
Flood Prediction with Graph Neural Networks

Arnold Kazadi¹, James Doss-Gollin¹, Antonia Sebastian², Arlei Silva¹

¹Rice University ²University of North Carolina at Chapel Hill

{arnoldnyamabo, jdoss, gollin, arlei}@rice.edu, asebastian@unc.edu

Abstract

Climate change is increasing the frequency of flooding around the world. As a consequence, there is a growing demand for effective flood prediction. Machine learning is a promising alternative to hydrodynamic models for flood prediction. However, existing approaches focus on capturing either the spatial or temporal flood patterns using CNNs or RNNs, respectively. In this work, we propose FloodGNN, which is a graph neural network (GNN) for flood prediction. Compared to existing approaches, FloodGNN (i) employs a graph-based model (GNN); (ii) operates on both *spatial* and *temporal* dimensions; and (iii) processes the water flow velocities as vector features, instead of scalar features. Experiments show that FloodGNN achieves promising results, outperforming an RNN-based baseline.

1 Introduction

Climate change is driving changes in the intensity, frequency, and spatiotemporal structure of heavy precipitation, which is anticipated to increase urban flood hazard in many regions [1, 2]. Predictive modeling can support adaptation in many ways, such as through early warning systems or by mapping hazards across space and time [3]. Fluid mechanics, such as the 3D Navier Stokes equations, describe the physics of flooding. In practice, urban flooding is typically modeled through simplified 2D models with sophisticated numerical methods to balance accuracy and computational costs.

Recently, machine learning (ML) has been presented as an alternative to physics-based flood prediction models in many settings [4]. For instance, ML has been applied for real-time flood forecasting [5], continental-scale flood risk assessment [6, 7], high-resolution flood extent prediction [8], and resource-constrained prediction [9]. As expected, many of these approaches apply deep learning due to their expressive power [10] and scalability. However, these models either focus on the spatial or the temporal dimension [1, 11, 12, 13, 14, 15, 16, 17, 18]. More specifically, spatial models, which are based on Convolutional Neural Networks (CNNs) or feed-forward neural networks, predict only the maximum water depth at each location. On the other hand, temporal models apply Recurrent Neural Networks (RNNs) to model the evolution of water depths over time [19, 20, 21, 22, 23].

This work investigates spatiotemporal models for flood prediction. We focus on graph (or mesh) based models, where the raster map of a region is represented as a graph with nodes/cells as locations and edges as spatial proximity. Graphs are more flexible than image-based representations, as they support irregularly-sampled cells, while still being able to capture physical relations between nearby locations [24]. We propose FloodGNN, a Graph Neural Network architecture for flood prediction. At each time step, FloodGNN predicts the water depths and velocities—i.e. the state of the flood—based on previous depths and velocities and also other static features. Velocities are processed as vector features using geometric vector perceptrons [25]. Experiments based on a simulation of Hurricane Harvey, in Houston, TX, show that FloodGNN achieves promising results in terms of accuracy.

2 Problem definition and approach

Problem. We are given a set of regions $\mathcal{R}_L = \{R_g\}_{g=1}^K$, each represented as a graph with a series of states R_g^1, \dots, R_g^T . At time t , a graph $R_g^t = (V, E)$ has its nodes/cells $v_i \in V$ associated with vector features \mathbf{V}_i^t and scalar features \mathbf{s}_i . As vector features, we consider $\mathbf{V}_i^t = [\mathbf{a}_i^t / \|\mathbf{a}_i^t\|, \mathbf{b}_i^t / \|\mathbf{b}_i^t\|]^T \in \mathbb{R}^{2 \times 2}$, which are velocities in the x -axis and y -axis registered at the interfaces of grid cells (See Figure 1b). As scalar features, we consider $\mathbf{s}_i^t = (e_i, n_i, d_i, \|\mathbf{a}_i^t\|, \|\mathbf{b}_i^t\|, w_i^t)^T \in \mathbb{R}^6$, where e_i is the ground elevation, n_i is the Manning’s friction coefficient, and d_i is the distance from v_i to the closest river/stream— e_i, n_i , and d_i are static. Our goal is to learn a model that, given the current state R_g^t of region R_g , can predict the depth w_i^{t+1} and velocity \mathbf{V}_i^{t+1} for each node $v_i \in V$ at time step $t+1$.

