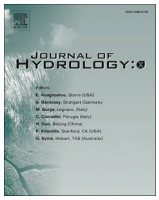
Journal of Hydrology X 22 (2024) 100167

Contents lists available at ScienceDirect

Journal of Hydrology X

journal homepage: www.sciencedirect.com/journal/journal-of-hydrology-x



Research papers

Data driven real-time prediction of urban floods with spatial and temporal distribution

Simon Berkhahn *, Insa Neuweiler

Institute of Fluid Mechanics and Environmental Physics in Civil Engineering, Leibniz Universität Hannover, Appelstrasse 9a, 30167 Hannover, Germany

ARTICLE INFO

Keywords:
Artificial neural network
Recursive prediction
Convolutional neural network
Temporal distribution
Real-time forecast
Urban flooding

ABSTRACT

The increase in extreme rainfall events due to climate change, combined with urbanisation, leads to increased risks to urban infrastructure and human life. Physically based urban flood models capable of producing water depth maps with sufficient spatial and temporal resolution are generally too slow for decision makers to react in time during an extreme event. We present a surrogate model with high temporal and spatial resolution for real-time prediction of water levels during a pluvial urban flood. We used machine learning techniques to achieve short computation times. The recursive approach used in this work combines convolutional and fully coupled multilayer architectures. The database for the machine learning was pre-simulated results from a physically based urban flood model. The forcing input of the prediction is precipitation and the output is water level maps with a temporal resolution of 5 min and a spatial resolution of 6 x 6 meters. The prediction performance can be considered promising for testing the model in real operational applications.

1. Introduction

Extreme weather events have challenged humanity throughout history. Due to climate change, extreme weather events will become more frequent (IPCC et al., 2021). Heavy rainfall events can lead to pluvial flooding, which is one of the most severe natural hazards for urban life (Zanchetta and Coulibaly, 2020). Adaptation of existing urban drainage systems is often not economically feasible. Therefore, flood forecasting tools are becoming increasingly important to prevent financial losses or personal injury (Rözer et al., 2021).

Since precipitation is the driving force behind urban flooding, a good weather forecast is needed. Predicting precipitation is a challenging task. The life cycle of relevant storm events can be short, requiring very fast methods. Since rain gauges cannot capture the spatial extension of a storm event, radar-based rain forecasting is now the most common tool (Nguyen et al., 2021). In the present study, the rain forecast is considered as a given input.

Physically based hydrodynamic models, in particular coupled 1D-2D models, are commonly used to generate urban flood maps based on a predicted rainfall event. The resulting flood maps have good temporal and spatial resolution, but have the disadvantage of long computation times. The main reason for this is the number of degrees of freedom in the system of equations. Even with a relatively coarse grid of 6 by 6 meters for a domain of one square kilometer, the temporal evolution of about 27800 water levels has to be solved for a medium-sized city. For complex urban structures, finer grids are required. Depending on the size of the domain, this leads to a large number of grid cells in a numerical model, resulting in long computation times.

Flood forecasting measures are becoming increasingly important, and many leading authorities are active in improving measures. For example, many authorities in Germany have recently requested static flood maps applied on large length scales. For example, two static maps are available on a web portal for the whole state of North Rhine-Westphalia, one for an event with a return period of 100 years and one for an event with an intensity of 90 liters per square meter per hour (LANUV, 2022). This type of flood map is useful for providing an overview of the potential flood hazard on a larger scale. However, real rainfall events can have much more complex spatial and temporal structures than those used to generate such static maps. For detailed early warning and operational contingency planning, the use of real-time precipitation forecasts as input to flood models is often required. Since the relevant heavy rain events have short lead times, real-time flood forecasting requires very short computation times. Therefore, detailed physically based models are not suitable for real-time purposes (Henonin et al., 2013). As a result, various approaches have been developed for surrogate models that are fast enough to be operational

* Corresponding author.
E-mail address: berkhahn@hydromech.uni-hannover.de (S. Berkhahn).

https://doi.org/10.1016/j.hydroa.2023.100167
Received 22 December 2022; Received in revised form 6 October 2023; Accepted 26 November 2023
Available online 20 December 2023

(Zanchetta and Coulibaly, 2020).

Simplified physically based models, such as a 1D/1D model that couples a 1D sewer model with a 1D surface model, have been tested as simplified fast models (Leandro et al., 2011). Information such as runoff and sewer overflow volume can be simulated with such a model. However, the extent of surface flooding is not part of the model. Without this information, it is difficult to produce hazard maps and implement early warning systems. The aim of many studies is therefore to combine the ability of detailed physically based models with the fast computational times of surrogate models to produce reliable flood maps needed for real-time forecasting.

In recent years, many data-driven models have been developed to achieve this goal (Sun et al., 2020). A comprehensive overview of deep learning methods used for flood mapping is given in Bentivoglio et al. (2022). For example, methods for the generation of maximum water level maps with artificial neural networks (ANN) have been presented in several studies (Berkhahn et al., 2019; Lin et al., 2020b; Hofmann and Schüttrumpf, 2021). All of these models used pre-simulated scenarios from a hydrodynamic model as training data for the ANN. Although the trained ANNs are able to predict flood maps in good agreement with hydrodynamic models, they are limited by the fact that they are trained for a fixed location and lack the temporal distribution.

To overcome the constraint of fixed locations, in Löwe et al. (2021) a model for maximal flood depth prediction was developed using convolutional neural networks (CNN). The aim of this model is to generate flood maps for areas which were not included in the training process. Using the NSE as performance indication, the prediction of water levels showed good results at many locations. In Seleem et al. (2023), a similar CNN-based approach was compared with a random forest approach for three study areas in Berlin, Germany. The authors found that CNN outperformed random forest in terms of generalisation. The studies by Löwe et al. (2021) and Seleem et al. (2023) did not use a detailed sewer model and no temporal resolution was considered. A model with temporal resolution was presented in Lin et al. (2020a). The authors used an ANN model to predict urban flooding with a spatial resolution of 4 by 4 meters and a temporal resolution of 1 h. The input to the model was the discharge at the inflow points to the urban catchment studied at time step t. The output was the water level map at time step $t+3$ hours. A time distribution was generated by rerunning the model with a sequence of start times. For example, if results for time step $t+4$ are required, the simulation is started at time step $t+1$. This approach allows a time sequence to be achieved, but the model output for a given time is not influenced by the previous states of the model. The prediction results were found to be good for short events up to three hours in duration. For longer events, the prediction performance decreased significantly. Recurrent neural networks could be a way to overcome this problem. In this type of ANN structure, the output at one time step is used as input for the next time step. In Burrichter et al. (2023) a real-time urban flood prediction model with high temporal and spatial distribution including the pipe network based on artificial neural networks is presented. The forecast model uses precipitation forecasts and manhole spilling forecasts as input. The output of the model is the complete sequence of flood maps for the whole catchment with a temporal resolution of 5 min. The forecast of spilling of manholes is, however, not always feasible. Instead of generating a complete sequence of outputs, many authors in the field of urban hydrology have used recurrent structures to produce time-distributed forecasts at individual locations [e.g. Chang et al., 2014; Gude et al., 2020]. Recursive structures allow intermediate results to be adjusted, for example, to integrate measurement data. A method based on a recursive neural network structure, which incorporates the main ideas of physically based models such as hydraulic gradients, is presented in the preprint Bentivoglio et al. (2023). Although such an approach allows very good transferability to unknown areas, the temporal resolution of several hours is still a drawback for real time warning systems. To the author's best knowledge, no model for urban flood prediction with real-time forecasting abilities with high temporal and spatial distribution that is based on a recursive structure can be found in literature. The main challenge for such a model is the huge amount of output data. Representing a medium sized urban catchment with sufficient spatial distribution leads to an immense number of cells. In addition for the temporal distribution parts of the flood history are needed.

In this paper we present an ANN approach to predicting water level maps (6 by 6 meters) at an urban catchment scale with a temporal resolution of 5 min. As one of the claims of this work is to develop a model that can be trained and used on a well-equipped standard computer, such as those used by local authorities, we have adopted a strategy to limit the memory consumption. In this paper we present a detailed description of the model setup and training process. The methods have been evaluated for a catchment scale test case.

More generally, machine learning has been used extensively in the wider field of hydrology in recent years. For example, Nguyen et al. (2021) improved radar-based rainfall forecasting using LSTM neural networks. Many studies have also used machine learning techniques for drought modelling in recent years (Sundararajan et al., 2021). A good overview of the use of machine learning in hydrology is given in (Xu and Liang, 2021).

2. Methods

As outlined in the introduction, information on the temporal evolution of spatially distributed urban flood heights is very beneficial for an early warning system. Therefore, this section outlines a fast machine learning based model for predicting time series of flood maps. The model proposed in this work is a surrogate for the model output of a specific physically based hydrodynamic model. To generate the training database for the machine learning based surrogate model, the hydrodynamic model was used to simulate a large number of flood events. For this purpose, an ensemble of rainfall events was used as input.

2.1. General concept

In order to build the real-time urban flood prediction model presented in this study, three main challenges had to be faced. (1) Using a sufficiently fine spatial resolution for a flood map of an urban catchment can easily exceed the main memory limits of a standard PC during training. To cope with the memory limitations of a standard PC, the catchment area had to be divided and the data contained in a flood map had to be compressed using autoencoders. (2) A suitable structure for the ANN representing the time series prediction of compressed flood maps had to be found. (3) The hyperparameters of the machine learning model, such as the length of the input rainfall time series and the network topology, need to be tuned in a reasonable way.

As written above in (1), flood maps on a catchment wide scale are too large for time series predictions. For this reason, the model is structured in two parts. One part is for the compression and decompression of the flood maps. It is shown in Fig. 1. The other part is for the time series prediction applied to the compressed maps and it is shown in Fig. 2. Both parts are trained independently and the first part (Fig. 1) has to be developed first to be used in the second part (Fig. 2). The input and output for compression/decompression is the results of the hydrodynamic model. The input to the time series prediction is compressed flood maps from previous and current time steps and the rainfall time series. The output is a compressed flood map for the next time step. The proposed approach for the time series prediction (Fig. 2) includes the following steps:

1. Start with a physically meaningful water level map (e.g. the dry map) in compressed form.
2. Use rainfall data as input to the model.
3. Use a NARX model for recursive time series prediction of compressed flood maps. This involves using both rainfall data and compressed flood maps from previous time steps.

