ICFM 10
Adapting to Global Change:
Innovative Approaches to Flood Management and Resilience
May 20-22, 2026 | Western University, London, Ontario, Canada

FLOOD HAZARD FORECASTING WITH CONVOLUTIONAL NEURAL NETWORK BASED SPATIO-TEMPORAL FLOOD PREDICTION

Yoshua Widijatmoko[1], Xilin Xia[1], and Nigel Wright[1]
School of Engineering, University of Birmingham, Birmingham, United Kingdom[1]
E-mail: yxw1385@student.bham.ac.uk

ABSTRACT

Real time flood hazard forecasting is critical part of early warning systems, and machine learning approach has emerged as viable way for rapid flood prediction, but generalisability and explainability is still a challenge. This study aims to develop a surrogate model for flood prediction using machine learning approach that can generalise to different resolution. The model is developed using a physics informed neural network (PINN), and a physics guided approach to attain generalisability. Flood events of Birmingham and Carlisle in England are used as study area to develop the model. The model is trained and validated using flood model at 50m resolution and tested on 10m resolution. The results show that the model manages to maintain the performance at lower resolution on higher resolution, and is faster than hydrodynamic model, whilst also being computationally inexpensive to be developed at commercially accessible platforms.

KEYWORDS: Rapid flood prediction, Surrogate models, CNN, Physics informed

1 INTRODUCTION

Over the years, a machine learning approach has offered an option to speedup the flood modelling process to real-time level at the cost of lower accuracy, but generalisability and nature of the model hinder its application (Karim et al., 2023; Teng et al., 2017). Fraehr et al. (2024) developed a model for converting flood map from low to high resolution, and a surrogate model for direct flood prediction. Model for converting resolution is flexible, while surrogate model has better speed, but both apply black box approach that is difficult to explain (Fraehr et al., 2024). Despite the drawbacks, surrogate model offers practicality for rapid flood prediction, especially for early warning system (Cao et al., 2024).

The development a flexible surrogate model is reflected through much research in the area (Karim et al., 2023). However, none has been as versatile as a hydrodynamic model that can generalise to different case or resolution. While the attempt to explain such model led to approach such as physics informed neural network (PINN) that aims to guide black box model using the continuity equation of physics law as part of the training loss function (Donnelly et al., 2024). However, the guiding model to follow physics law can also be done via imposing physics knowledge in framework formulation (Muralidhar et al., 2020).

Ideally, a surrogate model should be able to predict the temporal evolution of flood water depths and velocities. It should also generalise to any case at any resolution, and operate in line with the governing physics law, while maintaining a much lower computational burden. Such a surrogate model is beneficial for preliminary rapid flood prediction, especially within the framework of early warning system. To this end, this study seeks to contribute to the field of flood hazard forecasting by exploring surrogate model generalisability to different resolution to answer the following research questions: Can a surrogate model generalise to different resolution of the same case while maintaining performance? How to facilitate the development of such a model?

To answer these questions, a new framework of surrogate model will be developed by applying PINN and physics guided approach. Both physics approaches leverage physics knowledge on model

framework to allow generalisability across resolution. The rest of the paper is organised as follows: section 2 will discuss the materials and methodology applied, section 3 shows the result and discussion of the model, and finally conclusion is drawn in section 4.

2 MATERIALS AND METHODS

2.1 Study Area and Scenarios

Surrogate model is typically developed with single flood driver, either from rainfall or river flow (Fraehr et al., 2024; Situ et al., 2024), but in this study both are considered and the area of Birmingham and Carlisle in England are chosen as study area. From Birmingham, the watershed of River Rea is chosen, it has drainage area of 88 km^2 and flood is driven solely by rainfall. While from Carlisle, urban area along the River Eden is chosen, it has a domain of 109 km^2 and driven by both rainfall as well as river flow from four upstream. Figure 1 shows the location of both study area.

