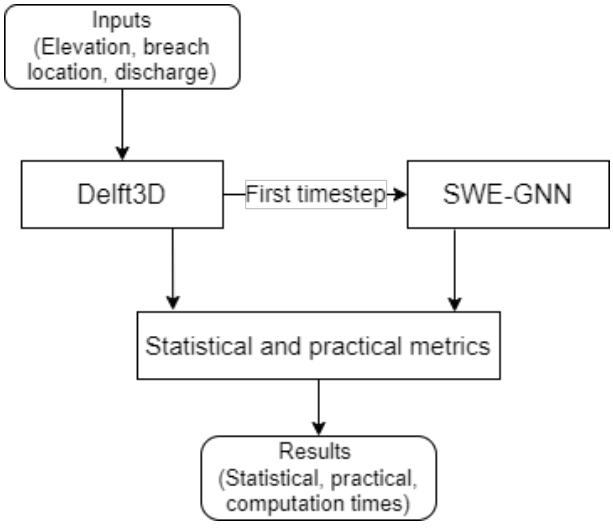


Figure 4. Visualization of the general workflow of this project. Test simulations will be run on both Delft3D and the SWE-GNN, the results will then be processed through the statistical metrics for assessment and the damage models for practical applicability.

With this workflow in place, where the results of the SWE-GNN on the test datasets will be compared to the Delft3D results, we will research the potential of the SWE-GNN for real world application.

3.2 Study areas

Two study areas are considered: dike rings 43 and 49 (Figure 5). The areas are located in the east of the Netherlands in the province of Gelderland. The dike rings are overseen by the local waterboards, Waterschap Rivierenland for dike ring 43 and Waterschap Rijn en IJssel for dike ring 49. Dike ring 43 is encapsulated by the Waal River in the south and the Nederrijn in the north. Dike ring 49 borders the Rhine and the IJssel. Although close in distance, the spatial properties of the two case studies differ. Dike ring 49 roughly drains in northwest direction while dike ring 43 is more flat. Additionally, dike ring 43 has a different distribution of spatial structures such as small lakes and elevated 1-d structures such as secondary dikes and highways. Dike ring 43 is divided in to two parts, west and east. The surface area of this eastern half is more comparable to dike ring 49, with the original complete dike ring 43 being more than twice the size. This renders it too large for a fair comparison with a training dataset of the same size as for dike ring 49. For this purpose we focus on the eastern part of the dike ring. The domain is clipped, and the training dataset is therefore of the same size to fairly assess the model, to maintain a good complexity of the GNN, and to attempt to limit overfitting.

3.3 Dataset creation

Data on the dike rings was obtained from the responsible authorities; Waterschap Rivierenland and Waterschap Rijn en IJssel. This includes information such as dike ring extent, topography, different calculation grids, pumps and sluices. The information was loaded into Delft3D to create the flood simulations used in this research. The resulting Delft3D simulations will serve as basis for this project. This means both the creation of a training datasets as well as testing dataset for performance comparison with the SWE-GNN.

- Training dataset 1 comprises 60 flood simulations on dike ring 49. The breach locations are randomized in the training dataset, with dry bed conditions and a constant flood discharge of 200 $\mathrm{m^3 s^{-1}}$ as initial and boundary conditions. This discharge was obtained from the overseeing waterboard and deemed relevant for further post-processing of the resulting flood maps. This was chosen since the main focus of the research is to assess the real-world applicability of the SWE-GNN.

- Training dataset 2 comprises 60 flood simulations on the eastern part of dike ring 43. The breaching discharge is set at 200 $\mathrm{m^3 s^{-1}}$ in accordance with the relevant authorities.

- The testing dataset will comprise 10 flooding simulations on dike ring 49 and 5 on the eastern part of dike ring 43. This is to test the generalizability to unseen breach locations. The breach locations are chosen in accordance with the overseeing authorities or obtained from the LIWO (Landelijk Informatiesysteem Water en Overstromingen) which is a database with the most critical flood scenarios per dike ring for The Netherlands (Rijkswaterstaat, 2022). The breaching discharge is set at 200 $\mathrm{m^3 s^{-1}}$ for both dike rings.

The training and validation split of the two training datasets will be 75% for training and 25% for validation. This results in 60 training samples and 20 validation samples on the first dataset. For the second dataset, this amounts to 45 samples for training and 15 for validation. The temporal resolution of all simulations is Δt = 1 h.

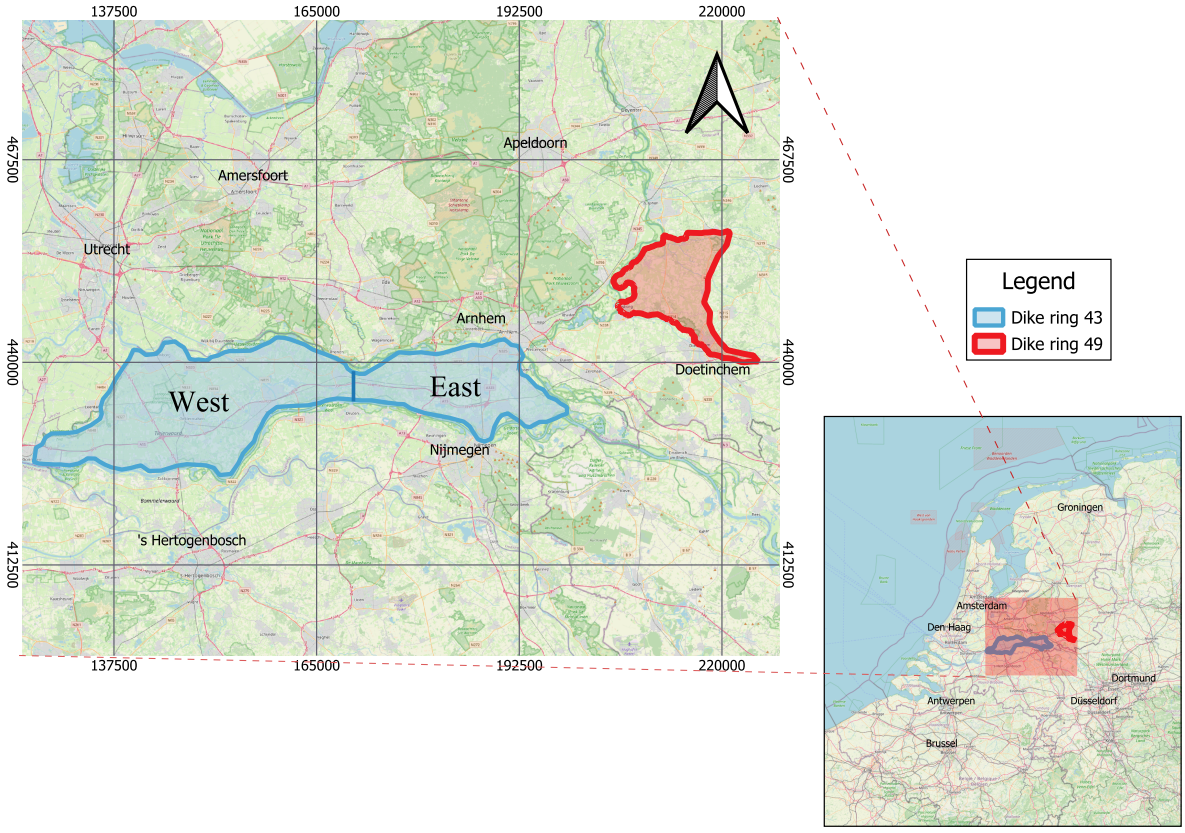


Figure 5. Case study areas for this project, dike rings 43 and 49 in the east of The Netherlands. Data obtained from waterboards Rivierenland and Rijn en IJssel respectively. Coordinate reference system set on EPSG:28992 Amersfoort/RD New.

3.4 Training setup

We train the SWE-GNN via the Adam optimization algorithm (Kingma and Ba, 2014). Training was carried out for 150 epochs, with an early stopping function if the training shows to not result in improvements. The learning rate is 0.009 in combination with a step decay strategy. The reduction factor is 0.7 and the step size is set at 15 epochs. The number of rollout steps during training is initially set at one time step ahead. This number will be increased during training based on the curriculum learning strategy that is updated every 25 epochs. A more detailed description of the training strategy is provided in Bentivoglio et al. (2023) in section 3.3.

The scripts are written in Python 3.10 (Inden, 2022) and the models are trained utilizing Pytorch version(1.13.1) (Paszke et al., 2019) and Pytorch Geometric (version 2.2) (Fey and Lenssen, 2019). Weights & Biases is used to keep track of training progression as well as execute the hyperparameter search (Biewald et al., 2020). In terms of hardware training the SWE-GNN is done on a NVIDIA A100 80GB GPU (Delft High Performance Computing Centre, 2022). Testing the SWE-GNN as well as creating the datasets with Delft3D is done on an Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz - 2.71 GHz.

4 Results and Discussion

In this chapter, the results of the SWE-GNN models trained on dataset 1 and dataset 2 will be presented. We will asses whether the model is sufficiently trained for the testing data from both datasets separately by evaluating the results with the discussed metrics and comparing them to the Delft3D results through the workflow discussed in chapter 3. All SWE-GNN model testing and Delft3D simulations were run on CPU in order to fairly assess the model capabilities.

A hyperparameter search was conducted with the goal of finding the configuration which maximizes the models performance on the specific task at hand. The chosen metric was the validation CSI which was calculated during training on the validation section of the dataset. The results and more details of this evaluation for both datasets is presented in Appendix B. The final optimal models chosen are as follows:

- Dataset 1 : K = 16 layers, 128 Hidden features
- Dataset 2 : K = 20 layers, 128 Hidden features

4.1 Statistical performance

In this section we will discuss the results from the SWE-GNN models on both test cases. The numerical performance of the models predictions on the test simulations will be expressed through the aforementioned statistical metrics.

4.1.1 SWE-GNN on dataset 1

First, we will discuss the results of the SWE-GNN on dike ring 49. Figure 6 provides an overview of the dike ring with the breaching locations used to train, validate, and test.

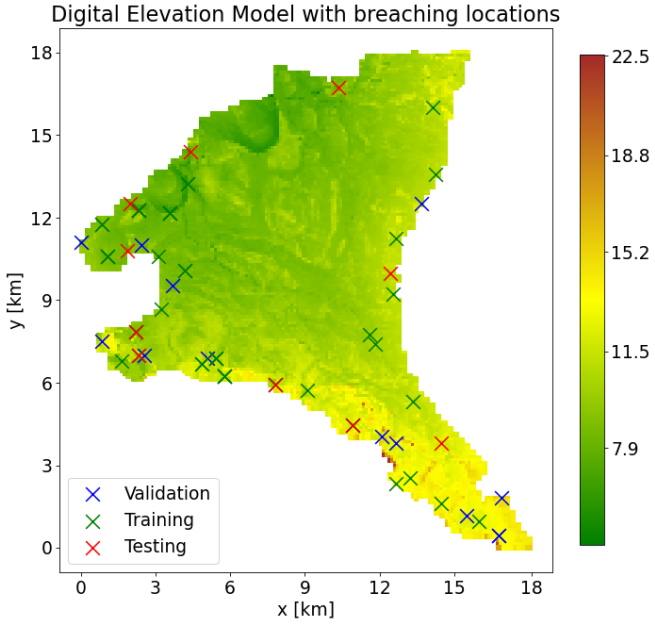


Figure 6. Digital Elevation Model (in meters) of dike ring 49 with the spatial distribution of the breach locations. Blue crosses mark validation simulations, green crosses mark training simulations and red crosses are testing simulations.

The average CSI score per time step for the test dataset is displayed in Figure 7. The metric remains consistently high for the first 48 hours of the simulation. This implies that the model is able to correctly predict the spatio-temporal evolution of the flood over time. The average CSI score is 0.85 ± 0.05 for the 0.05 m extent and 0.81 ± 0.05 for the 0.3 m extent. Generally, the CSI displays a small drop at the beginning and then increases again to a stable level. When observing the results this is explained by the initial propagation of the flood for certain breach locations and the topography of the dike ring. The domain roughly drains towards the northwest which means that the flood in some test cases travels a relative high distance between time steps at the beginning. This is further solidified by plotting the CSI for a model with less than 16 layers, where the drop off is larger at the start (Appendix D). The improvement of the score over time is attributed to a recurrent flood pattern for the domain. Water tends to concentrate at the northwestern corner of the dike ring which leads to little spatio-temporal errors later on in the simulation.

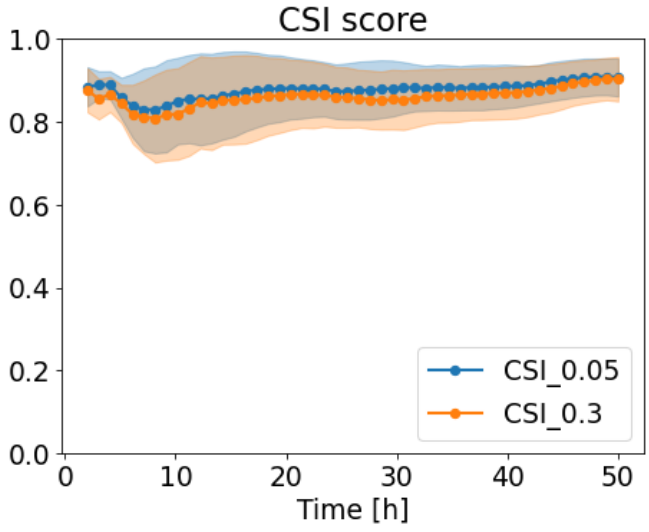


Figure 7. Temporal evolution of the CSI scores for the final model on dike ring 49. The dotted line represents the average score, the confidence band is set at one standard deviation from the mean result.

The statistical metrics employed over the quality of the predictions show that the model is able to accurately depict the flood depth and flow velocity in a certain cell. Over the 10 test simulations the model achieves a MAE of 0.027 m on water depth and 0.007 m^2/s on unit discharges over the entire dataset. In addition, the MAE is plotted per time step as presented in Figure 8. The MAE increases over time meaning that the general accuracy declines. However, the observed increase does not pose any significant concerns, as the increase happens sub-linearly and scores remain consistently high even after a duration of 48 hours. Part of this performance decrease is also due to the way the metric is evaluated; in the first time steps when the domain is mostly dry the error will be lower.

The SWE-GNN model predictions for water depth (a) and discharge (b) on one of the test samples is presented in Figure 9.

