



Research papers

Real time probabilistic inundation forecasts using a LSTM neural network

Fedde J. Hop^{a,b,*}, Ralf Linneman^b, Bram Schnitzler^b, Anouk Bomers^a, Martijn J. Booij^a^a University of Twente, Faculty of Engineering Technology, Water Engineering and Management, Enschede, the Netherlands^b HydroLogic, Amersfoort, the Netherlands

ARTICLE INFO

This manuscript was handled by Corrado Corradini, Editor-in-Chief, with the assistance of Saket Pande, Associate Editor

Keywords:

Probabilistic inundation forecast
LSTM neural network
Pluvial flood
Real time forecasting
Deep learning

ABSTRACT

Probabilistic inundation forecasts that consider the uncertainty in rainfall predictions are crucial for flood early warning systems to effectively manage and reduce potential risks posed by pluvial flood events. Timely generation of such forecasts is challenging with physically-based numerical models due to computational demands. In contrast, data-driven models have a relatively low computational cost and can generate results quickly, making them a promising alternative to overcome this issue. This study proposes a long short-term memory (LSTM) neural network that can predict inundation progression over time at a high spatial resolution. The network is trained on 1600 hydraulic simulations conducted using a 1D2D hydraulic model. With the trained network, probabilistic inundation forecasts are generated by combining the deterministic inundation predictions of 50 ensemble members of the rainfall forecast. The model is successfully tested for temporally varying rainfall events in a rural study area, and can generate accurate probabilistic inundation forecasts within seconds.

1. Introduction

Pluvial flooding occurs when precipitation intensity exceeds the capacity of natural and engineered drainage systems. It is expected to increase in frequency, severity and impact through the 21st century due to the combined effects of climate change and urbanization (Rosenzweig et al., 2018; Trenberth, 2011). Research in recent years has demonstrated an increasing interest in probabilistic pluvial flood inundation estimation, using ensembles of inputs to account for uncertainty (Cloke et al., 2013; Gomez et al., 2019; Wu et al., 2020).

For pluvial flooding the rainfall has a large influence on the inundation (Wu et al., 2020). Forecasting rainfall is challenging due to the chaotic nature of the atmosphere, leading to large uncertainties in rainfall forecasts. To include these uncertainties in the inundation forecasts, probabilistic inundation forecasts can be performed (Zarzar et al., 2018). In a probabilistic inundation forecast an ensemble of rainfall forecasts is considered, and for each ensemble member a deterministic inundation prediction is made. The deterministic inundation predictions for each ensemble member are then combined to create a probabilistic inundation forecast.

Numerical simulations play a key role in predicting inundation due to heavy rainfall (Goodarzi et al., 2019; Mei et al., 2020). Many studies

have shown the effectiveness of physically-based hydraulic models in predicting inundation depths (Seyoum et al., 2012; Shahapure et al., 2010). To make inundation forecasts, models numerically solve a set of partial differential equations to simulate the flow of water (Brutsaert, 2006). However, high levels of detail in numerical simulations come with large computational costs, and these increase as the demand for accuracy increases (Wang et al., 2019). For this reason, operational inundation forecasting has yet to become the standard (Wu et al., 2020). Operational probabilistic inundation forecasts provide an even greater challenge since they consist of a combination of many deterministic numerical simulations, resulting in even higher computational costs.

A commonly used method to reduce computational costs is surrogate modelling, which provides a second-level abstraction from the original system. Response surface surrogate models, such as machine learning (ML) algorithms, are data driven models trained based on the input–output relations of a physically based model or field measurements (Kilsdonk et al., 2022). Once trained, ML models can predict the output for unseen input data at low computational costs. While there are many types of ML techniques available, in recent years, the application of artificial neural networks (ANNs) has emerged as a promising tool for speeding up inundation modeling due to their ability to learn complex nonlinear patterns (Bentivoglio et al., 2022).

Abbreviations: ML, Machine Learning; ANN, Artificial Neural Network; LSTM, Long-Short Term Memory; RR, Rainfall Runoff; MSE, Mean Squared Error; MAE, Mean Absolute Error; RMSE, Root Mean Squared Error; NSE, Nash–Sutcliffe model Efficiency coefficient.

* Corresponding author at: University of Twente, Faculty of Engineering Technology, Water Engineering and Management, Enschede, the Netherlands.

E-mail addresses: fedde.hop@hydrologic.com (F.J. Hop), a.bomers@utwente.nl (A. Bomers), m.j.booij@utwente.nl (M.J. Booij).

<https://doi.org/10.1016/j.jhydrol.2024.131082>

Received 8 June 2023; Received in revised form 31 December 2023; Accepted 29 February 2024

Available online 22 March 2024

0022-1694/© 2024 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

ANNs have been applied to pluvial flooding in order to predict maximum inundation depths (Berkhahn et al., 2019; Guo et al., 2021) and inundation depths at multiple time steps (Chang et al., 2018, 2014; Kao et al., 2021; Kilsdonk et al., 2022). ANNs have also been applied for predicting fluvial floods (Kabir et al., 2020; Lin et al., 2020; Zanchetta and Coulibaly, 2022; Zhou et al., 2021). However, in case flood extents over time are predicted, typically only a relatively small dataset (in the order of 10–30) is used to train the neural network (Chu et al., 2020; Kabir et al., 2020). The literature lacks a systematic evaluation of the effect of the size of the training dataset on the performance of the neural network. Furthermore, despite the significant uncertainty in deterministic inundation predictions due to factors such as uncertain rainfall forecasts, ANNs are commonly applied to generate deterministic inundation maps while probabilistic inundation forecasts are of great importance. In this research, a LSTM neural network is applied to predict inundation depths over time at a high spatial resolution. LSTM networks have successfully been applied to predict temporal pluvial flood behaviour (Fang et al., 2021; Kilsdonk et al., 2022; McSpadden et al., 2024) and represent the most popular recurrent neural network for flood mapping applications (Bentivoglio et al., 2022). A LSTM network is typically used to model sequential data that exhibits short- and long-term dependencies. The objective is to accurately predict the inundation probabilities over time for the entire study area. The LSTM network in this study uses rainfall per hour as an input, and predicts inundation for each hour. This is essential for making accurate probabilistic inundation forecasts, since different ensemble members of the rainfall forecast can feature different distributions of rainfall over time. Due to the insufficient availability of observational inundation records, the LSTM network is trained on hydraulic simulations.

2. Materials and methods

2.1. Study area

The study area is polder de Tol (Fig. 1), located north-west of Utrecht in the Netherlands. The area encompasses approximately 12.5 square kilometers and is predominantly characterized by rural landscapes featuring meadows, a high density of ditches, and soil composed primarily of peat. Polder de Tol is mainly used for agricultural purposes. In the north-west some urban area is present, which is the village of Kockengen. In total, around 3,200 inhabitants live in the study area. The area is representative for polders in the west and north of the Netherlands. In terms of water management, the study area is a relatively closed system that runs off to pumping station de Tol. The area experiences little influences by processes outside of the model domain. This is helpful when validating the numerical model since most of the water that leaves the study area discharges through the pumping station, for which discharge data is available. There are inlets present in the study area, however these only let in water during periods of low rainfall, which is not relevant when there is a risk of inundation.

2.2. Numerical model

For this study, a 1D2D hydraulic model with a coupled rainfall runoff module is developed with the Delft3D modelling software (Deltares, 2023). The rainfall-runoff (RR) module is used for simulating the rainfall-runoff processes (Prinsen et al., 2010). The entire area is represented by 16 sub-catchments. Each sub-catchment can be divided into four nodes: paved, unpaved, open water and greenhouses. Within each node the water balance is calculated. The processes considered are: precipitation, evaporation, drainage, surface runoff, infiltration and seepage. Rainfall is based on external forcing data, which can be both historical data or synthetic events. The soil type is assumed to be peat for

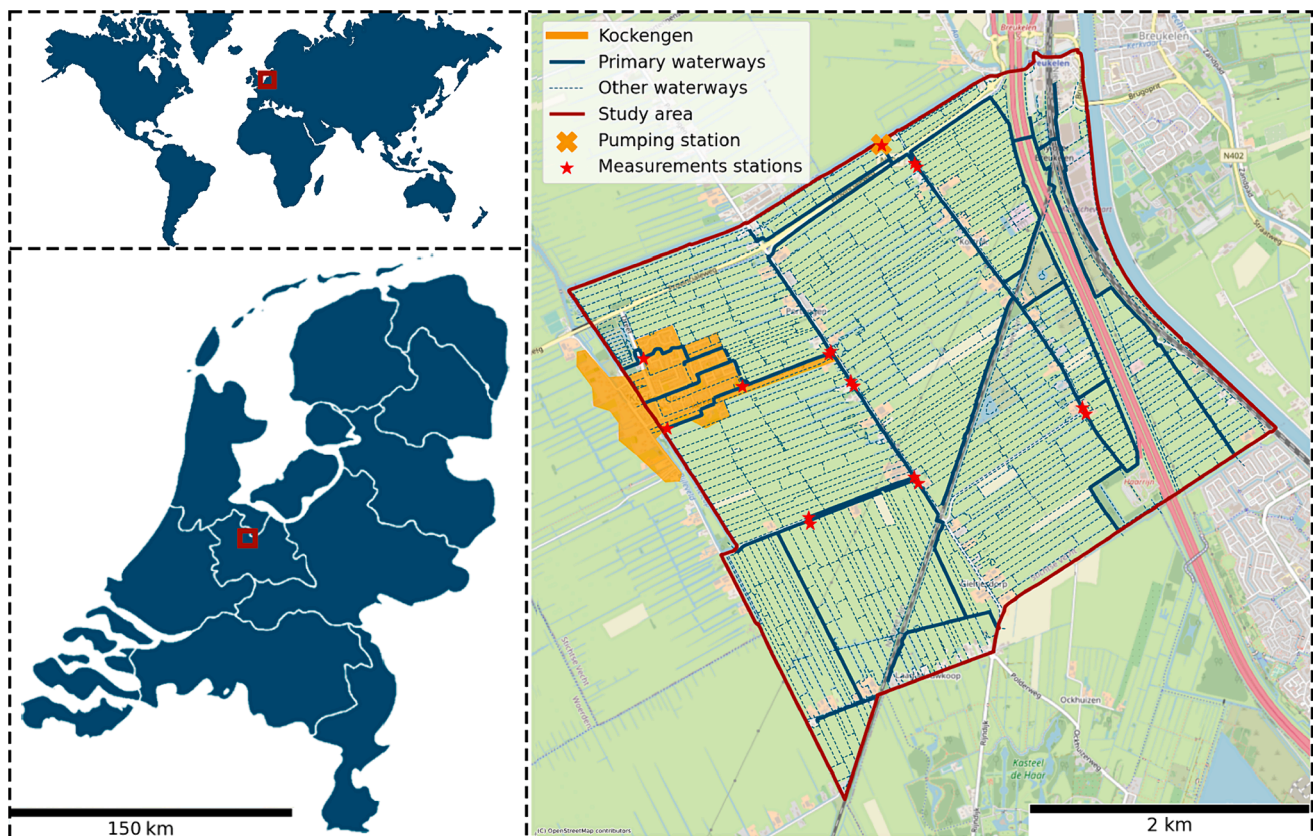


Fig. 1. Location of the study area, the waterways, the measurement stations and hydraulic structures in the study area.

the entire study area. Each sub-catchment exchanges water with the 1D component of the model.

