A deep convolutional neural network for rapid fluvial flood inundation modelling

Preprint · June 2020

8 authors, including:

Syed Rezwan Kabir
Loughborough University
4 PUBLICATIONS   **1** CITATION

SEE PROFILE

Xilin Xia
Loughborough University
27 PUBLICATIONS   **196** CITATIONS

SEE PROFILE

Qiuhua Liang
Loughborough University
135 PUBLICATIONS   **2,347** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project   REMATCH: Building REsilience to Multi-source Flooding in South/Southeast Asia Tech-informed Community-based approacH View project

Project   Climatic influence on the magnitude of COVID-19 outbreak: a stochastic model-based global analysis View project

A deep convolutional neural network for rapid fluvial flood inundation modelling

Syed Kabir[1,2]*, Sandhya Patidar[2], Xilin Xia[1], Qiuhua Liang[1], Jeffrey Neal[3] and Gareth Pender[2]

[1]School of Architecture, Building and Civil Engineering, Loughborough University, Loughborough, United Kingdom.

[2]School of Energy, Geoscience, Infrastructure and Society, Heriot-Watt University, Edinburgh, United Kingdom.

[3]School of Geographical Sciences, University of Bristol, Bristol, United Kingdom

*Corresponding author: Syed Kabir (S.R.Kabir@lboro.ac.uk)

Abstract

The two-dimensional (2D) hydrodynamic models are often infeasible for real-time operations. In this paper, a deep convolutional neural network (CNN)-based method is presented for rapid fluvial flood modelling. The CNN model is trained using outputs from a two-dimensional hydraulic model (i.e. LISFLOOD-FP) to predict water depths. The pre-trained model is then applied to simulate the flooding event that occurred in Carlisle, UK, in January 2005. The predictions are compared against the outputs produced by the calibrated LISFLOOD-FP. The performance of the CNN is also compared with a support vector regression (SVR)-based method. The results show that the CNN model outperforms SVR by a large margin. The model is highly accurate in capturing flooded cells as indicated by several quantitative assessment matrices, e.g.,

the estimated error for the peak flood depth is 0-0.2 meters for 97% cells of the domain when 99% confidence level is drawn. The proposed method offers great potential for real-time applications considering its simplicity, superior performance and computational efficiency.

Keywords

Rapid flood modelling; deep learning; convolutional neural network; machine learning; flood inundation

1. Introduction

Two-dimensional (2D) hydraulic/hydrodynamic models have been widely applied across a range of hydrological applications. These models are designed to simulate and predict the complex hydrological processes and hydrodynamics of a flood event. Recent advancements in the computing technology along with the increased availability of remotely sensed data, such as terrain elevation and river morphology in the data-scarce areas, are enabling hydrodynamic models to be implemented from regional to global scales (e.g., Yamazaki *et al.*, 2011, de Paiva *et al.*, 2013). However, due to their high computational demand, it is often challenging to use these physically-based sophisticated modelling approaches (standard hydrodynamic models hereafter) for real-time flood forecasting in practice (Bhola *et al.,* 2018).

A considerable amount of research has been done to-date to improve the overall performance of the standard hydrodynamic models for large-scale flood modelling. For example, the computational efficiency of a model may be improved by implementing parallel computing algorithms to take advantages of multiple processors simultaneously (Neal *et al.* 2018, Sanders and Schubert 2019). Xia *et al.* (2019) developed a new framework that utilizes the state-of-the-art high-performance computing facilities for

2

modelling fluvial flooding processes from rainfall-runoff to inundation at a higher

spatial resolution (~ 5m) across a large catchment of 2500 km^2. The so-called High-

Performance Integrated hydrodynamic Modelling System (HiPIMS) solves the full

shallow water equations (SWEs) and is accelerated by multiple modern graphics

processing units (GPUs). However, despite the advances in high-performance

computing technology and development of subsequent computational methods,

significant challenges still exist regarding the application of sophisticated 2D

hydrodynamic models for operational flood forecasting. One of the key challenges is

that while it is now possible to make a single model run in real-time, it is still

technically challenging to run a model multiple times using ensemble rainfall forecasts

to provide reliable forecasts with an acceptable lead time. Ensemble simulations are

essential for uncertainty quantification and data assimilation in a real-time forecasting

system.

An alternative approach is to use an offline method for operational flood

inundation mapping as proposed in Bhola *et al.* (2018). The proposed offline method

requires the construction of a database using pre-run inundation maps and river

discharges. During or before a flood event, the inundation maps with a matching

discharge are requested and retrieved from the database to provide forecasts (Leedal *et

al.*, 2010, Bhola *et al.*, 2018). Although such a system does not require live 2D

simulations, preparing the database of inundation maps can be labour intensive and

requires storing a large volume of data. Furthermore, due to changing environment, e.g.

land-use change, gemmological change and new engineering construction, the flood

scenarios may become outdated and new simulations may be needed to regularly update

the database, creating extra effort and resources for maintenance.

The ideal solution to these various technical challenges would be to develop a new model that can entirely relax the mentioned computational burden while still being able to generate practically useful and statistically significant information to support real-time flood forecasting. A possible solution could be the use of machine learning (ML) or deep learning (DL) models that can emulate the outputs of 2D hydraulic models. Application of ML techniques for rainfall-runoff forecasting has been investigated for a few decades. In contrast, research on the application of ML/DL for flood inundation modelling remains very limited and only a handful of studies have been reported to date to explore the potential of these techniques. Of particular interest to this work, Liu & Pender (2015) developed an inundation model that utilised a support vector regression (SVR) algorithm coupled with a coarse grid model (CGM) for predicting the outputs of a fine grid model (FGM). In their approach, SVRs were first trained using a small number of the outputs from FGM. The trained SVRs were then integrated within the CGM to assign predicted water depths and velocities to each grid. In Chang *et al.* (2010, 2014) and Shen & Chang (2013), hybrid ML techniques were successfully used for flood modelling. Chang *et al.* (2018a) developed a self-organizing map (SOM), an artificial neural network (ANN) designed for clustering operations, integrated with nonlinear autoregressive exogenous (R-NARX) networks for flood modelling at regional scale, giving flood forecasting of up to 12h ahead in the Kemaman River Basin, Malaysia. This approach was further improved in Chang, *et al.* (2018b), where the authors developed an intelligent hydroinformatics integration platform (IHIP) to provide a user-friendly web interface for improved capability in online forecasting and flood risk management. Bermúdez *et al.* (2019) presented a least squared-support vector machine (LS-SVM) based method for spatial prediction of flood hazards. In their study, the authors applied the ML-based model to compute the spatial

distribution of the maximum water depth and velocity in a coastal urban area using three sets of data for upstream flow and tidal level. Berkhahn *et al.* (2019) proposed an ensemble neural network model for real-time prediction of urban flooding. The model successfully predicted the maximum water levels during a flash flood event induced by spatially uniform rainfall. Wu *et al.* (2020) developed a DL algorithm, namely gradient boosting decision tree (GBDT), to predict flood depths in urban areas. In their work, a data warehouse was first constructed using available structured and unstructured flood data, which was then used to fit the regression model. A data-driven 3h ahead probabilistic flood inundation mapping framework was proposed by Kabir *et al.* (2020). These studies have demonstrated that computationally much less expensive ML-based approaches may be used to effectively predict comparable flood variables once they are trained and calibrated appropriately.

However, most of the existing ML algorithms are not suitable for multi-output scenarios, i.e. predicting flood variables (e.g., depth) for multiple cells through a single model. For example, Bermúdez *et al.* (2019) selected 25,000 points within a 16.3 km^2 domain and calibrated a LS-SVM at each point. Artificial neural networks (ANNs) may potentially be used to deal with multi-output problems. Over the last decade, as a type of ANN, convolution neuron network (CNN) has gained an unprecedented success in solving machine vision problems due to its great ability to extract unknown features and learn compact representations and is currently leading the DL paradigm. In recent years, one-dimensional CNN(1D-CNN) has achieved high-level performance in various scientific fields, for example, biomedical data classification (Zihlmann *et al.,* 2017) and structural damage detection (Abdeljaber *et al.,* 2018). However, the efficiency/ potentiality of this algorithm is yet to be tested for modelling high-resolution flood

inundation. At high spatial resolution, training a model involving millions of cells can be computationally expensive and challenging.

We propose, in this work, an innovative rapid fluvial flood modelling approach for predicting water depths using a CNN for real-time applications, e.g., flood forecasting. This is for the first time, the capability of a CNN model, accelerated by modern GPUs, is investigated to estimate the water depths of a fluvial flood event in a domain covered by over half of a million cells. The rest of the paper is organised as follows: the CNN modelling methodology is described in Section 2; the experimental setup and application are presented in Section 3; Section 4 provides and discusses the results; and finally the conclusions are drawn in Section 5.

2. Methodology

2.1 Overall research strategy

The strategic order of developing and assessing the new CNN model follows five core steps as illustrated in Fig. 1. In the first step, we randomly generate a number of synthetic hydrographs representing different hydrological conditions for each of the connected upstream locations in the study site. In the next step, these hydrographs are used as the driving boundary conditions for the 2D-hydraulic model (i.e. LISFLOOD-FP in this study). Water depth sequences are derived for the entire duration of different hydrological conditions represented by the synthetic hydrographs. In step three, we generate training and validation data for the candidate DL model (i.e. CNN). The upstream synthetic hydrographs are used as input variables and the outputs from LISFLOOD-FP model (i.e. water depth) are used as the target variable. In the following step, we develop the candidate model and optimize the parameters. Finally, the approach is assessed by reproducing the water depths for a real flood event (i.e. 2005

6

flood event) with the outputs from LISFLOOD-FP produced using observed

hydrographs. It is worth noting that the observed hydrographs in this case may be

replaced by the predictions from a forecasting model or ensembles.