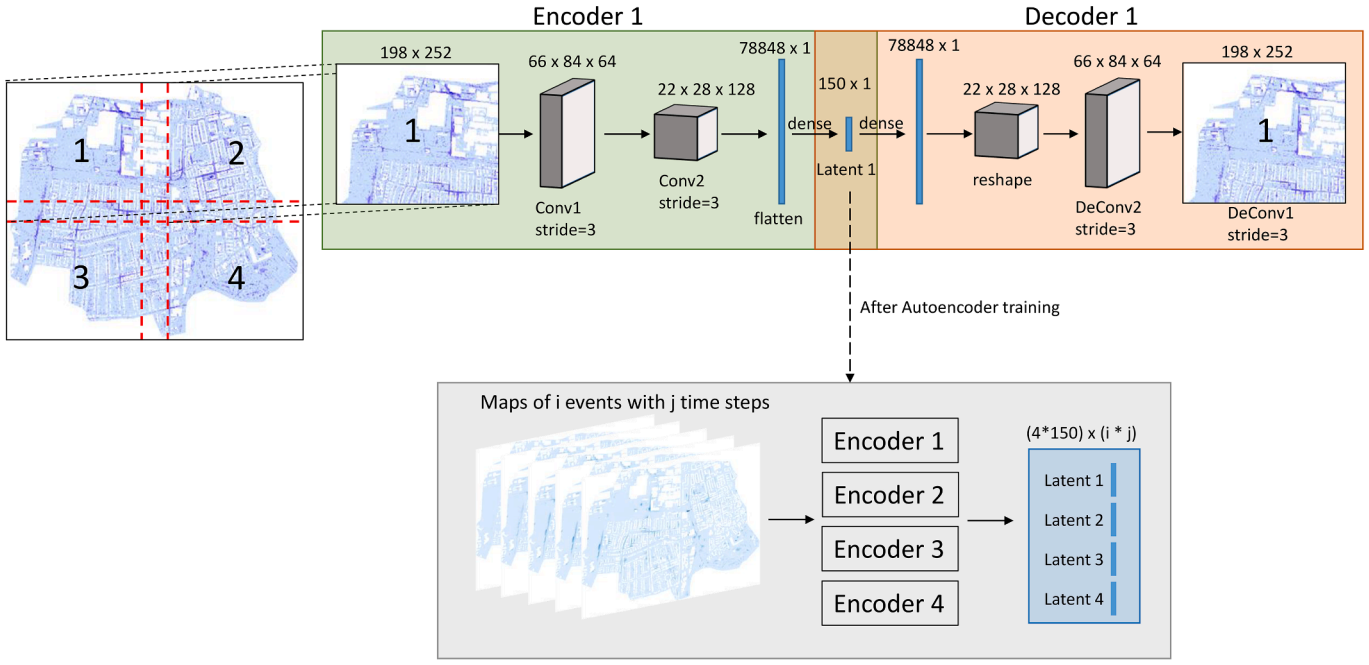


Fig. 1. Structure of the map compression and decompression part, with 4 subdomains of the flood map. Each is compressed to a vector of 150. The index i refers to the number of events and j to the number of time steps.

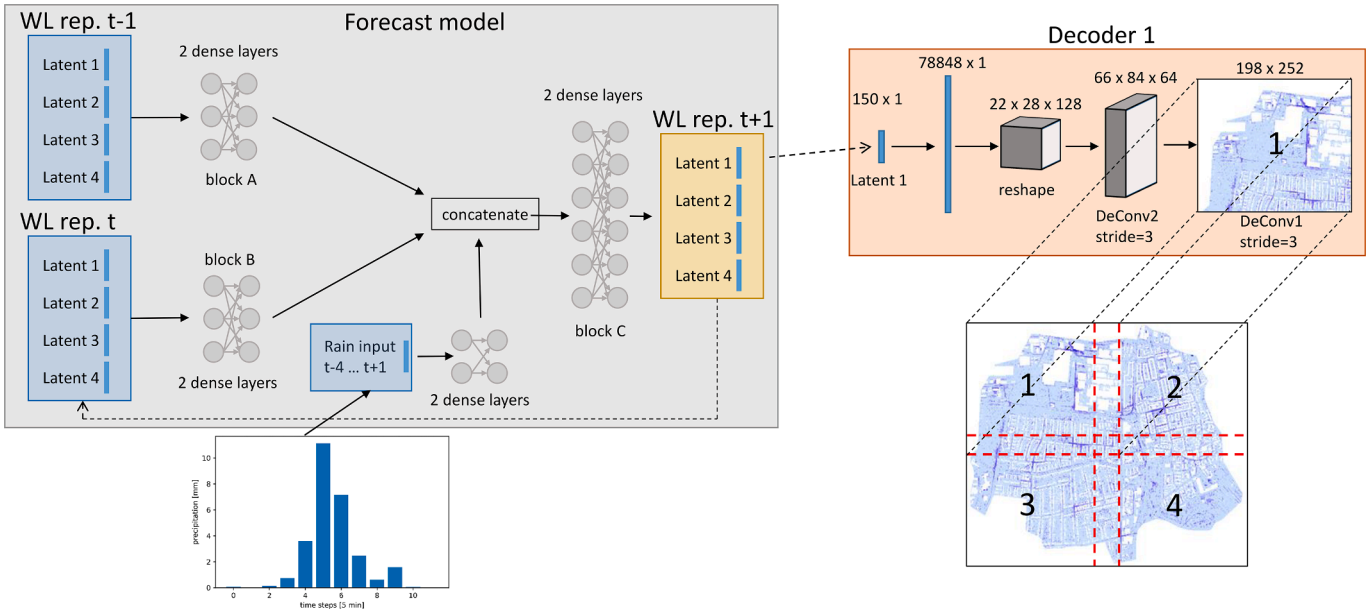


Fig. 2. General structure of time series forecast model using compressed maps. The results can be decompressed if the full maps are needed.

4. If necessary, decompress the flood map for visualisation purposes.

2.2. Flood map compression

In the proposed water level time series prediction model, the compressed flood maps are part of the input and the output of the model. Compression and decompression is here done using an autoencoder approach. A very clear introduction into autoencoders can be found in Dumoulin and Visin (2016). The equations representing the autoencoder for the flood maps compression purpose read as

$$g(\widehat{WL}) = l_{WL} \quad \text{and } f(l_{WL}) = WL \quad \text{combined to}$$
$$f(g(\widehat{WL})) = WL \quad \text{with the goal } \widehat{WL} = WL. \tag{1}$$

Here, the function g is the encoder and f the decoder part. The WL denotes a water level map and l_{WL} the latent space representation for that water level map. The $\widehat{WL}$ indicates the expected water level map. The latent space representation of the flood maps l_{WL} will be used for the time series prediction with the NARX.

The general structure of the autoencoders used in this study is based on the U-Net architecture used in many other studies for urban hydrology purposes (e.g. Li et al., 2021; Löwe et al., 2021; Seleem et al., 2023). In these studies, the focus was not on data compression, but on feature recognition and transferability. Since the aim of the present study is to use the autoencoders for data compression, a simplified version of the U-Net architecture is used. The skip connections that are

used in other applications of the U-Net architecture to improve training convergence, are omitted in the present study. The compression part of the U-Net architecture consists of a series of convolutional layers that reduce the spatial resolution of the input image while increasing the number of features that are extracted. This part keeps the global context of the input image and generates the latent space representation. In the decompression part the latent space vector is expanded through transposed convolution to increase the spatial extend and reach the original image size. This U-Net architecture is based on 2D convolutional layers and 2D deconvolutional layers. A good introduction into these layer types is given in Bentivoglio et al. (2022). These types of layer use rectangular pixel images as input. Therefore, machine learning is performed on the PC's GPU, which is optimised for processing pixel images.

In order to avoid a limitation of the catchment area size due to the memory size of the GPU, the catchment area is subdivided into subdomains which overlap at their boundaries. It has already been illustrated in Berkhahn et al., 2019 that for a well-trained data-driven model the segmentation does not lead to discontinuities at the boundaries. For the overlapping parts, the machine learning model predicts multiple values that do not have to be the same. To obtain a flood map for the whole catchment, the overlapping parts are averaged.

The way a 2D deconvolutional layer functions requires a restriction of the size of the input images. As illustrated in Bentivoglio et al., 2022, a filter matrix is step wise slid over the input image. The step size of this sliding is called stride in the context of autoencoders. Each dimension of the input image must be an integer divisible by the stride value. Since the proposed simplified U-Net architecture uses multiple layers of the 2D convolution, the restriction for the input size must be fulfilled in each layer. The mathematical expression for this restriction reads as

$$nn_x = \left\lfloor \left(\frac{n_x}{s^{n_f}} + 0.5 \right) s^{n_f} \right\rfloor , \tag{2}$$

where nn_x is the new number of pixels of a subdomain in x-direction, n_x is the original number of pixels of a subdomain in x-direction, s is the stride and n_f is the number of filter layers. Since n_f and s are positive integers, Eq. (2) always gives an $nn_x \geqslant n_x$. The same equation as (2) applies to the y-direction, replacing the index x by y. If no division of the catchment into subdomains is needed, $nn_x - n_x$ pixels with the value zero are equally placed on the left and right hand side of the flood map. The same applies on the upper and lower side with $nn_y - n_y$ pixels. For cases where a division of the catchment is needed, no zero pixels are added but overlapping pixels of the subdomains are used to satisfy Eq. (2) and its modification for y. Fig. 3 illustrates this strategy.

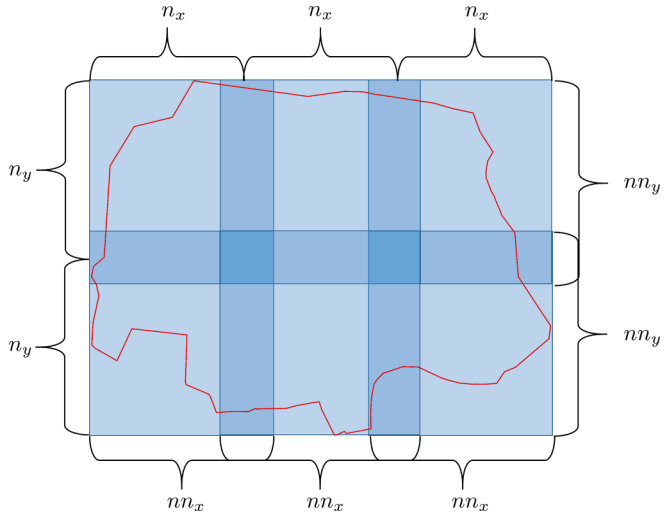


Fig. 3. Schematic representation of the division of the pixel image of dimension $n_x \times n_y$ of a catchment area (boundary shown in red) into overlapping subdomains of dimension $nn_x \times nn_y$ to satisfy the dimensional restriction of the autoencoder training.