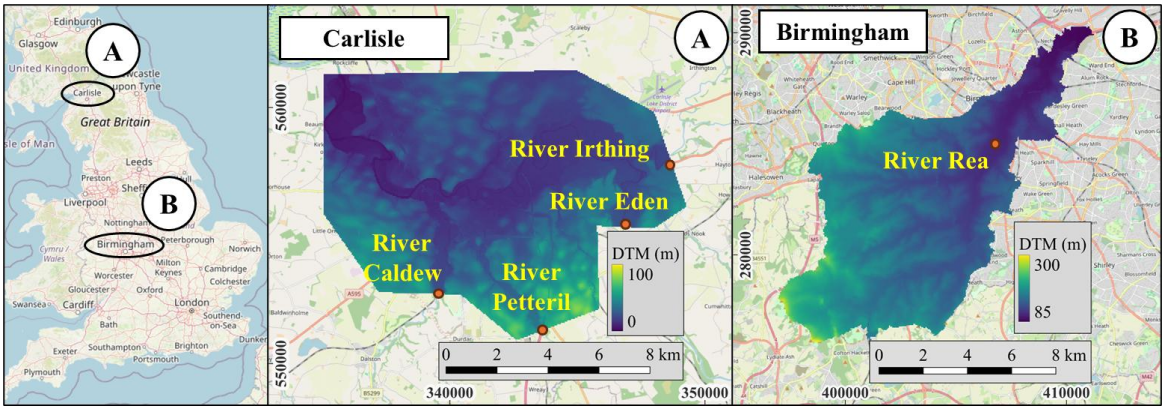

Figure 1: Location of Study Area

Aside from historical data, synthetic data is commonly used to increase availability and variability (Liao et al., 2023; Wang et al., 2024), and it is often generated from historical analysis or design pattern (Situ et al., 2024; Song et al., 2025). In this study, terrain and boundary condition data is collected from UK Department of Environment Food and Rural Affairs, synthetic data is generated from return period analysis, and flood is simulated using SynxFlow that solve 2D Shallow Water Equations (Xia and Ming, 2025). The data used in the study are summarised in Table 1.

Synthetic hyetograph is applied in pluvial and compound flood cases, but not for dry state and fluvial cases. Friction coefficient is uniformly set in all simulation to simplify modelling process as the focus of this study is replicating hydrodynamic modeller performance. Moreover, only simulation output from the 24 hours duration of synthetic flood event is included as data pool to reduce computational burden of training stage while maintaining key learning feature. Furthermore, data within training, validation, and testing set are engineered to optimise model development. Training set contain wide range of flood case and return periods to encourage generalisability. Validation set is engineered to validate performance at interpolating storm conditions. While testing set is to test model performance at unknown resolution.

2.2 Surrogate Model

In this study, fully convolutional network (CNN) is used as the framework since it allows processing of different input sizes (Wang et al., 2024) and therefore also applicable for different resolution since different resolution can be interpreted as expressing the same case over different size. The underpinning principle applied within the model is image-to-image translation where image output

and input are different but still maintain a level of resemblance (Guo et al., 2021). Hence the input and output of the model is limited to 2D format, and 1D inputs are processed into 2D spatial data that resembles hydrodynamic output to allow model to learn better. Accordingly, hyetograph and hydrograph are processed into 2D spatial data by imposing inversed min-max scaled DTM as spatial distribution pattern. This enables lower elevation to have more distribution weight than higher elevation that resemble hydrodynamic model water depth output since area near river channel will have deeper water than flood plain. Hyetograph is processed into rain intensity distribution map by calculating total rainfall inflow and distribute it according to the distribution pattern. While hydrograph is processed into river flow distribution map by identifying elevation bellow inlet surface elevation as inundation extend, then scaling distribution pattern within extend with discharge value. Hydrograph transformation does not maintain mass conservation due to technical difficulties and is left for the machine learning framework to resolve. Figure 2 shows example of rain intensity distribution map and river flow distribution map.