The performance of the proposed CNN model will be further evaluated by

comparing with the SVR approach, one of the popular ML methods used in the earlier

studies (e.g., Lin *et al.*, 2013, Liu & Pender, 2015, Jhong *et al.*, 2018, Bermúdez *et al.*,
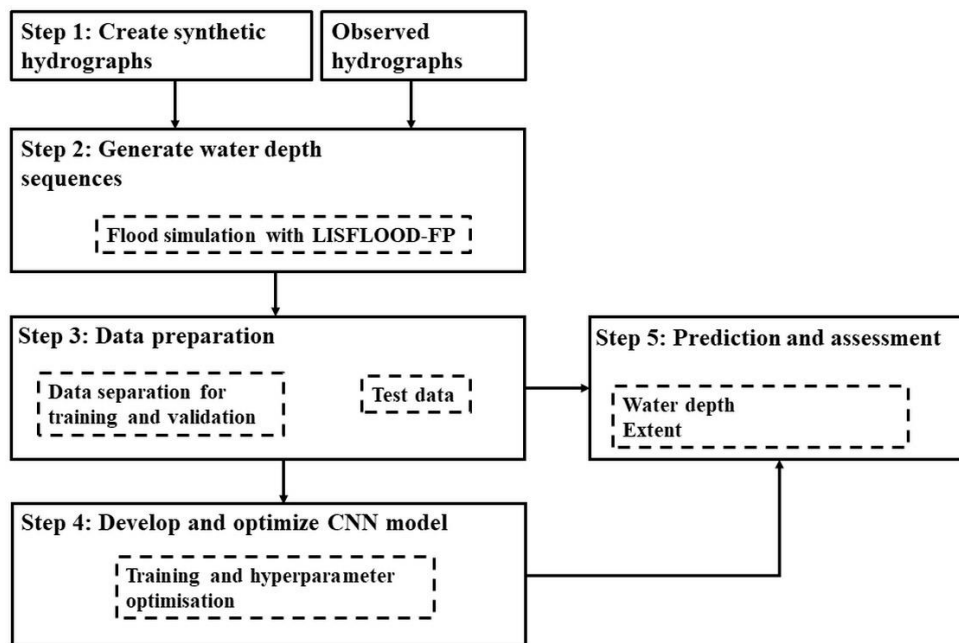
2019).

Figure 1. Flowchart of the core steps involved in developing and testing the proposed
CNN flood model.

2.2 Models and materials

In this section we describe the hydraulic and the ML models used to construct the rapid

flood modelling system. Two ML models are considered to compare their performance.

2.2.1 Convolutional Neural Network (CNN)

Since its inception in 1990, CNNs have become a research hotspot and the *de facto* standard for various ML and computer vision (CV) applications in the last decade. Specifically, the annual ImageNet large scale visual recognition challenge (ILSVRC) 2012 has changed the course of image classification problems through the application of deep CNNs. In this competition, Krizhevsky *et al.* (2012) proposed the winning AlexNet, a deep CNN, for large-scale image classification. In their seminal work, they also proposed the rectified linear units (ReLU) activation function instead of traditional sigmoid or hyperbolic tangent functions. However, the key factor which has made the deep CNNs extremely popular is the ever-increasing computational power of the modern computers.

CNNs are, in general, feed-forward neural networks with alternating convolutional and subsampling layers, and predominantly trained in a supervised manner (Kiranyaz *et al.*, 2019). Deep CNNs are exclusively developed to operate on 2D data (images and videos) and commonly known as '2D-CNN'. A 2D-CNN can extract features and learn complex objects from large volume of labelled data. A detailed description of the 2D-CNN training algorithms can be found in Kiranyaz *et al.* (2016). As the classical CNNs were developed specifically for 2D signals, for sequential data analysis (1D signal) the traditional ML techniques were preferred. However, Kiranyaz *et al.* (2015) proposed the first 1D-CNN to handle sequential data and has gained significant popularity in recent years, which is adopted in this study.

The graphical representation of the 1D-CNN used in this study is shown in Fig. 2. The network has five hidden layers that include two convolutional and three dense layers. The dense layers are the fully connected layers that act like a multi-layer perceptron (MLP) network. The output layer of the network contains nodes equal to the

number of cells in the simulation domain (i.e. 581,061 for the selected case study site), and the input layer receives the upstream flow values in a 3D array like shape. A detailed description of developing and training the CNN is presented in Section 3.3-3.5.
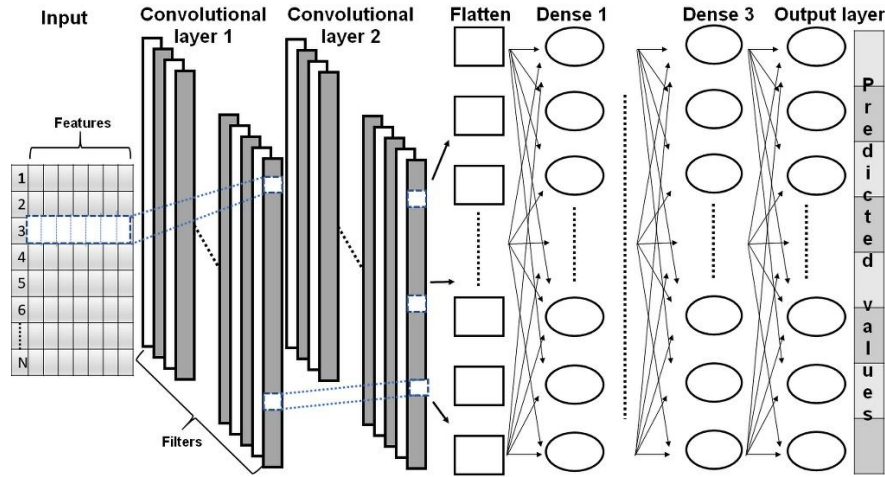

Figure 2. Structure of the CNN used in this study for water depth prediction.

2.2.2   LISFLOOD-FP flood inundation model

As illustrated in Fig. 1, the current flood inundation modelling framework uses the physically based LISFLOOD-FP model to generate training samples for the data-driven predictive models considered in this work. First reported by Bates & De Roo (2000), LISFLOOD-FP is a raster-based flood inundation model for simulating fluvial or coastal flood spreading. The model has been improved significantly over the last two decades and tested successfully in numerous case studies across the globe (e.g., Knijff *et al.*, 2010, Amarnath *et al.*, 2015, Komi *et al.*, 2017).

At the core, the model uses an explicit forward difference scheme to solve a local zero-inertial approximation of the 1D Saint Venant equations that represent continuity of mass and partly conservation of momentum on a staggered grid over a 2D plane. A 1D kinematic-wave river flow model is also imbedded in LISFLOOD-FP to

9

simulate flood wave propagation along river channels. Detailed introduction of the model may be found in Bates & De Roo (2000). Specific to the present study, the LISFLOOD-FP version 6.3.1 is used, which was implemented with an 'acceleration' solver (Bates *et al.*, 2010, de Almeida *et al.*, 2012) for calculating the floodplain inundation and a 'sub-grid' solver (Neal *et al.,* 2012) for the channel flow.

2.2.3 *Support vector regression (SVR)*

The SVR is developed from the original support vector machine (SVM) method, introduced by Vladimir Vapnik and Alexey Chervonenkis in early 1960s, to solve regression problems. It is a kernel-based supervised learning method and the basic feature of it is to map the input space into a high dimensional feature space using a non-linear mapping function (for solving non-linear problems) where the non-linearity of input vectors becomes linearly separable (Raghavendra and Deka 2014).

The standard SVR, in the context of present study, uses Vapnik's ε-insensitive loss function that defines the deviation of an estimated function from the predefined non-linear regression function to be mapped in the feature space (Eq. 1).

$$g(w, b) = w \cdot \emptyset(x) + b \tag{1}$$

where, x is the input vector, $\emptyset(\cdot)$ is a non-linear mapping function, w is the vector of weights and b is the bias term. To solve the problem, the regression function is expressed as a convex optimisation problem. The concept of non-linear SVR in the feature space is illustrated in Figure 3.
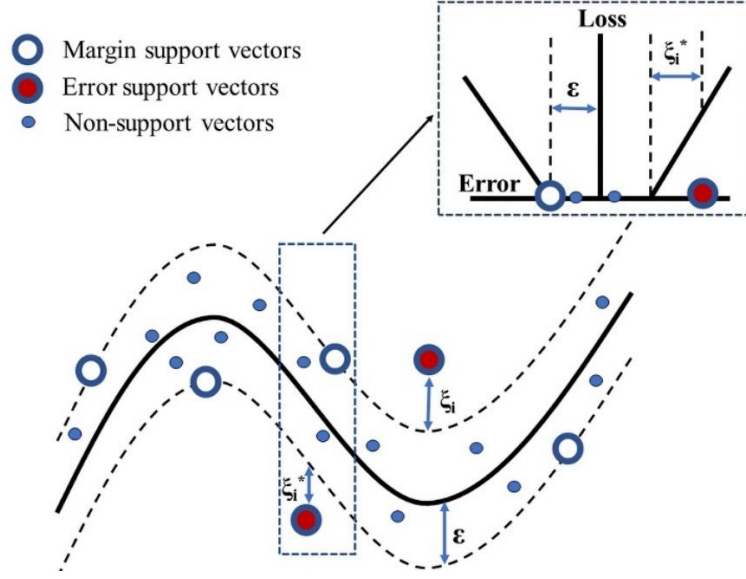
Figure 3. Non-linear SVR with ε-insensitive zone in the feature space.

In Figure 3, ξ_i and ξ_i^* are called the slack variables and these non-negative variables are introduced to estimate the deviation of training data samples lying outside the ε-insensitive zone. The effectiveness of an SVR model mainly depends upon several parameters, e.g., kernel parameters, cost parameter, and the width of the ε-insensitive zone. These parameters are mutually dependent, and hence, changing the value of one parameter could affect the value of other mutually linked parameters. The cost parameter is a positive constant and it checks for smoothness of the approximation function. A larger value of this parameter could overfit the training data and a smaller value could underfit. The ε parameter controls the width of the insensitive zone, influencing the number of support vectors, thus influencing the overall generalization ability of an SVR model.