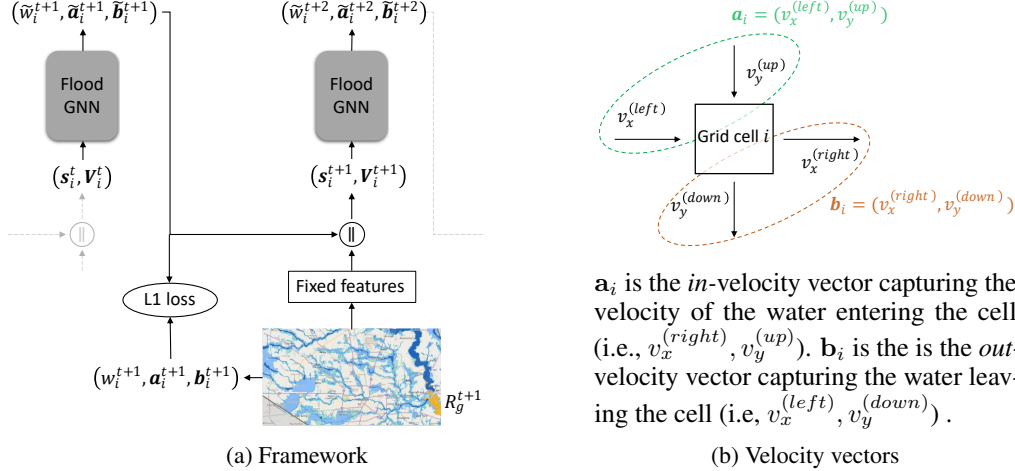


Figure 1: Overview of FloodGNN (1a) its velocity vectors (1b). At each time t , the region R_g is in state R_g^t with scalar features \mathbf{s}_i^t and vector features \mathbf{V}_i^t for each node/cell v_i . These are processed through a GNN (recursively) to produce estimates of the next water depth \tilde{w}_i^{t+1} and velocities $\tilde{\mathbf{a}}_i^{t+1}$ and $\tilde{\mathbf{b}}_i^{t+1}$. The L1 loss function between $\tilde{w}_i^{t+1}, \tilde{\mathbf{a}}_i^{t+1}, \tilde{\mathbf{b}}_i^{t+1}$ and their ground truth values $w_i^{t+1}, \mathbf{a}_i^{t+1}, \mathbf{b}_i^{t+1}$ is used for parameter learning in FloodGNN. We also show how the in-velocity (\mathbf{a}_i) and out-velocity (\mathbf{b}_i) vectors are generated. The time step superscript t is dropped for clarity

Method. As velocities are vectors, we would like to preserve their geometry and not treat them as scalar features. Thus, we apply geometric vector perceptrons (GVP) [25] for feature transformation. GVPs are an extension of standard dense layers (MLPs) that operate on collections of Euclidean vectors. They consist of two separate linear transformations, for the scalar and vector features, followed by non-linearities $\sigma, \sigma+$. Before the scalar features are transformed, they are concatenated with the L2 norm of the transformed vector features to extract rotation-invariant information from the input vectors. In short, a GVP takes two input types (i.e., scalar and vector features) and returns their transformations, that is $(\mathbf{s}', \mathbf{V}') = \text{GVP}(\mathbf{s}, \mathbf{V})$ (where $\mathbf{s}', \mathbf{s} \in \mathbb{R}^n$; and $\mathbf{V}', \mathbf{V} \in \mathbb{R}^{m \times p}$).