One hyperparameter that needs to be set for the proposed autoencoder architecture is the dimension of the filter matrix, also called the kernel size. Larger kernel sizes can be beneficial for capturing larger spatial features in an image. The drawback of using large kernel sizes is the increasing number of weights that need to be trained. As shown in Löwe et al. (2021), kernel sizes larger than 3×3 show no improvement in prediction performance for flood maps. Therefore, in this study, the kernel size is set to 3×3. In order to reduce the number of weights that need to be trained, the stride value is set to the edge length of the filter matrix and therefore equal to $s = 3$. With this strides and kernel size combination, each location is scanned once by the filter.

Further hyperparameters are the number of filters per layer denoted as b_f and the number of filter layers n_f. Following the U-Net architecture the number of filters is doubled for each convolution step towards the bottleneck. This approach reduces the number of hyperparameters to tune, as only the number of filters in the first layer needs to be set. The influence of different b_f and n_f combinations was investigated for the test case (Section 3) and shown in Table 2.

The output of the bottleneck, the latent space representation, is later used as input to the time series model (see Fig. 2). It is not a 2D map, but a vector. This is achieved by flattening the map from the last convolutional layer to a vector and process it through a dense layer with sigmoid activation. The advantage of sigmoid activation is that it scales the latent space from zero to one for later use as input to the NARX model. For reconstruction of the flood map in the output layer of the autoencoder a linear activation function is used. As in other studies (e.g. Löwe et al., 2021; Guo et al., 2022), for all other layers in the autoencoder the Leaky ReLu activation is applied.

A loss function is used to evaluate the mismatch between input to the encoder and output of the decoder during the autoencoder training (Bentivoglio et al., 2022). In flood maps, there are typically cells in the database that never exceed zero. Since ANN should not be used for extrapolation, a non-zero prediction for these cells would not be meaningful. Essentially, this means that the correlation between precipitation and flooding specific to these cells is not part of the information contained in the training dataset. Therefore, an adapted form of the absolute error function is implemented for use as the loss function. With the loss function written as

$$E = |x_n - y_n| \; \forall n(x_n > 0) , \tag{3}$$

where x_n is the value from a specific grid cell in the uncompressed water level map and y_n the autoencoder output for the same location, only cells with values greater than zero are included in the loss calculation. With this approach, the autoencoder is not trained on cells with values that are always zero and will not produce good results for these cells. To obtain reasonable flood maps, masking of the always-zero cells is applied to the output of the decoder part. This is achieved by multiplying, cell by cell, with a matrix in which the non-zero cells are represented by a one and the always-zero cells are represented by a zero.

For more robust training and better results, all input data to the autoencoders must be scaled. Commonly data is scaled from zero to one when using the sigmoid and the leaky ReLu activation function. Due to the use of the loss function shown in Eq. (3), the input for the autoencoders was scaled differently. Each cell in the catchment where the water level exceeds zero at least in one flood map in the training data is scaled from 0.2 to one and cells that are never flooded are set to zero. Thus one can differentiate between a predicted zero and an always-zero. Together with the loss function shown in Eq. (3), only cells with potential to be flooded are considered in the optimization. Cells that are predicted with a negative water depth are set to zero.

2.3. Time series prediction

After successful training of the autoencoders for each sub-domain of the catchment, the encoder part can be used to generate the input (compressed maps of water level heights) for the time series prediction as shown in Fig. 2. As shown in the lower left part of Fig. 1, flood maps of the subdomains are fed to the corresponding encoder to obtain the latent space representation for the subdomain. The outputs of all the encoders for the subdomains are then concatenated into one vector for each time step of the flood event. With this approach, one gets a vector representing the corresponding flood map for each time step in each event. The latent representation of the water level map for the j-th time step of event number i is denoted as $\overrightarrow{wl}^{*}_{i,j}$. This vector is used as the state vector that is propagated through time by the NARX model to make prediction of a latent representation of the map of the next time step. In order to distinguish whether the flood is rising or falling, the flood map from the time step $t-1$ is useful information and therefore this flood map is used as an additional input. The main driver of urban flooding is the rainfall amount. Therefore, a time series of rainfall amount (for the choice of length see below) is used as an exogenous input to the NARX model. The proposed model structure of the NARX model is shown on the left hand side of Fig. 2. For the last layer in the time series prediction part of the model, the sigmoid activation function was used to rescale the data from 0 to 1 to match the scaling of the input to the decoders. For all other layers, leaky ReLu was used as the activation function.

The two flood map representations of the current t and the previous $t-1$ time step are used as input and are processed in separate streams of dense layers (called block A and B in Fig. 2). In a separate stream, the rainfall input is processed in dense layers. Although a recursive layer type could also be used to process the rainfall time series, a dense layer was chosen. The results showed good performance and as it simplifies the model architecture compared to recursive layers, it reduces the risk of overfitting that can occur when introducing unnecessary complexity.

The processed rainfall information and the processed flood information are concatenated into one vector. This vector is processed in a dense layer structure (block C) to obtain the compressed flood map representation of the time step $t+1$. During the operational prediction mode, the predicted flood map representation can be used as input for the next time step. To obtain the complete and readable flood map, the trained decoders are used as shown on the right hand side of Fig. 2. Noise layers using Gaussian noise were used to improve the robustness of the model against error propagation by feeding back the water level representations to the next time step. To avoid disturbing the scaling from zero to one, the noise layer is used as a second layer to block A and block B. The only hyperparameter to be adjusted in the noise layer is the standard deviation. Assuming that the uncertainty for time step $t-1$ is larger than for time step t, the standard deviation for block A is chosen to be twice as large as that for block B.

To decide about the length of the rainfall time series input $\overrightarrow{p}_{i,j}$ one needs to know how many time steps of rainfall from the past have a high influence on the flood map in the next time step. For this purpose, a correlation analysis was conducted for the test case presented in Section 3. The correlation coefficient between the rainfall time series and the time series of total flood volume was calculated for each event. By shifting the rainfall time series stepwise into the future, one can see the correlation between the past rainfall and the current flood volume. As correlation measure the Spearman's rank correlation was used. The formula for Spearman's rank correlation coefficient reads

$$\rho_{sp} = 1 - \frac{6\sum d_i^2}{m(m^2-1)}, \tag{4}$$

where d_i is the difference between the ranks resp. position of corresponding elements in two sorted datasets, and m is the number of values in each dataset. The factor of 6 in the denominator is a normalization constant that ensures that the coefficient takes on values between -1 and $+1$. A high Spearman's rank correlation coefficient value shows that the relationship between rainfall and flood volume can be well described by a monotonic function (Gaál et al., 2015).

2.4. Optimisation algorithm and training process

Two different training algorithms were used to train the autoencoder and the NARX model structure. The Adam algorithm (Kingma and Ba, 2017) was used to train the autoencoders. This type of stochastic gradient descent optimisation algorithm is recommended in Ruder (2016) for training large datasets. In addition to the current gradient, the Adam algorithm incorporates momentum to take into account past gradients. This helps to avoid getting stuck in local minima. For training the NARX model, the stochastic gradient descent (SGD) optimisation algorithm without momentum (gradients from past iteration) was found to be the most efficient. Other optimisation algorithms were tested for both model parts with similar loss values, but led to longer computation times. In initial test runs for both model parts, it was found that if the learning rate was small enough, all training runs converged to the same loss value. Therefore, the autoencoders were trained with a constant learning rate of 0.001. For the NARX training, a learning rate schedule was used to speed up the training process. An exponential decay function was used with an initial learning rate of 0.1. This value was decayed every 20000 steps with a base of 0.8.

In each iteration during the training process, a batch of training samples is processed. The error gradient for adapting the weights is calculated for the whole batch. The larger the batch size, the better the approximation of the gradient. Therefore, the largest batch size possible with the given memory limitation was used. The autoencoders were trained on all flood maps in the training dataset. The NARX model was trained on all time steps of all events in the training dataset.

To avoid overfitting, an early stopping approach was used as the stopping criterion of the training process for both model parts. Training is stopped after 5 iterations with no improvement in the validation loss for the autoencoders and 500 iterations for the NARX model. The weights for the iteration with the lowest validation loss are then saved.

2.5. Validation metrics

When evaluating model results, especially for data-driven models, the choice of measures to quantify performance is crucial. In the case of urban flood prediction, it is useful to consider only cells in the flood map with a certain flood depth threshold (Jamali et al., 2019). The reasoning is that only these cells indicate potential damage, so bad predictions in the small level ranges are not problematic. Therefore, cells are relevant if they exceed the threshold either in the hydrodynamic model results or in the prediction model output. In this study, a threshold of 5 cm was used to define a cell as flooded (see Jamali et al., 2019). For some evaluations, validation measures of hotspots were needed. Hotspots were here quantified as locations exceeding a threshold value of 30 cm. All validation metrics used in this study are shown in Table 1. During the hyper-parameter analysis we used the *root mean squared error* (RMSE) of the output variables for evaluation. To save time, we compared the latent space vectors during the hyperparameter analysis for the time series prediction model, instead of using the decompressed maps. This eliminates the need to use the decoder for each flood map in each variant. A high RMSE value is a good indicator for bad representation, but a low RMSE does not necessarily indicate a good prediction of a flood map. As we are averaging over space when calculating the RMSE, bad prediction for hot spots can be missed if the rest of the map is in good agreement. We therefore use the dimensionless *nash–sutcliffe efficiency* (NSE) for creating maps of prediction accuracy calculated from all maps. Further, we use the *critical success index* (CSI) (Löwe et al., 2021). This measure addresses the application of a flood map in such a way that an alarm is given when a level rises above a certain threshold. A CSI of 1

Table 1
Validation Metrics Used for Comparing the Flood Prediction with the Hydrodynamic Model.