In this study, to draw a fair comparison between the CNN and SVR, identical training and testing datasets are used for both models. We also use same parameter optimisation technique to search for the model hyperparameters.

11

2.3 Evaluation criteria

To evaluate model performance, the results from the ML predictive models are compared with the outputs from LISFLOOD-FP at 18 pre-selected ground control points (GCPs). The evaluation metrices including the root mean squared error (RMSE) and the Nash-Sutcliffe model efficiency (Nash and Sutcliffe 1970) coefficient (NSE) are used to quantitatively evaluate the performance of models. The RMSE and NSE are defined as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(O_i - P_i)^2}{N}} \tag{2}$$

$$NSE = 1 - \frac{\sum_{i=1}^{N}(O_i - P_i)^2}{\sum_{i=1}^{N}(O_i - \bar{o})^2} \tag{3}$$

where, N is the sample size, O_i and P_i are the 'observed' and the predicted values, and $\bar{o}$ is the mean of 'observed' values.

In addition, precision, recall and F1 metrics are also considered to demonstrate how precisely the model predicts 'true positives' (correctly predicted as flooded by the predictive models) to the total predicted positives (the sum of correctly and wrongly predicted as flooded) and to the total actual positives (predicted as flooded by the LISFLOOD-FP). Note that, the primary objective of this study is to investigate the efficiency of the predictive models in emulating the outputs of a 2D hydraulic model. Therefore, the output of the LISFLOOD-FP model is considered as the reference to assess their performance.

$$Precision = \frac{True\ positive}{Total\ predicted\ positive} \tag{4}$$

$$Recall = \frac{True\ positive}{Total\ actual\ positive} \tag{5}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{6}$$

Finally, we compare the errors in the modelled peak water depths between the ML models and LISFLOOD-FP against the observed values in some of the GCPs.

3. Model setup and application

3.1 Case study

The city of Carlisle is used as the case study and the 2005 flood event is simulated in this study to demonstrate the performance of the DL-based fluvial flood modelling framework. The following subsections introduce the key hydrometric data, topographic information, and assumptions to support the simulations.

3.1.1 Site description

The city of Carlisle, located at the downstream of River Eden, is one of the most flood prone areas in the UK. The study domain considered in this study covers an area of about 14.5 km^2 within the city (Fig. 4). There are three river channels within the area of interest, i.e. River Eden, Petteril and Caldew, which drive the flood dynamics. River Eden confluences with its two tributaries as it passes through the domain.

In 2005, the River Eden caused unprecedented flooding on 8th January due to persistent rainfall that started on the 7th January and was extended onto the nearby highlands draining into the river network (Roberts *et al.,* 2009). The onset of the event was slow with initial flooding occurring in the early hours of the day before the peak arrived around noon. The inundation predominantly occurred in the residential/ commercial zones along the channels and low-lying rural areas situated on the North-East.
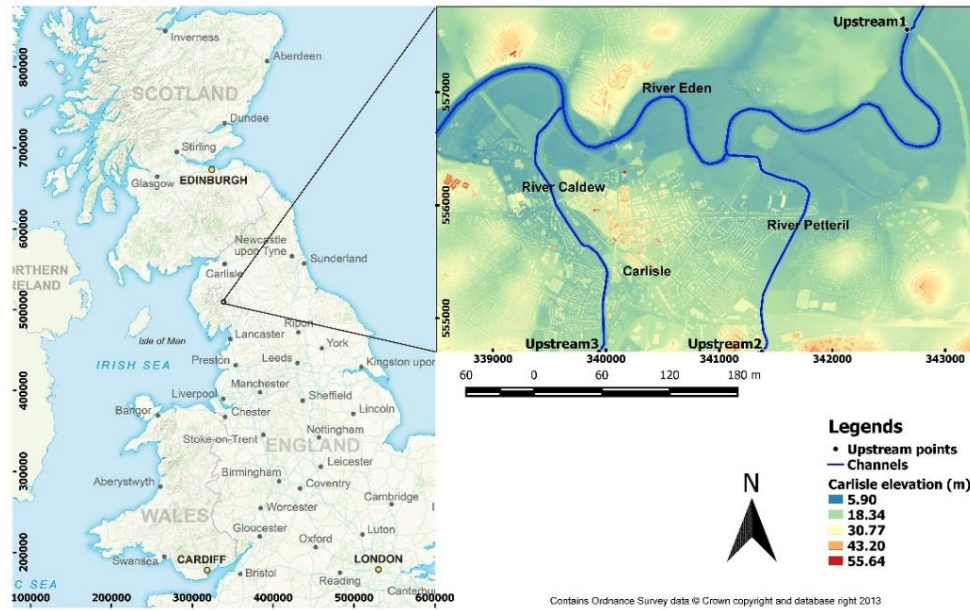
Figure 4. Study domain and topography.

3.1.2 Hydrometric and topographic data

The data required for setting up the LISFLOOD-FP model includes upstream boundary

conditions, digital elevation model (DEM) and river bathymetry. All of these data are

provided by the Environment Agency (EA) and collated at the School of Geographical

Sciences, University of Bristol, for LISFLOOD-FP modelling purposes. The DEM has a

5m spatial resolution and is further processed to remove the bridges. The gauge stations

Upstream 2 (i.e. River Petteril) and 3 (i.e. River Caldew) are located at the South edge

of the domain while Upstream 1 (i.e. River Eden) is located under the M6 bridge in the

North-East (see Fig. 4).

The available 15-minute upstream discharge hydrographs covering the whole

duration of the 2005 Carlisle flood event starts at 00:00 hours on 7th January 2005 as

time step 0 and ends at 20.15 hours on 9th January 2005. The three observed upstream

hydrographs are used to drive and test the performance of the predictive models.

Therefore, it is necessary to generate a substantial number of input (discharge) and

14

output (water depth) instances to train the models. This is done by producing additional hydrographs with various peaks and durations to represent different flooding scenarios. We randomly generated 15 synthetic hydrographs (five for each of the upstream locations) to provide enough training samples for the predictive models (Fig. 5). When generating these hydrographs, the flow in the River Eden is made to be substantially larger than the flows in the two tributaries to reflect the reality. These hydrographs are presented in Figure 5, in which "A" represents the observed hydrograph creating the 2005 flood event with the rest being the randomly generated hydrographs. A brief description of these hydrographs is presented in Table 1.
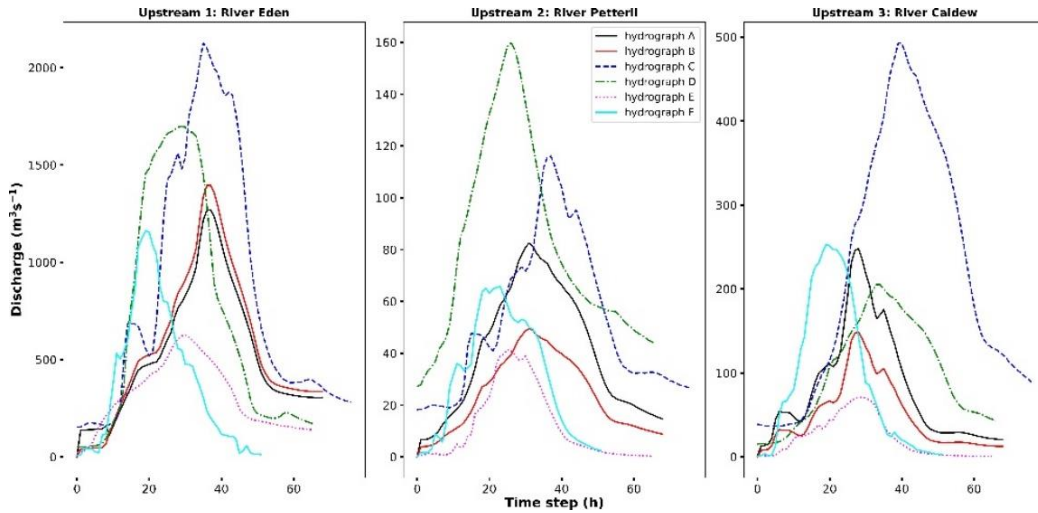

Figure 5. The synthetic and observed hydrographs used to train and test the predictive models.

Table 1. Summary of the hydrometric data used in this study.

Name	Type	Purpose	Upstream 1 peak [m³/s]	Upstream 2 peak [m³/s]	Upstream 3 peak [m³/s]
Hydrograph A	Observed	Test	1273.00	82.57	248.75
Hydrograph B	Synthetic	Train	1400.30	49.54	149.25
Hydrograph C	Synthetic	Train	2126.70	116.22	495.42
Hydrograph D	Synthetic	Train	1696.85	159.74	206.43
Hydrograph E	Synthetic	Train	626.48	41.55	70.84
Hydrograph F	Synthetic	Train	1176.17	66.66	252.90

3.2 Generating target data for training and testing

We first set up the driving files containing all of necessary parameters and data required to run the LISFLOOD-FP model. A uniform friction coefficient value of 0.055 is used across the simulation domain, as also used by Xia *et al.* (2019) in their baseline simulation over Eden catchment. The model outputs are saved in an interval of every 15 minutes. The simulation domain is defined by the 5m resolution DEM, consisting of 581,061 cells. It is worth mentioning that LISFLOOD-FP was calibrated and applied to reproduce this selected flood event and the detailed model calibration and quantification of uncertainties are not the focus of this study and will not be discussed further.

The LISFLOOD-FP model is run six times using the aforementioned inflow hydrographs to produce different flooding conditions in the site. The output files contain geographically distributed water depth values and are in raster format. Sequences of these water depth files cannot be used directly as target variables. In order to make it usable, we extract all the cell values and stack them vertically. This stacked 2D-array of water depth values has 1509 rows (samples) and 581061 columns. Each row of the array represents time varying depth and each column represents a cell within the domain (Table 2). Finally, a depth threshold of 0.2m is applied to neglect the insignificant depth values from the target data. This value is selected based on the flood depth thresholds (0.15m for transport links and 0.3m for buildings) used by Aldridge *et al.* (2016) to build their impact library for flood risk assessment in the UK. Finally, 1243 samples (out of 1509) corresponding to the synthetic inflow hydrographs are used for training whilst 266 samples corresponding to the observed inflow hydrographs are used for testing purposes.