The 1D component of the model simulates channel flow in the study area, including culverts, weirs, dams, inlets, and the pumping station (see Fig. 1). Computational nodes are positioned at intervals of 20 m, and are also placed upstream and downstream of a hydraulic structure. For the developed hydraulic model there is the assumption that the pumping station has full discharge capacity available at all times. However, this may not be realistic during extreme rainfall events when downstream water levels can increase significantly, potentially requiring the pump to shut down.

The 2D model is used to simulate overland flow where there is no predetermined flow path. A structured rectangular grid with a resolution of 10 m is used, resulting in a total of 123,993 grid cells. The roughness is dependent on the land use per grid cell according to relations between land use and Manning roughness (Papaioannou et al., 2018). The 1D and 2D model can exchange water at 1D2D embedded links (Deltares, 2023). At every computational node of a 1D waterway, a link is placed between the 1D waterway and the nearest 2D grid cell.

To calibrate and validate the model, five heavy precipitation events between 2010 and 2022 are selected. During these events, the discharge through the pumping station and water levels at 16 locations were measured at 15 min intervals. Four of the selected events are used for the calibration, and one for validation. The model is calibrated with a Monte Carlo simulation (Rientjes et al., 2013). A total number of 2470 simulations is carried out to calibrate six uncertain model parameters. These six parameters and their values can be found in Table 1. Within the calibration there are three objectives:

1. Maximizing the Nash–Sutcliffe model efficiency (NSE) coefficient of the measured pumping station discharge and the modelled pumping station discharge (weight = 0.25).
2. Minimizing the mean absolute error (MAE) of the measured pumping station discharge and the modelled pumping station discharge (weight = 0.25).
3. Minimizing the MAE of the measured and modelled water levels at various locations in the study area (weight = 0.5).

The selection of MAE was driven by its effectiveness in evaluating the overall fit of a model, while NSE was chosen for its ability to assess the performance of hydrologic forecasts (Jackson et al., 2019). The resulting NSE and MAE values are normalized between 0 and 1, and then the score is calculated by summing these normalized values according to the weight of the respective objective. The parameter set with the highest combined score is used in this study.

The calibrated 1D part of the hydraulic model is validated by comparing the measured and modelled discharge through the pumping station, as well as the water levels within the waterways at various locations. The discharge through the pumping station is predicted accurately with an NSE value of 0.84 on the validation event, and an average NSE of 0.77 for the calibration events. The water levels in the waterways

Table 1
Resulting values for model parameters after calibration.

Parameter	Unit	Calibration range	Calibrated value
Channel width*	m	1.5–4	2.75
Channel height*	m	1–1.5	1.2
Channel roughness	$\frac{1}{m^3}$	8–40	12.6
Drainage resistances**	–	0.3–2	1.57
Land storage	mm	5–20	18
Surface runoff resistance	day	0.3–2	1.21

*Channel width and height for all non-primary waterways in Fig. 1.

**Multiplication factor for values of 30, 200 and 10000 days for soil layer 1, 2 and 3 respectively.

are compared to measurements at different locations in the study area. The average mean absolute error for these locations is 6 cm for the validation event, and on average 7 cm for the calibration events. Even though no data was available to validate the 2D part of the model, the validation of the 1D part gives confidence in the accuracy of the model. Furthermore, many studies showed the applicability of a 1D-2D coupled model (e.g. Domeneghetti et al., 2013; Liu et al., 2015) and of the shallow water equations (e.g. Coulibaly et al., 2020; García-Navarro et al., 2019) for flood modelling purposes. Therefore, it is reasonable to assume that this 1D-2D coupled model is also capable of simulating flood extents as a result of overflow with sufficient accuracy.

2.3. Training data

The calibrated hydraulic model is used to simulate the inundation for 2,000 different rainfall events. To expedite the generation of training data, cloud computing is employed for all simulations. Each simulation produces inundation depths for all grid cells at twelve time steps with one-hour intervals. It is crucial to maintain a balanced training dataset to prevent the network from exhibiting a bias towards a certain type of event. Additionally, using a neural network for extrapolation beyond the extent of the training data is not reliable (Sajikumar and Thandaveswara, 1999). Therefore, the training dataset needs to contain heavier rainfall events than those expected in the near future.

Historically, there has been a prevalence of events with light to moderate rainfall, and fewer instances of very heavy rainfall. To address this imbalance, more heavy rainfall events are generated by scaling historic events. An outline of the steps involved is detailed below and illustrated in Fig. 2.

1. **Data Selection:** Utilize rainfall data recorded at the Bilt (15 km from study area) between 1906 and 2020 as the foundation for generating training events.
2. **Climate Adjustment:** Adjust the historical dataset to reflect the expected conditions in 2050 under a heavy climate change scenario (Attema et al., 2014).
3. **Event Identification:** Identify instances within the adjusted dataset where the recorded rainfall exceeds 50 mm over a 12-hour period.
4. **Total Rainfall List Creation:** Create a list of total rainfall amounts per event. This list is uniformly distributed and ranges from 0 to 180 mm.
5. **Random Event Selection:** For each simulation in the training dataset, randomly select an event from the pool identified in step 3.
6. **Temporal Shift:** Introduce a random temporal shift to the selected event. This introduces variability in the timing of peak rainfall.
7. **Hourly Rainfall Scaling:** Scale the hourly rainfall rates proportionally based on the desired total precipitation for the selected training event. The scaling factors are determined by the total rainfall values from the list created in step 4.
8. **Simulate:** Perform a simulation with the calibrated hydraulic model. All input parameters other than the rainfall remain constant. Each simulation serves as a sample in the training dataset.

The samples (simulations) are then split into training, testing and validation datasets. The training data is used to train the neural network, while the validation data is used to assess the network's performance during training and hyperparameter optimization. Finally, the testing data is used to evaluate the network's performance in the final analysis. The datasets are split using a stratified approach, with 80 % for training, 10 % for validation and 10 % for testing (Zanchetta and Coulibaly, 2022; Zhou et al., 2021). This results in 1600 simulations for training, 200 simulations for validation, and 200 simulations for testing. All simulations are sorted based on total input rainfall. Testing and validation samples are then picked at fixed intervals, such that the testing and validation sets are not biased towards samples of higher or lower rainfall. This also ensures that the events with the highest and lowest total

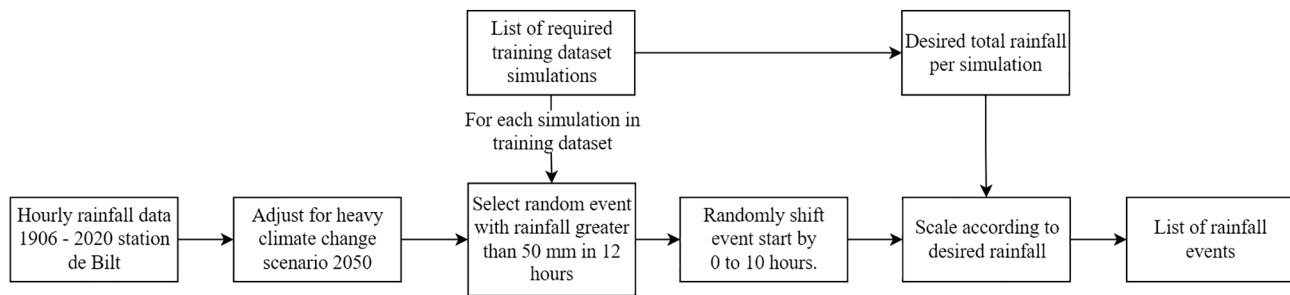


Fig. 2. Illustration of the steps taken to generate the input rainfall data for generating training data.

rainfall are not used for testing and validation, since the neural network's ability to extrapolate beyond the training data cannot be guaranteed (Sajikumar and Thandaveswara, 1999). The testing dataset used to generate the results of this study contains 200 samples with rainfall ranging between 7.2 mm and 156.4 mm in twelve hours.

Before the output of the hydraulic model can be used for the neural network, it first has to be preprocessed. Neural networks perform best when the input data is normalized between -1 and 1 (Rafiq et al., 2001). Therefore, both the input and output data are normalized. Normalization is done between 0 and 1 , since both the input (rainfall) and output (water depth) can physically not be below 0 . This will prove to be useful in section 2.4.