Validation netric	Formula	Range	Optimal score
RMSE [cm]	$\sqrt{\frac{1}{k}\sum_{n=1}^{k}(wl_{n,t,thr} - wl'_{n,t,thr})^2}$	$[0,\infty)$	0
NSE [-]	$1 - \frac{\sum_{i=0}^{k}\sum_{j=0}^{t}(wl_{i,j,n,thr} - wl'_{i,j,n,thr})^2}{\sum_{i=0}^{k}\sum_{j=0}^{t}(wl_{i,j,n,thr} - \overline{wl'}_{n,thr})^2}$	$(-\infty,1]$	1
CSI [-]	$\frac{H_{thr}}{H_{thr} + M_{thr} + FA_{thr}}$	$[0,1]$	1
ROTAf [-]	$\frac{A}{A'} = \frac{H_{0.05} + FA_{0.05}}{H_{0.05} + M_{0.05}}$	$[0,\infty)$	1

Descriptions:
- $wl_{n,t,thr}$: Water level in the n-th grid cell predicted by the forecast model for time step t and threshold thr.
- $wl'_{n,t,thr}$: Hydrodynamic model result for the same location and time step with threshold thr.
- H_{thr}: Number of cells where the water level is above the threshold thr in the prediction model and the hydrodynamic model results.
- M_{thr}: Number of cells where only the hydrodynamic model result has a water level above the threshold thr.
- FA_{thr}: Number of cells where only the prediction model has a water level above the threshold thr.
- A: Flooded area in the prediction model.
- A': Flooded area in the hydrodynamic model result.

would mean that no alarm was missed and no false alarms occurred. In Löwe et al. (2021) the ratio of total area flooded (ROTAf) is also used to compare the results. A cell is considered flooded when a threshold of 0.05 meters is exceeded.

2.6. Recursive forecast mode

During the training process time series forecast model shown in Fig. 2, the predicted water levels are not fed back to the input. This is known as a series–parallel NARX architecture (Hermansah et al., 2022). Only in the recursive forecast mode the predicted water level from one time step t are used as input for the next time step $t + 1$. For all tested events in the present study this means that the recursive forecast run is started with a dry map and used only the precipitation data as external input to predict the complete time series of water level maps.

3. Test case

Among a suitable neural network model structure, a sufficient database is needed to generate meaningful results. For training and testing the model we used a database presented in Rözer et al. (2021). Since realistic rainfall events are required for real-world applications of an urban flood model, observation data have been used to obtain natural rainfall events. This work was not part of the present study, but will be briefly presented. A more detailed description can be found in the supplementary material to Rözer et al. (2021). Historical rainfall events from different locations were matched to the statistics of the rainfall gauges closest to the investigated catchment used in this study. All available rainfall gauges within the radar range of the radar station nearest to the investigated catchment were considered. In total 914 rainfall events were obtained using this approach. The duration of the events varies from 10 min to 12 h. The total precipitation of the events varies from 14.3 mm to 84.8 mm. This results in return periods from 10 years to more than 100 years for the given test catchment.

These rainfall events, with a temporal distribution of five minutes, were used as input to a hydrodynamic 1D-2D coupled model to generate the database consisting of 914 flood events. The hydrodynamic model that was used to generate the scenario database used in this work is Hystem-Extran 2D (itwh, 2017). The model combines a 1D pipe flow part with a 2D surface flow part. The surface flow model uses an explicit finite volume method on an unstructured triangular grid to solve the shallow-water-equations. Buildings are masked from the surface model and all rainfall from roofs is directed to the pipe network. The pipe network and the surface model are bidirectionally coupled. The study area of 5 km^2 represents a typical northern German urban catchment. A more detailed description of the catchment can be found in Rözer et al. (2021).

Flood maps for the 914 events are available with a temporal distribution of 5 min on a triangular grid of 1190276 cells. As in Berkhahn et al. (2019) we used a reduced spatial resolution with quadratic cells of 6 by 6 meters. For the given catchment the resulting number of quadratic cells is 186690. We divided the catchment into 4 subdomains, each with 198 x 252 cells, as shown in Fig. 1.

To train and test the model, we randomly divided the 914 events into three sets. One set of 215 events is used only for training the neural networks. The second set of 225 events is used to validate the training process and to select the hyperparameter set. The third set of 474 events is completely excluded from the training process and is only used to test the final model structure.

4. Hyperparameter tuning

In this section we show how the hyperparameters of the proposed model are tuned based on the test case described above. All parameters in an ANN structure that are not trained during training, but need to be considered to obtain good results, are called hyper-parameters. The list of hyperparameters includes for example parameters that define the complexity of the network. In the present study, the following hyperparameters were varied:

1. The number of filter layers in the autoencoders.
2. The number of filters in the first autoencoder layer.
3. The size of the latent space vector.
4. The length of the rainfall time series used as input.
5. The number of layers in the time series model (block A, B and C in Fig. 2).
6. The standard deviation used in the noise layers.

As all these hyperparameters are dependent on the specific application, a general statement on how to choose values with the best performance is not possible (Hutter et al., 2015).

4.1. Hyperparameters of the autoencoder part

The results of the analysis of different hyperparameters for the autoencoder are shown in Table 2. We performed the autoencoder training with the same hyperparameter settings for all sub-domains. As a performance measure, we used the mean RMSE with respect to the validation dataset. As we evaluated on scaled data, the RMSE is dimensionless. The order of the combinations in the table reflects the strategy for finding a good hyperparameter setting. We started with a reasonably small network and changed the bottleneck size from 100 to

Table 2
Hyperparameters analysis for the autoencoders with the number of filter layers n_f, the number of filters in the first layer b_f and the mean RMSE for all subdomains.

n_f	b_f	latent space vector length	mean RMSE [-]
3	16	100	0.0489
3	16	150	0.0472
3	16	200	0.0532
3	32	150	0.0418
3	64	150	0.0236
2	64	100	0.0237
2	64	150	0.0195
2	64	200	0.0219

150 respectively 200. As a bottleneck size of 150 gave the lowest mean RMSE, we fixed this number and tested a higher number of convolutional units in the first layer. Using 64 instead of 16 units halves the RMSE. As we want to keep the network as small as possible, we tested whether the performance remained high with a lower number of layers. With two layers instead of three, performance was 17 percent higher. As a final step, we double-checked the bottleneck size after changing the other hyperparameters by testing 100 and 200. After this procedure, the final hyperparameter setting for the map compression part is two layers with 64 convolutional units in the first layer and a bottleneck size of 150. With the explored hyperparameter set for the autoencoders, the reproduced flood maps generally show good agreement with the expected values. The unscaled mean RMSE for all maps in the validation set is 4.61 mm. The scatter plot in Fig. 4 shows the water levels reconstructed by the autoencoder part for all validation events. Each point represents a reconstructed water level on the y-axis over the expected water level on the x-axis. The high density on the diagonal indicates good overall performance. However, there are points where the autoencoders over- or underestimate.

4.2. Hyperparameters of the time series forecast part

The forcing input of the proposed model is a rainfall time series. In order to find out how many time steps of past rainfall are needed to make a good water level prediction, we performed a correlation analysis similar to Schmid and Leandro (2023). The results of the correlation analysis are shown in Fig. 5. Only events from the training dataset are used for the correlation analysis. For each time shift, we calculated the correlation coefficient between the time series of flood volume and rainfall for each event. We then calculate the mean absolute value of the correlation coefficients per time shift value. As shown in Fig. 5, the maximum correlation is reached at a time shift of 25 min. We therefore considered a time series length of 6 (current time step plus 5 past time steps) for the precipitation input to the forecast model.

For the time series forecast model, the number of layers and the number of neurons per layer must be defined. To simplify the process, we decided to keep the number of layers the same for each of the blocks shown in Fig. 2. For example, in Fig. 2 we used 2 layers for each block. We excluded the rain input processing path from the search and kept 2 layers constant for this path. The results of the parameter analyses are shown in Table 3. All results are given in relation to a reference topology. As a reference, we used a topology with 300 neurons per layer in blocks A and B and 600 neurons for block C. For each block, we used one fully coupled layer. We increased the number of neurons per layer until the relative error stopped decreasing. We then added layers for each number of neurons until the relative error stopped decreasing. We ended up with a topology with 1.75 times the number of reference neurons and four additional layers compared to the reference topology.

With the resulting topology shown above, the influence of different standard deviation values for the noise layers in Block A and B was tested. The standard deviation for block A was varied from 0 to 0.45. As written above, the standard deviation for block B is half that of block A. The results of this analysis, shown in Table 4, indicate an improvement in validation loss when using noise. The best improvement was achieved with a layer adding noise with a standard deviation of 0.30 for block A and 0.15 for block B.

5. Results

In this section we show the results for the test case described above. To test the predictive power of the model, we strictly excluded the test data examples from the hyperparameter tuning process and from training. All results shown for the final model include only events from the test data set and were therefore not previously seen by the model. We tested the proposed model with the hyperparameter settings shown above in a virtual operational mode as described in Section 2.6. In order to better assess the given validation metrics, we give an example result for one event at the beginning of this chapter. The prediction performance for this result, based on the validation metrics from Table 1with $RMSE_{30cm} = 0.15$ cm, number of cells with an negative NSE of 106, $CSI_{30cm} = 0.833$ and ROTAf$= 1.044$, is approximately in the middle range of all test events. A comparison of expected and predicted total flood volume is shown in Fig. 6. An example on how the flooding propagates in space and time is shown in Fig. 7 for both prediction and hydrodynamic model results.

5.1. Spatial error distribution

To evaluate the spatial error distribution, we used the NSE measure for each cell that exceeded a water level of 5 cm for at least one time step. The resulting map is shown in Fig. 8a. The map shows large contiguous areas with NSE values above 0.75 or even above 0.9. There

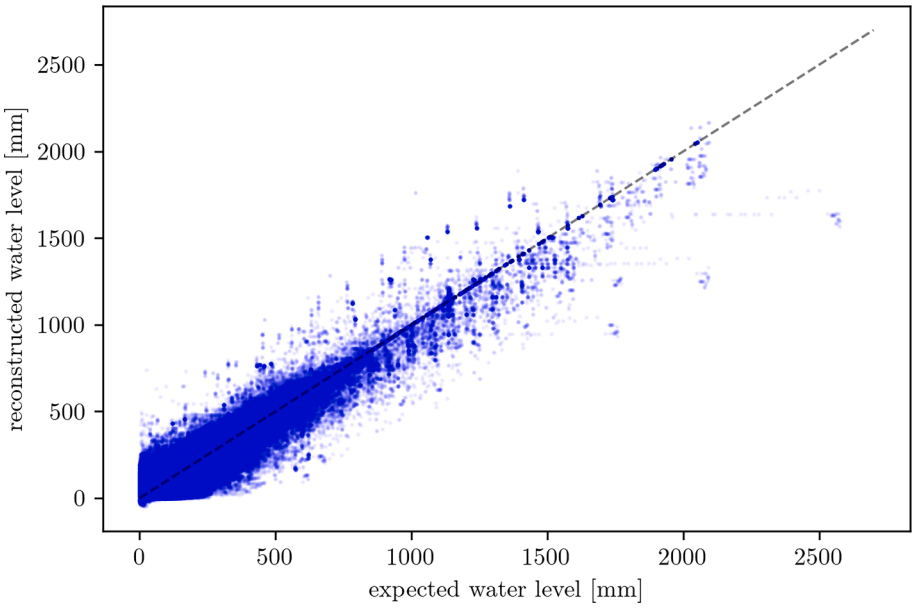


Fig. 4. Scatter plot of reconstructed water levels (output of autoencoders) over expected water levels.