Table 2. An example of the target data structure. Every row represents the simulated water depths of the LISFLOOD-FP at 15-minute interval for all 6 flooding scenarios.

Index	Cell 1	Cell 2	Cell 3	..	Cell 851	Cell 852	Cell 853	..	Cell 581061
1	0	0	0	..	0	0	0	..	0
2	0	0	0	..	0	0	0	..	0
:	:	:	:	..	:	:	:	..	:
143	0	0	0	..	3.12	3.12	3.11	..	0
144	0	0	0	..	3.12	3.11	3.11	..	0
:	:	:	:	..	:	:	:	..	:
1509	0	0	0	..	0	0	0	..	0

3.3 Constructing the predictive models

In this study, the proposed CNN is developed in Python programming language using Keras layer within the Tensorflow 2.1 framework. We use sequential application programming interface (Keras Sequential API) to build the model. The sequential API allows one to create models layer-by-layer and is feasible for most problems. Initially we constructed a simple network with one convolutional layer and two fully connected dense layers to stop the model from overfitting the training data and running out of GPU memory. While networks with many convolutional layers may produce finer results, they require much larger training samples and demand vast memory because of the sheer quantity of network parameters. We then increased the network complexity by adding one additional convolutional and one dense layer. This improved overall model results and was also memory efficient. The performance of the model is further enhanced by optimising the hyperparameters.

In addition to the CNN, we also develop an SVR method for comparison. Unlike CNN, the SVR method does not allow training and making predictions for all of the cells at a time. It is also not viable to calibrate an SVR model at each cell. Bermúdez *et al.* (2019) trained 25,000 models and then the predicted results were linearly

interpolated to produce the distributed water depth maps. In their study they used a

DEM at 1m spatial resolution. In this study, 500 locations are randomly selected within

the domain and a separate model is fitted to each location. 500 locations are selected

because the area of the domain is similar to that of Bermúdez *et al.* (2019) but the

spatial resolution of the DEM is 5m. Then we use the regression kriging (RK)

interpolation method (Hengl *et al.* 2007) to predict depths at the unsampled locations.

The SVR models are built using the Scikit-learn library in Python. More details on

parameter selection and optimisation is described in the following sub-sections.

3.4 Hyperparameter optimisation

The objective of hyperparameter optimisation is to find the best set of parameters of a

given model that returns the optimum performance measured on the validation data. The

challenge of optimising these parameters in DL models is a serious impediment in

scientific research and often makes it difficult to reproduce the published results

(Bergstra *et al.,* 2011). Different methods are applied to search for the best set of

hyperparameters, e.g., manual perturbation, grid search, random search, Bayesian

optimisation. Bayesian approaches are efficient in contrast to grid or random search

methods (Snoek *et al.,* 2012).

In this study we adopt the Bayesian optimisation approach to tune each of the

predictive models. In particular, the 'Hyperopt' hyperparameter optimisation library

(Bergstra *et al.,* 2015) is used to find the best set of CNN hyperparameters (i.e. filter

size, number of neurons, optimiser and batch size). We use the five synthetic flood

scenarios, i.e. associated with hydrographs B-F, for the hyperparameter optimisation

task and keep hydrograph A and the food outputs independent for testing the models.

We also optimise the 18 SVR models for each of the GCPs. This means that at this

stage we have 18 different sets of SVR parameters. We then run the SVR models

iteratively using these parameter sets to find the global parameters. The set of

parameters that produced the lowest error for all GCPs is chosen as the global SVR

parameters. This is essential because optimising 500 SVR models is not feasible, and

hence, we use a global parameter set. A summary of the searched hyperparameter sets is

presented in Table 3.

Table 3. Hyperparameters of the predictive models.

Model	Filter size	Neurons	Optimiser	Batch size
CNN	[32, 128]	[32, 256, 512]	Adam	10
	Kernel	**Cost**	**Epsilon**	**Gamma**
SVR	Radial basis	25.296	0.031	0.016

3.5 Training the models and making predictions

The CNN and baseline SVR models are trained using input (predictor) and output

(target) variables from the five flood scenarios (hydrographs B-F). The training process

is completed in two steps.

In the first step, the input and output variables are defined. The time series of

water depths are used as the target variable for the models as described earlier (Sect.

3.2). It is critical to select the minimum input variables that best characterise the

underlying input-output relationship (Galelli *et al.,* 2014). The key input variables used

in the previous ML-based inundation modelling studies include antecedent upstream

flow, downstream water level, rainfall, recursive water depth, peak discharge, and

duration (e.g. Chang *et al.* 2014, Liu and Pender 2015, Bermúdez *et al.* 2019). In this study, we explicitly use discharge values with eight antecedent time-steps (corresponds to 2 hours of LISFLOOD-FP model initialisation time) for each of the upstream gauge points and the corresponding observation time as the primary inputs for predicting water depths. We then apply an input perturbation algorithm, introduced by Breiman (2001) in his seminal work on random forests, to assess the feature importance. This is done to evaluate the importance of individual input variables. Any input variable with lower importance value (e.g. <0.1) is considered to be insignificant and therefore, can be removed from the list of input variables (i.e. dimensionality reduction). The result shows that all these input variables (28 in total) are significant with at least a value of 0.33 (Fig. 6). The version of input perturbation algorithm used in this work is adapted from Heaton *et al.* (2017). The algorithm is model independent and can be used for any supervised learning methods

Secondly, we use the early stopping criteria during the training process to stop the models from being overfitted to the training data. The SVR models are also trained using the same set of input variables at the sampled locations, sequentially. The CNN is trained on an NVIDIA Tesla P100 GPU while the SVR models are trained using Intel I5-9400 CPU with six cores.

The trained models are then saved for further testing and evaluation. The test data (Carlisle 2005 flood event) is ingested to the pre-trained models sequentially to predict water depths for the entire domain at a 15-minute interval.
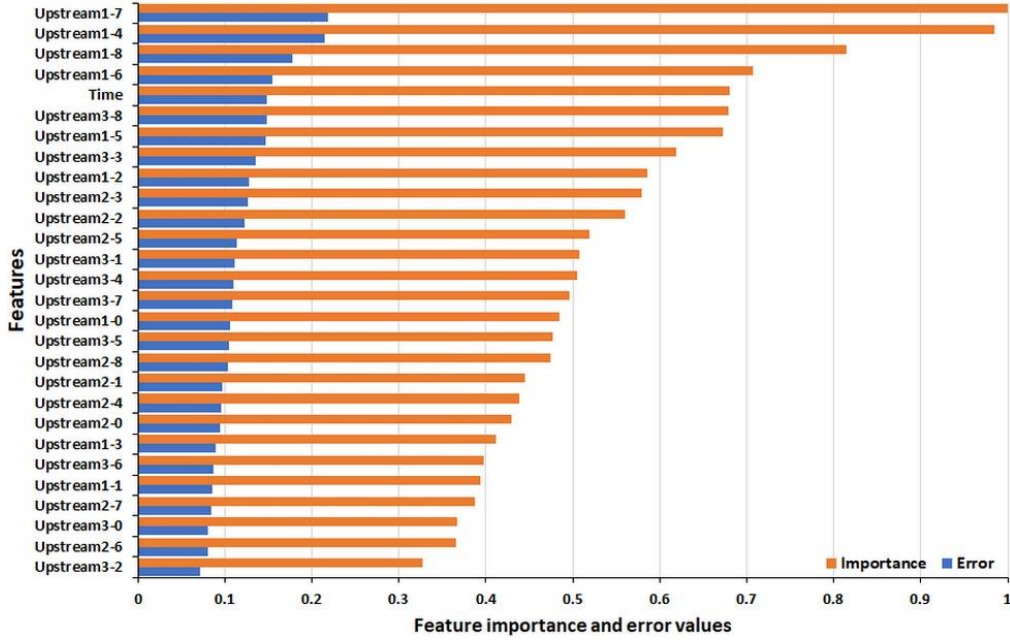
Figure 6. Input feature importance rank. Shuffling of most important features produce less accurate results. The *x*-axis shows the upstream variables in Upstream X-Y format, where, X is upstream id (1, 2, 3) and Y is the discharge time lags (0 – 8).

4. Results and discussion

4.1 Point-wise comparison between CNN, SVR and LISFLOOD-FP

The outputs of the CNN and SVR models are compared against the outputs from LISFLOOD-FP at the control locations as indicated in Figure 7 using the aforementioned multiple error measures (i.e. NSE and RMSE). The NSE value generally varies between 0 to 1, where a value of 1 represents a perfect fit between the reference and predicted data. A negative NSE value indicates that the model has failed to reproduce the test case. An RMSE value of 0 refers to a perfect fit between the modelled and reference data. The means of the error statistics are illustrated in Table 4.

21

Figure 7. Ground control locations used for assessing the model performance.

Note that cells which have water depths less than a threshold of 0.2m are considered as noise and set to 0 during the training process, the same threshold is used in the reference LISFLOOD-FP data during testing. However, it is observed that if we increase this threshold value marginally the overall predictive accuracy may be further improved for the CNN. In contrast, this action has an opposite effect on the overall SVR results. Figure 8 compares the results between CNN and SVR when different threshold values are used.

Table 4. Mean accuracies of CNN and SVR models compared at the ground control locations.