Normalization of the rainfall is done by scaling linearly based on the maximum value. For the inundation depths, a similar scaling procedure is used. However, there are a few cells that have significantly higher inundation depths than almost all other cells. If these "outliers" in inundation depth are used as normalization boundaries, this would result in the majority of data not being normalized between 0 and 1 , but between 0 and 0.4 . It was found that excluding outliers in determining the normalization boundaries improves the accuracy of the inundation predictions. The network was tested with different values for the outlier exclusion (ranging from 0.1% until $10^{-8}\%$). Excluding the largest 0.00001% when determining the normalization boundary of the inundation depths resulted in the smallest mean squared error on the validation dataset. So the smallest 99.99999% of inundation depths is used to determine the normalization boundaries.

The hydraulic model outputs inundation depth per grid cell for each time step. In the study area, certain grid cells remain dry during all simulations. For these specific locations, a neural network prediction of inundation is not required, as these cells are never inundated in the training data. For this reason, grid cells that are never inundated are removed from the training, testing and validation data, and will not be further considered inside the neural network. This reduces the number of outputs of the neural network from $123,993$ to $53,381$ for this model. By having fewer outputs, the network will have to train less parameters, which makes training faster and less memory-intensive.

2.4. Network architecture

Recurrent neural networks (RNN) have mostly been employed to model temporally varying floods, where they can exploit their sequential inductive bias. However, the basic recurrent layer suffers from the problem of vanishing and exploding gradients (Hochreiter and Schmidhuber, 1997). This problem is solved via the use of Long-Short Term Memory (LSTM) layers (Hochreiter and Schmidhuber, 1997), which is also the most popular RNN layer for flood mapping applications (Bentivoglio et al., 2022). Because of their sequential inductive bias and previous success in flood mapping (Kao et al., 2021; Kilsdonk et al., 2022; Zhou et al., 2021), this study uses a LSTM network to predict inundation depths for every grid cell in the hydraulic model.

Uniform rainfall over the area is assumed because the study area is relatively small compared to the spatial resolution of the probabilistic

rainfall forecasts ($7.5\text{ km} \times 7.5\text{ km}$). The input for the LSTM network is the rainfall over the entire domain at each time step. With a forecasting time of twelve hours, this means that the network has a total of twelve inputs, each representing the rainfall intensity during a one hour time step. The output of the network are the inundation depths for each of the grid cells, at each time step. With $53,381$ grid cells for which inundation needs to be predicted, and 12 time steps, this results in a total of $640,572$ outputs.

To find a well-performing network architecture, a hyper parameter optimization is executed. To do this, multiple grid searches (Bergstra et al., 2011) are carried out. Due to the high computational complexity of the problem, multiple separate grid searches with smaller search spaces are done rather than a full grid search. Starting with the most influential parameters, grid searches are performed to find the optimal setup for this parameter. The base network is updated with this new information, and the next parameter set is tested using a grid search. In this way, insight in how different parameters affect model convergence and number of trainable parameters is gathered, and the final model will be both accurate and efficient. The grid searches performed during the optimization are listed below:

1. The number of LSTM layers and number of units per layer are optimized with a grid search.
2. The number of dense layers and number of neurons per layer are optimized.
3. The dropout per layer is optimized.
4. The learning rate and batch size are optimized.

The decisions made during the hyper parameter optimization are elaborated in appendix A. For the LSTM layers, a hyperbolic tangent activation function is used. Changing to other activation functions did not show significant improvements, and the hyperbolic tangent activation function is the most popular one since it allows for GPU acceleration, which decreases the computational time. For the dense layer, a rectified linear unit activation function (ReLU) is used. From testing, this was found to be the most suitable one. It also makes sense physically, since the output (water depths) can never be below zero, just like the rectified linear unit activation function. Other activation functions that were tested include the sigmoid, linear and hyperbolic tangent activation functions. The final network setup is shown in Fig. 3, and the parameters are listed in Table 2. The network used to generate the final model for this study was trained for 2000 epochs, which took approximately 10 h without utilizing a GPU.

LSTM networks are known to show improved performance when the training dataset is larger. However, the effect diminishes with increasing size of the training dataset (Boulmaiz et al., 2020). The neural network in this study is trained using a total of 3.2 million samples, which corresponds with 2000 epochs on the full training dataset of 1600 samples. To analyse the effect of the training dataset size, the number of unique samples is varied meaning that there will be more repetition in unique samples when the size of training dataset is smaller. The performance of five trained neural networks are compared in which the number of

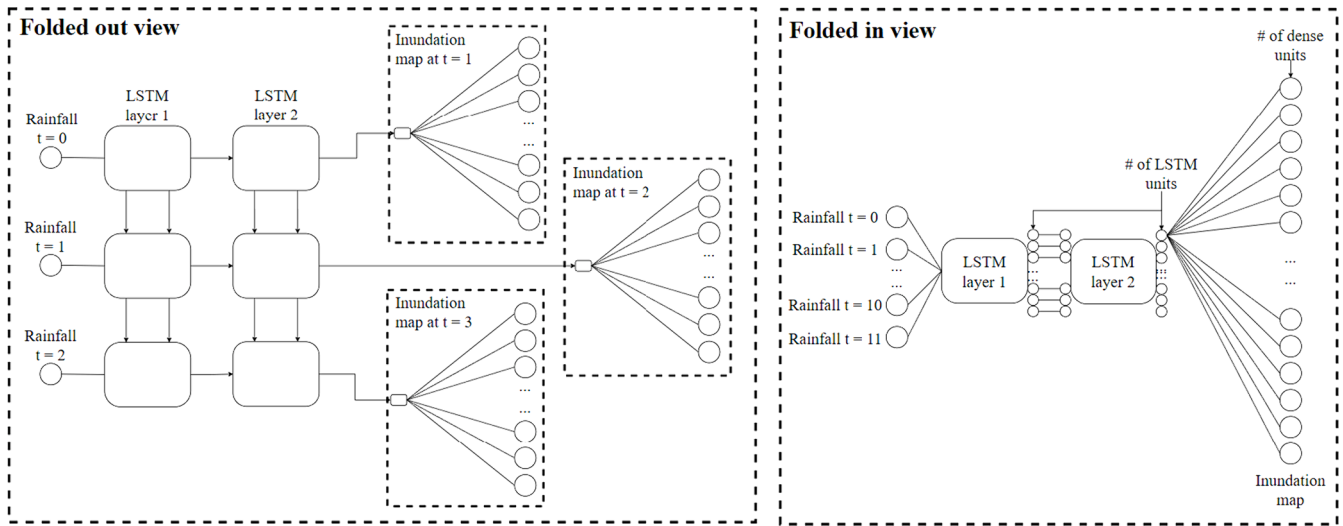


Fig. 3. Graphical representation of the neural network architecture.

Table 2
Neural network architecture resulting from the hyper parameter optimization.

Hyper parameter	Final value
LSTM layer 1 units	256
LSTM layer 2 units	256
LSTM layer activation function	hyperbolic tangent
Dense layer units	53.381 (number of grid cells)
Dense layer activation function	rectified linear unit
Dropout after layer 1	0.02
Dropout after layer 2	0.2
Learning rate	0.001
Batch size	8

unique samples considered in the training phase are 100, 200, 400, 800 and 1600.

2.5. Probabilistic forecasts

The neural network is trained to replicate single simulations, where this research aims to predict probabilistic inundation maps based on ensemble rainfall forecasts. To make a probabilistic prediction, the neural network predicts the inundation patterns for each of the ensemble members of the rainfall forecast. A threshold of 5 cm is then used to determine whether a cell is inundated or dry (Zanchetta and Coulibaly, 2022). The inundation probability for each grid cell is calculated according to equation (1). In equation (1) $P(I)_i$ is the inundation probability at grid cell i , $n_{t,i}$ is the number of ensemble members where inundation occurs at grid cell i and N_{ens} is the total number of ensemble members considered.

$$P(I)_i = \frac{n_{t,i}}{N_{ens}} \quad (1)$$

To test the performance of the neural network on generating probabilistic inundation forecasts, testing events are required. In this study, historical ensemble forecasts for the Netherlands between the 15th of April and the 15th of October in 2022 are available. However, for accurate model testing a focus on heavy rainfall predictions is essential, aligning with the model's intended application.

In the period for which the ensemble forecasts are available, no sufficient heavy rainfall events within the study area are available. Therefore, to generate these heavy rainfall forecasts, the following steps are taken:

1. Download the ensemble forecasts for the Netherlands for all days with a measured rainfall larger than 5 mm at measurement station the Bilt. The ensemble forecast is published every 6 h, forecasting 2 days ahead. It has a spatial resolution of 7.5 km * 7.5 km, meaning that the entirety of the Netherlands is represented by 758 grid cells.
2. From the ensemble forecasts from step 1, select the three events (both in time and space) with the highest average rainfall over all the ensemble members.
3. Scale each of the three events such that the ensemble member with the most rainfall in 12 h corresponds to the rainfall of a T1000 event (Beersma et al., 2019). A T1000 event is an extreme event that is expected to occur once every 1000 years. This corresponds to a rainfall amount of 139.2 mm in 12 h.

The resulting scaled rainfall patterns for the three ensemble forecasts can be seen in Fig. 4.

2.6. Performance indicators

The accuracy of the neural network is assessed by comparing the neural network predictions to those of the hydraulic model. The neural network can never perform better than the hydraulic model, since it is trained on the output of the hydraulic model.

The MAE, MSE and RMSE (equation (2), 3 and 4) are used to assess the difference between the neural network and the hydraulic model. The MSE is used as a minimization function for the neural network during the training process, since it provided the best results during testing. The other performance indicators are only used for testing.

$$MAE = \frac{1}{n^*t} \sum_{t=1}^T \sum_{i=1}^n |y_{pred,i,t} - y_{obs,i,t}| \quad (2)$$

$$MSE = \frac{1}{n^*t} \sum_{t=1}^T \sum_{i=1}^n (y_{pred,i,t} - y_{obs,i,t})^2 \quad (3)$$

$$RMSE = \sqrt{\frac{1}{n^*t} \sum_{t=1}^T \sum_{i=1}^n (y_{pred,i,t} - y_{obs,i,t})^2} \quad (4)$$

In equation (2), 3 and 4 $y_{pred,i,t}$ is the inundation according to the neural network at grid cell i and timestep t . $y_{obs,i,t}$ is the inundation according to the hydraulic model at grid cell i and timestep t . T is the total number of timesteps and n is the total number of grid cells that are considered.