Table 1 Summary of Data Used in the Study

Description	Birmingham	Carlisle
Data Collected		
Terrain data	2022 LiDAR DTM at 2 m resolution	2022 LiDAR DTM at 2 m resolution
Daily rainfall	2003-2024 Saltley, Waseley Hills, Alvechurch, and	1997-2024 Willow Holme and Cumwhinton
15 min rainfall	2003-2024 Frankley	1997-2024
15 min hydrograph		1997-2024 Cummersdale - River Caldew (E: 339489; N: 552735)
		2006-2024 Newbiggin Bridge - River Petteril (E: 343561; N: 551259)
		1996-2024 Great Corby - River Eden (E: 346825; N: 555339)
		1975-2024 Greenholme - River Irthing (E: 348618; N: 558072)
Synthetic Data		
Terrain data	Averaging DTM data to 50m and 10m resolution	Averaging DTM data to 50m and 10m resolution
Return Period	2, 5, 8, 10, 20, 50, 80, 100, 200, 500 years	2, 5, 8, 10, 20, 50, 80, 100, 200, 500 years
Simulation		
Output	Hourly Water Depth (m) and Velocity at x and y directions (m/s)	Hourly Water Depth (m) and Velocity at x and y directions (m/s)
Total simulation time	72 hours (24 leads, 24 floods, 24 lags)	72 hours (24 leads, 24 floods, 24 lags)
Manning's (n)	0.055 $sm^{(-1/3)}$	0.055 $sm^{(-1/3)}$
Flood Scenarios	70 Pluvial Floodings, 56 Dry State	70 Compund Floodings, 56 Fluvial Floodings
Data Pool		
Training Set	112 data	112 data
- Resolution	50 m	50 m
- Return period	2, 5, 10, 20, 50, 100, 200, 500 years	2, 5, 10, 20, 50, 100, 200, 500 years
- Scenarios	56 Pluvial Floodings, 56 Dry State	56 Compund Floodings, 56 Fluvial Floodings
Validation Set	14 data	14 data
- Resolution	50 m	50 m
- Return period	8, 80 years	8, 80 years
- Scenarios	14 Pluvial Floodings	14 Compund Floodings
Testing Set	14 data	14 data
- Resolution	10 m	10 m
- Return period	8, 80 years	8, 80 years
- Scenarios	14 Pluvial Floodings	14 Compund Floodings

Aligning with physics guided approach (Muralidhar et al., 2020), this study proposes context convolution mechanism where terrain data is projected into context vectors that later are used as kernel weights within a convolutional layer operation. Moreover, to embody the concept that flood depth

increases rapidly within river channel and slower as it flows out to flood plain, a custom output function is proposed and shown in Eqs. (1). Figure 3 shows the framework used in this study.

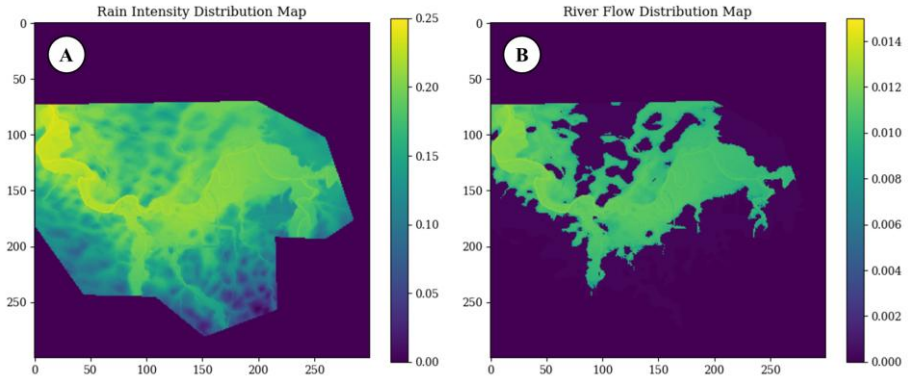

Figure 2: Rain Intensity Distribution Map (a) and River flow Distribution Map (b) of Carlisle Study Area