As can be observed from the figure, the SWE-GNN is able to predict how the water depth and discharge evolve accurately over time on dike ring 49. Upon closer inspection of the results, the errors are mostly related to relatively small

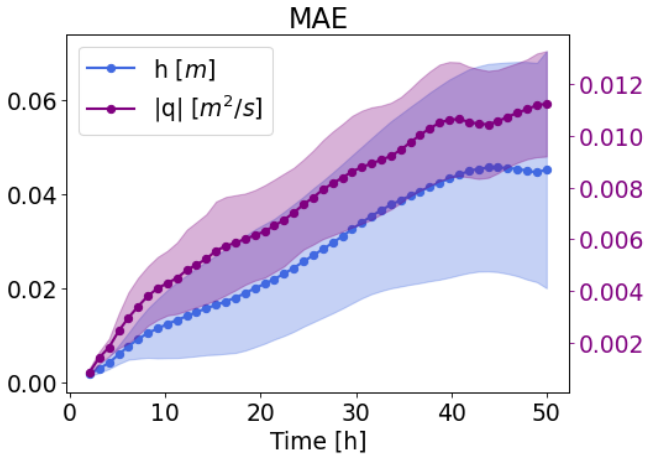


Figure 8. Evolution of the MAE over time for the test dataset samples on dike ring 49. The dotted line represents the average score, the confidence band is set at one standard deviation from the mean result.

over- or under-estimations due to lagging and incorrect flow routes. The numerical test simulations are inspected for supercritical cells attributed to higher Froude numbers. Supercritical flows can result in complex flow patterns and non-linear flow dynamics, this can introduce numerical instability (Neal et al., 2012). Based on our test cases, we observed that supercritical flows occur only during the initial time steps at the breach location. Importantly, these supercritical flows did not lead to any incorrect predictions by the SWE-GNN model. Subsequently, the rest of the domain predominantly displayed sub-critical flow conditions. This is likely due to the gentle terrain slopes and relative lower flow velocities attributed.

For all test cases, the difference is calculated and plotted per cell by subtracting the Delft3D simulation from the SWE-GNN prediction. This results in a positive value where the SWE-GNN over predicts water depth or discharge and a negative value where it under predicts, as can be observed in Figure 9. We conducted a residual analysis on these differences to assess if there were significant model errors. The result is displayed in Appendix C, where the totality of the domain is considered for the calculation. This means that both cells that are only flooded by one of the models as well as cells that are flooded by both, but by a different water depth, contribute to the bias. The violin plots show that there is a tendency to slightly over predict in the test cases after 48 hours.

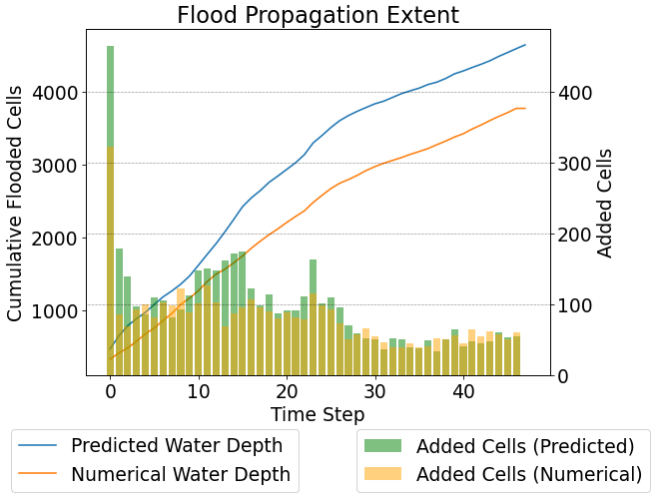
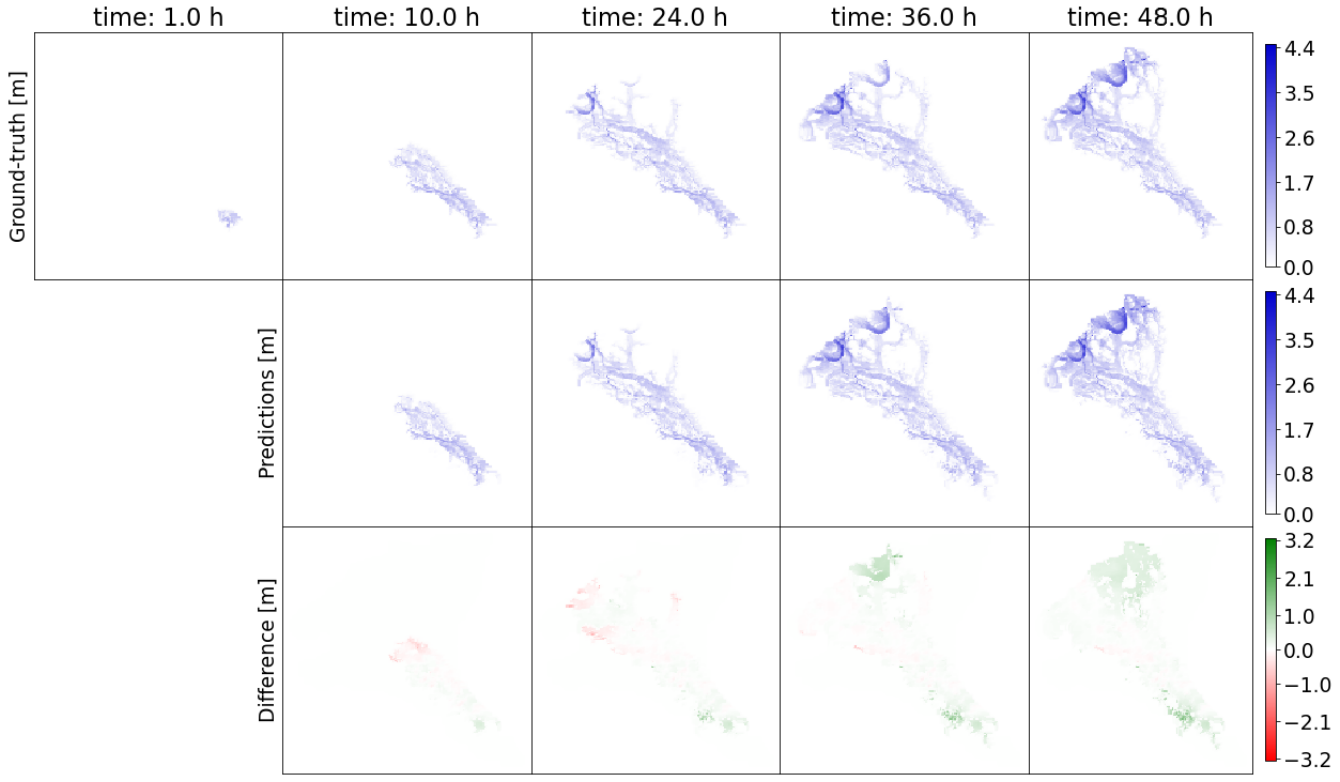


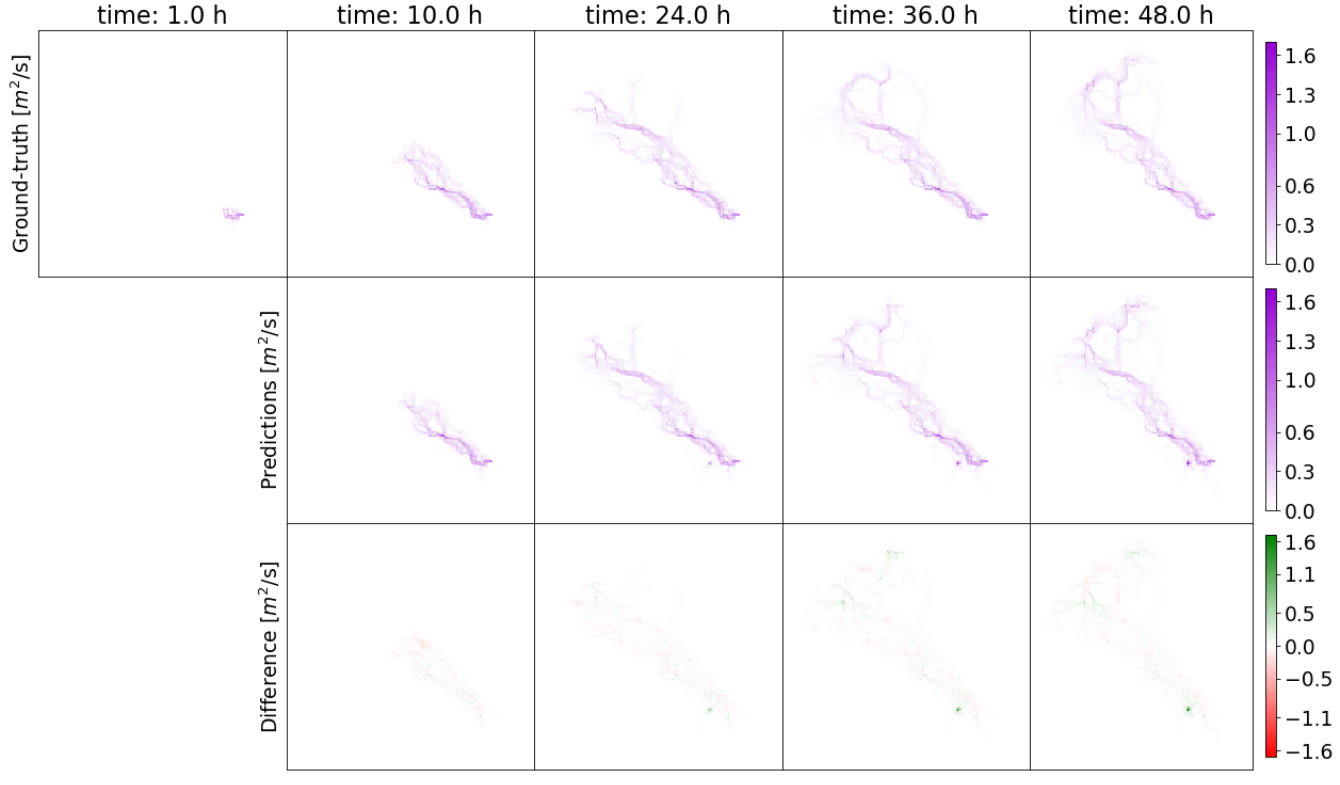
Figure 10. Flood propagation presented as cells flooded over time.

Additionally, we analyzed the flood arrival times for the different test cases. The SWE-GNN predictions slightly lag behind towards the outer edges of the flood propagation front of the Delft3D results. This was already observable for test case 2 in Figure 9, where the largest water depth differences for the first few time steps are located the furthest away from the breaching location. Figure 10 visualizes the number of cells flooded over time both for the SWE-GNN as well as Delft3D for test case 3. The line indicates the cumulative amount of flooded cells and the bars represent newly flooded cells for both models. This might be caused by a lack of preferential path that the flow follows, leading to an underestimation in the direction of preferential flows and an overestimation in the lateral flows.

From analyzing the SWE-GNN results it becomes clear that both the extent as well as the water depth are modestly overestimated, this is visualized in Figure C3 of Appendix C, the difference plot of test case 2 is divided in to two figures: the residuals corresponding to cells where both models simulate water, and the residuals associated with cells where only one of the two predicts water. This overprediction is likely due to the nature of the SWE-GNN; where water is constantly being added to the domain in the training data and the water depths in cells is mostly increased during the training phase. This phenomenon could be countered by adding conservation equations to the loss function, this is a common practice in physics embedded neural networks known as soft constraints. Hard constraints could in our context translate to a strict limitation on the maximum water level allowed in any given node or cell within the graph. This constraint would directly enforce a physical limit on the water level, ensuring that it never exceeds a predefined threshold, regardless of the model's predictions or the optimization process (Márquez-Neila et al., 2017).



(a) Water depths over time



(b) Discharges over time

Figure 9. Spatio-temporal performance of the model on test simulation 2 for dike ring 49. The first 2 days after the dike breach are visualized.

4.1.2 SWE-GNN on dataset 2

Breaching locations for the training, validation and testing dataset on dike ring 43 are presented in Figure 11. The final model was executed on the five test simulations marked as critical flood locations by the LIWO. As observed, the complete training and testing was conducted on the eastern half of the dike ring, this was explained in more detail in chapter 3. The exact area extent was selected because the maximum flood extent of the five test simulations did not reach beyond this domain.

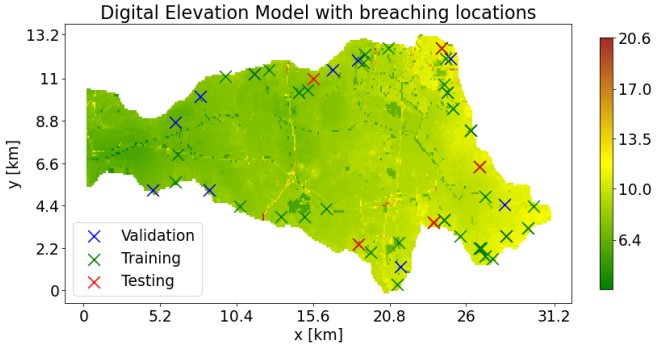


Figure 11. Digital Elevation Model (in meters) of dike ring 43 with the spatial distribution of the breach locations. Blue crosses mark validation simulations, green crosses mark training simulations and red crosses are testing simulations.

The final model trained on dataset 2 shows impressive results for the test simulations. Figure 12 displays the CSI score for the two flooding thresholds over time. The CSI score remains relatively stable at 0.75 +- 0.1 for the 0.05 m extent and 0.7 +-0.1 for the 0.3 m extent. These scores, however promising, are lower than the CSI scores for dataset 1. The standard deviation on the scores is also larger than for the model on dike ring 49. This implies that the performance for the different test simulations varies more than on the first test case. This variation in the score could be explained by the fact that dike ring 43 is more flat than dike ring 49, where all the test simulations drained to the northwest and converged in a similar final flow pattern after 48 hours. The test simulations on dike ring 43 do not follow a similar pattern which can result in vastly different flood maps between the training and testing data. Since the errors for the SWE-GNN on dike ring 49 generally come from incorrect flow routing, this will be more prevalent for the test dataset on dike ring 43. Where the flow routes are more unique between the different breach locations, it is possible that the SWE-GNN has to simulate a certain flow route for the first time during the testing. This model is therefore more prone to minor inaccuracies than the model from dataset 1.