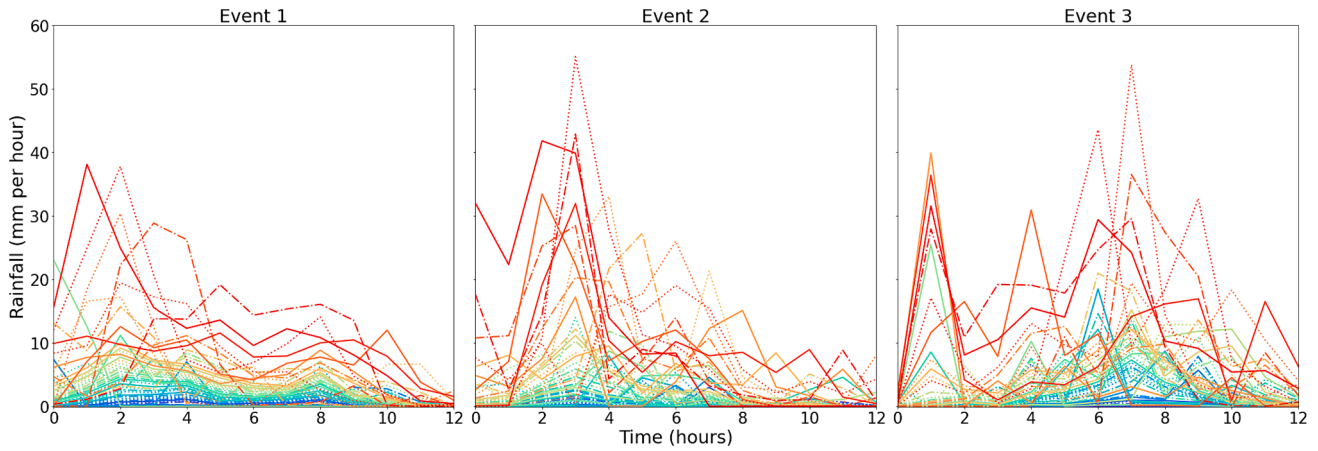


Fig. 4. Rainfall over time during the generated events. The different colored lines represent the different ensemble members, there is a total of 50 ensemble members per event.

The NSE is a normalized statistic that determines the relative magnitude of the residual variance compared to the measured data variance. A NSE of 1 corresponds to a perfect match between model and data. $NSE = 0$ indicates that model predictions are as accurate as predicting the mean. A NSE between 0 and $-\infty$ indicates that the mean would be a better predictor than the model. The formula for the NSE is shown in equation (5) (Nash and Sutcliffe, 1970).

$$NSE_i = 1 - \frac{\sum_{t=1}^T (y_{pred,i,t} - y_{obs,i,t})^2}{\sum_{t=1}^T (y_{obs,i,t} - y_{mean,i})^2} \quad (5)$$

In equation (5) NSE_i is the NSE at grid cell i , $y_{pred,i,t}$ is the inundation according to the neural network at timestep t at grid cell i , $y_{obs,i,t}$ is the inundation according to the hydraulic model at timestep t at grid cell i , $y_{mean,i}$ is the mean inundation according to the hydraulic model at grid cell i and T is the total number of time steps that is considered.

The CSI measures the model’s ability to correctly predict flooding. Some cells are infrequently flooded, so determining the accuracy of these cells would typically yield a high accuracy, even if the model always predicts no flooding. To counter this problem, the CSI is used, which does not consider true negatives (or in this case: predicting a dry cell when it is actually dry). The CSI is calculated according to equation (6) (Schaefer, 1990).

$$CSI = \frac{true\ positives}{false\ negatives + false\ positives + true\ positives} \quad (6)$$

To assess the performance of the probabilistic inundation forecasts, the inundation probabilities of the network are compared to those of the hydraulic model. The quality is assessed by calculating the Brier score, plotting a reliability diagram and calculating the percentage of agreement between the hydraulic model and the neural network.

The reliability diagram is a 2D density plot of the neural network inundation probabilities and the hydraulic model inundation probabilities for all grid cells at all time steps. The resulting points are colored based on how often they occur. The Brier score is equivalent to the mean squared error as applied to predicted probabilities, and is calculated according to equation (7) (Brier, 1950).

$$Brier\ score = \frac{1}{T} \frac{1}{n} \sum_{t=1}^T \sum_{i=1}^n (f_{i,t} - o_{i,t})^2 \quad (7)$$

Where $f_{i,t}$ is the neural network probability of inundation of grid cell i at time step t , $o_{i,t}$ is the hydraulic model probability of inundation of grid cell i at time step t , n is the total number of grid cells that is considered, and T is the total number of time steps that is considered.

3. Results

3.1. Performance on single simulations

In Fig. 5 a scatter plot of the inundation depth forecasts can be seen. A comparison is made for each grid cell, each time step, and each testing sample. The color represents the density of points. At some points in the scatter plot point density is higher than in the color bar. For example, the density is very high at (0,0), because this point represents the occasions where a cell is correctly predicted as not inundated, which occurs frequently during low rainfall.

From Fig. 5 and Table 3 it can be seen that the neural network performs very well in predicting the inundation depths across all events. The ReLU activation function used for the final dense layer ensures that predictions are always larger than or equal to 0. The density around the $x = y$ prediction line is high, and going further from this line the density drastically reduces. From the figure, there is no large bias towards either under- or over prediction of inundation depths.

However, when zooming in towards (0,0), and changing the color scale, a clear pattern is present (Fig. 6). The neural network often tends to predict inundation depths of 0, while in the hydraulic model there was a small amount of inundation (generally < 1 cm). This is caused by

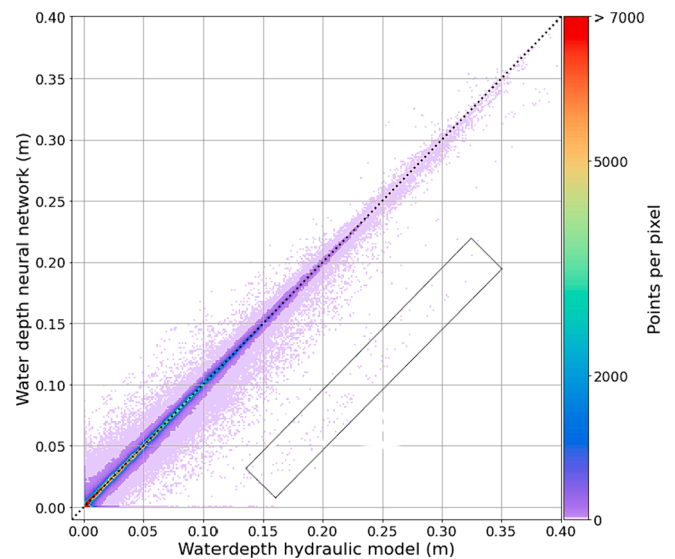


Fig. 5. Inundation predictions of the neural network versus the inundation predictions of the hydraulic model (testing data).

Table 3

Performance on different metrics for the entire test dataset of 200 events, with rainfall ranging between 7 and 157 mm in 12 h.

Metric	MSE	RMSE	MAE	CSI
Score	$6.98 \times 10^{-7} m^2$	$8.36 \times 10^{-4} m$	$1.15 \times 10^{-4} m$	0.964

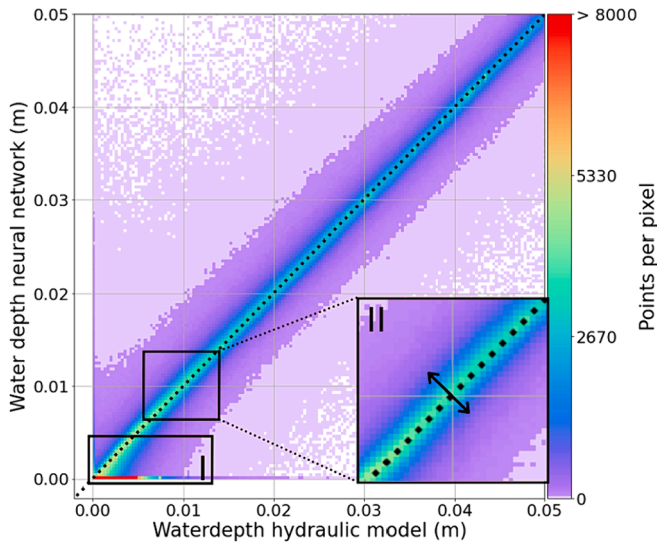


Fig. 6. Zoomed in view of Fig. 5 near (0,0).

the combination of the ReLU activation function and MSE as an optimization objective. Because of the ReLU activation, predicting 0 for the network is “easy”, in the sense that everything below 0 is set to zero in the last layer. When the network is optimizing, it will quickly learn to not predict 0 when inundation is large, because this yields high MSE values. But when inundation is very small (below 1 cm), predicting 0 is not penalized harshly. When the inundation depth for a specific cell is frequently zero and occasionally very small, the neural network lacks incentive to learn this particular pattern, as predicting zero already yields a low MSE. As a result the network does not always improve these errors. However since these errors are small this is not considered as a significant issue. The MSE puts more emphasis on large errors, which is desired for achieving accurate inundation forecasts.

Another pattern which is present in the scatter plot is highlighted with the rectangle in Fig. 5. Within this rectangle there is a trend line in the predictions, which indicates the network is making a systematic prediction error. More of these “trend lines” are present when inspecting Fig. 5 closely, however the others are closer to the $x = y$ line and therefore less conspicuous. Such a trend line is caused by an event where the inundation depths are estimated inaccurately in a particular area. This happened for some events where there was a large peak in rainfall, and right after this peak the network underestimates the increase in inundation depths the rainfall causes. The relative prediction errors are similar in a region, and therefore these “trend lines” emerge in Fig. 5.