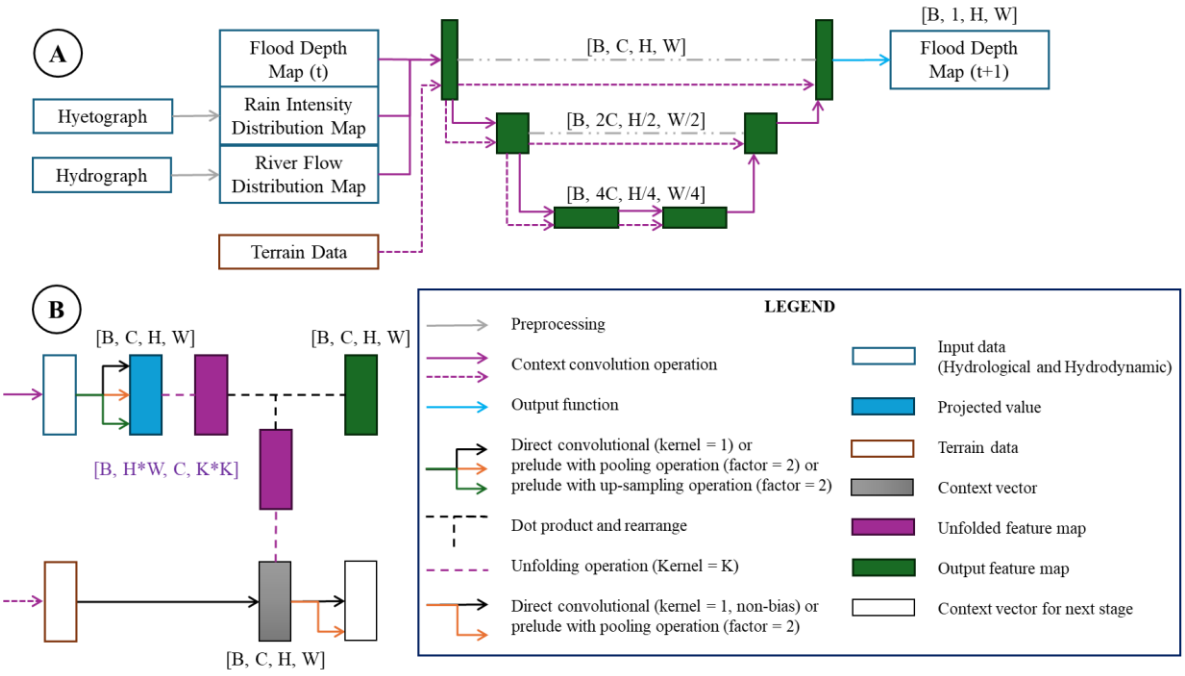

Figure 3: Surrogate Model Framework (a) and Context Convolutional Mechanism (b)

$$f(\boldsymbol{X}) = ReLU\left(W_2 \frac{\ln(ReLU(\boldsymbol{X}) + 1)}{\ln(W_1)}\right) \quad (1)$$

Where $ReLU$ is rectified linear function, $\boldsymbol{X}$ is the input tensor, W_1 and W_2 are learnable weights. The model receive input that consist of water depth map, rain intensity distribution map, and river flow distribution map of current time step together with DTM data to predict the next time step of water depth map. Furthermore, the model works autoregressively with hourly time step where the output of previous step is used as input for current step until the prescribe time step is finished.