At each time-step t we perform a node regression task by predicting the next water depth \tilde{w}_i^{t+1} and velocities $\tilde{\mathbf{a}}_i^{t+1}$ and $\tilde{\mathbf{b}}_i^{t+1}$ using a message-passing graph neural network (GNN). Scalar ($\mathbf{m}'_{j \rightarrow i}$) and vector ($\mathbf{M}'_{j \rightarrow i}$), messages are computed as:

$$\mathbf{m}_{j \rightarrow i} = [\mathbf{s}_i^t \parallel \mathbf{s}_j^t], \quad \mathbf{M}_{j \rightarrow i} = [\mathbf{V}_i^t \parallel \mathbf{V}_j^t] \quad \forall j \mid v_j \in \mathcal{N}_{(v_i)}$$

$$(\mathbf{m}'_{j \rightarrow i}, \mathbf{M}'_{j \rightarrow i}) = \text{GVP}(\mathbf{m}_{j \rightarrow i}, \mathbf{M}_{j \rightarrow i})$$

where \parallel is the concatenation operator, and $\mathcal{N}_{(v_i)}$ is the set of neighbors of node v_i .

The node update operation is performed as follows:

$$(\mathbf{s}'_i, \mathbf{V}'_i) = \text{GVP}\left([\mathbf{s}_i^t \parallel \sum \mathbf{m}'_{j \rightarrow i}], [\mathbf{V}_i^t \parallel \sum \mathbf{M}'_{j \rightarrow i}]\right)$$

Finally, the values of \tilde{w}_i^{t+1} , $\tilde{\mathbf{a}}_i^{t+1}$, and $\tilde{\mathbf{b}}_i^{t+1}$ are predicted as:

$$(\tilde{w}_i^{t+1}, \mathbf{P}) = \text{GVP}(\mathbf{s}'_i, \mathbf{V}'_i) \quad \mathbf{P} \in \mathbb{R}^{2 \times 2} \quad \Rightarrow \quad \tilde{\mathbf{a}}_i^{t+1} = \mathbf{P}_{[:,0]}, \quad \tilde{\mathbf{b}}_i^{t+1} = \mathbf{P}_{[:,1]}$$

where \tilde{w}_i^{t+1} , $\tilde{\mathbf{a}}_i^{t+1}$ and $\tilde{\mathbf{b}}_i^{t+1}$ are used to construct input features \mathbf{s}_i^{t+1} and \mathbf{V}_i^{t+1} for the next time-step ($t + 2$). Our proposed GNN, FloodGNN, is run recurrently, with the same parameters. The L1 loss is used to compare predictions \tilde{w}_i^{t+1} , $\tilde{\mathbf{a}}_i^{t+1}$, $\tilde{\mathbf{b}}_i^{t+1}$ and their respective ground truth values w_i^{t+1} , \mathbf{a}_i^{t+1} , \mathbf{b}_i^{t+1} to update the model parameters. The overall architecture of FloodGNN is shown in Figure 1a.

3 Experiments

Our experiments are based on data from a model simulation for Hurricane Harvey in Houston, Texas. Precipitation was collected from the Harris County Flood Control District (HCFCD) rain gages and spatially averaged over the model domain. Flood depths and velocities were simulated at hourly time steps for 24 simulation days using the LISFLOOD-FP hydrodynamic model (version 7) [26]. Channel parameters were derived from the Hydrologic Engineering Center’s River Analysis System (HEC-RAS) available through the HCFCD Model & Map Management (M3) System.

The dataset, represented as a raster map of size 1961×1636 cells (30-meter resolution), was divided into smaller non-overlapping sub-regions of sizes $\approx 50 \times 40$ in order to generate different regions for training. There were 1531 grid-based graphs from which we randomly selected 70% for training, 15% for validation, and 15% for testing. All the scalar features were normalized to zero mean and unit variance based on the training set statistics.