To be able to make probabilistic inundation forecasts, cells are classified as either dry or wet according to the inundation threshold. With this classification, the confusion matrix is made, which can be seen in Table 4.

Table 4

Confusion matrix of the neural network predictions.

		Neural network	
		Wet	Dry
Hydraulic model	Wet	2.850.161	59.596
	Dry	46.736	125.157.907

When excluding the correctly predicted dry cells, 96.4 % (CSI) of the test data is correctly classified. The main source of error in classifying inundation is near the inundation threshold value. For instance, if the neural network predicts an inundation depth of 4.99 cm and the hydraulic model predicts 5.01 cm, the classification would be incorrect, even though the network’s prediction is very accurate. To address this issue, a margin of safety can be implemented around the threshold value. If both the hydraulic model and neural network predict an inundation depth within the margin of safety, then the classification of this grid cell will not be taken into account in calculating the CSI. The resulting CSI for different margins of safety can be seen in Table 5. It can be seen that by taking a margin around the threshold, the CSI improves significantly. This proves the classification error indeed mainly comes from the predictions that are very close to the inundation threshold. For the remainder of this study, the margin of safety is not used.

There is a very slight bias towards predicting false negatives (predicted dry but is wet) over predicting false positives (predicted wet but is dry). When looking closely, the reason for this can be seen in Fig. 7. At the 5 cm inundation threshold there is a very slight bias towards predicting lower inundation depths. When the classification between wet/dry is made, predictions close to the threshold are biased to be slightly lower, and therefore a larger part of them is classified as dry when they should be wet.

The inundation threshold has an influence on this classification bias. For example, if the inundation threshold was set at 1 cm instead of 5 cm, the bias would be the other way around, which can be seen in Fig. 6 at II. Inundation thresholds below 1 cm are unreliable, because below this threshold the tendency for the network to predict 0 at small inundation depths becomes a problem. Inundation predictions below 1 cm are often not even considered (Kabir et al., 2020; Zanchetta and Coulibaly, 2022; Zhou et al., 2021), and therefore this prediction error at very low inundation depths is not very relevant for inundation forecasts.

With the inundation threshold, the first time step at which a grid cell is inundated can be calculated, which is known as the arrival time. In general, the network predicts the arrival times very well. For 94 % of all grid cells the arrival time is predicted correctly. Important to note is that the small errors near the inundation threshold are the main cause of the errors in the arrival time. Because of this, the error mostly is either one hour too late or one hour too early. Prediction errors of more than one hour are rare, only occurring 0.2 % of the time.

The neural network is employed to predict inundation patterns across 12 time steps. To analyse the performance across different timesteps, the MAE and relative error are calculated for each time step, and illustrated in Fig. 8. Notably, the observed trend reveals a higher MAE for later time steps, which can be attributed to the extended temporal window allowing increased rainfall and subsequently higher inundation depths.

In response to this trend, the relative error, expressed as the MAE relative to the average inundation, is computed. The higher relative errors observed in the earlier time steps are attributed to the neural network’s optimization objective being the minimization of the MSE. Given that MSE is influenced by the extent of inundation, the network prioritizes locations with larger MSE for improvement. Consequently, at earlier time steps with lower inundation, there is a diminished potential for substantial MSE improvement compared to later time steps with more inundation, influencing the observed higher relative errors at earlier timesteps.

For each of the grid cells and for every testing event, the NSE is calculated. A NSE of 1 indicates perfect forecasting skill. The percentage

Table 5

CSI for different margins around the inundation threshold.

Margin	0 cm	0.1 cm	0.5 cm	1 cm
CSI	96.4 %	98.6 %	99.7 %	99.9 %

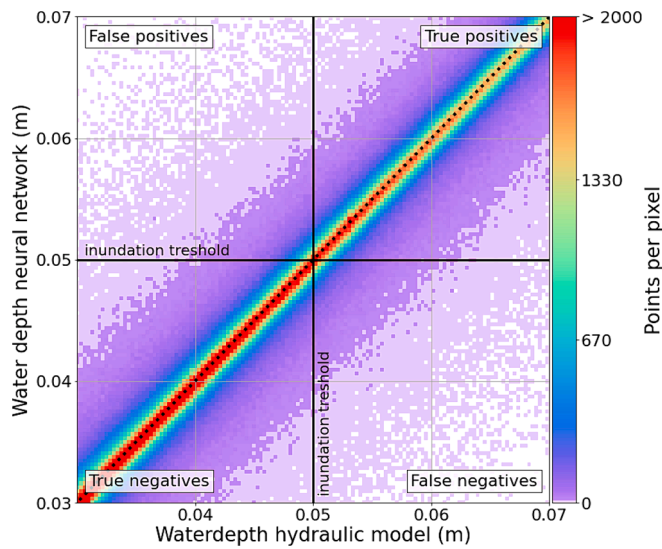


Fig. 7. Zoomed in view of Fig. 5 near the inundation threshold.

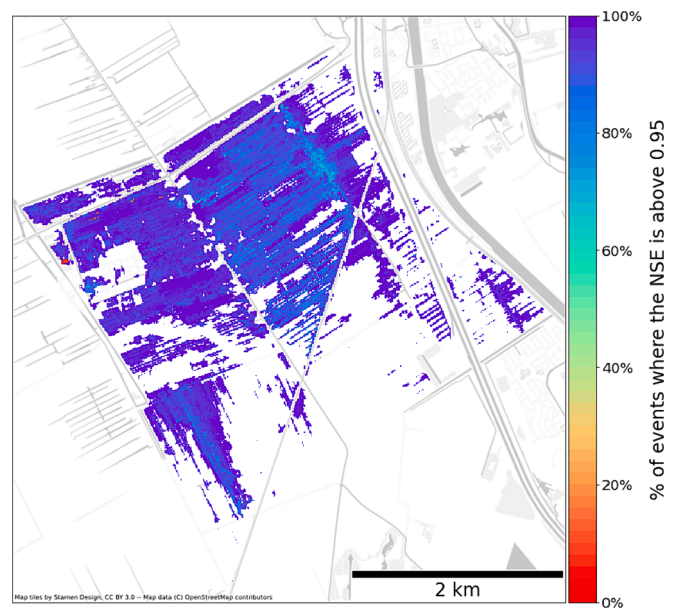


Fig. 9. The percentage of events for each grid cell where the NSE was higher than 0.95. Grid cells that are never inundated in the training dataset are not shown in this figure.

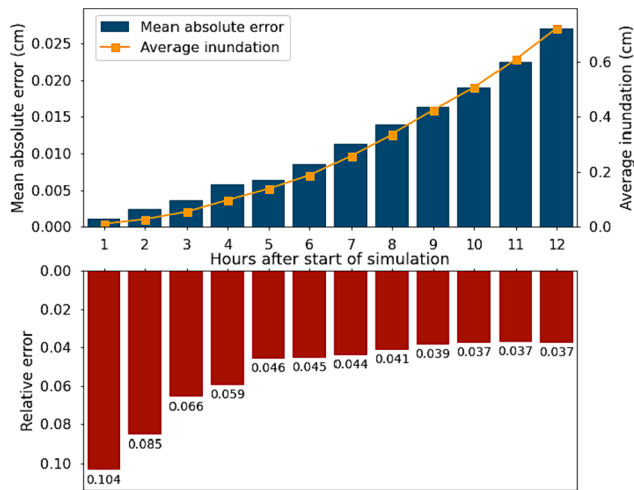


Fig. 8. The MAE and relative error of the inundation predictions at different time steps.

of events where the NSE of a grid cell is above 0.95 is shown in Fig. 9. For most of the grid cells, the NSE is above 0.95 for the majority of the events. Over the entire domain, the NSE is larger than 0.95 in 94.5 % of the cases, which indicates near perfect forecasting skill in these cases. Here, a case refers to the NSE of a grid cell in one testing event.

In 2.3 % of the cases, the NSE is between 0 and 0.95. This generally is caused by low inundation (on average 0.4 cm for these cells). With such low inundation, the mean would already be a decent predictor. This is difficult to improve upon by the neural network, as the network has little incentive to improve in areas where the objective function value (MSE) is small. At these locations the network performs better than the mean, however it does not perfectly predict the inundation depths.

In 3 % of the cases the NSE is lower than 0. These are locations with very low inundation depths (on average 0.1 cm), where the network has a tendency to always predict zero, as shown in Fig. 6 at I. Because of this tendency, the mean is sometimes a better predictor than the neural network, despite that the actual prediction error is very small at these locations (Fig. 10)

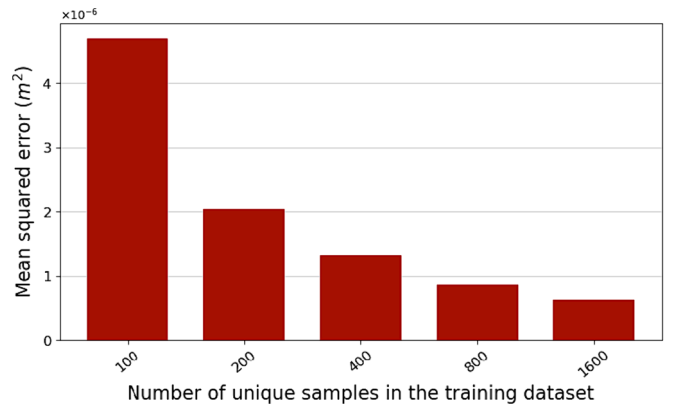


Fig. 10. The performance of the neural network for differently sized training datasets.

3.2. Performance on probabilistic forecasts

Fig. 11 displays the reliability diagrams for each of the three testing events, and in Table 6 the Brier scores are shown. Both the reliability diagrams and Brier scores demonstrate the neural network’s impressive ability to generate accurate probabilistic inundation forecasts. The performance is expected, since the network already has shown a good performance on inundation classification for deterministic simulations. Specifically, the network’s predictions matched the probabilities predicted by the hydraulic model for 91 % of the grid cells. For 99.6 % of the grid cells the neural network inundation probability is within 2 % of the probability predicted by the hydraulic model. These numbers exclude the grid cells where both the neural network and hydraulic model predicted a 0 % chance of inundation. From these results it can be concluded that the neural network can accurately predict the probabilities of inundation.