To further develop model performance, PINN is also implemented by directly applying law of conservation of mass in shallow water equations (Eleuterio, 2001) to form the PINN loss function (L_{PINN}). Eqs. (2) - (3) shows the PINN loss function implemented. The model is also trained separately using

Mean Square Error (MSE) loss function (L_{MSE}) to explore the effectiveness of PINN (Donnelly et al., 2024).

$$S_{(i,j)} = \frac{1}{\Delta t}\left(h^{t+1}_{(i,j)} - h^{t}_{(i,j)}\right) + \frac{1}{\Delta x}\left(hu^{t}_{\left(i+\frac{1}{2},j\right)} - hu^{t}_{\left(i-\frac{1}{2},j\right)}\right) + \frac{1}{\Delta y}\left(hv^{t}_{\left(i,j+\frac{1}{2}\right)} - hv^{t}_{\left(i,j-\frac{1}{2}\right)}\right) \quad (2)$$

$$L_{PINN} = \frac{1}{|\Omega|}\sum_{(i,j)\in\Omega}\left(\left(h^{t+1}_{(i,j)} - \hat{h}^{t}_{(i,j)}\right) + \frac{\Delta t}{\Delta x}\left(\hat{h}u^{t}_{\left(i+\frac{1}{2},j\right)} - \hat{h}u^{t}_{\left(i-\frac{1}{2},j\right)}\right) + \frac{\Delta t}{\Delta y}\left(\hat{h}v^{t}_{\left(i,j+\frac{1}{2}\right)} - \hat{h}v^{t}_{\left(i,j-\frac{1}{2}\right)}\right) - \Delta t S_{(i,j)}\right)^{2} \quad (3)$$

Where $\hat{h}$ is predicted water depth, h, u, and v are ground truth value of water depth, velocity due x and y direction respectively, Ω is valid region of cell coordinate (i, j), t and Δt are time step and its duration, Δx and Δy are grid cell size.

2.3 Experiment Setup

The surrogate model is developed using Python in Google Colab using PyTorch libraries and details of model setups are summarised in Table 2. All input data are pre-processed to value between 0 to 1 to ensure learning stability (Wang et al., 2024). DTM data is pre-processed differently for the same reason stated in Section 2.2 concerning spatial distribution pattern. Furthermore, training regulations are imposed to safeguard learning failure due to overfitting and gradient explosion or vanishing (Zhao et al., 2024). During development stage, it is found that PINN loss function only works well when trained with input that involve hyetograph, hence PINN model is trained using half of the training set data with pluvial and compound flood cases only (see Table 1).

Table 2 Summary of Model Setup in the Study

Description	MSE Model	PINN Model
Traning Strategy	Stochastic gradient descent	Stochastic gradient descent
Optimiser	Adaptive moment estimation (Adam)	Adaptive moment estimation (Adam)
Training Regulation	Batch processing, Batch normalisation, Early stopping	Batch processing, Batch normalisation, Early stopping
Loss Function	MSE	PINN
Learning Rate	0.0001	0.001
Learning Epoch	25	25
Preprocessing Data		
- DTM Data	Inverse min-max scaling	Inverse min-max scaling
- Other Input Data	Min-max scaling	Min-max scaling
Hyperparameter Optimisation		
- Batch Size	12	12
- Base Channel Size	64	64
- Framework Depth	2	2
- Kernel Size K	3	3

3 RESULTS AND DISCUSSION

During development of the model, it is found that loss value calculated for MSE and PINN model had different scales of magnitude, therefore both training and validation loss graph are ploted at log scale for better comparison and is shown in Figure 4. Comparing Figure 4a and 4b, training and validation loss graph of PINN model is more similar in trend with each other than MSE model, this indicates that PINN loss function allows the model to corelate better to unknown storm conditions than MSE loss function. This is expected since PINN is based on shallow water equations while MSE is purely a statistical evaluation. Based on validation loss, the best performing models are both found at 15 epochs at loss value of 2.260 and 7.765 point for MSE and PINN model respectively and is used for subsequent evaluation.