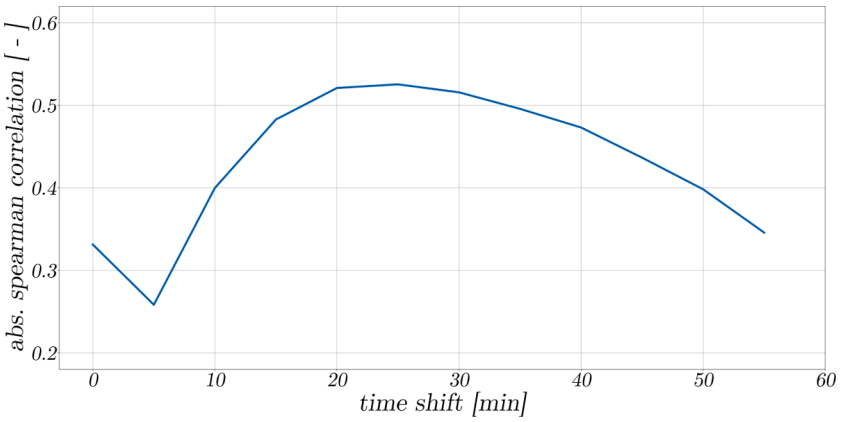


Fig. 5. Correlation analysis between rainfall and flood volume.

Table 3
Relative validation loss values for different topologies of the time series prediction model part with respect to a reference topology. Values lower than one are representing improvements over the reference.

		num. of neurons per layer in rel. to reference case							
		0.25	**0.50**	**0.75**	**1.00**	**1.25**	**1.50**	**1.75**	**2.00**
additional layers	**0**	1.18	1.09	1.02	1.00	0.99	0.98	0.97	0.97
	1	1.03	0.98	0.95	0.91	0.92	0.84	0.85	0.84
	2	0.95	0.85	0.82	0.80	0.76	0.74	0.73	0.72
	3	0.90	0.82	0.77	0.74	0.73	0.70	0.69	0.68
	4	0.90	0.82	0.75	0.73	0.70	0.68	**0.66**	0.66
	5	0.91	0.80	0.75	0.73	0.70	0.69	0.67	0.66
	6	-	0.81	-	-	-	-	-	-

Table 4
Relative validation loss compared to no noise variant for different standard deviations used in block A σ_A and block B σ_B of the time series prediction model part.

σ_A [-]	σ_B [-]	relative validation loss [%]
0.00	0.00	100
0.15	0.075	89.6
0.30	0.15	87.6
0.45	0.225	95.5

are also areas with negative NSE. An overview of how many hot spot cells are in each NSE category is given in Fig. 9. It is noteworthy that only 5.5 percent of these hot spot cells have an NSE lower than 0.5 and almost 48 percent have an NSE higher than 0.9. Low or negative NSE values can result from over- or underestimation of water levels. A time shift in the forecast can also lead to low or negative NSE values. For such an event, one would say that a forecast is good, as the patterns match, but show a time shift. However, the mismatch calculated by the NSE measure for some locations is large because the time shift persists throughout the event. In order to assess whether the negative NSE value is the result of a mismatch in water level height or in the timing of the peak, we calculated the time and water level difference for the peak at each location. The resulting maps are shown in Figs. 8b and 8c. The maps show that most cases of negative NSE values are due to a shift in the timing of the peak.

5.1.1. CSI

Fig. 10 shows the CSI_{30cm} values for all test events. The mean is 0.716 and the median is 0.738. Some events have very low CSI values, with a minimum of 0.143. Most of these low CSI values are due to the very small number of cells reaching the 30 cm threshold for these events.

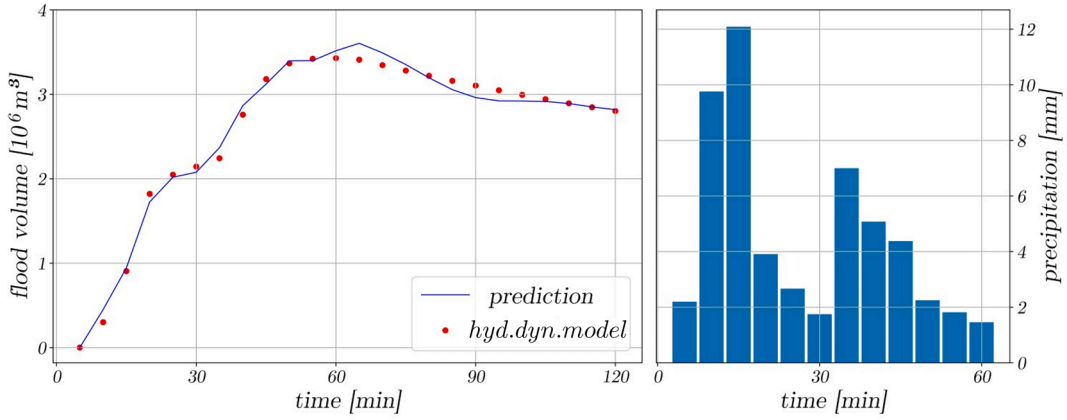


Fig. 6. left: flood volume time series of the ann prediction and hydrodynamic model results, right: precipitation timeseries of an event with avarage performance.

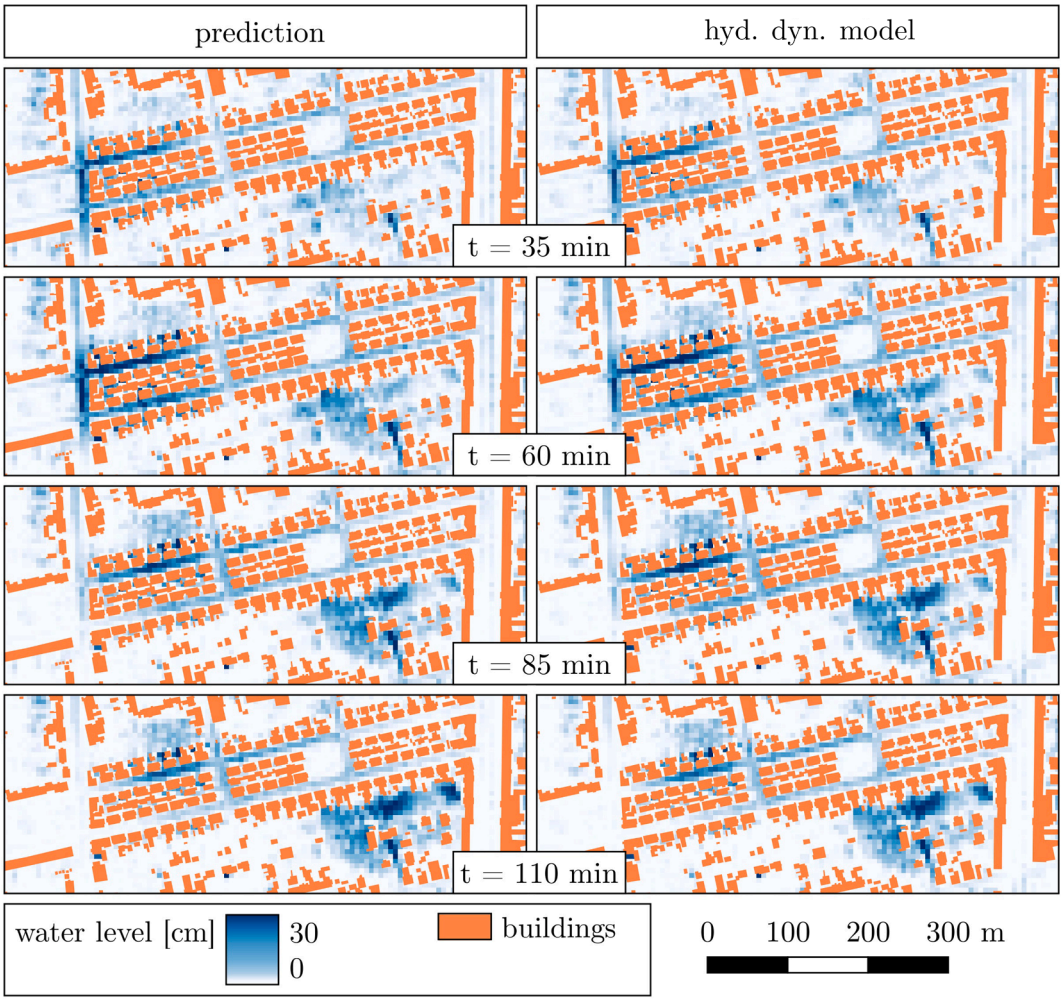


Fig. 7. Propagation of the flood in space and time for an event with average performance. Left: ANN prediction, right: hydrodynamic model results.

Therefore, a single missed or false alarm cell has a large impact on the CSI value. However, there are also events with small CSI values due to missed alarms. In general, we see more missed alarms than false alarms.

5.1.2. *Ratio of total area flooded*

The ROTAf (see Table 1) as an indicator of the extent of flooding was evaluated for all test events. The results showed an almost normal distribution with a mean of 0.956 and a median of 0.958. This indicates that the prediction model tends to slightly underestimate flooding, which is consistent with the CSI results.