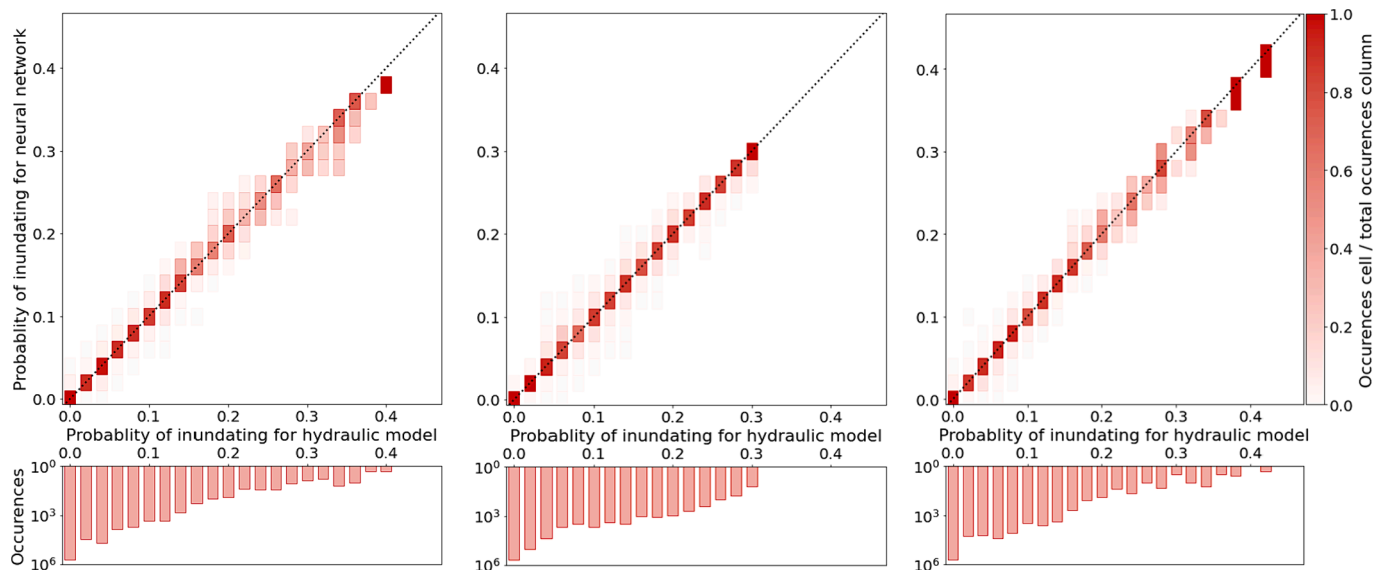


Fig. 11. Reliability diagrams for all three events. The occurrences show how many times a certain probability occurs according to the hydraulic model.

Table 6
Brier score for each of the three events.

Event	1	2	3
Brier Score	4.9×10^{-6}	7.4×10^{-6}	7.3×10^{-6}

The performance of the neural network on probabilistic inundation forecasts is mainly limited by the accuracy around the inundation threshold as shown in Fig. 7. When a significant number of rainfall events in a given ensemble forecast result in inundation near the threshold value, there is an increase in classification errors. When inundation is either significantly higher than the threshold, or significantly lower, the classification error reduces substantially, making the neural network more accurate in generating probabilistic inundation forecasts.

3.3. Sensitivity to size of training dataset

Investigating the impact of varying training dataset sizes on the performance of the developed neural network reveals a strong sensitivity of the Mean Squared Error (MSE) on the testing dataset to the number of unique samples in the training dataset. This sensitivity is illustrated in Fig. 9.

Further analysis of the outcomes highlight that, as the training dataset diminishes in size, the network is sensitive to overfitting. When employing 100 unique samples, the onset of overfitting is observed after processing 70,000 samples. Similarly, for training datasets comprising 200 unique samples, overfitting manifests after processing 200,000 samples. A further increase to 400 unique samples results in overfitting after 640,000 samples. Training datasets consisting of 800 and 1600 unique samples do not show significant overfitting within the allocated training time.

While acknowledging the influence of training dataset size on performance, the results display the diminishing returns of performance improvement as the size of the training dataset increases. This observation aligns with expectations, as noted in the work of Boulmaiz et al., 2020. The neural network’s performance may exhibit further enhancements with a training dataset exceeding the currently employed 1600 samples, however the performance improvements are expected to be minor.

4. Discussion

Several studies have predicted inundation depths with neural networks trained on hydraulic models (e.g. Berkhahn et al., 2019; Chang et al., 2018, 2014; Guo et al., 2021; Kao et al., 2021; Kilsdonk et al., 2022; Zanchetta and Coulibaly, 2022). Comparing the accuracy between studies is challenging, since the case study used has a significant impact on the accuracy. However compared to other studies, the network proposed in this study provides a very low prediction error while making inundation predictions at a high spatial resolution. Furthermore, the network is able to predict inundation progression over time, which can provide additional insight compared to studies that only predict maximum inundation depth (e.g. Berkhahn et al., 2019; Guo et al., 2021). The high accuracy of the network developed can partly be attributed to the large training dataset. Because cloud computing was used, there were no time limitations in generating this training dataset.

A drawback of using a neural network for inundation prediction is its rigidity compared to a hydraulic model. Changes in the study area can be easily incorporated into a hydraulic model by adjusting the model schematization. This allows for analyzing the effect of specific measures on the inundation. In contrast, the neural network is trained on a large number of hydraulic simulations that all need to be performed beforehand. When frequent predictions are necessary and minimal adjustments to the model schematization are needed, there is no immediate issue. However, heavy rainfall events are infrequent, which means that inundation predictions are needed less regularly. In certain cases this could make it more efficient to perform the hydraulic simulations when predictions are needed, instead of training a neural network to do so. There is a trade off between computation time and cost associated with conducting hydraulic simulations for a probabilistic forecast and the cost of generating training data.

Generally, insufficient real-world inundation data is available to train a neural network. Therefore, typically hydraulic simulations are used to create the training dataset, as was also done in this study. Consequently the accuracy of neural networks is limited to the accuracy of the hydraulic model. Furthermore, using field observations as training data may negatively affect the performance of a neural network due to the presence of many other factors beyond rainfall that impact inundation.

While some of these variables can be incorporated as inputs for the network, additional variables would require more training data to make accurate predictions. Additionally, there may be variables that influence

inundation but cannot be included as inputs for the neural network, resulting in a level of unpredictability in the inundation that cannot be captured by the network, potentially resulting in a lower accuracy than obtained in this study. For this reason, further research is suggested on the impact of additional input variables on the accuracy of the neural network and the required training data. Also, the neural network has only been trained for forecasting inundation 12 h ahead. It would be interesting to analyse the capability of the neural network for generating forecasts with lead times exceeding 12 h.

An important note is that within the final network architecture there is no spatial coherence between adjacent cells. Within the neural network architecture, there is no difference between cells laying next to each other and cells laying on the opposite side of the domain. Instead, the network will learn patterns for each cell, and if it learns them well enough, the spatial coherence of the original dataset is replicated. Furthermore, the network proposed in this study predicts the inundation for every grid cell in the study area at a 10 m resolution. This provides a good insight over the entire study area, and also shows the capabilities of a neural network. However if a network would be used operationally it might not be necessary to predict inundation for every grid cell. Locations of interest can be selected beforehand, and the network could be trained to only predict inundation at these locations. This would allow for less complex network structures (e.g. Zhou et al., 2021), which reduces the training time. A challenge for further research lies in researching the possibilities of a generalized neural network for inundation predictions, which can be applied without needing to be trained specifically for a new area. The input of this network would need to include geospatial information. Training a generalized network is significantly more difficult than training a network for a specific area. Rather than remembering relations between a grid cell and predetermined input variables, the network would need to have an understanding of how the geography of an area affects inundation. It is unlikely that such a network would reach the same accuracy as a network that is specifically trained for an area. However, it would allow for rapid inundation predictions without needing to setup a hydraulic model. Furthermore, such a generalized neural network can be used in data-scarce regions with limited measurements available to train a neural network specifically for an area.

Convolutional neural networks or physics-informed neural networks might be better suited for the task at hand, given their ability to interpret rasters and incorporate physical equations in training (Bentivoglio et al., 2022; Cai et al., 2021). Furthermore, recent research has shown that graph neural networks are effective in predicting the spatio-temporal distribution of floods and can generalize well to unseen topographies (Bentivoglio et al., 2023). However, computation times in the order of minutes for a single flood event, limits the use of graph neural networks

Appendix A. Hyper parameter optimization

The results of the hyper parameter optimization can be seen in the figures below. Note that these figures show the tests performed for the 20 m resolution network. The tests have been performed again with a narrowed down search space on the 10 m resolution network, which did yield the same hyper parameters.

for ensemble predictions for large-scale areas.

Lastly, the initial state of the hydraulic system is an important source of uncertainty. With the low computation times of the neural network, the possibility arises of including the uncertainty in the initial conditions within the probabilistic forecast. In this case the network would need to be able to predict inundation based on rainfall and the initial state. When considering an ensemble of initial states, together with the ensemble of rainfall forecasts, the neural network could create probabilistic inundation forecasts that do include both the uncertainty in the initial state of the system, and the uncertainty in the rainfall forecasts.

5. Conclusions

Computational cost has been a bottleneck for performing probabilistic inundation forecasts using hydraulic models. This study proposes a LSTM network trained on 1600 inundation simulations of a 1D2D hydraulic model, which can predict inundation progression over time with a spatial resolution of 10 m. By combining the inundation predictions for all ensemble members of a rainfall forecast, a probabilistic inundation forecast is generated. The developed network predicts the inundation with a high accuracy. With the network a probabilistic inundation forecast consisting of 50 ensemble members can be generated in less than a second, making it suitable for real time forecasting.