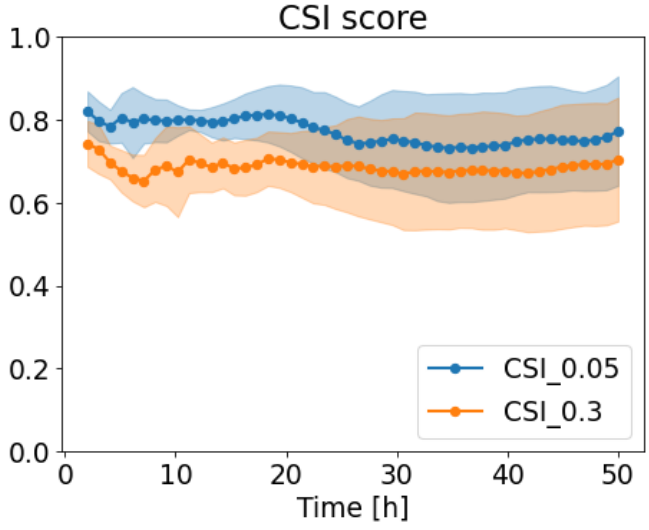


Figure 12. Temporal evolution of the CSI scores for the final model on dike ring 43. The dotted line represents the average score, the confidence band is set at one standard deviation from the mean result.

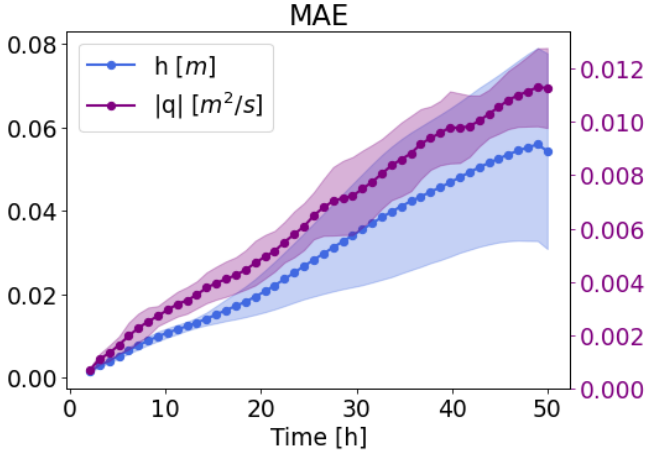
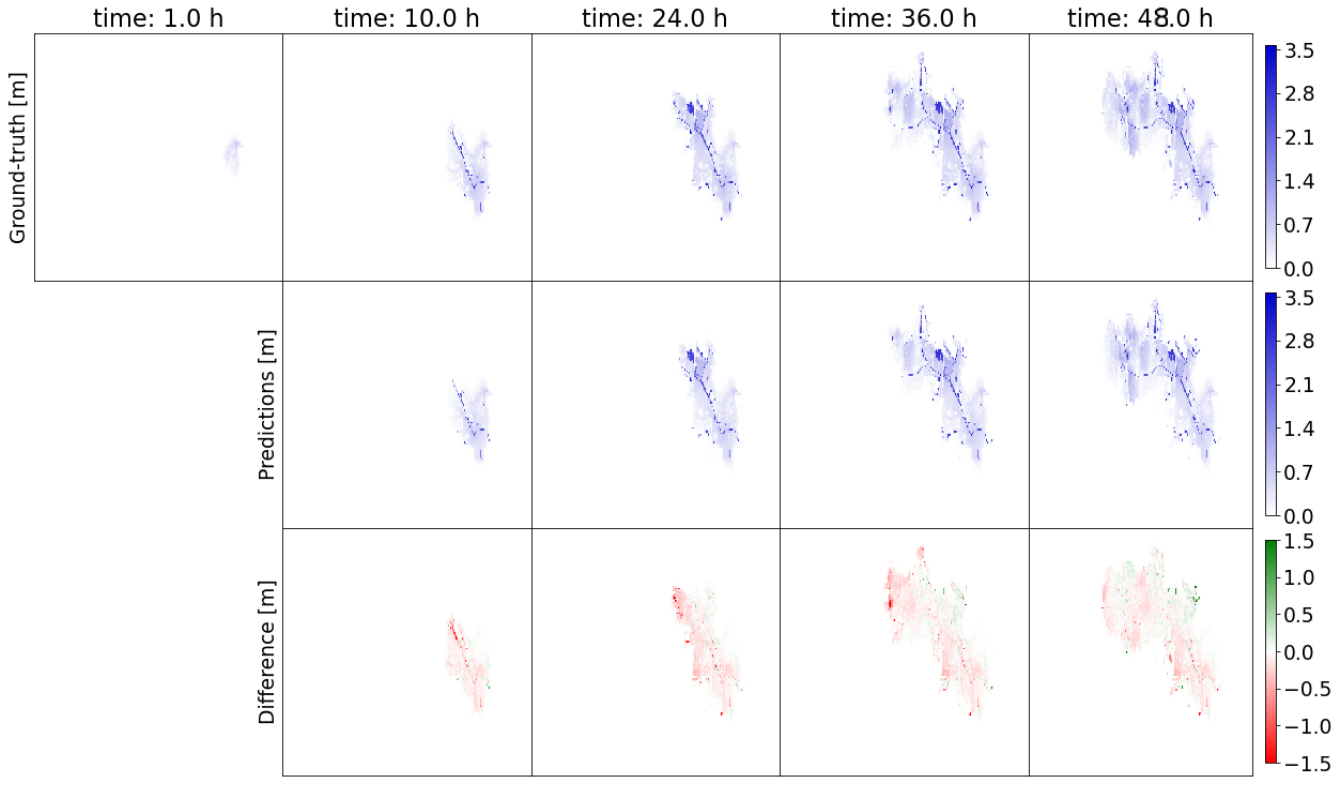


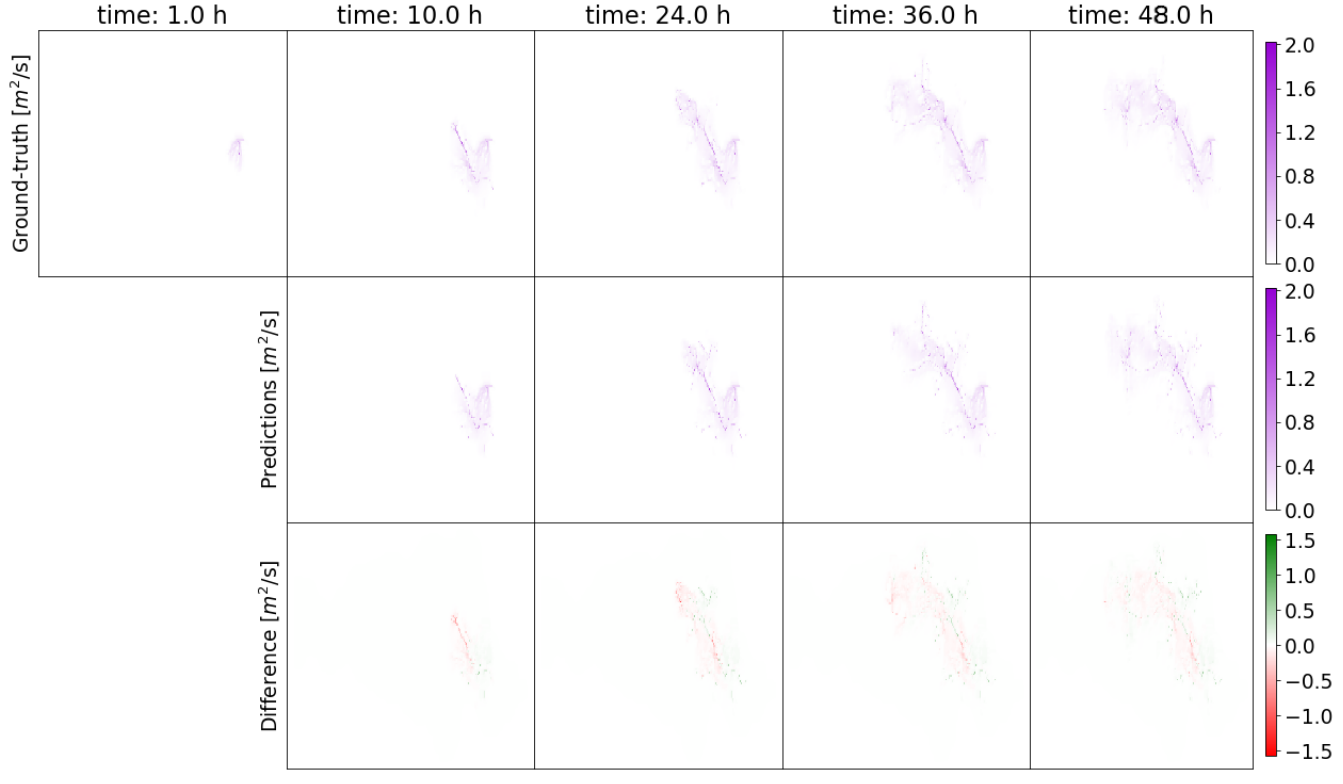
Figure 13. Evolution of the MAE over time for the test dataset samples on dike ring 43. The dotted line represents the average score, the confidence band is set at one standard deviation from the mean result.

The MAE over time for the test simulations is visualized in Figure 13. Over the 5 test simulation the model achieves a MAE of 0.029 m on water depth and 0.006 m^2/s on unit discharges over the entire dataset. The residual analysis and the statistical results of all test simulations are presented in Appendix B.

Figure 14 illustrates the spatio-temporal performance of the SWE-GNN for test case 0. The water depth and discharges match the ground truth from Delft3D successfully



(a) Water depths over time



(b) Discharges over time

Figure 14. Spatio-temporal performance of the model on test simulation 4 for dike ring 43. The first 2 days after the dike breach are visualized.

over time. As can be observed from the differences plots, the largest underestimations per timestep are located at the outer edges of the flooding front, implying that the SWE-GNN prediction is slightly lagging behind. This underestimation disappears in the following timesteps where the SWE-GNNs predictions catch on to the numerical flood extent. This could be the result of a combination of insufficient GNN layers to properly match the flood propagation per time step by message passing and lags originating from ponding phenomena. The number of GNN layers was obtained through the hyperparameter search. The accuracy could benefit from more GNN layers to convey information between more distant nodes, possibly improving the lagging inaccuracies for the first few timesteps. However, the increased number of GNN layers does lead to a more complex model which ultimately did not aid the overall validation CSI as found through the hyperparameter evaluation.

The term ponding phenomena (Carrivick, 2006) is used to describe when a certain part of the domain gets filled with water and forms a lake. This is for example visible in the water depth extent on Figure 14 from 24 hours onwards. Generally, deep learning models can struggle with the ponding phenomena since the discharge contribution to the training loss gets limited. This can lead to delays in flood propagation before the model continues the correct flow paths. The error is stopped from propagating over time due to the multi-step ahead loss function, which is applied during training.

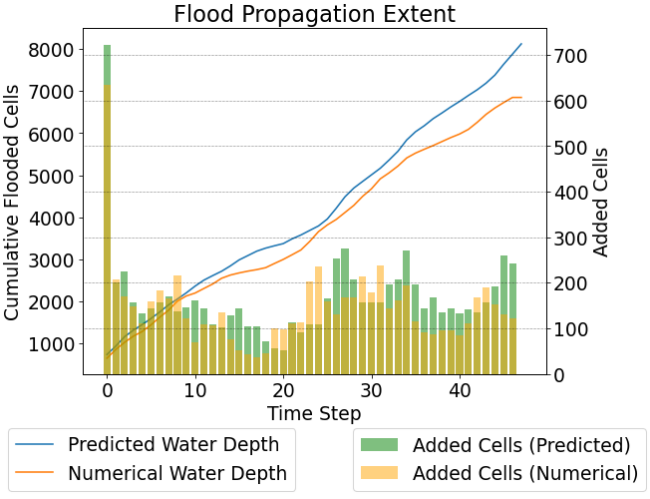


Figure 15. Flood propagation presented as cells flooded over time.

Figure 15 visualizes the number of cells flooded over time both for the SWE-GNN as well as Delft3D for test case 0. Even though the amount of cells flooded by the SWE-GNN align more closely to the numerical model than for dataset 1, a slight over prediction is still observable.

Despite the higher variation in flow routing and the addition of 1-D structures and ponding phenomena to the domain, the SWE-GNN is able to correctly predict the flood propagation from the unseen breaching locations on dataset 2. Across the different test simulations, the model is able to simulate the flooding extent as well as the water depths accurately.

4.2 Practical applicability performance

The resulting flood maps from the SWE-GNN on the test simulations are analyzed through the workflow (Figure 4) together with the Delft3D flood maps and compared on a few categories.

4.2.1 SWE-GNN on dataset 1

The direct damages on dike ring 49 are presented in Table 1. The average error is 4% over the entirety of simulations. Simulation 2 shows to be an outlier with an overestimation of the damages by 100%. Upon closer inspection of the prediction it shows a slight overestimation of the flood extent at the first few time steps. Coincidentally these cells where water was wrongly predicted were in the city of Doetinchem, located inside dike ring 49. This explains the overestimation in the cost of direct damages.

Simulation	SWE-GNN	Delft3D	Difference %
0	410	460	-10.8
1	220	250	-12
2	540	270	100
3	220	240	-8.3
4	310	290	6.9
5	260	230	13
6	220	240	-8.3
7	390	440	-11.4
8	230	240	-4.2
9	220	240	-8.3

Table 1. Direct damage calculations (in Millions of EUR) per simulation for the SWE-GNN and Delft3D results on dike ring 49. These damages are obtained from the maximum depth maps in combination with the max water discharges. Differences are expressed in percentages.

As per fatalities, the results of the SWE-GNN and Delft3D runs align closely, with the exception of test simulation 2 which is discussed earlier. For the affected people, the SWE-GNN overpredicts the number with some significance. This is related to the result of the residual analysis, which shows that the model on average after 48 hours predicts a larger flooding extent than the Delft3D runs. The deadly casualties, as well as the number of affected people for the SWE-GNN and Delft3D models, are visualized in Figure

16.