5.2. *Overall prediction performance of the prediction model*

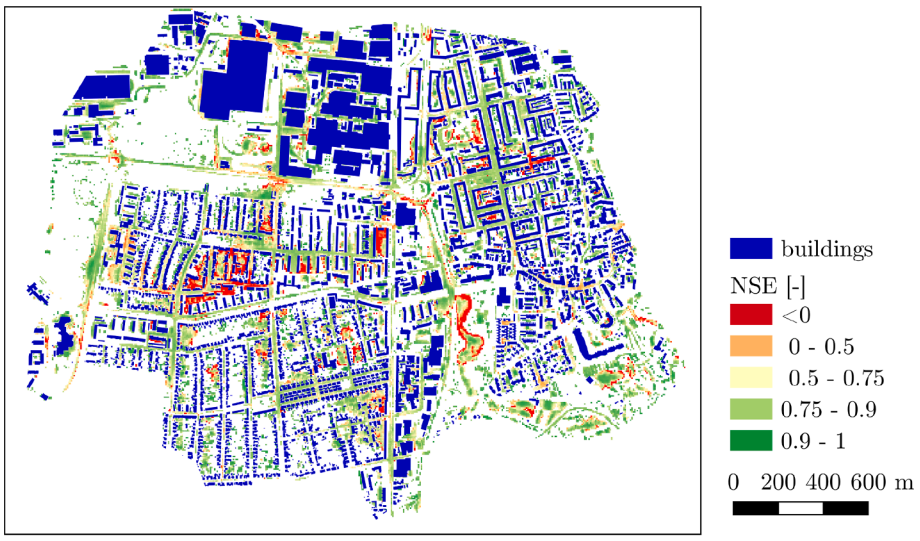
Evaluation of all test events with the validation metrics given in Table 1 shows worst results for events with total precipitation and event length on the edges of those included in the dataset. The worst prediction in terms of the validation measures used is also the event with the highest total precipitation in the test dataset. In Fig. 12 we show the time series of the predicted flood volume compared to the hydrodynamic model result and the time series of the precipitation used as input. The peak of the flood volume in the forecast is overestimated and delayed. After the peak, the flood volume is continuously slightly overestimated. However, the predicted start time of the flood and the increase in flood volume are close to the hydrodynamic model result. Fig. 13 shows the temporal and spatial distribution of the water depth for the same event to illustrate a prediction with poor performance. The predictions shown for time steps t = 135 min and t = 180 min are in very good agreement with the hydrodynamic model result. At time step t = 540 min, small areas of underestimated water depth and larger areas of overestimated water depth can be seen. The prediction does not show any unreasonable values, such as outstanding single cells.

To understand the relationship between the characteristics of the rainfall events used as input and the performance of the prediction model, the four scatter plots in Fig. 11 are used. Plotted is the $RMSE_5 cm$ for each event over (a) the duration of the event, (b) the rainfall peak, (c) the sum of the rainfall that occurs before the rainfall peak and (c) the sum of the rainfall that occurs after the rainfall peak. All values are scaled from 0 to 1 and are therefore dimensionless. For the relationship between event length and prediction performance and for the relationship between pre-peak rainfall and performance, no clear dependence can be shown. A tendency that the prediction performance with respect to the RMSE decreases with increasing rainfall peak can be shown in Fig. 11 (b). Also for increasing rainfall after the peak a decreasing prediction performance can be seen in Fig. 11(d).

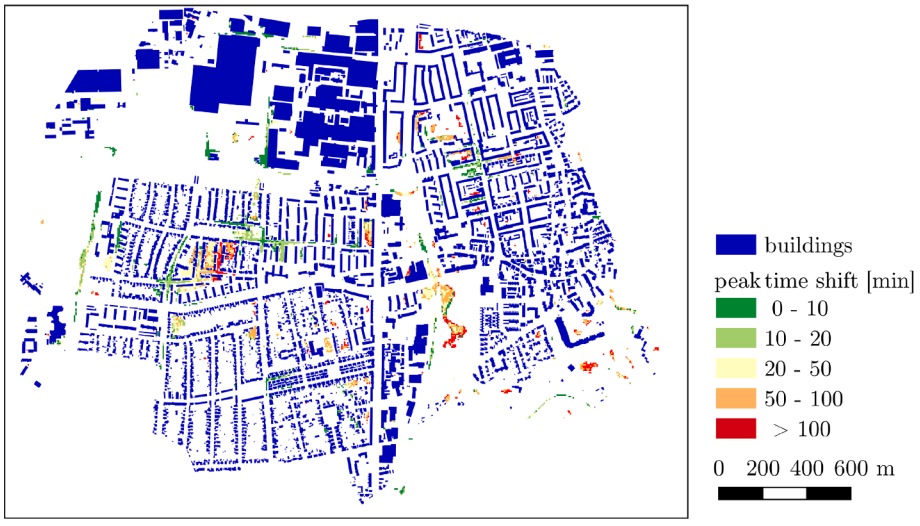
5.3. *Computation efforts*

The computation time for the 474 events from the test dataset with a total of 22292 time steps was 207153.99 minutes for the hydrodynamic model[1] and 79.13 min with the surrogate model. The computation time

[1] Note that the simulations were distributed across several different standard PCs.



(a) cellwise calculated NSE values



(b) mean time shift of the peak

(c) mean water level deviation at peak time

Fig. 8. Maps of cellwise calculated (a) NSE, (b) mean time shift and (c) mean water level deviation values for all test events with a threshold of 5 cm.

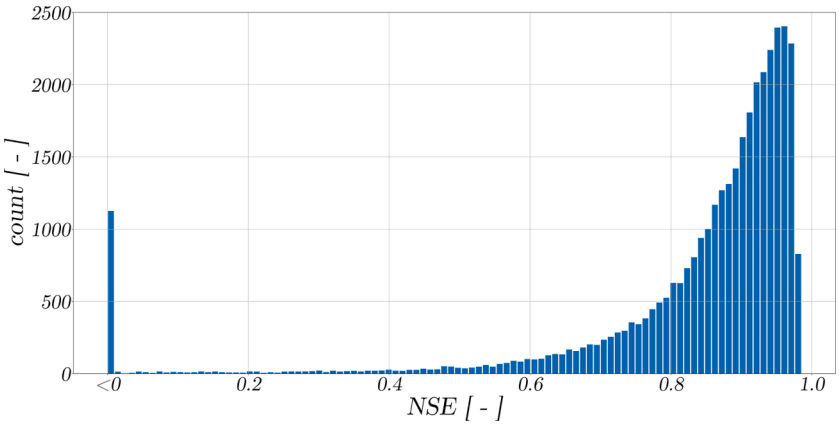


Fig. 9. Number of cells over NSE values for all test events with a threshold of 30 cm.

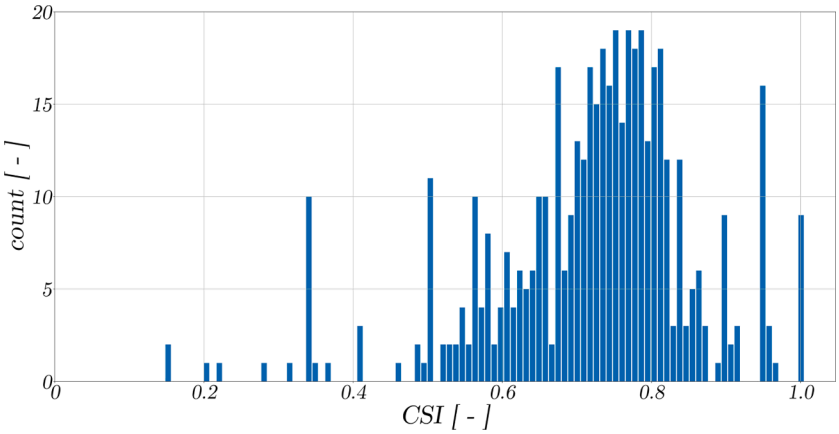


Fig. 10. Number of events over CSI_{30cm} value for all test events.

of the surrogate model includes the time series prediction part, the decompression part at the end of the prediction and the rescaling of each water level map. The resulting average computation time for a simulation time step (5 min) is 0.21 s. The speedup gaint by using the surrogate model is therefore about 2618. The training time for the surrogate model, including hyperparameter analyses, was in the range of 24 h.

6. Discussion and Outlook

The proposed model for predicting water depth time series shows reasonable results for all tested flood events. Overall, the validation metrics used in this work indicate a good performance of the presented model. With a CSI score of about 0.7 and a mean RMSE value of about 5 mm, the model shows better results compared to other studies, for example the recent study by Löwe et al. (2021). However, it is difficult to compare the results as the approaches are different. While we used natural rainfall events as in Löwe et al. (2021), we only considered predictions of water levels for fixed locations. Transferability of the model to other locations, as in Löwe et al. (2021) and Seleem et al., 2023, was not considered. In addition to predicting maximum water levels with spatial distribution, we also considered the temporal distribution of the flood event.

Results for some regions in the tested catchment had shown negative NSE values. Although by looking on the maps, many of these regions are local sinks, other local sinks are well predicted. The results showed that in most cases a negative NSE value is associated with a large time lag between the prediction and the hydrodynamic model result.

For the events tested, the approach used to divide the catchment into rectangular subdomains did not lead to related error patterns. There were also no sharp edges or obviously unreasonable results at the overlapping zones. This is in good agreement with the results for the maximum water level prediction model shown in Berkhahn et al. (2019).

From the investigation of the relationship between rain characteristics and prediction performance, no clear conclusion can be drawn regarding which processes cannot be accurately represented. Overall, the time series of inundation in the tested flood events were well captured by the prediction model for increasing water levels. However, the prediction model showed weaknesses in predicting the decreasing water levels after the flood peak in the catchment tested. With regard to the application of the prediction model in a real-time warning system, the increasing part of the flood event can be considered as the more important one. In the training database generated with the physically-based hydrodynamic model used in this study, evaporation and infiltration were not implemented. For urban catchments in the cool temperate climatic zone, the effect of evaporation is less important than that of infiltration. As shown in Tügel (2023), the infiltration can have a high influence on the flood course. As a result of neglecting infiltration, a complete recession of the flood volume is not simulated and cannot be represented in the prediction based on the ANN. However, for a continuous prediction, it should also be possible to represent the falling water levels in a meaningful way. A pragmatic approach in a real application might be to reset of all water levels to the dry state after a user defined time.