CRedit authorship contribution statement

Fedde J. Hop: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Ralf Linneman:** Conceptualization, Funding acquisition, Resources, Writing – review & editing. **Bram Schnitzler:** Conceptualization, Funding acquisition, Supervision, Writing – review & editing. **Anouk Bomers:** Conceptualization, Methodology, Supervision, Writing – review & editing. **Martijn J. Booij:** Conceptualization, Methodology, Supervision, 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

Data will be made available on request.

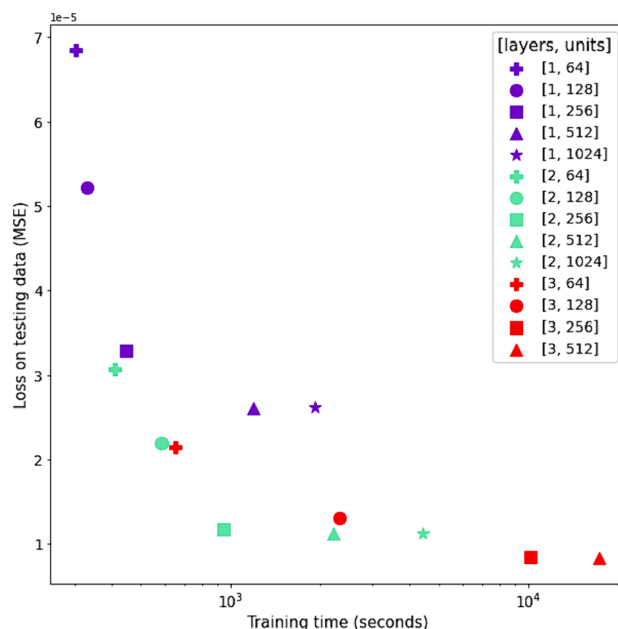


Fig. A.1. Number of LSTM layers and number of units per LSTM layer. 2 layers with density of 256 are chosen since accuracy does not improve significantly with more complexity. Using different numbers of units per layer was also tested, but did not improve MSE and are therefore not shown.

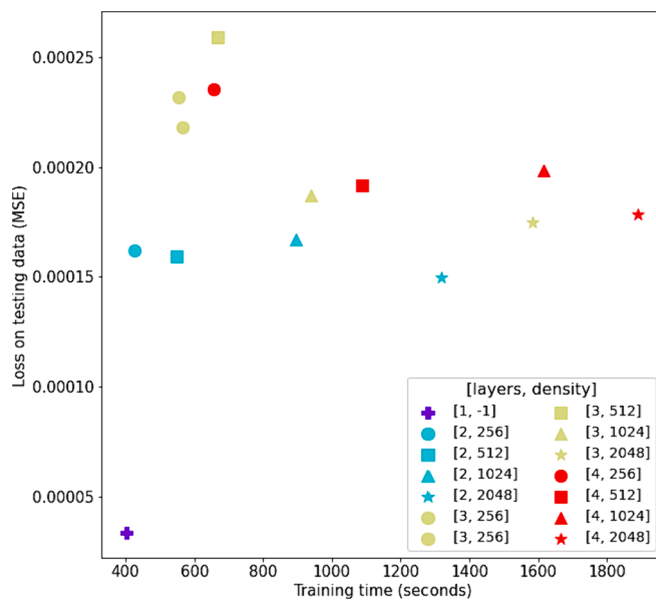


Fig. A.2. Number of dense layers and number of neurons per layer. Last layer always has the density of output size (number of grid cells), -1 density stands for default output size. Adding additional dense layers does not improve MSE, but it does increase training time. Therefore, no additional dense layers are used.

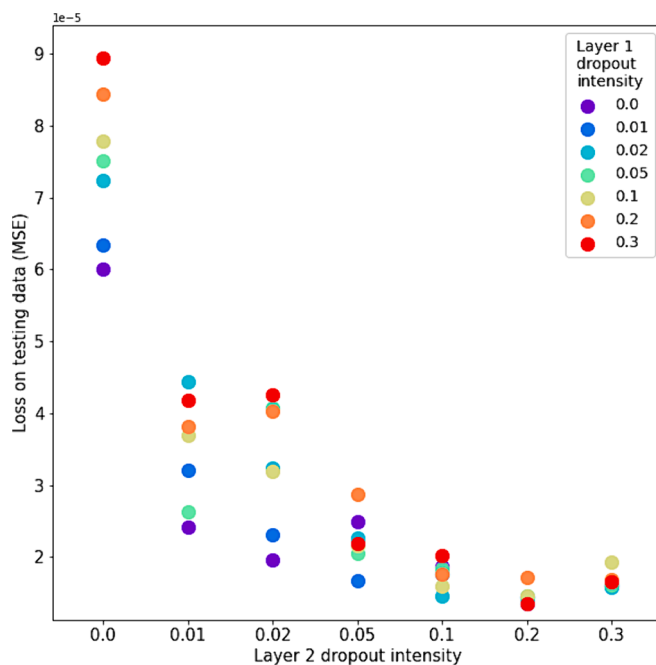


Fig. A.3. Dropout after LSTM layer 1 and after LSTM layer 2. Dropout after LSTM layer 2 is more influential than after layer 1, with a dropout of 0.2 after layer 2 being the best option tested. Layer 1 dropout is chosen at 0.02, but influence of this is small. Note that small variations in MSE can also be due to stochastic nature of the training, not necessarily caused by changing parameters.

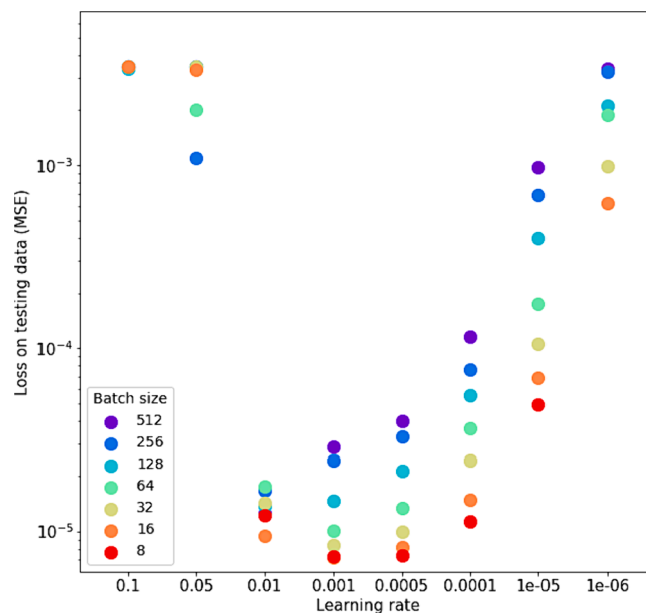


Fig. A.4. Learning rate and batch size. Final decision is to use learning rate of 0.001, since it results in the lowest MSE. For the batch size, smaller batches perform better, as expected from literature (Wilson and Martinez, 2003). However, larger batches can significantly improve training times especially when calculations are performed on a GPU. A batch size of 8 is used.

References

Attema, J., Bakker, A., Beersma, J., Bessembinder, J., Boers, R., Brandsma, T., van den Brink, H., Drijfhout, S., Eskes, H., Haarsma, R., others, 2014. KNMI'14: Climate Change scenarios for the 21st Century—A Netherlands perspective [WWW Document]. URL http://www.klimaatsscenarios.nl/brochures/images/KNMI_WR_2014-01_version26May2014.pdf (accessed 8.26.22).

Beersma, J., Hakvoort, H., Jilderda, R., Overeem, A., Versteeg, R., 2019. Neerslagstatistiek en reeksen voor het waterbeheer 2019. Stichting Toegepast Onderzoek Waterbeheer.

Bentivoglio, R., Isufi, E., Jonkman, S., Taormina, R., 2023. Rapid Spatio-Temporal Flood Modelling via Hydraulics-Based Graph Neural Networks. [Preprint]. <https://doi.org/10.5194/egusphere-2023-284>.