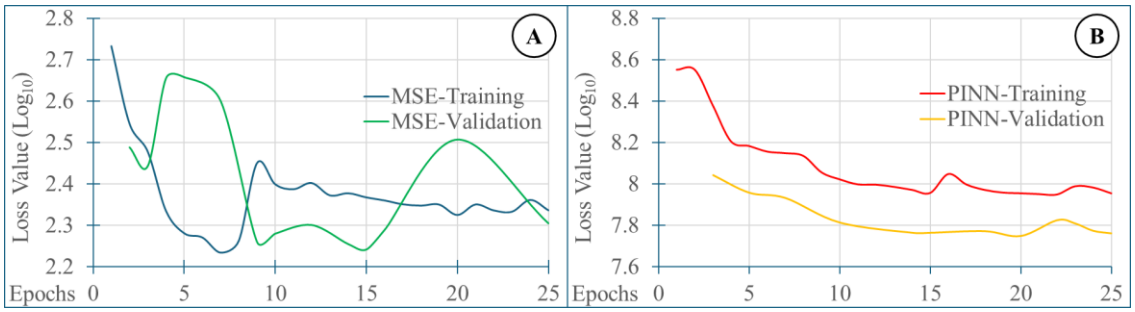

Figure 4: Comparing training and validation loss for MSE model (a) and PINN model (b)

The model performance is evaluated by using regression metrics of Root Mean Square Error (RMSE) and Mean Absolute Error (MAE), and classification metrics of precision, recall, and F1 value with 0.05 m threshold to screen wet and dry cells. Performance is evaluated on validation and testing set and the results are given in Table 3. Based on the results, model trained with MSE loss function outperform model with PINN, but compared with other studies (Donnelly et al., 2022; Guo et al., 2021; Liao et al., 2023), the regression performance is limited while classification performance is decent. The failure to perform better can be attributed to the level of generalisability that the model aim to achieve and the inability of the framework to accommodate it. However, as performane only declined within the range of 2.1~10.4% when resolution increase from 50 to 10 m, this shows that the model maintain initial performance across different resolution reasonably well.

Table 3 Surrogate Model Performance Summary

Case Set	Cases	Duration (hr)	Resolution (m)	Loss Function	RMSE (m)	MAE (m)	Precision	Recall	F1
Validation	28	24	50	MSE	2.405	1.548	0.685	0.784	0.725
Test			10		2.656	1.659	0.639	0.744	0.681
Percentage Difference					10.40%	7.20%	6.70%	5.10%	6.10%
Validation	28	24	50	PINN	4.378	2.968	0.827	0.672	0.714
Test			10		4.685	3.277	0.844	0.625	0.684
Percentage Difference					7.00%	10.40%	2.10%	7.00%	4.20%

Figure 5 shows prediction sample of CNN model trained with MSE loss function compared to ground truth water depth map from SynxFlow. Inline with the performance report, prediction results resemble the outputs from SynxFlow but with lower accuracy. Comparing predictions at 50m and 10m resolution (Figure 5b-c and 5e-f), the model maintain visual coherency when tested at higher resolution. Moreover, considering flood flows towards Northeast in Birmingham (Figure 5a-c) and Northwest in Carlisle (Figure 5d-f), this shows that model can also recognise river direction. These overall success can be atributed to the use of DTM data within the model framework, as effectively the model predict flood in the image of DTM data. However, the use of DTM data within the framework also impose limitation along the boundry region near the outlet area as shown by Figure 5a-c. Boundary region around outlet area are clearly predicted as flooded by the CNN model but not so by SynxFlow, these can be caused by elevation around the oulet area that are mostly very low when compared to the rest of the domain area and therefore have unrealistic distribution weight during prediction process. Nonetheless, this limitation is deemed acceptable as the framework has allowed surrogate model to emulate the ability of hydrodynamic model to generalise different resolution and river direction. This also reasonably improve explainability of the model since it now behave similarly to hydrodynamic model.