Events with return periods greater than 100 years are rare in the training data set and the prediction performance seems to partly reflect the selection of precipitation events in the training dataset. The results in our test case show that the prediction performance for these rare events

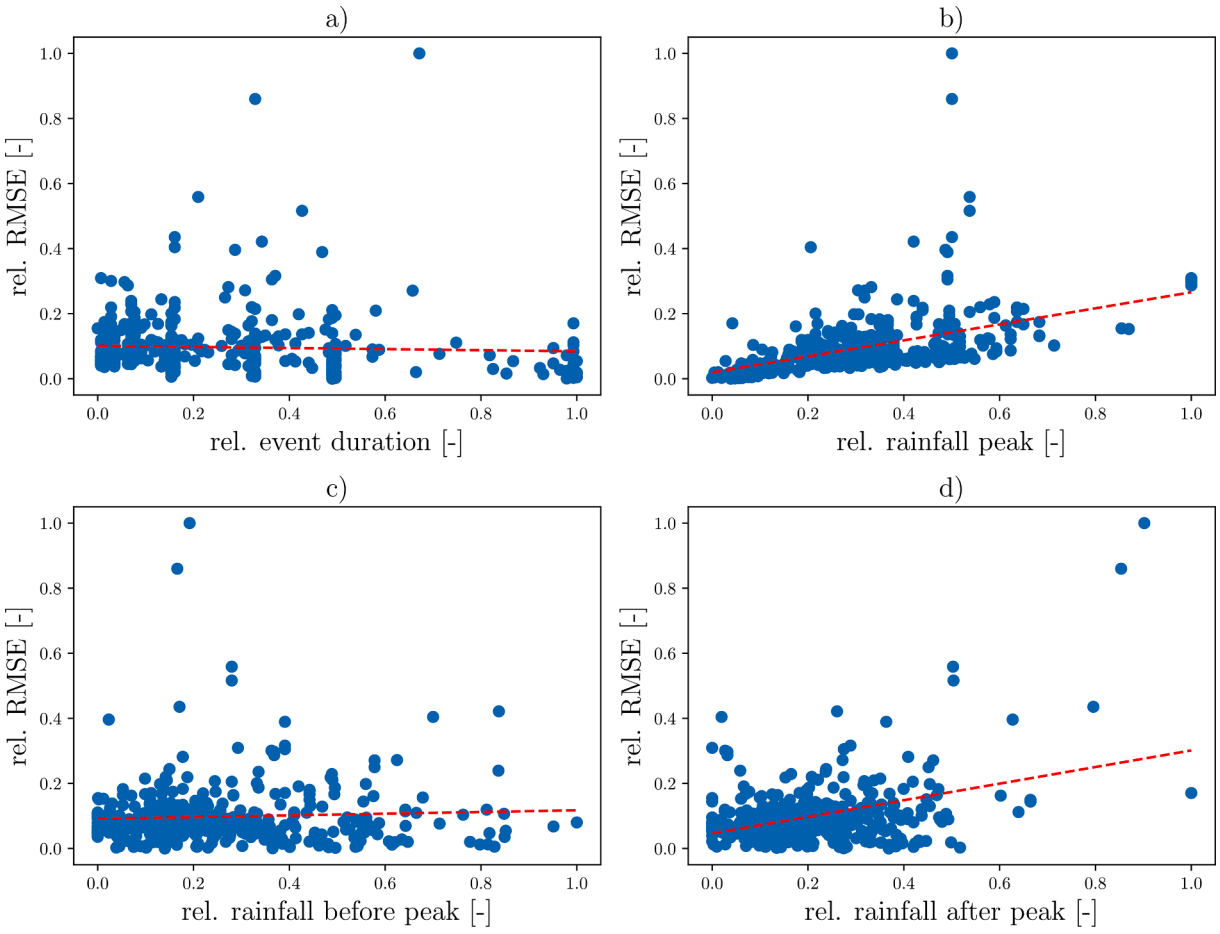


Fig. 11. Relationship between the different characteristics of the rainfall event and the performance of the prediction model for all test events.

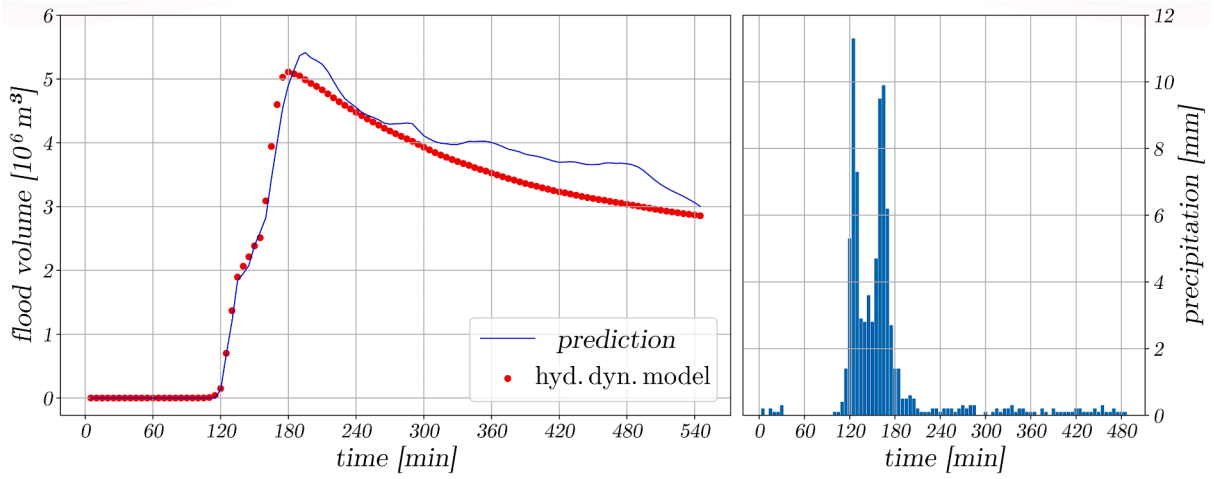


Fig. 12. left: flood volume time series of the ann prediction and hydrodynamic model result, right: precipitation timeseries of worst predicted event in the test data.

is worse than for events with return periods of around 30 years. This is not unexpected as ANNs are good at approximation but not at extrapolation (e.g. Martius and Lampert, 2016). This means that predictions in ranges that are not well covered by training data cannot be expected to work well. This needs to be considered when implementing the model for real-time predictions for operational emergency planning. The scaling of the data used can be used to determine whether the model is approximating or extrapolating. If the scaled value for a water level is greater than one, this is an indication of extrapolation. For possible operational planning of public transport and fire services, the mismatches of these events may not be a problem for two reasons. (1) The very extreme events will not occur often and (2) if it is clear by the scaling that the predictions for these events are unreliable, no wrong conclusions would be drawn. Reliable real-time forecasts of more frequent events are well covered by the proposed model. If predictions for events with return periods much longer than 100 years are required, more of these events need to be included in the training data set.

The rainfall events used in this study were taken from a database

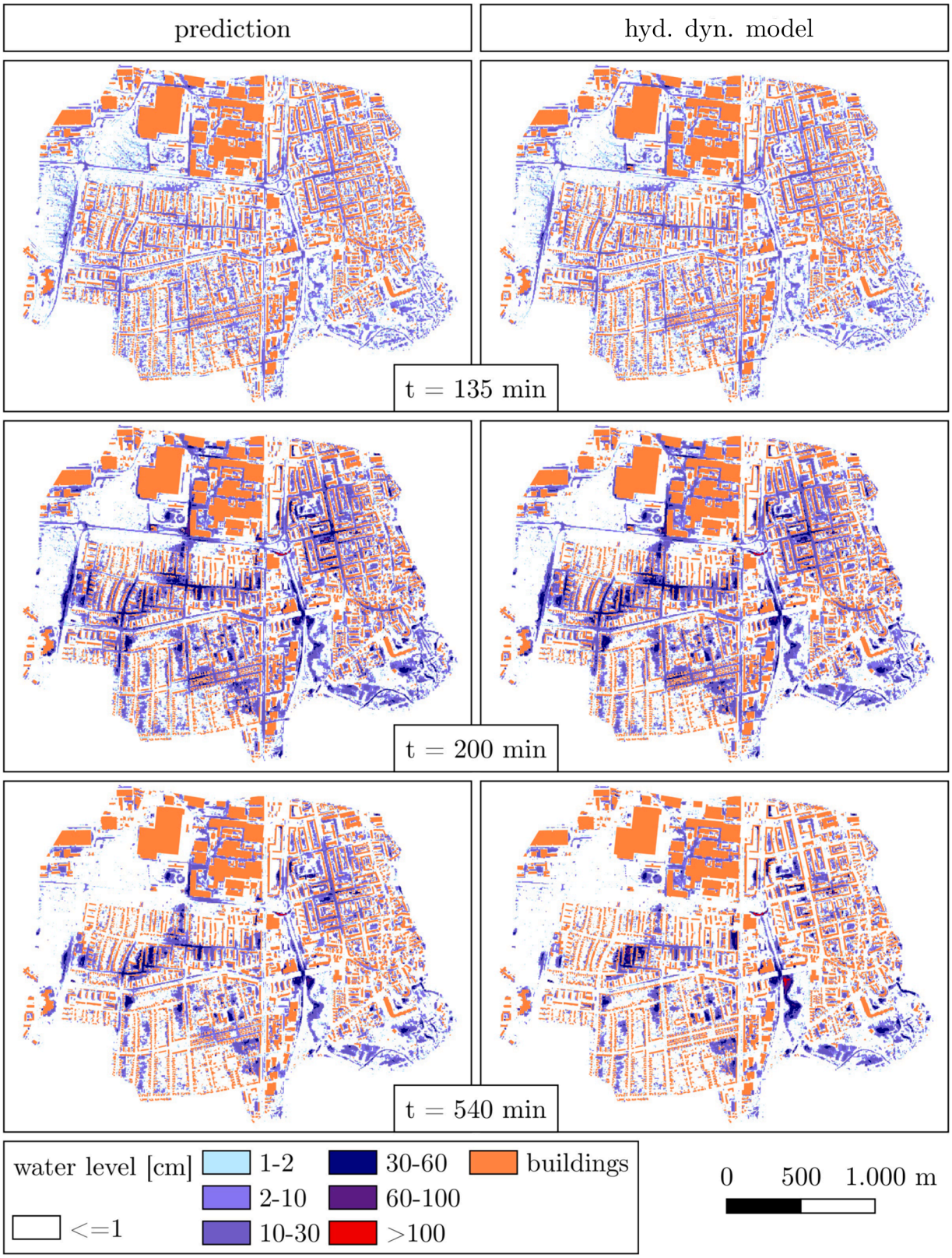


Fig. 13. Spatial and temporal distributed flood prediction for the worst predicted event in the test data.

generated from real gauge data and adjusted to match the catchment statistics. To overcome the problem of extrapolation discussed above, it might be beneficial to use a scenario database that does not reflect the statistics of the catchment, but rather the range that is supposed to be predicted. The choice of events for such a training database is not obvious. As shown in Sämann et al. (2022), it is challenging to find suitable metrics that relate different rainfall characteristics to flooding.

Uncertainty assessment is important for operational prediction models. Uncertainty in an urban flood prediction model can have many sources. Usually the most important source of uncertainty is the forcing of the flood prediction, i.e. the rainfall forecast. One way to deal with the uncertainty is to use ensemble forecasts. With the short computation time of the proposed model, it is feasible for application in ensemble approaches. The generation of rainfall forecast ensembles was not part of this study and the model was not tested for this purpose. However, it would be straightforward to implement in ensemble forecasting.