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. <https://doi.org/10.5194/hess-26-4345-2022>.
- Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B., 2011. Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems* 24 (NIPS 2011) 2546–2554.
- Berkhahn, S., Fuchs, L., Neuweiler, I., 2019. An ensemble neural network model for real-time prediction of urban floods. *J Hydrol (amst)* 575, 743–754. <https://doi.org/10.1016/j.jhydrol.2019.05.066>.
- Boulmaiz, T., Guermoui, M., Boutaghane, H., 2020. Impact of training data size on the LSTM performances for rainfall–runoff modeling. *Model Earth Syst Environ* 6, 2153–2164. <https://doi.org/10.1007/s40808-020-00830-w>.
- Brier, G.W., 1950. Verification of forecasts expressed in terms of probability. *Mon. Weather Rev.* 78, 1–3. [https://doi.org/10.1175/1520-0493\(1950\)078<0001:VOFEIT>2.0.CO;2](https://doi.org/10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2).
- Brutsaert, W., 2006. *Hydrology: an introduction*, Choice Reviews Online. Cambridge University, Cambridge. <https://doi.org/10.5860/choice.43-4036>.
- Cai, S., Mao, Z., Wang, Z., Yin, M., Karniadakis, G.E., 2021. Physics-informed neural networks (PINNs) for fluid mechanics: a review. *Acta Mechanica Sinica/Chinese Xuebao* 37, 1727–1738. <https://doi.org/10.1007/s10409-021-01148-1>.
- Chang, L.C., Shen, H.Y., Chang, F.J., 2014. Regional flood inundation nowcast using hybrid SOM and dynamic neural networks. *J. Hydrol. (amst)* 519, 476–489. <https://doi.org/10.1016/j.jhydrol.2014.07.036>.
- Chang, L.C., Amin, M.Z.M., Yang, S.N., Chang, F.J., 2018. Building ANN-based regional multi-step-ahead flood inundation forecast models. *Water (switzerland)* 10, 1283. <https://doi.org/10.3390/W10091283>.
- Chu, H., Wu, W., Wang, Q.J., Nathan, R., Wei, J., 2020. An ANN-based emulation modelling framework for flood inundation modelling: application, challenges and future directions. *Environ. Model. Softw.* 124, 104587. <https://doi.org/10.1016/j.envsoft.2019.104587>.
- Cloke, H.L., Pappenberger, F., van Andel, S.J., Schaake, J., Thielen, J., Ramos, M.H., 2013. Hydrological ensemble prediction systems. *Hydrol. Process* 27, 1–4. <https://doi.org/10.1002/hyp.9679>.
- Coulily, G., Leye, B., Tazen, F., Mounirou, L.A., Karambiri, H., 2020. Urban flood modeling using 2D shallow-water equations in Ouagadougou. *Burkina Faso. Water (basel)* 12, 2120. <https://doi.org/10.3390/w12082120>.
- Deltares, 2023. D-flow FM user manual (Delft3D FM suite 1D2D), revision, 7. ed. Deltares, Delft.*
- Domenghetti, A., Vorogushyn, S., Castellarin, A., Merz, B., Brath, A., 2013. Probabilistic flood hazard mapping: effects of uncertain boundary conditions. *Hydrol. Earth Syst. Sci.* 17, 3127–3140. <https://doi.org/10.5194/hess-17-3127-2013>.
- Fang, Z., Wang, Y., Peng, L., Hong, H., 2021. Predicting flood susceptibility using LSTM neural networks. *J Hydrol (amst)* 594. <https://doi.org/10.1016/j.jhydrol.2020.125734>.
- García-Navarro, P., Murillo, J., Fernández-Pato, J., Echeverriar, I., Morales-Hernández, M., 2019. The shallow water equations and their application to realistic cases. *Environ. Fluid Mech.* 19, 1235–1252. <https://doi.org/10.1007/s10652-018-09657-7>.
- Gomez, M., Sharma, S., Reed, S., Mejia, A., 2019. Skill of ensemble flood inundation forecasts at short- to medium-range timescales. *J Hydrol (amst)* 568, 207–220. <https://doi.org/10.1016/j.jhydrol.2018.10.063>.
- Goodarzi, L., Banihabib, M.E., Roozbahani, A., 2019. A decision-making model for flood warning system based on ensemble forecasts. *J Hydrol (amst)* 573, 207–219. <https://doi.org/10.1016/j.jhydrol.2019.03.040>.
- Guo, Z., Leitão, J.P., Simões, N.E., Moosavi, V., 2021. Data-driven flood emulation: speeding up urban flood predictions by deep convolutional neural networks. *J Flood Risk Manag* 14, 287–315. <https://doi.org/10.1111/jfr3.12684>.
- Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural Comput.* 9, 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>.
- Jackson, E.K., Roberts, W., Nelsen, B., Williams, G.P., Nelson, E.J., Ames, D.P., 2019. Introductory overview: error metrics for hydrologic modelling – a review of common practices and an open source library to facilitate use and adoption. *Environ. Model. Softw.* 119, 32–48. <https://doi.org/10.1016/j.envsoft.2019.05.001>.
- Kabir, S., Patidar, S., Xia, X., Liang, Q., Neal, J., Pender, G., 2020. A deep convolutional neural network model for rapid prediction of fluvial flood inundation. *J Hydrol (amst)* 590, 125481. <https://doi.org/10.1016/j.jhydrol.2020.125481>.
- Kao, I.F., Liou, J.Y., Lee, M.H., Chang, F.J., 2021. Fusing stacked autoencoder and long short-term memory for regional multistep-ahead flood inundation forecasts. *J Hydrol (amst)* 598, 126371. <https://doi.org/10.1016/j.jhydrol.2021.126371>.
- Kilsdonk, R.A.H., Bomers, A., Wijnberg, K.M., 2022. Predicting urban flooding due to extreme precipitation using a long short-term memory neural network. *Hydrology* 9. <https://doi.org/10.3390/hydrology9060105>.
- Lin, Q., Leandro, J., Gerber, S., Disse, M., 2020. Multistep flood inundation forecasts with resilient backpropagation neural networks: Kulmbach case study. *Water (switzerland)* 12, 3568. <https://doi.org/10.3390/w12123568>.
- Liu, Q., Qin, Y., Zhang, Y., Li, Z., 2015. A coupled 1D–2D hydrodynamic model for flood simulation in flood detention basin. *Natural Hazards* 75, 1303–1325. <https://doi.org/10.1007/s11069-014-1373-3>.
- McSpadden, D., Goldenberg, S., Roy, B., Schram, M., Goodall, J.L., Richter, H., 2024. A comparison of machine learning surrogate models of street-scale flooding in Norfolk, Virginia. *Machine Learning with Applications* 15, 100518. <https://doi.org/10.1016/j.mlwa.2023.100518>.
- Mei, C., Liu, J.H., Wang, H., Li, Z.J., Yang, Z.Y., Shao, W.W., Ding, X.Y., Weng, B.S., Yu, Y.D., Yan, D.Y., 2020. Urban flood inundation and damage assessment based on numerical simulations of design rainstorms with different characteristics. *Sci. China Technol. Sci.* 63, 2292–2304. <https://doi.org/10.1007/s11431-019-1523-2>.
- Nash, J.E., Sutcliffe, J.V., 1970. River flow forecasting through conceptual models. *J. Hydrol. (amst)* 10, 282–290.
- Papaioannou, G., Efstratiadis, A., Vasilades, L., Loukas, A., Papalexioiu, S.M., Koukouvinos, A., Tsoukalas, I., Kossieris, P., 2018. An operational method for flood directive implementation in ungauged urban areas. *Hydrology* 5. <https://doi.org/10.3390/hydrology5020024>.
- Prinsen, G., Hakvoort, H., Dahm, R., 2010. Neerslag-afvoermodellering met sobek-RR. *Stromingen* 15 (4), 8–24.
- Rafiq, M.Y., Bugmann, G., Easterbrook, D.J., 2001. Neural network design for engineering applications. *Comput. Struct.* 79, 1541–1552. [https://doi.org/10.1016/S0045-7949\(01\)00039-6](https://doi.org/10.1016/S0045-7949(01)00039-6).
- Rientjes, T.H.M., Muthuwatta, L.P., Bos, M.G., Booi, M.J., Bhatti, H.A., 2013. Multi-variable calibration of a semi-distributed hydrological model using streamflow data and satellite-based evapotranspiration. *J. Hydrol. (amst)* 505, 276–290. <https://doi.org/10.1016/j.jhydrol.2013.10.006>.
- Rosenzweig, B.R., McPhillips, L., Chang, H., Cheng, C., Welty, C., Matsler, M., Iwaniec, D., Davidson, C.I., 2018. Pluvial flood risk and opportunities for resilience. *Wiley Interdiscip. Rev. Water* 5. <https://doi.org/10.1002/wat2.1302>.
- Sajikumara, N., Thandaveswara, B.S., 1999. A non-linear rainfall-runoff model using an artificial neural network. *J Hydrol (amst)* 216, 32–55. [https://doi.org/10.1016/S0022-1694\(98\)00273-X](https://doi.org/10.1016/S0022-1694(98)00273-X).
- Schaefer, J.T., 1990. The critical success index as an indicator of Warning skill. *Weather Forecast.* 5, 570–575. [https://doi.org/10.1175/1520-0434\(1990\)005<0570:tcsiaa>2.0.co;2](https://doi.org/10.1175/1520-0434(1990)005<0570:tcsiaa>2.0.co;2).
- Seyoum, S.D., Vojinovic, Z., Price, R.K., Weesakul, S., 2012. Coupled 1D and noninertia 2D flood inundation model for simulation of urban flooding. *J. Hydraul. Eng.* 138, 23–34. [https://doi.org/10.1061/\(asce\)hy.1943-7900.0000485](https://doi.org/10.1061/(asce)hy.1943-7900.0000485).
- Shahapure, S.S., Kulkarni, A.T., Bharat, K.S.R., Eldho, T.I., Rao, E.P., 2010. Coastal urban flood simulation using fem-gis based model. *ISH Journal of Hydraulic Engineering* 16, 74–88. <https://doi.org/10.1080/09715010.2010.10515017>.
- Trenberth, K.E., 2011. Changes in precipitation with climate change. *Clim Res* 47, 123–138. <https://doi.org/10.3354/cr00953>.
- Wang, X., Kingsland, G., Poudel, D., Fenech, A., 2019. Urban flood prediction under heavy precipitation. *J. Hydrol. (amst)* 577. <https://doi.org/10.1016/j.jhydrol.2019.123984>.
- Wilson, D.R., Martinez, T.R., 2003. The general inefficiency of batch training for gradient descent learning. *Neural Netw.* 16, 1429–1451. [https://doi.org/10.1016/S0893-6080\(03\)00138-2](https://doi.org/10.1016/S0893-6080(03)00138-2).
- Wu, W., Emerton, R., Duan, Q., Wood, A.W., Wetterhall, F., Robertson, D.E., 2020. Ensemble flood forecasting: current status and future opportunities. *Wiley Interdiscip. Rev. Water* 7. <https://doi.org/10.1002/WAT2.1432>.
- Zanchetta, A.D.L., Coulily, P., 2022. Hybrid surrogate model for timely prediction of flash flood inundation maps caused by Rapid River overflow. *Forecasting* 4, 126–148. <https://doi.org/10.3390/forecast4010007>.
- Zarzar, C.M., Hosseini, H., Siddique, R., Gomez, M., Smith, V., Mejia, A., Dyer, J., 2018. A hydraulic MultiModel ensemble framework for visualizing flood inundation uncertainty. *J. Am. Water Resour. Assoc.* 54, 807–819. <https://doi.org/10.1111/1752-1688.12656>.
- Zhou, Y., Wu, W., Nathan, R., Wang, Q.J., 2021. A rapid flood inundation modelling framework using deep learning with spatial reduction and reconstruction. *Environ. Model. Softw.* 143. <https://doi.org/10.1016/j.envsoft.2021.105112>.