We compare FloodGNN to an RNN-based method (FloodRNN) that models only the temporal data. We also evaluate FloodGNN-NoV, a variant of FloodGNN that treats velocities as scalar features. FloodRNN takes the same inputs as our method but predicts the water depth (flooding) of each cell/node independently. For a fair comparison, we use the same training, validation, and test sets for all the methods. We compare these methods in terms of root mean square error (RMSE) and correlation of determination (R^2) score. The mean and standard deviation for 3 random runs of experiments on time-series of length 5 are shown in the Table 1. FloodGNN achieves lower RSME and higher R^2 scores at each time step, implying that flood prediction results are much closer to the ground truth. This can be attributed to the combination of spatial and temporal information in our method, which is able to learn more complex flood dynamics than FloodRNN. Notice also that FloodGNN-Nov achieves worse results than FloodGNN, which is evidence of the importance of geometric information for approximating the physics of flood simulations.

RMSE (lower better)					
time-step t	1	2	3	4	5
FloodRNN	.22 ± .030	.33 ± .030	.41 ± .031	.48 ± .034	.55 ± .038
FloodGNN-NoV	.25 ± .030	.40 ± .031	.51 ± .021	.60 ± .007	.66 ± .006
FloodGNN (Ours)	.17 ± .031	.27 ± .043	.34 ± .040	.39 ± .037	.44 ± .036
R^2 (higher better)					
time-step t	1	2	3	4	5
FloodRNN	.95 ± .0160	.87 ± .0390	.79 ± .0540	.72 ± .0620	.66 ± .0660
FloodGNN-NoV	.95 ± .0023	.88 ± .0069	.80 ± .0177	.71 ± .0356	.63 ± .0571
FloodGNN (Ours)	.98 ± .0028	.93 ± .0063	.89 ± .0083	.85 ± .0088	.80 ± .0091

Table 1: Comparative results.

We also conducted experiments with longer time series (length 10) and on relatively deeper water depths. The latter is to investigate how our method predicts significant flooding, which occurs at a limited number of cells relative to the entire area. Figure 2 compares the predictions over multiple steps for all cells (Figure 2a), for those with a depth larger than 0.02 meters (Figure 2b), and those with a depth larger than 0.05 meters (Figure 2c). We can observe that, even in deep water regions and for longer time series, our method performs better than the baselines.

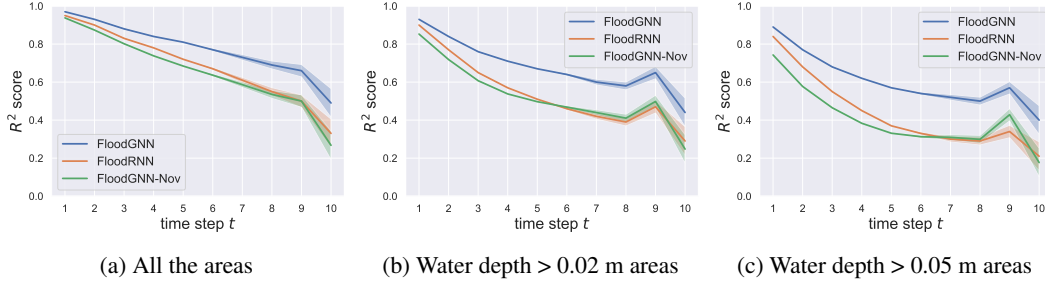


Figure 2: Prediction over time-series of length 10 and in deeper water depth areas.

Figure 3 shows flooding maps of a (sub-)region from the test set and the corresponding predictions from our method. We can see that FloodGNN predicts values close to the ground truth, even though the accuracy decays over time. Still, predictions enable the identification of flooded vs. non-flooded areas even after 10 time steps.