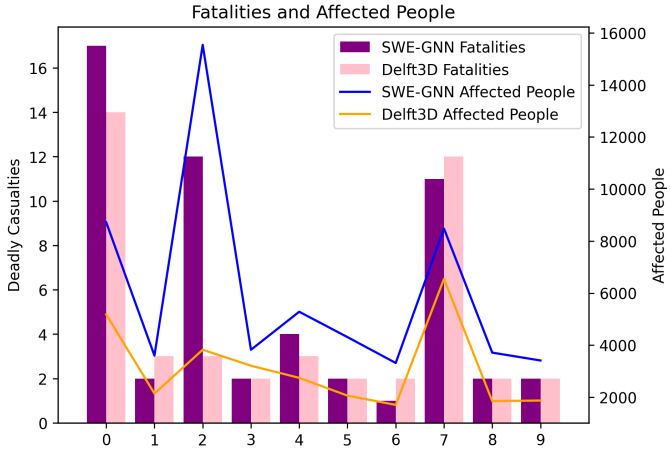


Figure 16. Fatalities and affected people per simulation for the SWE-GNN and Delft3D results. The fatalities are plotted by the bars and the affected people are visualized by the graph.

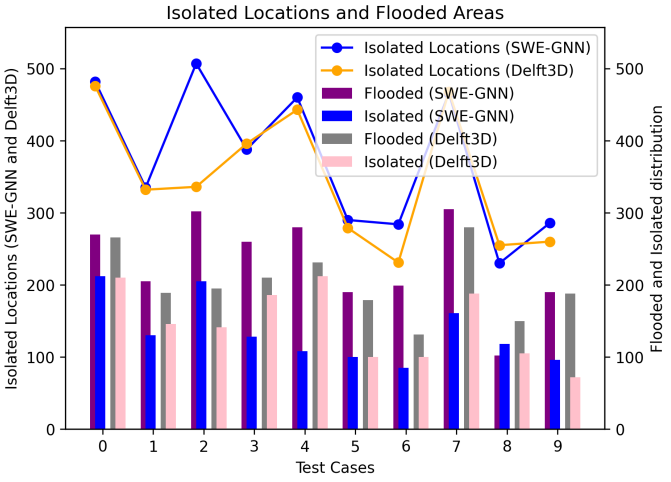


Figure 17. Number of isolated locations for dike ring 49, per test case for the SWE-GNN and Delft3D results. The line indicates the total number of isolated locations and the bar plots display the distribution of the cause of isolation.

For the indirect damage assessment, the mobility disruption to the transport network is quantified. We consider isolated locations as per Ra2Ce analysis. For the test dataset on dike ring 49, the SWE-GNN delivered good results. The flood maps proved to be of such quality that the effects on mobility on the domain are roughly identical. The number of isolated locations from the SWE-GNN matches up to an average of 90% with the Delft3D results. Isolated locations are grouped into two categories; isolation due to flooding at the location or isolation due to flooding on network parts further away, critical for mobility of a certain location. Here, some differences are observed which results from the flood extents of both models not being exactly identical. Despite that, the overall number of isolated locations align because of the way the mobility disruption is determined. If the SWE-GNN slightly under predicts flooding on a road or predicts the flooding in the cell next to where Delft3D simulates a flood, the road is still inaccessible. The results of this analysis are displayed in Figure 17. An example plot of a Ra2Ce analysis for test case 0 is visualized in E. Here, disruptions to the transport network are presented for both the SWE-GNN prediction and the D-Hydro flood scenario. The results indicate that the SWE-GNN performs well on the test samples for dataset 1. The direct damages are comparable and overall mobility disruption to the network align for both models. This demonstrates that the SWE-GNN offers reliable results in terms of practical applicability.

4.2.2 SWE-GNN on dataset 2

The direct damages on dike ring 43 are presented in Table 2. The average error is -1.25% over the entirety of simulations which means that the SWE-GNN minimally under predicts the damage costs. As per fatalities the results of the SWE-GNN and Delft3D runs align closely, with the exception of test simulation 2 which is discussed earlier. For the affected people, the SWE-GNN overpredicts the number with some significance. This is related to the result of the residual analysis, which shows that the model on average after 48 hours predicts a larger flooding extent than the Delft3D runs. The fatalities, as well as the number of affected people for the SWE-GNN and Delft3D models, are visualized in Figure 18. For the indirect damages, the Ra2Ce analysis is executed on the results of both the models for dike ring 43. These results are visualized in Figure 19. For the test dataset on dike ring 43, the SWE-GNN delivered good results. The disruption of the floods to the mobility of the domain are roughly identical for the test cases. The total isolated locations match up to an average of 93%. The main difference occurs in the distribution of isolation cause. Similarly to the results on dike

Simulation	SWE-GNN	Delft3D	Difference %
0	1400	1500	-6.7
1	2200	2200	0
2	1400	1500	-6.7
3	1500	1500	0
4	1400	1300	7.7

Table 2. Direct damage calculations (in Millions of EUR) per simulation for the SWE-GNN and Delft3D results on dike ring 43. These damages are obtained from the maximum depth maps in combination with the max water discharges. Differences are expressed in percentages.

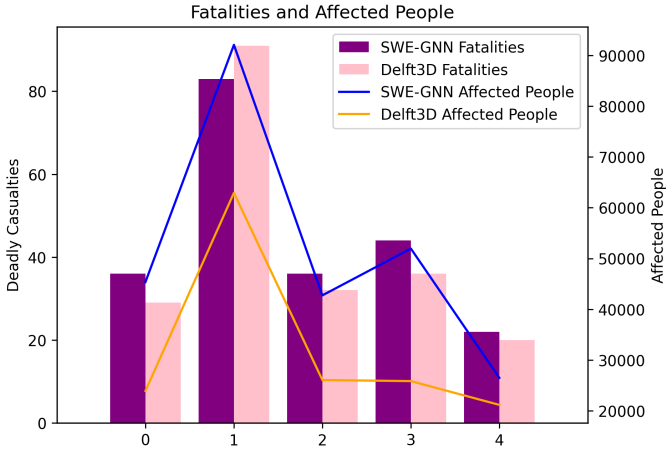


Figure 18. Fatalities and affected people per simulation for the SWE-GNN and Delft3D results. The fatalities are plotted by the bars and the affected people are visualized by the graph.

ring 49 the SWE-GNN predicts more flooded locations. This is in line with the larger flood extent observed in the results. Overall the mobility is affected similarly for the SWE-GNN and Delft3D simulations. Minor inaccuracies are compensated due to the nature of the analysis as mentioned before.

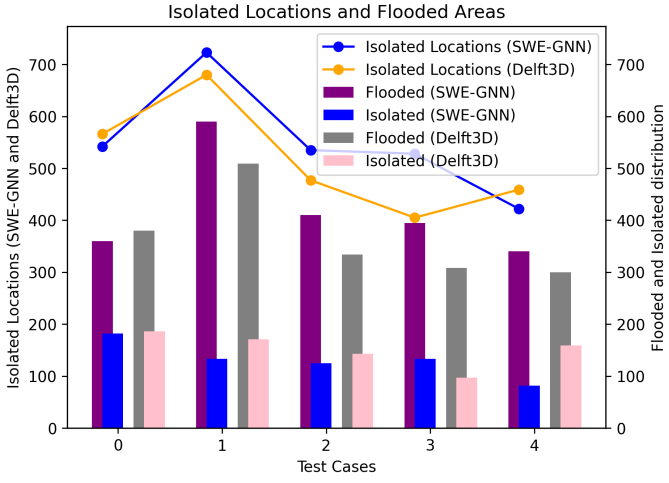


Figure 19. Number of isolated locations for dike ring 43, per test case for the SWE-GNN and Delft3D results. The line indicates the total number of isolated locations and the bar plots display the distribution of the cause of isolation.

4.3 Computational speed up

For fair comparison between the two models, testing for the SWE-GNN was done on CPU. The optimal model configuration for dataset 1, as per the hyperparameter search, displays an overall computational speed up of 8.2 times. For dataset 2 the SWE-GNN is 6.6 times faster than Delft3D. As mentioned earlier, the SWE-GNN for dataset 2 has a higher number of GNN layers, making the model slightly more complex and resulting in a reduction in computational efficiency. These speed ups are in line with the speed ups presented in the paper by Bentivoglio et al. (2023).

Model testing was also conducted on GPU, these results are not taken into consideration for the model analysis since the numerical model was only executed on CPU. On GPU the SWE-GNN demonstrated a speed up of two orders of magnitude when compared to the numerical model on CPU. Additionally, the observed speed up for dataset 2 is larger than for dataset 1, emphasizing the SWE-GNNs ability to efficiently scale up and better exploit the GPU hardware for larger graphs. Overall, the GPU speed up is in line with the computational efficiency presented in Bentivoglio et al. (2023). Utilizing the full potential of GPUs for parallel calculations.

4.4 Sensitivity Analysis

To test the sensitivity of the model to overfitting a sensitivity analysis was carried out. In this analysis, the number of training data samples was systematically decreased and the CSI on the validation samples of the training data was computed. The hyperparameters of the trained models are the same as for the rest of the project. For every number of training samples, the speed of the training and validation split was changed in order to randomize the specific simulations in the dataset. The average results are visualized in Figure 20. The SWE-GNN performance remains relatively stable with the decrease in training samples. Indicating that the model is able to capture the underlying patterns and relationships in the data effectively, even with a reduced amount of training data. This highlights the robustness of the SWE-GNN and means that the results are not overly dependent on certain samples in the dataset.

4.5 Generalizability to unseen domains

In this research, some first exploratory work has been conducted on the generalizability to unseen domains. In the paper Bentivoglio et al. (2023), the SWE-GNN showcases that it can generalize well to previously unseen topographies. For this research, the SWE-GNN models trained on dataset 1 and dataset 2 have been applied on the western, unseen half, of dike ring 43. These results are presented in Appendix F.

The results demonstrate that the SWE-GNN is only modestly able to generalize to topographies beyond the training data in real-world scenarios. At this point, the performance poses a limitation for practical application on unseen domains.

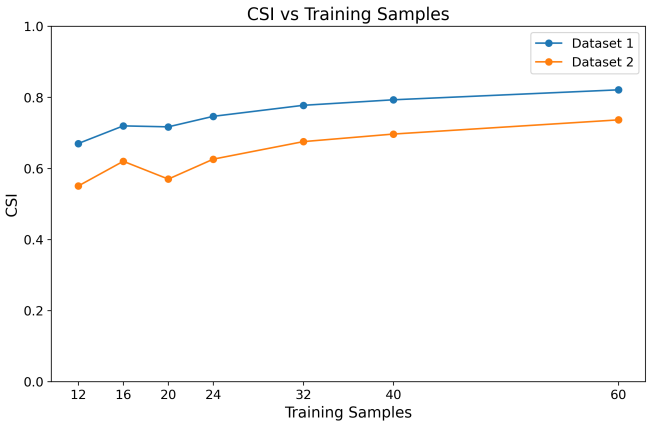


Figure 20. The validation CSI plotted against the number of training samples. The model performance gradually declines with fewer training samples, yet the overall performance remains stable.

5 Limitations and Future Research

Future work should focus on the generalizability limitations on unseen domains encountered in this research. The model could possibly benefit from more supervision on the training dataset selection, maintaining a certain threshold of complexity in the samples and ensuring that structures like, for example, ponds or 1-D structures are present if the model is required to simulate them for the tests. This became apparent in the tests presented in Appendix F, where the model trained on dataset 2 outperformed the model trained on dataset 1 for the unseen domain. This further solidifies that certain patterns and relations in the training dataset are pivotal for proper flood prediction on the test simulations (Kimura et al., 2019; Zhao et al., 2021). In addition, the SWE-GNN could benefit from transfer learning. Having a pre-trained model on a certain domain the performance on new domains can be increased by transferring knowledge from this pre-trained model to a new transfer learning model. Subsequently the remaining weights of the transfer learning model can then be trained on new samples and tasks (Olivas et al., 2009; Cache et al., 2024).

On the other hand, it might prove to be more straightforward to set up individual SWE-GNN models for each dike ring. The results of this research demonstrated the ability to generalize well to unseen breaching locations within the trained domain. The sensitivity analysis showed that the models performance remains consistent even when the number of training samples is reduced, indicating that model set up procedures could be simplified.

The Froude number analysis on the test cases displayed predominantly sub-critical flow conditions. Future research could explore the behavior of the SWE-GNN model in cells with higher Froude numbers. Since the SWE-GNN can be used for any flood applications where the SWE holds, it could be interesting to examine the performance for areas with supercritical flow regimes.

In this project, the analysis was carried out under constant breach discharge. Future research could explore the application of variable discharge as boundary condition on real-world dike ring scenarios. This could be achieved by employing ghost cells and assigning them with boundary conditions (LeVeque, 2002). This new addition does not only remove the dependency on a numerical model for the first time step, but also marks a valuable step towards better representing real world dike breach scenarios. The computational speed up could make the tool of particular value for probabilistic calculations. During periods with high river levels, numerous uncertainties are present. The exact location of a dike failure and the corresponding hydrograph are unknown. At this stage it would be beneficial to run a considerable amount of simulations with different hydrographs on the entire section of the dike at risk (Vorogushyn et al., 2011; Kamrath et al., 2006). Therefore, future work could focus on implementing boundary conditions to the SWE-GNN, and assessing how a range of variable discharges in the training dataset performs for test cases with different hydrographs.

Furthermore, the model could be improved by exploring multi-scale methods to reduce computation time by passing information more efficiently (Lino et al., 2022). Another interesting direction could be incorporating some sort of adaptive modelling, where the SWE-GNN can adapt to real-time observations during a calamity as input. With the overall inaccuracy during simulation increasing over time, this could reset predictions with a corrected flood extent. Additionally, implementing real-time observations would correct flow paths and minimize any lag between the SWE-GNN predictions and the flood front.

Generally, the SWE-GNN proves to modestly over predict the flooding extent and the water depths. Future work could focus on adding conservation equations to the training loss function. This is a common practice in physics-based neural networks, where conservation laws such as the mass balance are employed to better enforce a more accurate representation of the physical processes.

6 Conclusion

In this research, we modified and applied the proposed SWE-GNN from Bentivoglio et al. (2023) on two real world case studies in The Netherlands. We assess the suitability of the physics-based GNN for generating reliable flood predictions and evaluate its efficacy as a practical tool in emergency situations. Besides measuring the performance of the model through numerical metrics, we also test the real world applicability of the results.