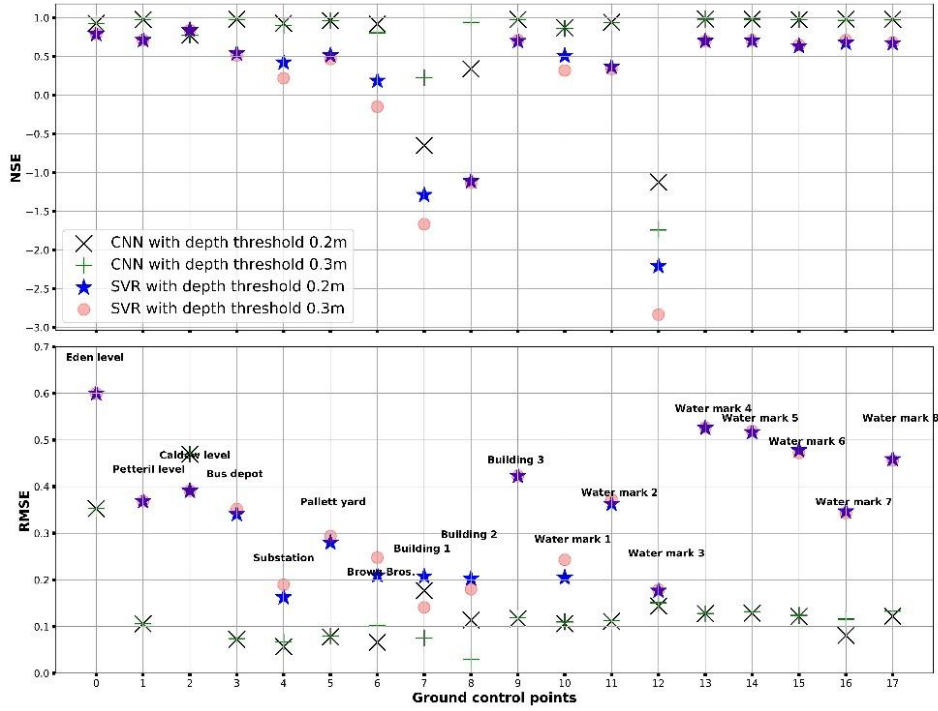|	CNN				SVR			
Threshold	0.2		0.3		0.2		0.3	
Error measure	NSE	RMSE	NSE	RMSE	NSE	RMSE	NSE	RMSE
Average value	0.707	0.142	0.744	0.138	0.243	0.345	0.143	0.350

Figure 8. Comparing the error statistics between CNN and SVR against the LISFLOOD-FP model.

The NSE and RMSE values, irrespective of threshold values, indicate that the CNN outperforms the SVR by a considerable margin. The CNN model performs better than the SVR models in all 18 locations. Furthermore, it is noticed that both the models performed badly in two locations (Building 1 and Water mark 3). However, in the case of CNN, the error statistics is improved slightly for Building 1 when the threshold value is set to 0.3 m. In fact, this improvement results in an increase of overall accuracy for CNN although the accuracy drops fractionally for most of the locations. Considering these two locations as outliers and thus, removing from the analysis results in much improved error statistics for the CNN model (NSE and RMSE values are 0.93 and 0.14, respectively). To better analyse these discrepancies, the predicted water depths are plotted at the control locations and compared with the LISFLOOD-FP references using the threshold of 0.3 m (Fig. 9). It can be clearly seen that the SVR models predicted

23

flood water (except for river levels in three locations) arriving earlier than that predicted by the CNN. The CNN model, in contrast, can satisfactorily predict maximum depths, arrival and receding times. It is remarkable that a single CNN model can make comparable predictions for half of a million cells in seconds. As it is pointed out by the accuracy table, the predicted water depths do not match well with the LISFLOOD-FP estimations at Building 1 and Water mark 3, where the maximum depth and flood duration are both low. This means that the states of these two locations change rapidly during the flood event, possibly providing insufficient training samples for the model to learn this feature. This suggestion is supported by the fact that the model performs well in other locations where the maximum depth and flood duration both become greater.
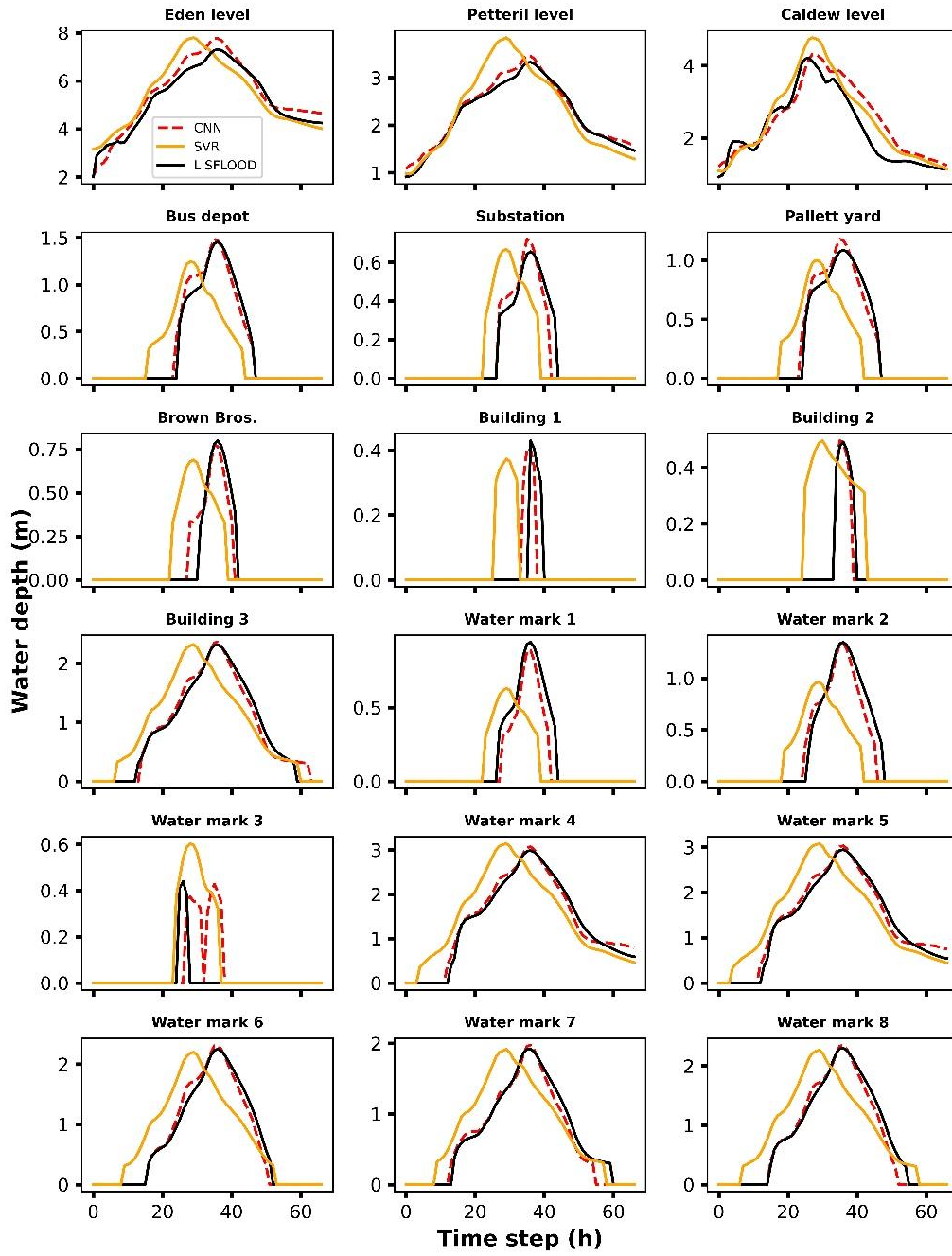
Figure 9. Comparing the water depths predicted by CNN and SVR against LISFLOOD-FP.

4.2 Comparison of inundation maps between CNN, SVR and LISFLOOD-FP

After comparing the pointwise predictions with the SVR models, we further evaluate

the performance of the ML models by comparing the spatial maps of water depths they

produce against the LISFLOOD-FP outputs. To do so, four maps at a time interval of 12 hours are selected to represent the early, growing, peak and receding periods of the flood event. We assume that the flood peak occurs at 12:00 hours on 8th January 2005 as occurred at the actual event. To reflect how the CNN, SVR and LISFLOOD-FP water depth maps compare with each other, descriptive statistics are calculated and presented in Table 5.

Table 5. Descriptive statistics for water depths estimated using CNN and LISFLOOD-FP during different stages of the event.

Date	Model	Max (m)	Mean (m)	St. dev. (m)
12:00	CNN	5.28	0.14	0.68
7th Jan 2005	SVR	4.94	0.21	0.69
	LISFLOOD-FP	4.95	0.12	0.62
00:00	CNN	6.99	0.42	1.04
8th Jan 2005	SVR	7.45	0.59	1.21
	LISFLOOD-FP	6.61	0.38	1
12:00	CNN	8.42	0.83	1.48
8th Jan 2005	SVR	8.01	0.72	1.35
	LISFLOOD-FP	8.1	0.81	1.44
00:00	CNN	7.01	0.49	1.09
9th Jan 2005	SVR	6.35	0.42	0.98
	LISFLOOD-FP	7.19	0.53	1.14

It can be noticed that the CNN model produced results that are consistent with the LISFLOOD-FP outputs for all four flood stages and results are almost identical during the peak. It is also observed that the CNN model slightly overestimated the water depths from the early stage to the peak then underestimated with equal measure during the recession.