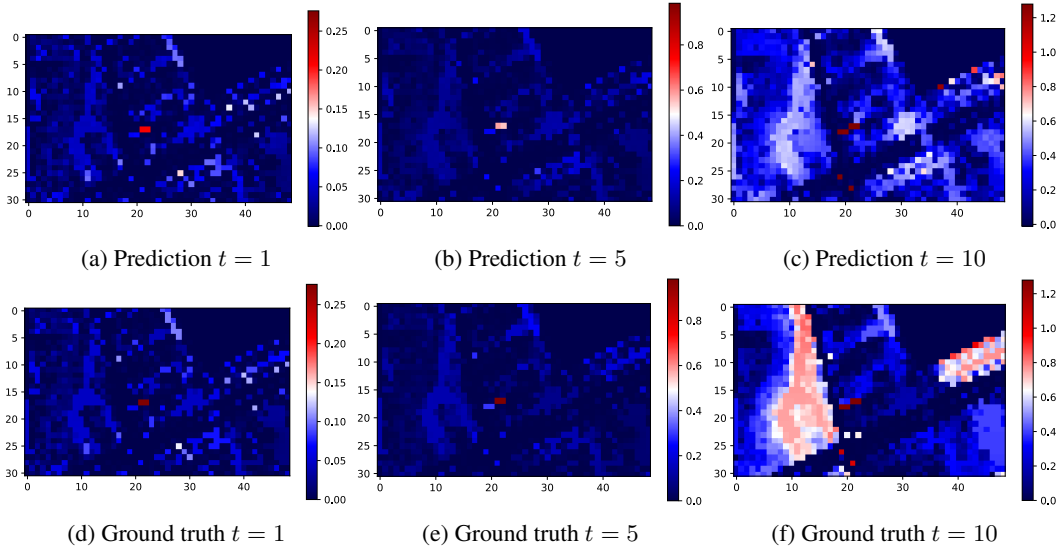


Figure 3: Comparison between real data (bottom row) and predictions from our model (top row).

4 Conclusions

We have presented a graph neural network method, named FloodGNN, for flood prediction using spatiotemporal data. FloodGNN is used recurrently over the graphical representation of a region, predicting water depths and velocity vectors at each time step in an auto-regressive manner. Our preliminary results—based on a simulation of Hurricane Harvey, in Houston, TX—have demonstrated empirically how FloodGNN can infer flooding extent on sub-regions unseen during training over time series with lengths of up to 10 time steps.

Our work opens several avenues for future research. First, we want to incorporate rainfall data into FloodGNN predictions using a representative set of rainfall events. Next, we want to incorporate physics knowledge (e.g., conservation of mass and momentum) into our model. Finally, we will investigate automatic re-meshing algorithms to adaptively sample cells/nodes at different regions.

References

- [1] Simon Berkahn, Lothar Fuchs, and Insa Neuweiler. An ensemble neural network model for real-time prediction of urban floods. *Journal of hydrology*, 575:743–754, 2019.