The SWE-GNN trained on dataset 1 and 2 showed very promising results on dike ring 49 and 43 respectively. Further highlighting the models capabilities to generalize to unseen breach locations. The model is able to correctly predict the spatio-temporal evolution of the flood and achieves good scores for the statistical metrics employed. For dike ring 49 it displays an average MAE of 0.027 m on water depth and 0.007 m^2/s on units discharge over the entire dataset, for dike ring 43 the average MAE is 0.029 m on water depth and 0.006 m^2/s on units discharge over the entire dataset. The post-processing on the flood predictions showed that the model is able to provide a viable tool in emergency scenarios. For direct damages the results of the SWE-GNN were satisfactory, aligning closely with the post-processed Delft3D results. On the indirect damage analysis the results of the SWE-GNN were excellent, further solidifying the viability of the model in emergency situations.

The SWE-GNN showcased its ability to deliver precise flood simulations while reducing computational times on CPU by factors of 8.2 and 6.6 for the respective test cases. Besides during calamities, these speed ups could be valuable for situations where multiple calculations are required in quick succession, such as for probabilistic calculations with uncertainties in the inputs.

The sensitivity analysis showed that by decreasing the training size by roughly 75% the performance remained consistently stable. Implying that the model has the expressiveness to capture relationships and patterns for the task at hand, despite the smaller training dataset. This means that any SWE-GNN model for a dike ring could be set up relatively easily to provide a valuable tool during a calamity for the early emergency response.

Concluding, in this research we proved that the SWE-GNN is a promising innovation on rapid spatio-temporal flood modelling. The SWE-GNN can successfully generalize to unseen breaching locations and provide fast and qualitative strong results which can be used for real world applications during a calamity.

References

Alcrudo, F. and Garcia-Navarro, P.: A high-resolution Godunov-type scheme in finite volumes for the 2D shallow-water equations, International Journal for Numerical Methods in Fluids, 16, 489–505, 1993.

Almeida, G. A. D., Bates, P., Freer, J. E., and Souvignet, M.: Improving the stability of a simple formulation of the shallow water equations for 2-D flood modeling, Water Resources Research, 48, https://doi.org/10.1029/2011WR011570, 2012.

Anees, M. T., Abdullah, K., Nawawi, M. N., Rahman, N. N. N. A., Piah, A. R. M., Zakaria, N. A., Syakir, M. I., and Omar, A. K. M.: Numerical modeling techniques for flood analysis, https://doi.org/10.1016/j.jafrearsci.2016.10.001, 2016.

Apel, H., Merz, B., and Thieken, A.: Influence of dike breaches on flood frequency estimation, Computers & Geosciences, 35, 907–923, 2009.

Bates, P. D.: Flood inundation prediction, Annual Review of Fluid Mechanics, 54, 287–315, 2022.

Bates, P. D., Horritt, M. S., and Fewtrell, T. J.: A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling, Journal of Hydrology, 387, https://doi.org/10.1016/j.jhydrol.2010.03.027, 2010.

Bentivoglio, R., Isufi, E., Jonkman, S. N., and Taormina, R.: Deep learning methods for flood mapping: a review of existing applications and future research directions, Hydrology and Earth System Sciences, 26, 4345–4378, https://doi.org/10.5194/hess-26-4345-2022, 2022.

Bentivoglio, R., Isufi, E., Jonkman, S. N., and Taormina, R.: Rapid spatio-temporal flood modelling via hydraulics-based graph neural networks, Hydrology and Earth System Sciences, 27, 4227–4246, https://doi.org/10.5194/hess-27-4227-2023, 2023.

Biewald, L. et al.: Experiment tracking with weights and biases, Software available from wandb. com, 2, 2020.

Bles, T., van Marle, M., Boonstra, H., van Muiswinkel, K., and de Bel, M.: Multi-hazard resilience assessment and adaptation planning for the Dutch highway network, Transportation Research Procedia, 72, 3801–3808, 2023.

Cache, T., Gomez, M. S., Beucler, T., Blagojevic, J., Leitao, J. P., and Peleg, N.: Enhancing generalizability of data-driven urban flood models by incorporating contextual information, Hydrology and Earth System Sciences Discussions, 2024, 1–23, 2024.

Carrivick, J. L.: Application of 2D hydrodynamic modelling to high-magnitude outburst floods: An example from Kverkfjöll, Iceland, Journal of Hydrology, 321, 187–199, 2006.

Chang, F. J., Chen, P. A., Lu, Y. R., Huang, E., and Chang, K. Y.: Real-time multi-step-ahead water level forecasting by recurrent neural networks for urban flood control, Journal of Hydrology, 517, https://doi.org/10.1016/j.jhydrol.2014.06.013, 2014.

Chen, P. A., Chang, L. C., and Chang, F. J.: Reinforced recurrent neural networks for multi-step-ahead flood forecasts, Journal of Hydrology, 497, https://doi.org/10.1016/j.jhydrol.2013.05.038, 2013.

Costabile, P., Costanzo, C., and Macchione, F.: Performances and limitations of the diffusive approximation of the 2-d shallow water equations for flood simulation in urban and rural areas, Applied Numerical Mathematics, 116, https://doi.org/10.1016/j.apnum.2016.07.003, 2017.

Delft High Performance Computing Centre: DelftBlue Supercomputer (Phase 1), https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase1, 2022.

Deltares: 2023.

Deltares: Delft3D-FLOW User Manual, https://oss.deltares.nl/web/delft3d/manuals, 2024.

Fey, M. and Lenssen, J. E.: Fast graph representation learning with PyTorch Geometric, arXiv preprint arXiv:1903.02428, 2019.

Guo, Z., Leitão, J. P., Simões, N. E., and Moosavi, V.: Data-driven flood emulation: Speeding up urban flood predictions by deep convolutional neural networks, Journal of Flood Risk Management, 14, https://doi.org/10.1111/jfr3.12684, 2021.

Guo, Z., Moosavi, V., and Leitão, J. P.: Data-driven rapid flood prediction mapping with catchment generalizability, Journal of Hydrology, 609, 127 726, https://doi.org/10.1016/J.JHYDROL.2022.127726, 2022.

Henonin, J., Russo, B., Mark, O., and Gourbesville, P.: Real-time urban flood forecasting and modelling - A state of the art, Journal of Hydroinformatics, 15, https://doi.org/10.2166/hydro.2013.132, 2013.

Horie, M. and Mitsume, N.: Physics-Embedded Neural Networks: Graph Neural PDE Solvers with Mixed Boundary Conditions, https://github.com/yellowshippo/penn-neurips2022., 2022.

Hu, R. L., Pierce, D., Shafi, Y., Boral, A., Anisimov, V., Nevo, S., and Chen, Y. F.: Accelerating physics simulations with tensor processing units: An inundation modeling example, International Journal of High Performance Computing Applications, 36, https://doi.org/10.1177/10943420221102873, 2022.

Inden, M.: Short introduction to python 3.10, in: Python Challenges: 100 Proven Programming Tasks Designed to Prepare You for Anything, pp. 635–643, Springer, 2022.

Jonkman, S. and Vrijling, J.: Loss of life due to floods, Journal of Flood Risk Management, 1, https://doi.org/10.1111/j.1753-318x.2008.00006.x, 2008.

Kamrath, P., Disse, M., Hammer, M., and Köngeter, J.: Assessment of discharge through a dike breach and simulation of flood wave propagation, Natural Hazards, 38, 63–78, 2006.

Kimura, N., Yoshinaga, I., Sekijima, K., Azechi, I., and Baba, D.: Convolutional neural network coupled with a transfer-learning approach for time-series flood predictions, Water, 12, 96, 2019.

Kingma, D. P. and Ba, J.: Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980, 2014.

Kok, M., Huizinga, H., Vrouwenvelder, A., and Barendregt, A.: Standard method 2004. Damage and casualties caused by flooding, Client: Highway and Hydraulic Engineering Department, 2004.

Koks, E. E., Bočkarjova, M., de Moel, H., and Aerts, J. C.: Integrated direct and indirect flood risk modeling: development and sensitivity analysis, Risk analysis, 35, 882–900, 2015.

LeCun, Y., Bengio, Y., and Hinton, G.: Deep learning, nature, 521, 436–444, 2015.

LeVeque, R. J.: Finite volume methods for hyperbolic problems, vol. 31, Cambridge university press, 2002.

Lino, M., Fotiadis, S., Bharath, A. A., and Cantwell, C. D.: Multi-scale rotation-equivariant graph neural networks for unsteady Eulerian fluid dynamics, Physics of Fluids, 34, 2022.

Löwe, R., Böhm, J., Jensen, D. G., Leandro, J., and Rasmussen, S. H.: U-FLOOD–Topographic deep learning for predicting urban pluvial flood water depth, Journal of Hydrology, 603, 126 898, 2021.

Márquez-Neila, P., Salzmann, M., and Fua, P.: Imposing hard constraints on deep networks: Promises and limitations, arXiv preprint arXiv:1706.02025, 2017.

Messner, F. and Meyer, V.: FLOOD DAMAGE, VULNERABILITY AND RISK PERCEPTION – CHALLENGES FOR FLOOD DAMAGE RESEARCH, https://doi.org/10.1007/978-1-4020-4598-1_13, 2007.

Nayak, P. C., Sudheer, K. P., Rangan, D. M., and Ramasastri, K. S.: Short-term flood forecasting with a neurofuzzy model, Water Resources Research, 41, https://doi.org/10.1029/2004WR003562, 2005.

Neal, J., Villanueva, I., Wright, N., Willis, T., Fewtrell, T., and Bates, P.: How much physical complexity is needed to model flood inundation?, Hydrological Processes, 26, 2264–2282, 2012.

Nicholls, R., Zanuttigh, B., Vanderlinden, J. P., Weisse, R., Silva, R., Hanson, S., Narayan, S., Hoggart, S., Thompson, R. C., de Vries, W., and Koundouri, P.: Developing a Holistic Approach to Assessing and Managing Coastal Flood Risk, Coastal Risk Management in a Changing Climate, pp. 9–53, https://doi.org/10.1016/B978-0-12-397310-8.00002-6, 2014.

Olivas, E. S., Guerrero, J. D. M., Martinez-Sober, M., Magdalena-Benedito, J. R., Serrano, L., et al.: Handbook of research on machine learning applications and trends: Algorithms, methods, and techniques: Algorithms, methods, and techniques, IGI global, 2009.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems, 32, 2019.

Penning-Rowsell, E., Johnson, C., Tunstall, S., Tapsell, S., Morris, J., Chatterton, J., and Green, C.: The benefits of flood and coastal risk management: a handbook of assessment techniques, ISBN 1904750516, 2005.

Petaccia, G., Leporati, F., and Torti, E.: OpenMP and CUDA simulations of Sella Zerbino Dam break on unstructured grids, Computational Geosciences, 20, https://doi.org/10.1007/s10596-016-9580-5, 2016.

Pregnolato, M., Ford, A., Wilkinson, S. M., and Dawson, R. J.: The impact of flooding on road transport: A depth-disruption function, Transportation Research Part D: Transport and Environment, 55, https://doi.org/10.1016/j.trd.2017.06.020, 2017.

Rijkswaterstaat: 2023.

Rijkswaterstaat, W. N. W.: 2022.

Rodrigue, J.-P.: 3.1 – Transportation and Economic Development, The Geography of Transport Systems, 2020.

Rose, A. and Liao, S.-Y.: Modeling regional economic resilience to disasters: A computable general equilibrium analysis of water service disruptions, Journal of regional science, 45, 75–112, 2005.

Sabbaqi, M. and Isufi, E.: Graph-time convolutional neural networks: Architecture and theoretical analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023.

Sanchez-Lengeling, B., Reif, E., Pearce, A., and Wiltschko, A. B.: A gentle introduction to graph neural networks, Distill, 6, e33, 2021.

Serre, D. and Heinzlef, C.: Assessing and mapping urban resilience to floods with respect to cascading effects through critical infrastructure networks, International Journal of Disaster Risk Reduction, 30, https://doi.org/10.1016/j.ijdrr.2018.02.018, 2018.

Slager, K. and Wagenaar, D.: Standaardmethode 2017: Schade en slachtoffers als gevolg van overstromingen, Deltares report, 2017.

Slager, K., Burzel, A., Bos, E., De Bruikn, K., Wagenaar, D., Winsemius, H., Bouwer, L., and Van der Doef, M.: User Manual Delft-FIAT version 1, 2016.

Taormina, R. and Galelli, S.: Deep-learning approach to the detection and localization of cyber-physical attacks on water distribu-

tion systems, Journal of Water Resources Planning and Management, 144, 04018 065, 2018.

Teng, J., Jakeman, A. J., Vaze, J., Croke, B. F., Dutta, D., and Kim, S.: Flood inundation modelling: A review of methods, recent advances and uncertainty analysis, https://doi.org/10.1016/j.envsoft.2017.01.006, 2017.

Thieken, A., Ackermann, V., Elmer, F., Kreibich, H., Kuhlmann, B., Kunert, U., Maiwald, H., Merz, B., Müller, M., Piroth, K., et al.: Methods for the evaluation of direct and indirect flood losses, in: RIMAX Contributions at the 4th International Symposium on Flood Defence (ISFD4), Deutsches GeoForschungsZentrum GFZ, 2009.

Vorogushyn, S., Apel, H., and Merz, B.: The impact of the uncertainty of dike breach development time on flood hazard, Physics and Chemistry of the Earth, Parts A/B/C, 36, 319–323, 2011.

Vreugdenhil, C. B.: Numerical Methods for Shallow-Water Flow, 1994.

Wagenaar, D. J., Dahm, R. J., Diermanse, F. L., Dias, W. P., Dissanayake, D. M., Vajja, H. P., Gehrels, J. C., and Bouwer, L. M.: Evaluating adaptation measures for reducing flood risk: A case study in the city of Colombo, Sri Lanka, International Journal of Disaster Risk Reduction, 37, https://doi.org/10.1016/j.ijdrr.2019.101162, 2019.

Wang, Y., Fang, Z., Hong, H., and Peng, L.: Flood susceptibility mapping using convolutional neural network frameworks, Journal of Hydrology, 582, https://doi.org/10.1016/j.jhydrol.2019.124482, 2020.

Ward, P. J., Winsemius, H. C., Kuzma, S., Bierkens, M. F. P., Bouwman, A., Moel, H. D., Loaiza, A. D., Eilander, D., Englhardt, J., Gilles, E., Gebremedhin, E., Iceland, C., Kooi, H., Ligtvoet, W., Muis, S., Scussolini, P., Sutanudjaja, E. H., Beek, R. V., Bemmel, B., Huijstee, J. V., Rijn, F. V., Wesenbeeck, B. V., Vatvani, D., Verlaan, M., Tiggeloven, T., and Luo, T.: Aqueduct floods methodology, World Resources Institute, 2020.