Spatial maps are also plotted in Figure 10 to visualise the range of water depths, in which the first column shows the CNN generated water depth maps and the second column depicts outputs of the LISFLOOD-FP model, and corresponding error maps are

presented in Figure 11. The depth maps generated by the CNN effectively capture the flooded zones and the error ranges are also considered to be reasonable. However, there is a high level of discrepancy (up to 1.8 meters) on the results predicted on 8th January 00:00 hours. The error mainly occurs near to the Sheepmount athletics stadium areas in the West side of the domain. These are the areas that sit behind embankments that are only just overtopped in the LISFLOOD-FP model, and once the overtopping occurred, they deepened very quickly. The CNN may have limitation in capturing these rapidly varying non-linear dynamics relative to the rest of the floodplain where the flood dynamics is less dramatic. The potential reasons for this will need further investigation in the future research.
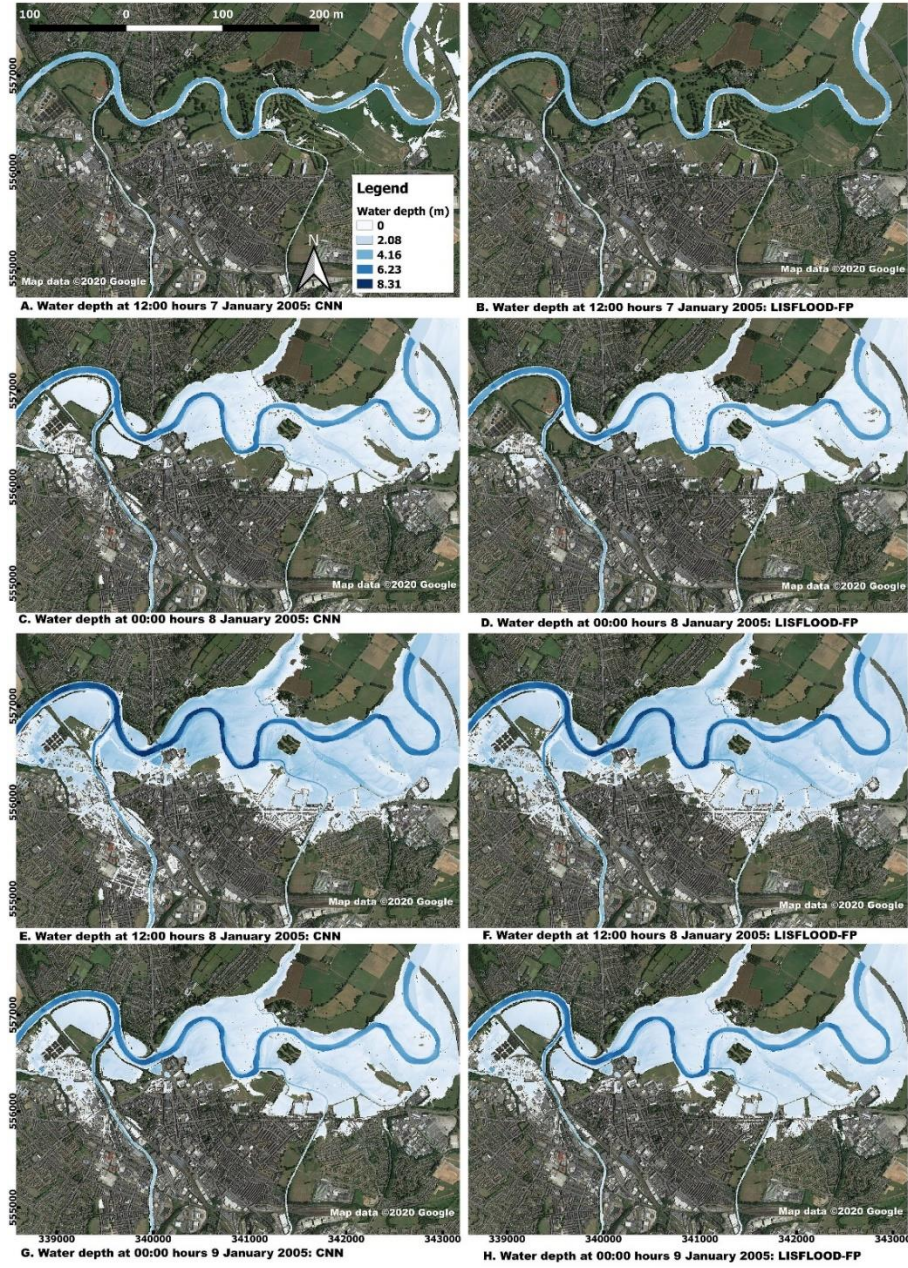
Figure 10. The maps for CNN and LISFLOOF-FP generated water depths during different stages of the flood event.

Since the CNN model overestimates the water depths during the rising period, certain locations of the floodplain may be overly flooded. It is possible to further filter the results by applying a 99% confidence level to the model errors. A 99% confidence interval reduces the maximum error of predictions on January 8th 00:00 hours to 1 meter and 97% of the error resides within 0 to 0.2-meter range. A 95% confidence interval

further reduces the maximum error to 0.4 meters. Fig. 12 shows the distribution of errors with a 99% confidence interval.
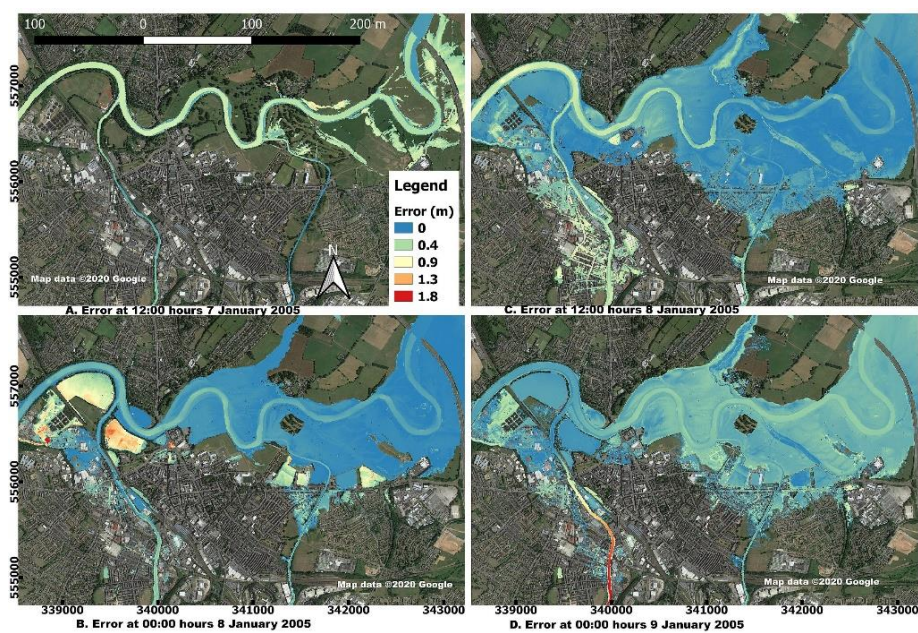

Figure 11. Spatial distribution of the error. The difference between CNN generated flood depths and the outputs from LISFLOOD-FP.
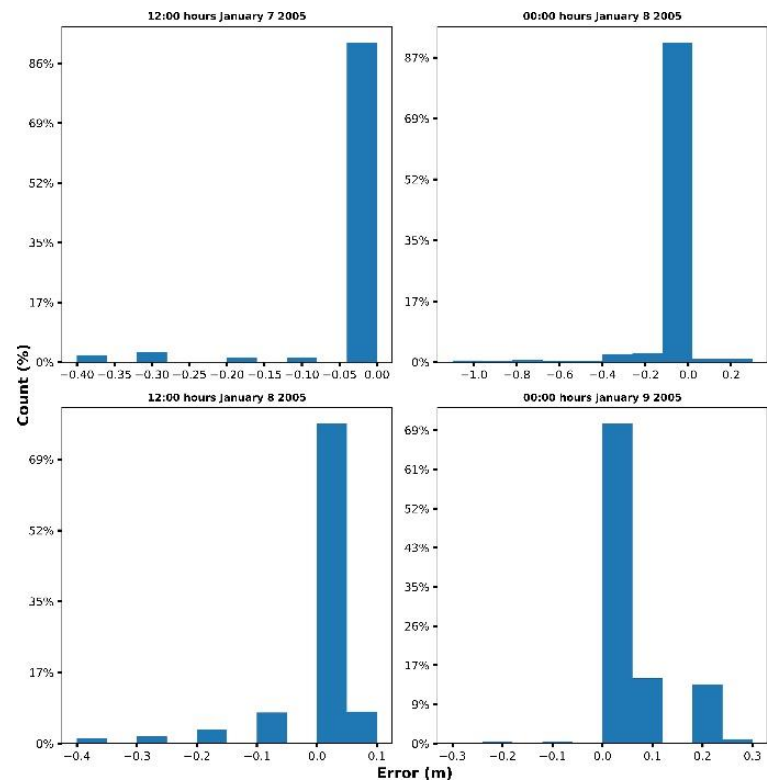

Figure 12. Error distribution of the CNN predicted water depths.

The flood depth map values predicted by the SVR models at the sampled location are spatially interpolated using the RK method. The interpolated depth map (top) during the peak and corresponding error map (bottom) are depicted in Figure 13. From the error map, the depth errors predicted by the SVR are significantly higher than those predicted by the CNN model (Fig 11). Therefore, it may be concluded that the results produced by the CNN method is significantly higher in accuracy and the CNN model is more robust in predicting depth maps for flood inundation.
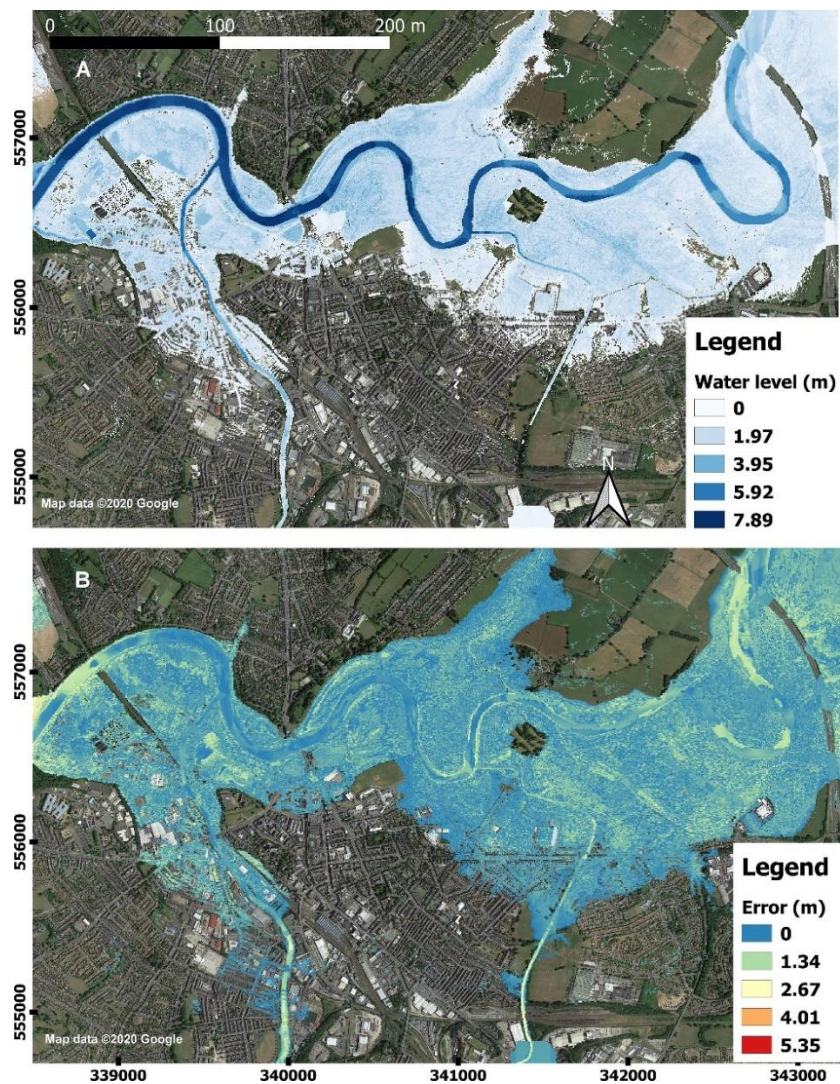