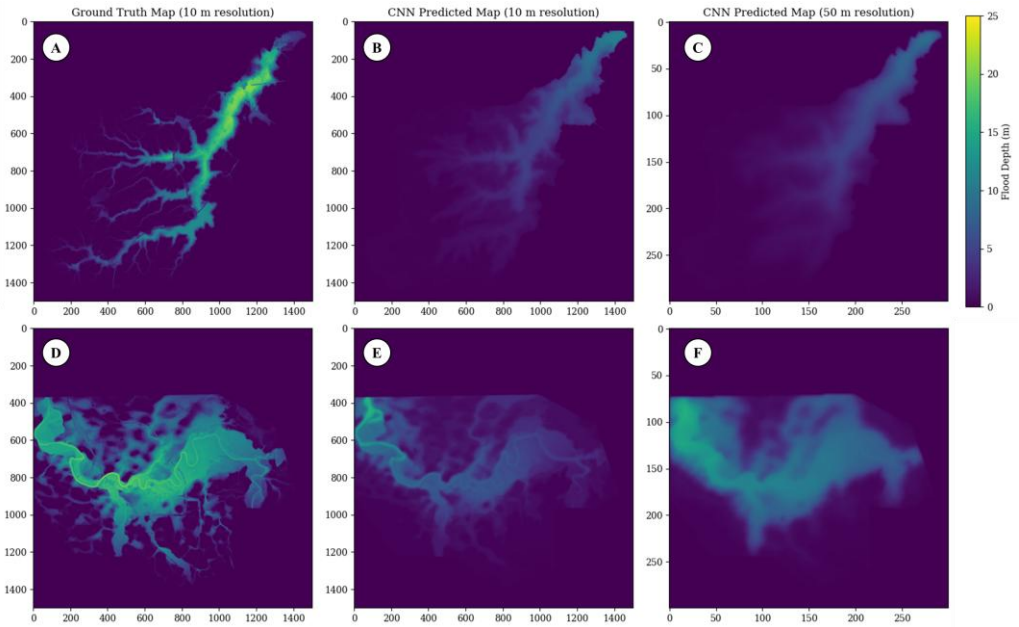

Figure 5: Prediction at t = 11 hour of return period 80 years for Birmingham (a-c) and Carlisle (d-f)

Computational burden of the model is compared to SynxFlow and results are shown in Table 4. The CNN model is faster than SynxFlow, and speed factor increases as resolution incease. This can be atributed due to the exponential increase in computing cell when moving from 50 m to 10 m resolution. Although faster, the time needed to train the model takes up to 137.6 hours excluding the time needed to generate data pool, which shows that substantial amount of preparation is needed. However, since it only required once, it can be argued that the model is still faster than hydrodynamic model and is suitable for rapid flood prediction. Furthermore, as the model is successfully developed in Google Colab, this shows that resources required to develop the model is not prohibitively expensive.

Table 4 Computational Burden Evaluation

Model	Device (Google Colab)	Training Time (hr)	Simulation Duration (hr)	Resolution (m)	Simulation Time (min)	Speed Factor
SynxFlow	T4 GPU	-	72	50	1.64	1
				10	111.47	1
CNN	CPU High Ram (Training)	137.6	24	50	0.12	4.6
	v5e-1 TPU (Validation & Testing)	-		10	3.7	10.03

4 CONCLUSION

This study shows that surrogate model can be developed to generalise different resolution of the same case while maintaining performance. The model manages to achieve this by leveraging terrain data to guide the process and mechanism within the framework. The model is computationally inexpensive to be developed on Google Colab platform and faster than hydrodynamic model, showing that it is accessible for stakeholders with limited resources while allowing certain degrees of rapid flood predictions. However, the model has limited predictive ability at water depth predictions but arguably functional at inundation extend. Nonetheless, this study has contributed to improve generalisability of surrogate model across resolution for flood predictions and open the possibility of training such model with low resolution to predict high resolution case. Future study should focus on improving ways to process data, enhance framework ability to predict accurately, and expand prediction to include velocity.