Yang, M., Isufi, E., and Leus, G.: Simplicial convolutional neural networks, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8847–8851, IEEE, 2022.

Yu, D., Yin, J., Wilby, R. L., Lane, S. N., Aerts, J. C., Lin, N., Liu, M., Yuan, H., Chen, J., Prudhomme, C., Guan, M., Baruch, A., Johnson, C. W., Tang, X., Yu, L., and Xu, S.: Disruption of emergency response to vulnerable populations during floods, Nature Sustainability, 3, https://doi.org/10.1038/s41893-020-0516-7, 2020.

Zhao, G., Pang, B., Xu, Z., Peng, D., and Zuo, D.: Urban flood susceptibility assessment based on convolutional neural networks, Journal of Hydrology, 590, https://doi.org/10.1016/j.jhydrol.2020.125235, 2020.

Zhao, G., Pang, B., Xu, Z., Cui, L., Wang, J., Zuo, D., and Peng, D.: Improving urban flood susceptibility mapping using transfer learning, Journal of Hydrology, 602, 126 777, 2021.

Zhou, Y., Wu, W., Nathan, R., and Wang, Q. J.: Deep Learning-Based Rapid Flood Inundation Modeling for Flat Floodplains With Complex Flow Paths, Water Resources Research, 58, https://doi.org/10.1029/2022WR033214, 2022.

Appendix A: SWE-GNN

In this section a short summary will be provided on the exact functioning of the SWE-GNN. For more in depth information on the SWE-GNN we refer to the original paper; Bentivoglio et al. (2023).

Encoder

The static node features ($\mathrm{X}_s \in \mathbb{R}^{N \times \mathrm{I}_{Ns}}$), the dynamic node features ($\mathrm{X}_d \equiv \mathrm{U}^{t-p:t} \in \mathbb{R}^{N \times O(p+1)}$) and the edge features ($\varepsilon \in \mathbb{R}^{E \times \mathrm{I}_\varepsilon}$) are processed in three separate encoders. I_{NS} refers to the number of static node features, O to the number of considered hydraulic variables, p the number of input previous time steps and finally I_ε the number of edge features as input. The encoded variables then can be expressed as

$$\mathrm{H}_s = \phi_s, \mathrm{H}_d = \phi_d, \mathrm{E}^{'} = \phi_\varepsilon(E) \tag{A1}$$

where ϕ represent MLPs that generate node matrices for static and dynamic node features. And for the edge features it processes and encodes them accordingly to maintain the relations. The MLPs consist of two layers, with a hidden dimension G and a parametric PReLU activation. These encoders are in place to expand the dimensionality of the inputs to facilitate higher expressivity of the features, where G represent the dimension of the node embeddings.

Processor

The processor consists of a GNN with L layers that takes the high-dimensional inputs provided by the encoder at time t which produces the spatio-temporal propagation of the flood evolution for for time = t + 1. This evolution is based on the Shallow Water Equation, where the dynamic features such as mass and momentum fluxes progress in space as a function of the input terms i.e. the static and dynamic features. The physics basis of the SWE-GNN constrains water to only propagate from cells with water, and the velocity of this propagation is dependent on the hydraulic gradients between adjacent cells. The updated features are as follows:

$$\mathrm{s}_{ij}^{(l+1)} = \psi(\mathrm{h}_{si}, \mathrm{h}_{sj}, \mathrm{h}_{di}^{(l)}, \mathrm{h}_{dj}^{(l)}, \varepsilon_{ij}^{'}) \odot (\mathrm{h}_{dj}^{(l)} - \mathrm{h}_{di}^{(l)}) \tag{A2}$$

$$\mathrm{h}_{di}^{(l+1)} = \mathrm{h}_{di}^{(l)} + \sum_{j \in \mathrm{N}_i} \mathrm{s}_{ij}^{(l+1)} \mathrm{W}^{(l+1)} \tag{A3}$$

In these equations ψ consists of an MLP with two layers, hidden dimension of 2G with a PReLU activation function. $\odot$ is the Hadamard product and w^l are the parameter matrices. $(\mathrm{h}_{dj}^{(l)} - \mathrm{h}_{di}^{(l)})$ presents the gradient of the hydraulic variables. The static and dynamic inputs are then incorporated by ψ to compute an estimate of the source terms on the nodes. This output is normalized along the embedding dimension to increase stability.

s_{ij} presents the fluxes between neighboring cells. The contribution of each layer is multiplied by $\mathrm{W}^{(l)}$.

The GNNs produce the output for the predicted hydraulic variables at time t+1 for the defined time step. Multiple GNN layers are stacked to increase the propagation space, enabling the model to capture dependencies from further away and correctly predict flood propagation relations per time step. The full processor for the L GNN layers can be described as

$$\begin{aligned}
\mathrm{h}_{di}^{(0)} &= \mathrm{h}_{di}\mathrm{W}^{(0)} \\
\mathrm{s}_{ij}^{(l+1)} &= \psi(\mathrm{h}_{si}, \mathrm{h}_{sj}, \mathrm{h}_{di}^{(l)}, \mathrm{h}_{dj}^{(l)}, \varepsilon_{ij}^{'}) \odot (\mathrm{h}_{dj}^{(l)} - \mathrm{h}_{di}^{(l)}) \\
\mathrm{h}_{di}^{(l+1)} &= \mathrm{h}_{di}^{(l)} + \sum_{j \in \mathrm{N}_i} \mathrm{s}_{ij}^{(l+1)} \mathrm{W}^{(l+1)} \\
\mathrm{h}_{di}^{(L)} &= \sigma(\mathrm{h}_{di}^{(L-1)} + \sum_{j \in \mathrm{N}_i} \mathrm{s}_{ij}^{(L)} \mathrm{W}^{L})
\end{aligned} \tag{A4}$$

The σ presents the Tanh activation function used at the output of the Lth layer to restrict exploding values due to numerical instabilities. The static node features and the edge features remain constant over the simulation since the topography of the domain does not change.

Decoder

When the GNN results are computed they are processed through the decoder, which consists of an MLP ϕ, shared across all nodes. The decoder updates the hydraulic variables at the next time step in accordance with the results of the GNN. Similar to the encoder, the decoder MLP consists of two layers and has a hidden dimension G followed by a PReLU activation. The MLPS both in the encoder and the decoder do not have the bias terms as this could result in adding non-zero values to areas that are not flooded, causing water to originate from cells.

Appendix B: Hyperparameter search

A hyperparameter search was conducted in order to find the optimal configuration for both dike rings. The considered hyperparameters are the number of GNN layers (K) and the hidden features.

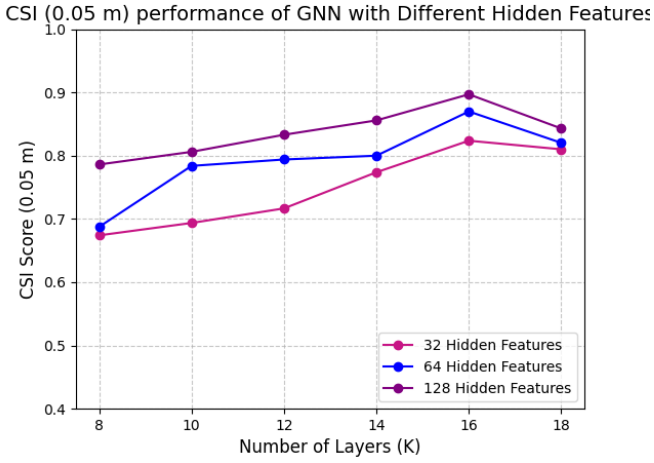
A hyperparameter search is a critical step in machine learning model development, as it aims at optimizing the models performance. The optimal configuration of hyperparameters depends on the task at hand and on the data used. Therefore, there is no universally best combination which applies to all scenarios. The hyperparameters chosen influence the complexity of the model. The best performing model is obtained by systematically exploring different combinations of hyperparameters.

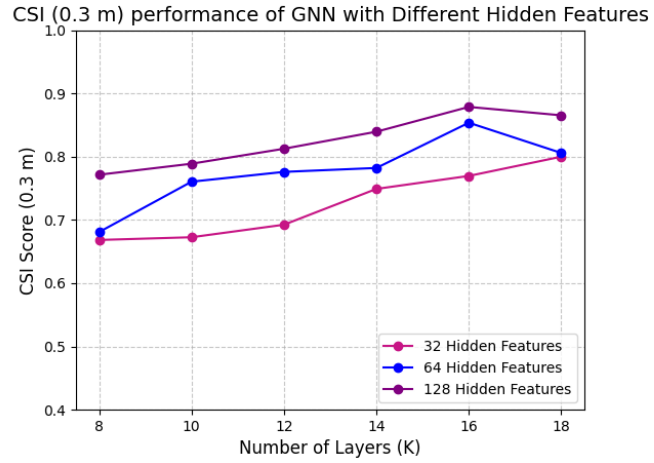
The results of the hyperparameter search are presented in Figure B2. The performance increases consistently with the amount of hidden features chosen. By increasing the number of hidden features we are essentially increasing the ability of the model to learn complex patterns and representations within the graph data. The hidden features represent node embeddings that encode information about the nodes and their neighborhoods. Increasing this value leads to a richer representation space, which causes the model to better capture nuances of the grid topology, node attributes, and allows for more expressive message passing.

The general spatio-temporal accuracy and quality of generated results also increases with the amount of GNN layers employed. A smaller number of layers is not able to correctly display the water propagation between time steps at the current discharge and on the case study area. Increasing the number of layers improves the information propagation through the graph. This results in each node to be able to capture and integrate dependencies from further away for each time step of the simulation. The performance roughly reaches an optimal state at K = 16 layers for dike ring 49 and K = 20 layers for dike ring 43. Higher numbers of layers result in a slight decrease of performance. This can be the result of overfitting after this number of layers due to the increased complexity of the model. For this reason the model chosen is the 16 layer 128 hidden features configuration for dike ring 49 and 20 layer 128 hidden features configuration for dike ring 43. Since these models displays the best numerical results.

Appendix C: Residual analysis

This section presents the results of the residual analysis for both datasets. Figures C1 and C2 visualize the statistics per simulation ranked by performance. The simulation id is displayed on the x-axis and the CSI, RMSE and residuals are plotted for the final time step of simulation. As can be observed from the figures, the residuals per simulation for dataset 1 show a slight positive trend in the water depth of approximately 0.18m. This is further investigated per test case. Figure C3 presents the residuals for test case 2. For the cells where both models simulate a flood as well as cells where only one of the two models simulate a flood the SWE-GNN slightly over predicts the water depth. The flooding extent, i.e. wet cells, are over predicted mainly at the breaching location. As discussed in the results section, this is a common finding in our test cases. The numerical model predicts the flood path in a certain direction and propagates the water rapidly over the grid cells. The amount of layers in the SWE-GNN restricts the water propagation over too many nodes per time step. This results in water dispersing around the breaching location for the first few time steps, as well as a slight lag in flood arrival in the general direction of the flood.





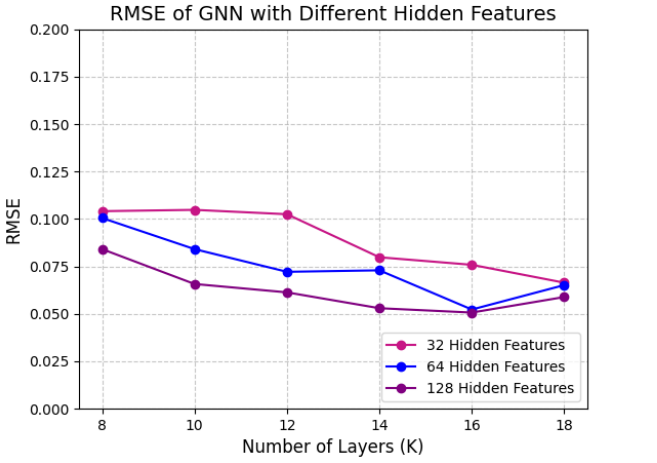
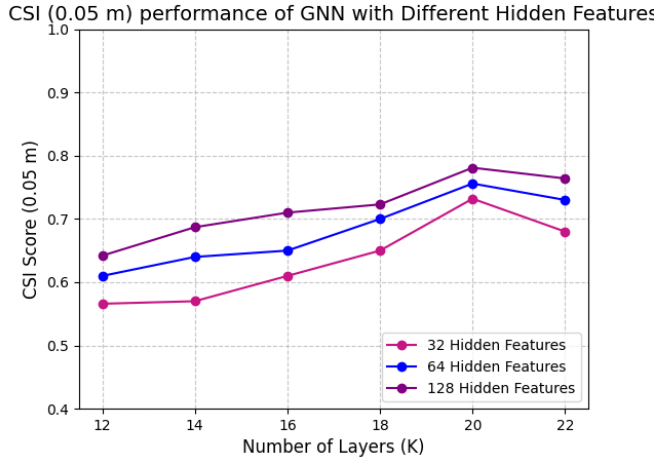
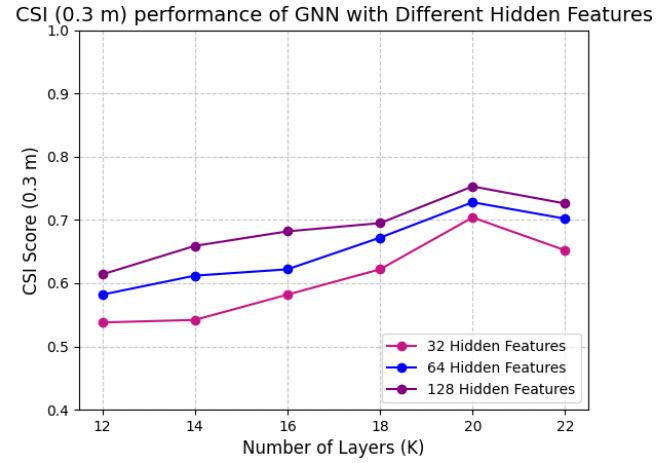


Figure B1. Visualization of the hyperparameter search results for different configurations of hidden features and number of GNN layers (K) for dike ring 49, dataset 1.