Figure 13. The SVR generated water depths and corresponding error maps for 8th January 12:00 hours 2005.

Finally, out of the 18 GCPs used for quantitative error analysis, observed maximum flood depths are available for 13 of them where we can directly compare the modelled maximum water depths against the observed data. The results are plotted in Figure 14, in which the modelled maximum depths are observed to be close to the observed values at most of the locations. The CNN model outputs match well with the LISFLOOD-FP predictions whilst the SVR method creates less satisfactory results in certain locations. It should be noted that the predictive models are trained using the outputs from the LISFLOOD-FP model, and the embedded error and uncertainty in the LISFLOOD-FP outputs are propagated to the predictive models which also have underlying model/parameter uncertainties. It is, therefore, essential to calibrate the 2D model prior to generating target data for the predictive models to reduce the error.
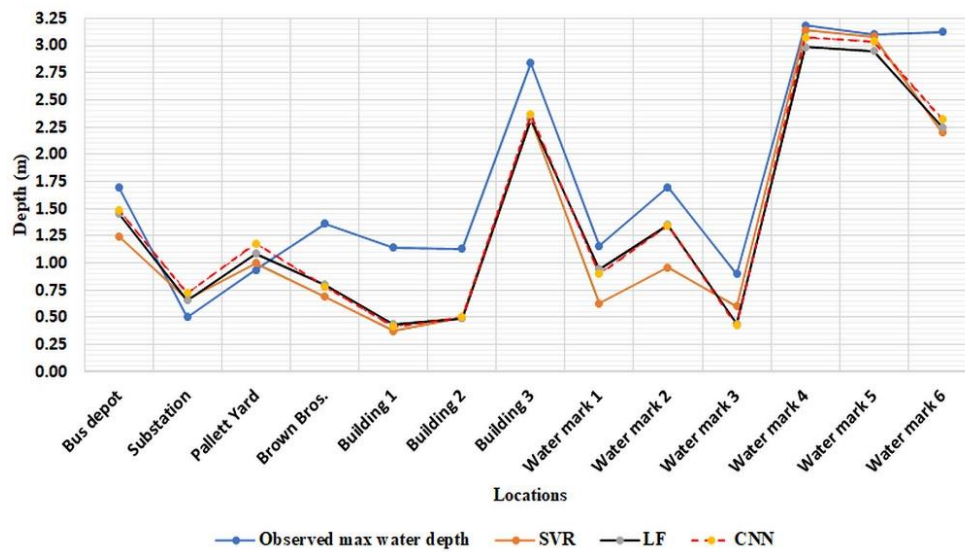

Figure 14. Comparison of maximum depths between the observed and modelled data.

4.3 Classification accuracy

In this subsection we derive the flood extents and estimate classification accuracy yielded by the CNN and SVR models by quantifying how many cells are correctly

classified as flooded by the models at different stages of the flood event, in comparison with the LISFLOOD-FP predictions. First, a threshold value of 0.3 was used to delineate wet-dry cells, and then recall, precision and F1 scores are calculated to quantify how well the predictive model outputs match the spatial predictions of the LISFLOOD-FP. Table 6 summaries the classification accuracies of the CNN and SVR models against LISFLOOD-FP.

Table 6. Classification accuracy scores of the CNN and SVR models against LISFLOOD-FP during flood initiation, growing, peak and recession stages.

Date	Precision		Recall		F1	
	CNN	SVR	CNN	SVR	CNN	SVR
12:00 7th Jan 2005	0.70	0.92	1.00	0.22	0.83	0.36
00:00 8th Jan 2005	0.86	0.98	0.98	0.69	0.92	0.81
12:00 8th Jan 2005	0.96	0.95	0.99	0.92	0.98	0.93
00:00 9th Jan 2005	0.99	0.89	0.96	0.90	0.97	0.89

A higher precision value implies that most of the cells that are predicted as flooded by the CNN and SVR models are also classified as flooded by LISFLOOD-FP. On the other hand, a higher recall value means cells that are classified as flooded are well captured by the CNN and SVR models. F1 is the harmonic mean of recall and precision metrics and defines how well the ML model predictions match the LISFLOOD-FP outputs. A score of 1 means a perfect fit between the models. The tendency of the CNN model to overestimate water depths during flood initiation to peak period is reflected in the recall values. Almost all the cells classified by LISFLOOD-FP as flooded are well captured by the CNN model during these periods. In contrast, nearly 100% cells classified by the CNN model as flooded are correctly captured by LISFLOOD-FP in the recession stage. This is due to the tendency of the CNN model to

underestimate the flooding during the recession stage. Overall, the CNN model agrees well with LISFLOOD-FP at a high level of accuracy throughout the whole flood event and performs better than the SVR model.

To further compare the results, the CNN predicted flood extents are overlaid on the LISFLOOD-FP results (Fig. 15). It is now possible to evidently identify the patches (marked in rectangular boxes) where the CNN model mostly did not match to the hydraulic model. During the flood initiation phase, the areas where overestimations largely occurs are found to be near to the upstream boundary 1 (near M6 motorway). It might be possible to define a minimum patch size to further filer the flood zones. Nevertheless, the predicted flood extents are highly accurate, compared with the reference LISFLOOD-FP extents except for some inconsistencies in certain small patches.
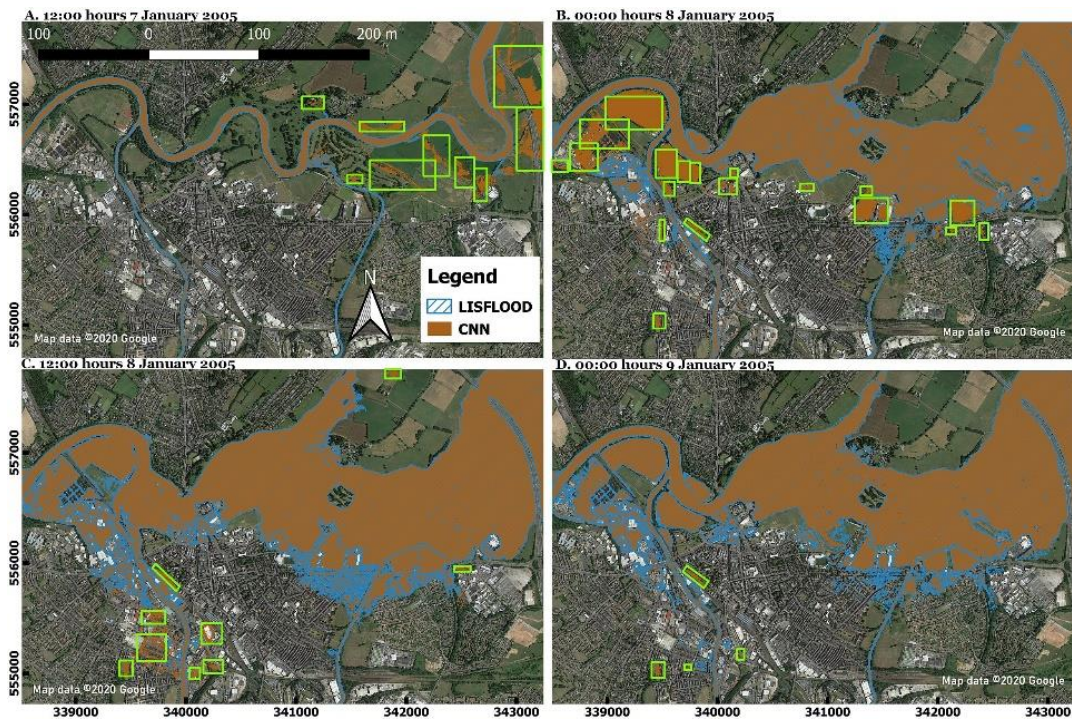

Figure 15. Comparing the CNN and LISFLOOD-FP derived flood extents.

4.4 Computational time

Computational cost is one of the key constraints of the 2D hydraulic models for real-time applications. In this study, the CNN model that is developed to predict flood depth maps reduces the computational time by many folds. It takes only a few seconds to generate the sequential water depth maps for the entire domain and a few minutes to train the model. It should be noted that, the training cost is only needed once. Once it is trained, the CNN model can be used to efficiently predict different flood scenarios. In Table 7, a comparison of the computational costs between the CNN and LISFLOOD-FP models is presented.

Table 7. The runtimes of the CNN and LISFLOOD-FP models for simulating the 2005 Carlisle flood event.