- [2] S. Yu. Schreider, D. I. Smith, and A. J. Jakeman. Climate change impacts on urban flooding. *Climatic Change*, 47(1):91–115, Oct 2000.
- [3] Paul D. Bates. Flood inundation prediction. *Annual Review of Fluid Mechanics*, 54(1):287–315, 2022.
- [4] Amir Mosavi, Pinar Ozturk, and Kwok-wing Chau. Flood prediction using machine learning models: literature review. *Water*, 10(11):1536, 2018.
- [5] Farzad Piadeh, Kourosh Behzadian, and Amir Alani. A critical review of real-time modelling of flood forecasting in urban drainage systems. *Journal of Hydrology*, page 127476, 2022.
- [6] Sean A Woznicki, Jeremy Baynes, Stephanie Panlasigui, Megan Mehaffey, and Anne Neale. Development of a spatially complete floodplain map of the conterminous United States using random forest. *Science of the total environment*, 647:942–953, 2019.
- [7] Elyssa L Collins, Georgina M Sanchez, Adam Terando, Charles C Stillwell, Helena Mitsova, Antonia Sebastian, and Ross K Meentemeyer. Predicting flood damage probability across the conterminous united states. *Environmental Research Letters*, 17(3):034006, 2022.
- [8] Qing Lin, Jorge Leandro, Wenrong Wu, Punit Bhola, and Markus Disse. Prediction of maximum flood inundation extents with resilient backpropagation neural network: case study of Kulmbach. *Frontiers in Earth Science*, page 332, 2020.
- [9] Sella Nevo, Efrat Morin, Adi Gerzi Rosenthal, Asher Metzger, Chen Barshai, Dana Weitzner, Dafi Voloshin, Frederik Kratzert, Gal Elidan, Gideon Dror, and others. Flood forecasting with machine learning models in an operational framework. *Hydrology and Earth System Sciences Discussions*, pages 1–31, 2021.
- [10] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. ISSN 0893-6080.
- [11] Gang Zhao, Bo Pang, Zongxue Xu, Dingzhi Peng, and Depeng Zuo. Urban flood susceptibility assessment based on convolutional neural networks. *Journal of Hydrology*, 590:125235, 2020. ISSN 0022-1694.
- [12] Dieu Tien Bui, Nhat-Duc Hoang, Francisco Martínez-Álvarez, Phuong-Thao Thi Ngo, Pham Viet Hoa, Tien Dat Pham, Pijush Samui, and Romulus Costache. A novel deep learning neural network approach for predicting flash flood susceptibility: A case study at a high frequency tropical storm area. *Science of The Total Environment*, 701:134413, 2020. ISSN 0048-9697.
- [13] Yi Wang, Zhice Fang, Haoyuan Hong, and Ling Peng. Flood susceptibility mapping using convolutional neural network frameworks. *Journal of Hydrology*, 582:124482, 2020. ISSN 0022-1694.
- [14] Zifeng Guo, Vahid Moosavi, and João P. Leitão. Data-driven rapid flood prediction mapping with catchment generalizability. *Journal of Hydrology*, 609:127726, 2022. ISSN 0022-1694.
- [15] Roland Löwe, Julian Böhm, David Getreuer Jensen, Jorge Leandro, and Søren Højmark Rasmussen. U-flood – topographic deep learning for predicting urban pluvial flood water depth. *Journal of Hydrology*, 603:126898, 2021. ISSN 0022-1694.
- [16] Julian Hofmann and Holger Schüttrumpf. floodGAN: Using Deep Adversarial Learning to Predict Pluvial Flooding in Real Time. *Water*, 13(16):2255, 2021.
- [17] Syed Kabir, Sandhya Patidar, Xilin Xia, Qiu-hua Liang, Jeffrey Neal, and Gareth Pender. A deep convolutional neural network model for rapid prediction of fluvial flood inundation. *Journal of Hydrology*, 590:125481, 2020. ISSN 0022-1694. doi: <https://doi.org/10.1016/j.jhydrol.2020.125481>.
- [18] Zifeng Guo, Joao P Leitao, Nuno E Simões, and Vahid Moosavi. Data-driven flood emulation: Speeding up urban flood predictions by deep convolutional neural networks. *Journal of Flood Risk Management*, 14(1):e12684, 2021.
- [19] Deng-Lin Chang, Sheng-Hsueh Yang, Sheau-Ling Hsieh, Hui-Jung Wang, and Keh-Chia Yeh. Artificial intelligence methodologies applied to prompt pluvial flood estimation and prediction. *Water*, 12(12):3552, 2020.
- [20] Li-Chiu Chang, Fi-John Chang, and Yen-Ming Chiang. A two-step-ahead recurrent neural network for stream-flow forecasting. *Hydrological Processes*, 18(1):81–92, 2004.
- [21] Qiao-Feng Tan, Xiao-Hui Lei, Xu Wang, Hao Wang, Xin Wen, Yi Ji, and Ai-Qin Kang. An adaptive middle and long-term runoff forecast model using eemd-ann hybrid approach. *Journal of Hydrology*, 567:767–780, 2018. ISSN 0022-1694.
- [22] Frederik Kratzert, Daniel Klotz, Mathew Herrnegger, Alden K. Sampson, Sepp Hochreiter, and Grey S. Nearing. Toward improved predictions in ungauged basins: Exploiting the power of machine learning. *Water Resources Research*, 55(12):11344–11354, 2019.
- [23] F. Kratzert, D. Klotz, G. Shalev, G. Klambauer, S. Hochreiter, and G. Nearing. Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. *Hydrology and Earth System Sciences*, 23(12):5089–5110, 2019.

- [24] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*, 2021.
- [25] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2021.
- [26] Paul D Bates. The lisflood-fp flood inundation model, 2013.