Verification of urban flood prediction models is a challenging task due to the lack of measurements. The hydrodynamic model used to generate the database for this study was tested on a real event in Rözer et al. (2021). The proposed model is trained to represent the water levels from the hydrodynamic model. In case of a discrepancy between reality and prediction, adjustments have to be considered for both the hydrodynamic model and the fast prediction model.

The short computation time achieved with the presented model comes with some limitations. As we have used a data-driven approach for predicting water levels, the whole data set has to be re-simulated and the model re-trained if something changes in the catchment. Physical constraints such as mass conservation are not implemented in the machine learning approach. The extent to which physical constraints are respected in the prediction of compressed water level maps is an open question.

The use of the last 25 min of rainfall as input in the present study is in good agreement with the response time of the catchment tested. Therefore, for small catchments, it may be plausible to relate the length of the rainfall time series used for prediction to the response time of the catchment in general. Further investigations should be done to test this hypothesis.

Also, the use of spatially distributed rainfall forecasts (e.g. for radar-based forecasts) as input was not considered in this study. For the small catchment tested in the present study, the assumption of spatially invariant rainfall, can be considered sufficiently accurate. When using the proposed model for larger catchments than in the present study, the spatiality of the rain input should be taken into account. In order to reduce the amount of training data required, some form of feature engineering is likely to be advantageous in the case of spatially distributed rain input (e.g. Ochoa-Rodriguez et al., 2015).

7. Conclusions

In this study, we presented a real-time prediction model for water levels in an urban catchment with spatial and temporal distribution. The performance of the model is promising in terms of computational time and low prediction error. We have also achieved the goal of usability on a well-equipped standard computer. While we used similar convolutional neural network approaches as in Gude et al. (2020) or Löwe et al. (2021), we added a temporal distribution to the flood prediction by implementing a recursive approach. With the ability to predict catchment-wide water levels with a temporal resolution of five minutes, the model has the potential to be used in operations control centres for fire services or public transport.

CRediT authorship contribution statement

Simon Berkhahn: Conceptualization, Data curation, Investigation, Methodology, Software, Writing - original draft. **Insa Neuweiler:** Conceptualization, Investigation, Methodology, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgements

We thank Felix Schmid[2] for discussions. Furthermore, we thank Johannes Steinhauer[3] for his help in creating the figures for this work. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions, which significantly improved the quality and clarity of this manuscript. The research is being conducted with financial support of the state of Lower Saxony and the BMU funded research project FURBAS (Entwicklung und Implementierung einer effizienten und nutzerfreundlichen Modellkette zur Frühwarnung von urbanen Sturzfluten in Hannover) [BMU, 67DAS224A].

References

Bentivoglio, R., Isufi, E., Jonkman, S.N., Taormina, R., 2022. Deep learning methods for flood mapping: a review of existing applications and future research directions. Hydrol. Earth Syst. Sci. 26, 4345–4378.

Bentivoglio, R., Isufi, E., Jonkman, S.N., Taormina, R., 2023. Rapid spatio-temporal flood modelling via hydraulics-based graph neural networks. EGUsphere 1–24.

Berkhahn, S., Fuchs, L., Neuweiler, I., 2019. An ensemble neural network model for real-time prediction of urban floods. J. Hydrol. 575, 743–754.

Burrichter, B., Hofmann, J., Koltermann da Silva, J., Niemann, A., Quirmbach, M., 2023. A spatiotemporal deep learning approach for urban pluvial flood forecasting with multi-source data. Water 15, 1760.

Chang, F.-J., Chen, P.-A., Lu, Y.-R., Huang, E., Chang, K.-Y., 2014. Real-time multi-step-ahead water level forecasting by recurrent neural networks for urban flood control. J. Hydrol. 517, 836–846.

Dumoulin, V., & Visin, F. (2016). A guide to convolution arithmetic for deep learning. arXiv preprint arXiv:1603.07285,.

Gaál, L., Szolgay, J., Kohnová, S., Hlavčová, K., Parajka, J., Viglione, A., Merz, R., Blöschl, G., 2015. Dependence between flood peaks and volumes: a case study on climate and hydrological controls. Hydrol. Sci. J. 60, 968–984.

Gude, V., Corns, S., Long, S., 2020. Flood prediction and uncertainty estimation using deep learning. Water 12, 884.

Guo, Z., Moosavi, V., Leitão, J.P., 2022. Data-driven rapid flood prediction mapping with catchment generalizability. J. Hydrol. 609, 127726.

Henonin, J., Russo, B., Mark, O., Gourbesville, P., 2013. Real-time urban flood forecasting and modelling–a state of the art. J. Hydroinformatics 15, 717–736.

Hermansah, Rosadi, D., Abdurakhman, Utami, H., & Darmawan, G. (2022). Differencing effect in series-parallel architecture of narx model for time series forecasting. In 7TH INTERNATIONAL CONFERENCE ON MATHEMATICS: PURE, APPLIED AND COMPUTATION: Mathematics of Quantum Computing AIP Conference Proceedings (p. 030022). AIP Publishing.

Hofmann, J., Schüttrumpf, H., 2021. floodgan: Using deep adversarial learning to predict pluvial flooding in real time. Water 13, 2255.

Hutter, F., Lücke, J., Schmidt-Thieme, L., 2015. Beyond manual tuning of hyperparameters. KI-Künstliche Intelligenz 29, 329–337.

IPCC, Masson-Delmotte, V., Zhai, P., Pirani, A., Connors, S., Péan, C., Berger, S., Caud, N., Chen, Y., Goldfarb, L., Gomis, M., Huang, M., Leitzell, K., Lonnoy, E., Matthews, J., Maycock, T., Waterfield, T., Yelekçi, O., Yu, R., & B. Zhou (eds.) (2021). Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change. Cambridge University Press, In Press.

itwh (2017). HYSTEM-EXTRAN 2D Modellbeschreibung (HYSTEM-EXTRAN 2D model description). Technical Report Institut für technisch-wissenschaftliche Hydrologie GmbH, Hannover.

Jamali, B., Bach, P.M., Cunningham, L., Deletic, A., 2019. A cellular automata fast flood evaluation (ca-ffé) model. Water Resour. Res. 55, 4936–4953.

Kingma, D.P., Ba, J., 2017. Adam: A Method for Stochastic Optimization.

[2] Research assistant at Department of Civil Engineering, Chair of Hydromechanics and Hydraulic Engineering, University of Siegen, Siegen, Germany

[3] Student assistant at the Institute of Fluid Dynamics and Environmental Physics in Civil Engineering, Leibniz Universität Hannover, Hanover, Germany

LANUV, L. (2022). Hochwasserschutz.
Leandro, J., Djordjević, S., Chen, A., Savić, D., Stanić, M., 2011. Calibration of a 1d/1d urban flood model using 1d/2d model results in the absence of field data. Water Sci. Technol. 64, 1016–1024.
Li, W., Li, Y., Gong, J., Feng, Q., Zhou, J., Sun, J., Shi, C., Hu, W., 2021. Urban water extraction with uav high-resolution remote sensing data based on an improved u-net model. Remote Sensing 13, 3165.
Lin, Q., Leandro, J., Gerber, S., Disse, M., 2020a. Multistep flood inundation forecasts with resilient backpropagation neural networks: Kulmbach case study. Water 12, 3568.
Lin, Q., Leandro, J., Wu, W., Bhola, P., & Disse, M. (2020b). Prediction of maximum flood inundation extents with resilient backpropagation neural network: case study of kulmbach. Frontiers in Earth Science, (p. 332).
Löwe, R., Böhm, J., Jensen, D.G., Leandro, J., Rasmussen, S.H., 2021. U-flood–topographic deep learning for predicting urban pluvial flood water depth. J. Hydrol. 603, 126898.
Martius, G., & Lampert, C.H. (2016). Extrapolation and learning equations. arXiv preprint arXiv:1610.02995.
Nguyen, D.H., Kim, J.-B., Bae, D.-H., 2021. Improving radar-based rainfall forecasts by long short-term memory network in urban basins. Water 13, 776.
Ochoa-Rodriguez, S., Wang, L.-P., Gires, A., Pina, R.D., Reinoso-Rondinel, R., Bruni, G., Ichiba, A., Gaitan, S., Cristiano, E., van Assel, J., Kroll, S., Murlà-Tuyls, D., Tisserand, B., Schertzer, D., Tchiguirinskaia, I., Onof, C., Willems, P., & ten Veldhuis, M.-C. (2015). Impact of spatial and temporal resolution of rainfall inputs on urban hydrodynamic modelling outputs: A multi-catchment investigation. Journal of Hydrology, 531, 389–407. Hydrologic Applications of Weather Radar.
Rözer, V., Peche, A., Berkhahn, S., Feng, Y., Fuchs, L., Graf, T., Haberlandt, U., Kreibich, H., Sämann, R., Sester, M., et al., 2021. Impact-based forecasting for pluvial floods. Earth's. Future 9, 2020EF001851.
Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747,.
Sämann, R., Graf, T., Neuweiler, I., 2022. Performance of nearest neighbour metrics for pluvial flood nowcasts in urban catchments. J. Hydrol. 604, 127225.
Schmid, F., Leandro, J., 2023. An ensemble data-driven approach for incorporating uncertainty in the forecasting of stormwater sewer surcharge. Urban Water J. 1–17.
Seleem, O., Ayzel, G., Bronstert, A., Heistermann, M., 2023. Transferability of data-driven models to predict urban pluvial flood water depth in berlin, germany. Natural Hazards Earth System Sci. 23, 809–822.
Sun, W., Bocchini, P., Davison, B.D., 2020. Applications of artificial intelligence for disaster management. Nat. Hazards 103, 2631–2689.
Sundararajan, K., Garg, L., Srinivasan, K., Bashir, A.K., Kaliappan, J., Ganapathy, G.P., Selvaraj, S.K., Meena, T., 2021. A contemporary review on drought modeling using machine learning approaches. CMES-Computer Modeling Eng. Sci. 128, 447–487.
Tügel, F. (2023). Flash flood modeling with a specific focus on arid regions and infiltration.
Xu, T., Liang, F., 2021. Machine learning for hydrologic sciences: An introductory overview. Wiley Interdisciplinary Reviews: Water 8, e1533.
Zanchetta, A.D., Coulibaly, P., 2020. Recent advances in real-time pluvial flash flood forecasting. Water 12, 570.