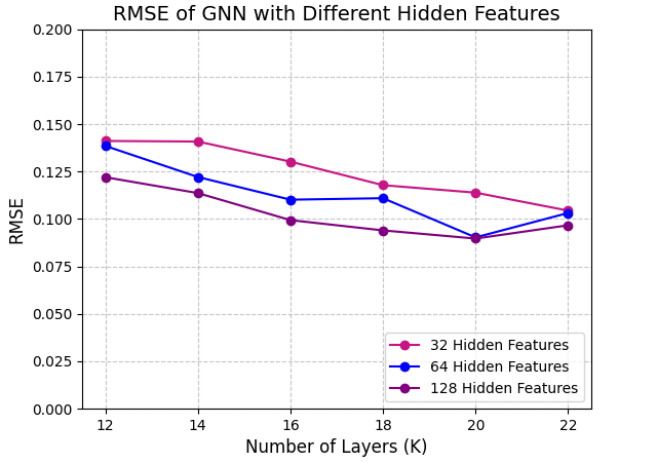


Figure B2. Visualization of the hyperparameter search results for different configurations of hidden features and number of GNN layers (K) for dike ring 43. dataset 2.

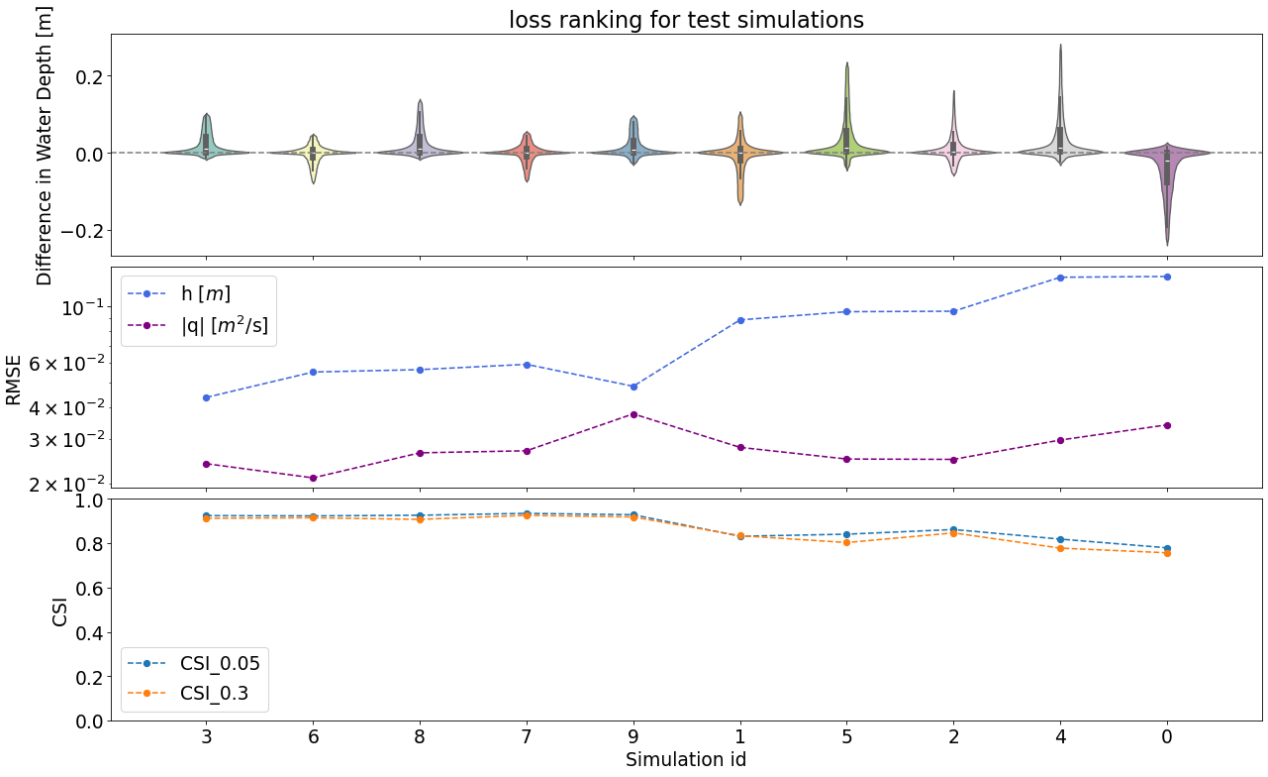


Figure C1. Statistical results per simulation for dataset 1, dike ring 49.

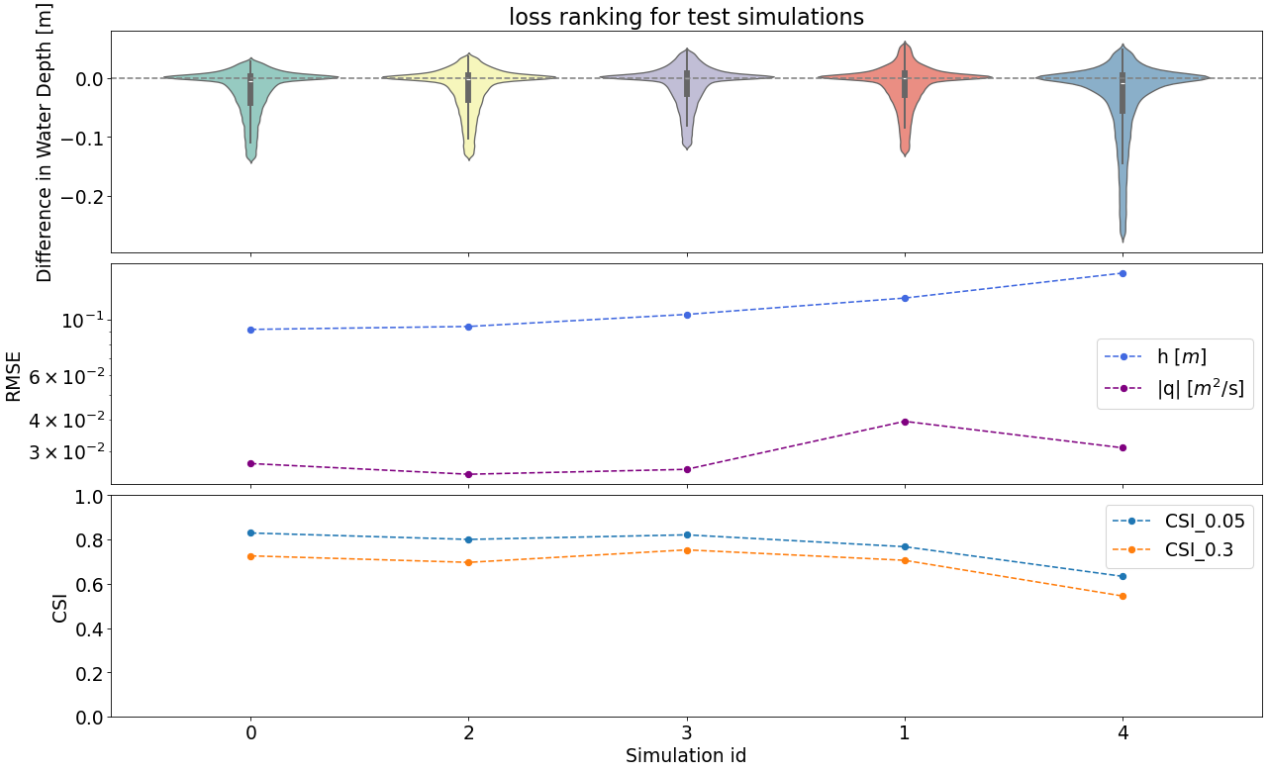


Figure C2. Statistical results per simulation for dataset 2, dike ring 43.

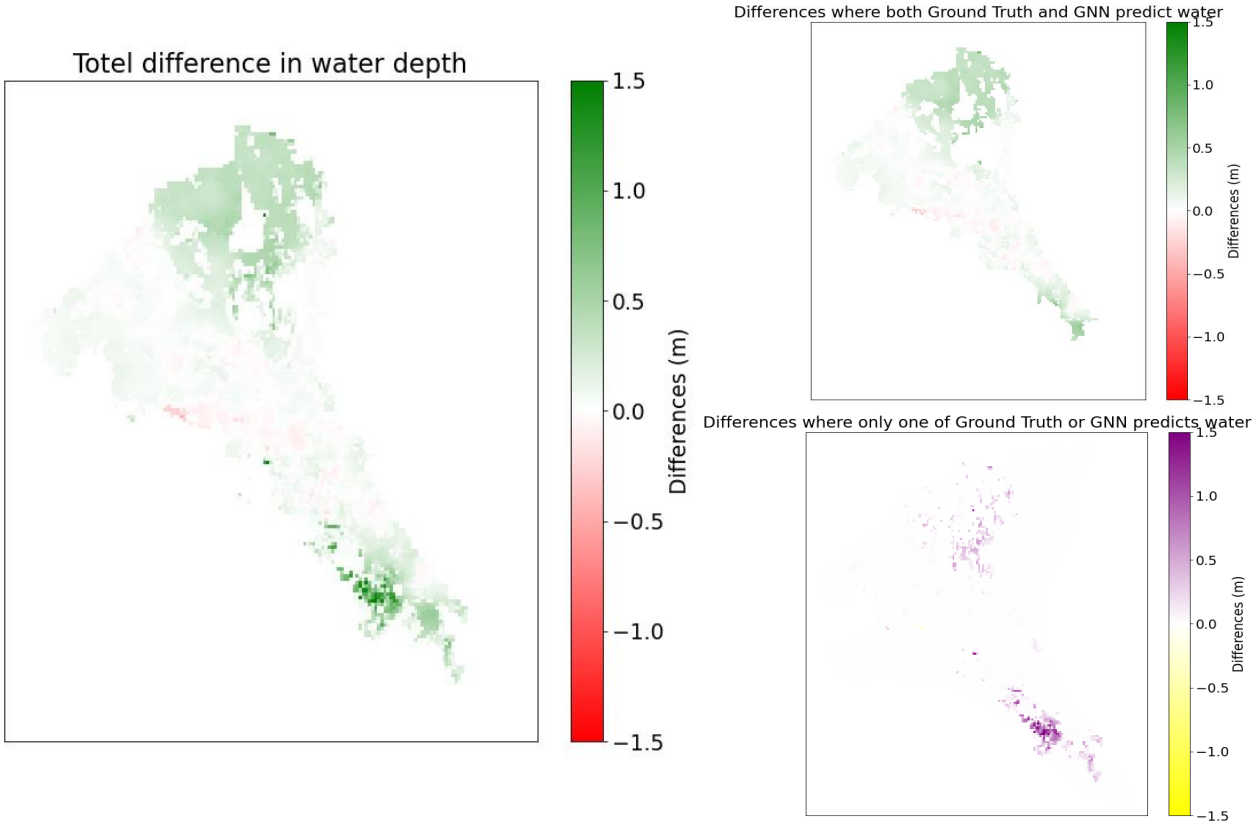


Figure C3. Plot of the residuals of dataset 1 for test case 2. The water depth differences at t=48 hours is analyzed for cells where both models predict flood and cells where only one of the two models predicts flood.

Appendix D: CSI Drop Off for fewer GNN layers

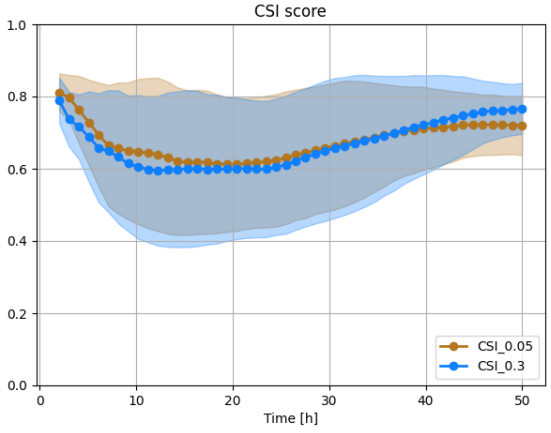


Figure D1. Initial drop off in performance for dataset 1 proves to be larger with fewer GNN layers before stabilizing to a certain CSI value

Figure D1 presents the drop off for the initial time steps as mentioned in the results section. The propagation of the flood in one direction according to Delft3D is too large, which leads to a lag in SWE-GNN prediction. Consequently the SWE-GNN inundates the grid cells around the breaching location, which leads to a lower CSI score. This phenomenon is more prevalent for models with fewer layers.

Appendix E: Ra2Ce analysis plots

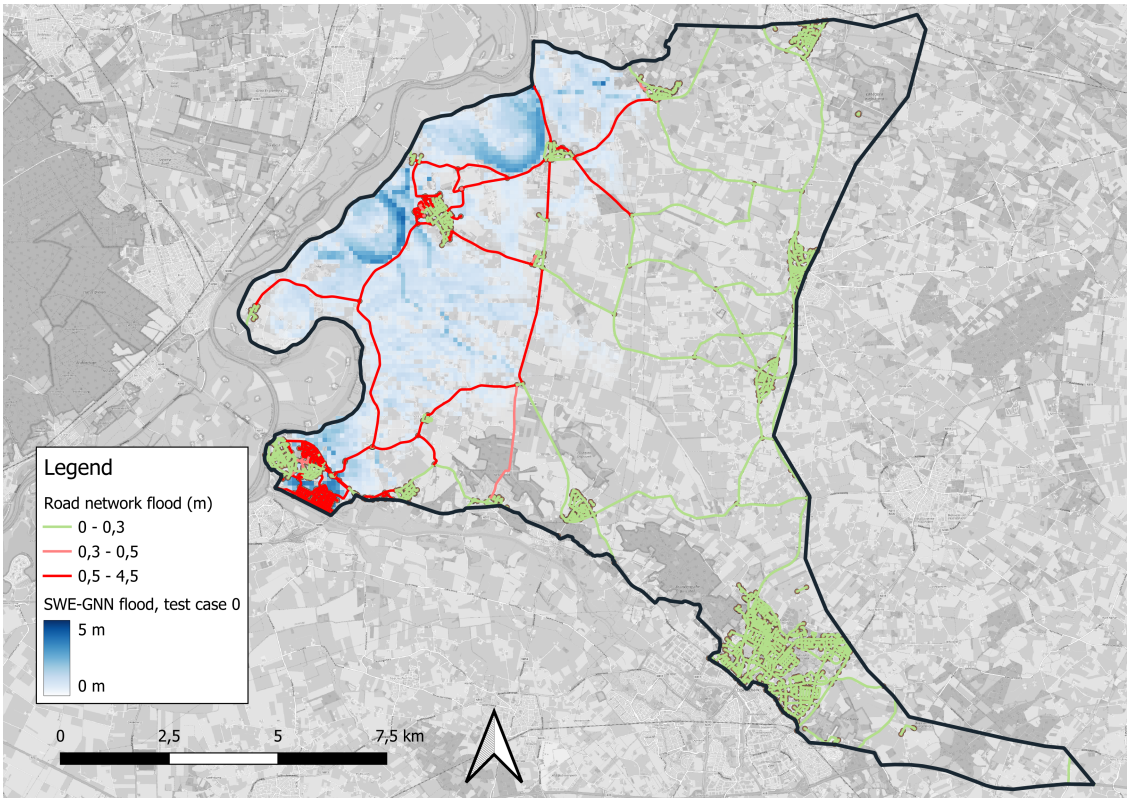


Figure E1. Plot of Ra2Ce analysis on SWE-GNN results. Test case 0 for dataset 1 visualized.