5 ACKNOWLEDGEMENTS

The first author acknowledges the support of the Indonesia Endowment Fund for Education LPDP (Grant No. 202407120105809).

REFERENCES

Cao, A., Nakamura, S., Otsuyama, K., Namba, M., & Yoshimura, K. (2024). Current status and challenges in operating flood early warning systems at the local level in Japan. International Journal of Disaster Risk Reduction, 112. https://doi.org/10.1016/j.ijdrr.2024.104802

Donnelly, J., Daneshkhah, A., & Abolfathi, S. (2024). Physics-informed neural networks as surrogate models of hydrodynamic simulators. Science of the Total Environment, 912. https://doi.org/10.1016/j.scitotenv.2023.168814

Eleuterio, F. T. (2001). Shock-Capturing Methods for Free-Surface Shallow Flows. Wiley.

Fraehr, N., Wang, Q. J., Wu, W., & Nathan, R. (2024). Assessment of surrogate models for flood inundation: The physics-guided LSG model vs. state-of-the-art machine learning models. Water Research, 252. https://doi.org/10.1016/j.watres.2024.121202

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. Journal of Flood Risk Management, 14(1). https://doi.org/10.1111/jfr3.12684

Karim, F., Armin, M. A., Ahmedt-Aristizabal, D., Tychsen-Smith, L., & Petersson, L. (2023). A Review of Hydrodynamic and Machine Learning Approaches for Flood Inundation Modeling. In Water (Switzerland) (Vol. 15, Number 3). MDPI. https://doi.org/10.3390/w15030566

Liao, Y., Wang, Z., Chen, X., & Lai, C. (2023). Fast simulation and prediction of urban pluvial floods using a deep convolutional neural network model. Journal of Hydrology, 624. https://doi.org/10.1016/j.jhydrol.2023.129945

Muralidhar, N., Bu, J., Cao, Z., He, L., Ramakrishnan, N., Tafti, D., & Karpatne, A. (2020). PhyNet: Physics Guided Neural Networks for Particle Drag Force Prediction in Assembly. In Proceedings of the 2020 SIAM International Conference on Data Mining (pp. 559–567). Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9781611976236.63

Situ, Z., Wang, Q., Teng, S., Feng, W., Chen, G., Zhou, Q., & Fu, G. (2024). Improving urban flood prediction using LSTM-DeepLabv3+ and Bayesian optimization with spatiotemporal feature fusion. Journal of Hydrology, 630. https://doi.org/10.1016/j.jhydrol.2024.130743

Song, W., Guan, M., & Yu, D. (2025). SwinFlood: A hybrid CNN-Swin Transformer model for rapid spatiotemporal flood simulation. Journal of Hydrology, 660. https://doi.org/10.1016/j.jhydrol.2025.133280

Teng, J., Jakeman, A. J., Vaze, J., Croke, B. F. W., Dutta, D., & Kim, S. (2017). Flood inundation modelling: A review of methods, recent advances and uncertainty analysis. In Environmental Modelling and Software (Vol. 90, pp. 201–216). Elsevier Ltd. https://doi.org/10.1016/j.envsoft.2017.01.006

Wang, Z., Lyu, H., Fu, G., & Zhang, C. (2024). Time-guided convolutional neural networks for spatiotemporal urban flood modelling. Journal of Hydrology, 645. https://doi.org/10.1016/j.jhydrol.2024.132250

Xia, X., & Ming, X. (2025). SynxFlow: A GPU-accelerated Python package for multi-hazard simulations. Journal of Open Source Software, 10(111), 7586. https://doi.org/10.21105/joss.07586

Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M., & Parmar, M. (2024). A review of convolutional neural networks in computer vision. Artificial Intelligence Review, 57(4). https://doi.org/10.1007/s10462-024-10721-6