Model	Train time	Total time required to simulate entire event	Output format	Computing device
CNN	~2-4 minutes	~4 minutes	Raster	NVIDIA Tesla P100 GPU
LISFLOOD-FP	__	~92 minutes	Raster	Intel I5-9400 2.90GHz CPU

5. Conclusions

The need for evidence-based flood management is greater than ever before due to rapid urbanisation and climate change that have already led to increased flood risk across the world. In this context, fast, reliable and robust modelling tools for real-time flood prediction/forecasting is important for assessing the multidimensional social and economic impacts of and providing reliable forecasts to enhance societal resilience to flooding. This work introduces a deep CNN approach for rapid fluvial flood modelling

that can potentially be leveraged for operational flood nowcasting or forecasting. The idea underpinning the proposed study is that cell-based water depths in a floodplain are a function of time varying discharge and the time of observation at the upstream. Therefore, a non-linear function can be fitted between sequences of historical geographically distributed water levels and observed discharge to predict water levels for the future flood events. The results show that the CNN model can effectively emulate the outputs (i.e. water depth) of a 2D hydraulic model to a high accuracy. This paper also shows that a single CNN model has the capability to make predictions for a domain consisting hundreds of thousands of cells. Due to the much-reduced computational cost, performance and simplicity, the proposed method offers a promising tool for real-time nowcasting/forecasting of flood inundation. In the future studies, model uncertainties should be quantified, and similar methods could be investigated for pluvial flooding.

Data availability

The models constructed in this paper and reproduceable data have been made available through https://github.com/SRKabir/Rapid_FloodModelling_CNN.

References

Abdeljaber, O., Avci, O., Kiranyaz, M.S., Boashash, B., Sodano, H., and Inman, D.J., 2018. 1-D CNNs for structural damage detection: Verification on a structural health monitoring benchmark data. *Neurocomputing*, 275, 1308–1317.

Aldridge, T., Gunawan, O., Moore, R.J., Cole, S.J., and Price, D., 2016. A surface water flooding impact library for flood risk assessment. *E3S Web of Conferences*, 7, 18006.

de Almeida, G.A.M., Bates, P., Freer, J.E., and Souvignet, M., 2012. Improving the stability of a simple formulation of the shallow water equations for 2-D flood modeling. *Water Resources Research*, 48 (5).

Amarnath, G., Umer, Y.M., Alahacoon, N., and Inada, Y., 2015. Modelling the flood-risk extent using LISFLOOD-FP in a complex watershed: case study of Mundeni Aru River Basin, Sri Lanka. *Proc. IAHS*, 370, 131–138.

Bates, P.. and De Roo, A.P.., 2000. A simple raster-based model for flood inundation simulation. *Journal of Hydrology*, 236 (1–2), 54–77.

Bates, P.D., Horritt, M.S., and Fewtrell, T.J., 2010. A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling. *Journal of Hydrology*, 387 (1–2), 33–45.

Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B., 2011. Algorithms for Hyper-Parameter Optimization. *In*: K.Q.W. J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, ed. *Advances in Neural Information Processing Systems*. 2546–2554.

Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., and Cox, D.D., 2015. Hyperopt: a Python library for model selection and hyperparameter optimization. *Computational Science & Discovery*, 8 (1), 014008.

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

Bermúdez, M., Cea, L., and Puertas, J., 2019. A rapid flood inundation model for hazard mapping based on least squares support vector machine regression. *Journal of Flood Risk Management*, 12 (S1), e12522.

Bhola, P., Leandro, J., and Disse, M., 2018. Framework for Offline Flood Inundation Forecasts for Two-Dimensional Hydrodynamic Models. *Geosciences*, 8 (9), 346.

Breiman, L., 2001. Random Forests. *Machine Learning*, 45 (1), 5–32.

Chang, L.-C., Amin, M., Yang, S.-N., and Chang, F.-J., 2018. Building ANN-Based Regional Multi-Step-Ahead Flood Inundation Forecast Models. *Water*, 10 (9),

1283.

Chang, L.-C., Chang, F.-J., Yang, S.-N., Kao, I.-F., Ku, Y.-Y., Kuo, C.-L., and Amin, I., 2018. Building an Intelligent Hydroinformatics Integration Platform for Regional Flood Inundation Warning Systems. *Water*, 11 (1), 9.

Chang, L.-C., Shen, H.-Y., and Chang, F.-J., 2014. Regional flood inundation nowcast using hybrid SOM and dynamic neural networks. *Journal of Hydrology*, 519, 476–489.

Chang, L.-C., Shen, H.-Y., Wang, Y.-F., Huang, J.-Y., and Lin, Y.-T., 2010. Clustering-based hybrid inundation model for forecasting flood inundation depths. *Journal of Hydrology*, 385 (1–4), 257–268.

Galelli, S., Humphrey, G.B., Maier, H.R., Castelletti, A., Dandy, G.C., and Gibbs, M.S., 2014. An evaluation framework for input variable selection algorithms for environmental data-driven models. *Environmental Modelling and Software*.

Heaton, J., McElwee, S., Fraley, J., and Cannady, J., 2017. Early stabilizing feature importance for TensorFlow deep neural networks. *In*: *2017 International Joint Conference on Neural Networks (IJCNN)*. 4618–4624.

Hengl, T., Heuvelink, G.B.M., and Rossiter, D.G., 2007. About regression-kriging: From equations to case studies. *Computers & Geosciences*, 33 (10), 1301–1315.

Jhong, Y.-D., Chen, C.-S., Lin, H.-P., and Chen, S.-T., 2018. Physical Hybrid Neural Network Model to Forecast Typhoon Floods. *Water*, 10 (5), 632.

Kabir, S., Patidar, S., and Pender, G., 2020. A Machine Learning Approach for Forecasting and Visualizing Flood Inundation Information. *Proceedings of the Institution of Civil Engineers - Water Management*, https://doi.org/10.1680/jwama.20.00002.

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., and Inman, D.J., 2019. 1D Convolutional Neural Networks and Applications: A Survey.

Kiranyaz, S., Ince, T., and Gabbouj, M., 2016. Real-Time Patient-Specific ECG Classification by 1-D Convolutional Neural Networks. *IEEE Transactions on Biomedical Engineering*, 63 (3), 664–675.

Kiranyaz, S., Ince, T., Hamila, R., and Gabbouj, M., 2015. Convolutional Neural

Networks for patient-specific ECG classification. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, 2015, 2608–2611.

Van Der Knijff, J.M., Younis, J., and De Roo, A.P.J., 2010. LISFLOOD: a GIS-based distributed model for river basin scale water balance and flood simulation. *International Journal of Geographical Information Science*, 24 (2), 189–212.

Komi, K., Neal, J., Trigg, M.A., and Diekkrüger, B., 2017. Modelling of flood hazard extent in data sparse areas: a case study of the Oti River basin, West Africa. *Journal of Hydrology: Regional Studies*, 10, 122–132.

Krizhevsky, A., Sutskever, I., and Hinton, G.E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. *In*: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. Red Hook, NY, USA: Curran Associates Inc., 1097–1105.

Leedal, D., Neal, J., Beven, K., Young, P., and Bates, P., 2010. Visualization approaches for communicating real-time flood forecasting level and inundation information. *Journal of Flood Risk Management*, 3 (2), 140–150.

Lin, G.-F., Lin, H.-Y., and Chou, Y.-C., 2013. Development of a real-time regional-inundation forecasting model for the inundation warning system. *Journal of Hydroinformatics*, 15 (4), 1391–1407.

Liu, Y. and Pender, G., 2015. A flood inundation modelling using v-support vector machine regression model. *Engineering Applications of Artificial Intelligence*, 46, 223–231.

Nash, J.E. and Sutcliffe, J.V., 1970. River flow forecasting through conceptual models part I — A discussion of principles. *Journal of Hydrology*, 10 (3), 282–290.

Neal, J., Dunne, T., Sampson, C., Smith, A., and Bates, P., 2018. Optimisation of the two-dimensional hydraulic model LISFOOD-FP for CPU architecture. *Environmental Modelling & Software*, 107, 148–157.

Neal, J., Schumann, G., and Bates, P., 2012. A subgrid channel model for simulating river hydraulics and floodplain inundation over large and data sparse areas. *Water Resources Research*, 48 (11).

de Paiva, R.C.D., Buarque, D.C., Collischonn, W., Bonnet, M.-P., Frappart, F., Calmant, S., and Bulhões Mendes, C.A., 2013. Large-scale hydrologic and hydrodynamic modeling of the Amazon River basin. *Water Resources Research*, 49 (3), 1226–1243.

Raghavendra. N, S. and Deka, P.C., 2014. Support vector machine applications in the field of hydrology: A review. *Applied Soft Computing*, 19, 372–386.

Roberts, N.M., Cole, S.J., Forbes, R.M., Moore, R.J., and Boswell, D., 2009. Use of high-resolution NWP rainfall and river flow forecasts for advance warning of the Carlisle flood, north-west England. *Meteorological Applications*, 16 (1), 23–34.

Sanders, B.F. and Schubert, J.E., 2019. PRIMo: Parallel raster inundation model. *Advances in Water Resources*, 126, 79–95.

Shen, H.-Y. and Chang, L.-C., 2013. Online multistep-ahead inundation depth forecasts by recurrent NARX networks. *Hydrol. Earth Syst. Sci.*, 17 (3), 935–945.

Snoek, J., Larochelle, H., and Adams, R.P., 2012. Practical Bayesian Optimization of Machine Learning Algorithms.

Wu, Z., Zhou, Y., Wang, H., and Jiang, Z., 2020. Depth prediction of urban flood under different rainfall return periods based on deep learning and data warehouse. *Science of The Total Environment*, 716, 137077.

Xia, X., Liang, Q., and Ming, X., 2019. A full-scale fluvial flood modelling framework based on a high-performance integrated hydrodynamic modelling system (HiPIMS). *Advances in Water Resources*, 132, 103392.

Yamazaki, D., Kanae, S., Kim, H., and Oki, T., 2011. A physically based description of floodplain inundation dynamics in a global river routing model. *Water Resources Research*, 47 (4).

Zihlmann, M., Perekrestenko, D., and Tschannen, M., 2017. Convolutional Recurrent Neural Networks for Electrocardiogram Classification.