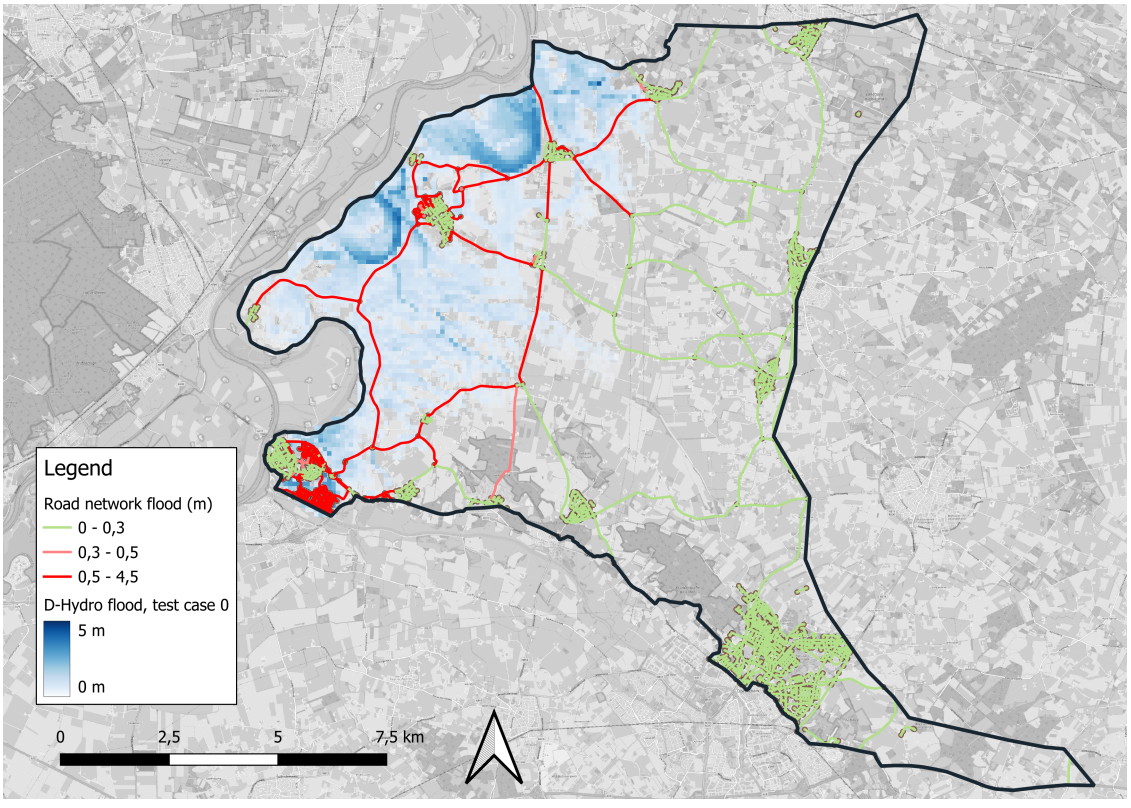


Figure E2. Plot of Ra2Ce analysis on D-Hydro results. Test case 0 for dataset 1 visualized.

Appendix F: Generalizability to unseen domains

To test the generalizability to unseen domains we tested the SWE-GNN trained on dataset 1 and 2 on 5 unseen test cases for dike ring 43. The locations of these test cases are presented in Figure F1.

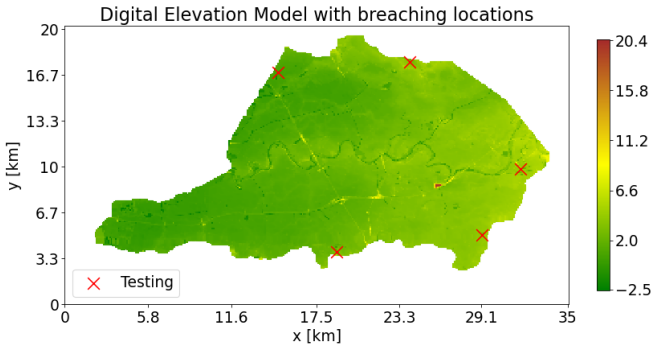


Figure F1. Digital Elevation Model (in meters) of dike ring 43 with the spatial distribution of the breach locations. Red crosses mark the locations of the breach.

The results indicate that the two SWE-GNN models are able to predict the flood propagation to a certain degree. The SWE-GNN trained on dataset 2 exhibits a slightly better CSI than the model trained on dataset 1 for the first 24 hours. This could be due to the increased complexity of the model, with more GNN layers. More complex models are better able to capture the representations and relations from the training data and effectively transfer them to the testing samples.

Overall, the model is modestly able to capture relevant patterns in the flood propagation and predict the water depths and associated discharges for the testing samples. This affirms the findings from Bentivoglio et al. (2023) to some extent, displaying the ability of the SWE-GNN to generalize to unseen topographies. For real world application during calamities, these results are insufficient at this stage. But the model does demonstrate some capabilities of performing on domains outside of the training data. For improving these predictions, model adaptations could be made such as employing multi-scaling methods to enhance the efficiency of message passing. In addition, better enforcing physics (Sabbaqi and Isufi, 2023) or generalizations to higher orders (Yang et al., 2022) may further improve the model accuracy, as stated by Bentivoglio et al. (2023).
Furthermore, the SWE-GNN might benefit from transfer learning (Olivas et al., 2009; Cache et al., 2024). Using a pre-trained model and transferring case specific information to that model could increase the accuracy, however this lies outside of the current scope of the project. At this point, it is more straightforward to create separate models for each domain to enhance their practical applicability.

All relevant plots for the SWE-GNN models on the unseen domain:

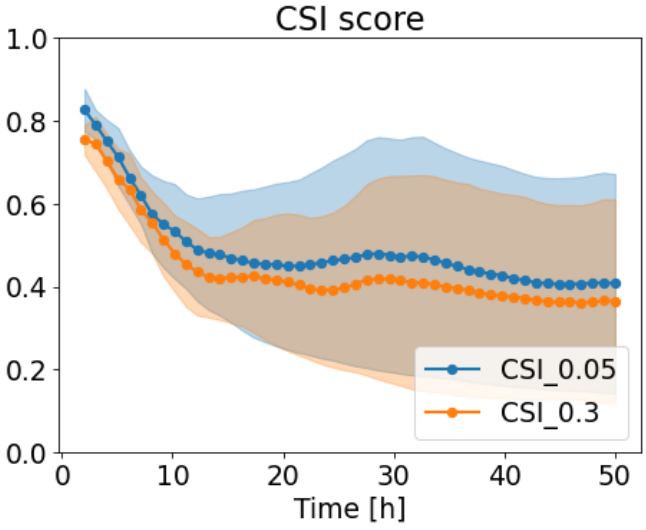


Figure F2. CSI for the SWE-GNN trained on dataset 1, tested on the unseen western half of dike ring 43.

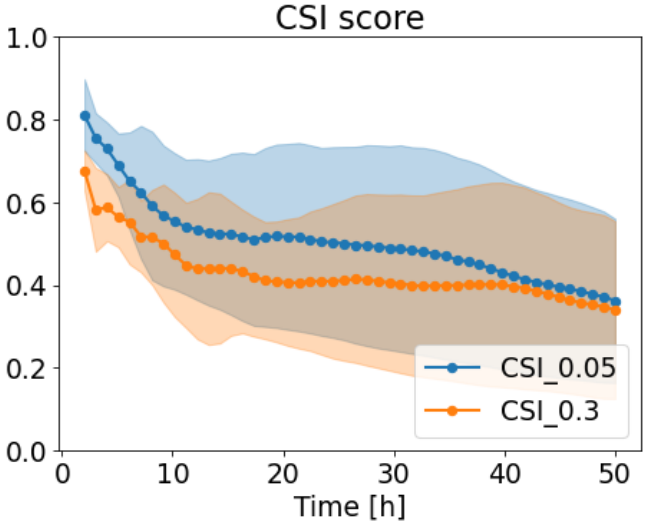


Figure F3. CSI for the SWE-GNN trained on dataset 2, tested on the unseen western half of dike ring 43.

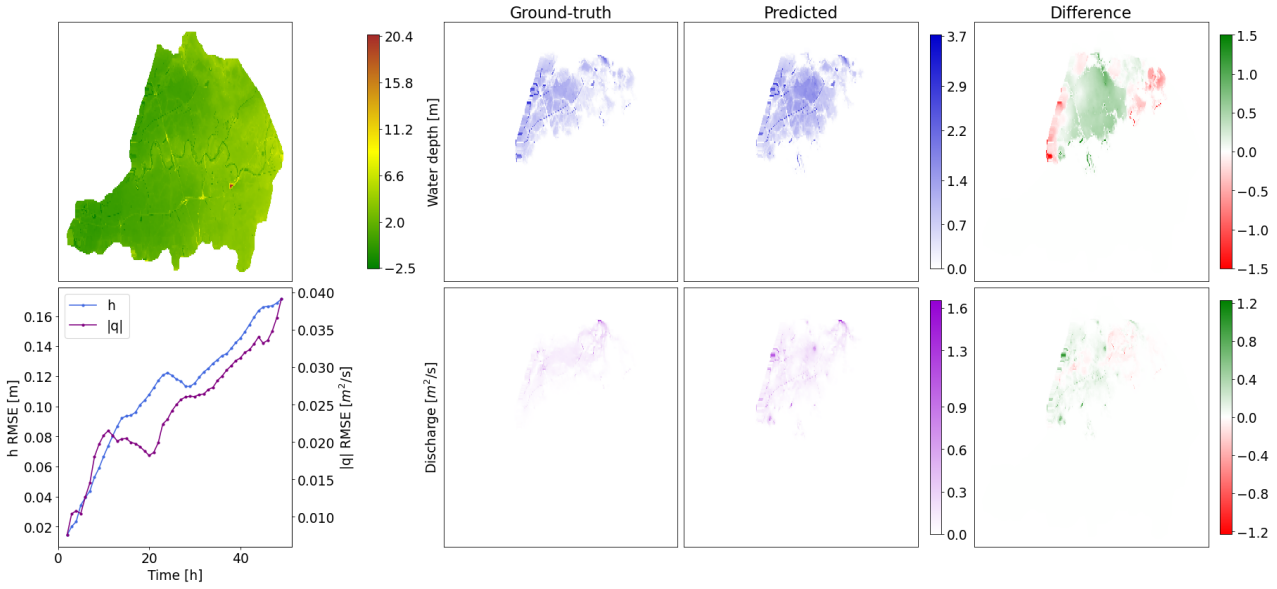


Figure F4. Plot for the SWE-GNN, trained on dataset 1, tested on the unseen western half of dike ring 43. The RMSE over time is plotted in the bottom left corner. The water depth and discharges are plotted for the final time step, t=48 hours. The total differences are plotted in the final column. Test case 0 of the testing dataset is visualized.

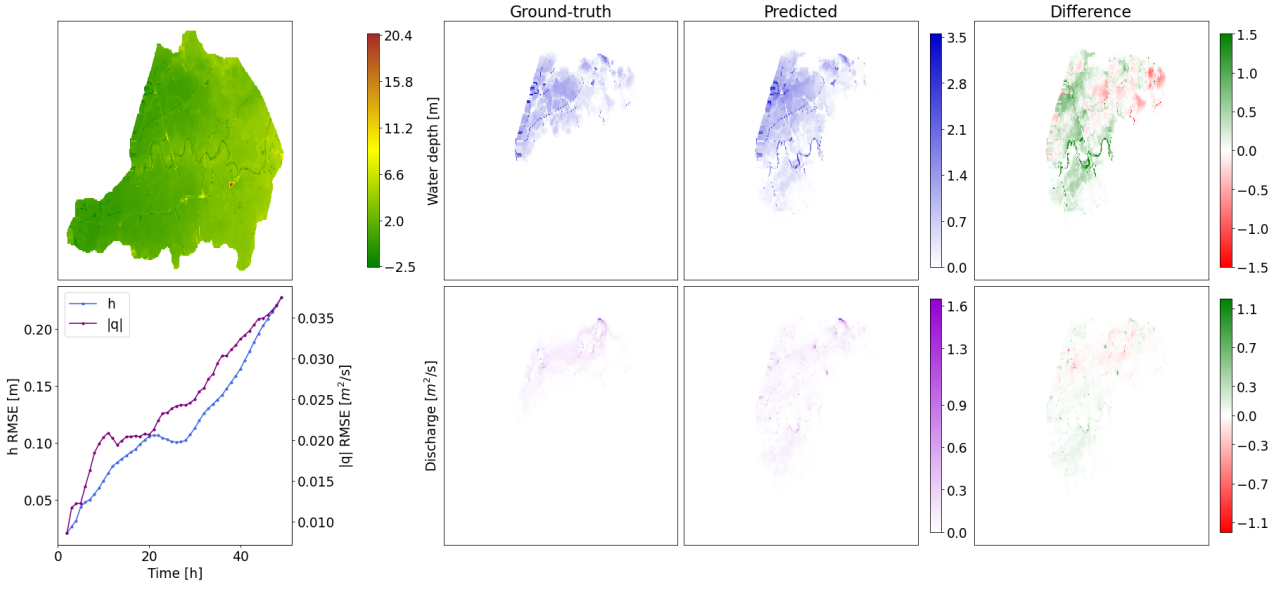


Figure F5. Plot for the SWE-GNN, trained on dataset 2, tested on the unseen western half of dike ring 43. The RMSE over time is plotted in the bottom left corner. The water depth and discharges are plotted for the final time step, t=48 hours. The total differences are plotted in the final column. Test case 3 of the testing dataset is visualized.