



Structured exploration of machine learning model complexity for spatio-temporal forecasting of urban flooding

Candace Agonafir & Tian Zheng

To cite this article: Candace Agonafir & Tian Zheng (03 Dec 2025): Structured exploration of machine learning model complexity for spatio-temporal forecasting of urban flooding, *Urban Water Journal*, DOI: [10.1080/1573062X.2025.2589078](https://doi.org/10.1080/1573062X.2025.2589078)

To link to this article: <https://doi.org/10.1080/1573062X.2025.2589078>



Published online: 03 Dec 2025.



Submit your article to this journal 



View related articles 



View Crossmark data 

RESEARCH ARTICLE



Structured exploration of machine learning model complexity for spatio-temporal forecasting of urban flooding

Candace Agonafir^{a,b} and Tian Zheng^{a,b}

^aLearning the Earth with AI and Physics (LEAP) and Department of Statistics, Columbia University, New York, NY, USA; ^bNational Science Foundation (NSF) Science and Technology Center, USA

ABSTRACT

Urban flooding disrupts socio-economic systems and endangers lives, necessitating the employment of prediction tools. This study applies spectral clustering to delineate flood-prone zones in New York City (NYC), followed by an evaluation of statistical and neural network models, including feed-forward and graph neural networks. Among these, the graph wavenet (GWN) excels due to its proficiency in capturing dynamic spatio-temporal relationships, improving mean R^2 by 0.15 and achieving up to R^2 of 0.72 in certain areas. The study emphasizes that augmenting spatio-temporal components and adopting graph-based architectures enhances predictive accuracy, particularly in data-scarce settings. By combining clustering and advanced modeling, this research provides novel insights into urban flood prediction. These findings equip NYC urban planners and emergency responders with a valuable framework to mitigate the socio-economic impacts of flooding and improve response strategies.

ARTICLE HISTORY

Received 14 December 2024
Accepted 7 November 2025

KEYWORDS

Urban flooding; neural networks; artificial intelligence; hydrology; natural disasters

SUSTAINABLE DEVELOPMENT GOALS

SDG 9: Industry, innovation and infrastructure; SDG 10: Reduced inequalities; SDG 11: Sustainable cities and communities; SDG 13: Climate action; SDG 15: Life on land

1. Introduction

Urban flooding, a natural disaster with dynamic ramifications, requires circumspect consideration. At the onset of rainfall, as runoff traverses through an urban setting, numerous obstacles, including an inability to permeate structures, sidewalks and streets, and restricted entry into drainage systems due to debris blockages, are experienced. Compounding the issue, additional rainfall in a brief timeframe overwhelms even the unobstructed drains, preventing water admission and exacerbating overflow departure attempts. Consequently, runoff either persists on the streets or seeps through gaps in homes and buildings. In the resulting circumstance of urban flooding, the community is harmed. Most egregiously, deaths may occur, as without adequate warning, individuals are not able to implement precautionary measures, such as avoiding outdoor travel or relocating to higher elevated areas when indoors. Veritably, this tragic outcome was demonstrated during the post-tropical depression Ida event, where many of the 44 deaths within the New York City (NYC) metropolitan region were caused either by exposure to outdoor hazards, such as vehicular drownings or being swept away by the waters, or indoor below-ground dangers,

such as drownings in flooded basements (Falconer 2021; Plumer 2021). Furthermore, in addition to fatalities, urban flooding exacts economic strain, as there may be destruction to the infrastructure, interruption to transportation services, and structural damage to the buildings and vehicles. Indeed, in the event of a large flooding disaster, direct costs may be incurred at the extent of billions of dollars; additionally, when examining smaller, frequent floods, long-term costs collect over the years by the chronic strains to the structural, plumbing and electrical systems (Agonafir et al. 2023). Hence, prioritizing the analysis of the diverse elements and influences on water behavior in an urban environment proves essential.

Now, as accurate flood forecasting models, allowing for the implementation of disaster deterrent measures, would offer significant health and financial relief, there is continuous progression in model development. Traditionally, hydrodynamic models have been widely used for flood prediction and risk assessment; however, the employment of these physics-based models is limited in certain metropolitans (Agonafir et al. 2023). Specifically, hydrodynamic models rely on extensive calculations to determine water flow, requiring detailed

drainage network plans; as thus, in certain urban cities, such as NYC, where drainage details are unobtainable to researchers (Al-Suhili, Cullen, and Khanbilvardi 2019), the implementation of physics-based models becomes infeasible. Therefore, there has been a turn towards data-driven techniques to provide insight into water behavior when existing physical information is limited. With the provision of influencing variables, via statistical calculations and artificial intelligence (AI) capabilities, the models possess the ability to assess an occurrence and then create forecasts or ascertain vulnerabilities. Hence, the objective of understanding urban flooding is met without the need to simulate the exact water path. Ultimately, due to the convenience of use, data-driven techniques, particularly AI methods, have risen drastically in flood literature (Mosavi, Ozturk, and Chau 2018). Accordingly, an in-depth study into the efficiency of emerging AI techniques, within the field of urban flooding, affords complementation to the trend.

The mission of this study is formulated in consideration of the dire human and economic devastations of urban flooding, the modeling limitations due to data availability, and the recent advances in data-driven models to remedy the issues. Accurately assessing the intricacies of urban flood occurrence in NYC, by the employment of physical and crowdsourced data, this research provides a unique analysis of added components by presenting a cascade of statistical and neural network models, each with ascending complexities. In the exploration, a preliminary step involves the delineation of zones based on urban flood characteristics, using the unsupervised machine learning technique, spectral clustering. Then, a particular set of models, the Poisson generalized linear regression (GLM), feed forward neural network (FFN), recurrent neural network (RNN), convolutional neural network (CNN), graph convolutional network (GCN) and the graph wavenet (GWN), is selected to assess the benefits of auxiliary, advanced spatio-temporal aspects, dynamic graph creation and convolutional node-messaging capabilities. The conduction of the experiment follows, such that every model, receiving identical time-series input data and undergoing training (8 years) and testing (2 years) using the same set of dates, is tasked with the production of daily predicted street flooding counts for the testing period. By goodness-of-fit determinations, model comparisons are performed and dissected to discover the impacts of additional complexities. Therefore, the comprehensive examination of diverse models imparts invaluable modeling guidance towards urban flood research in the data-driven era.

Regarding existing urban flood literature, there is limited utilization of graph-structured artificial neural

network (ANN) models. Moreover, of the few existing studies, an inclusion of pluvial, urban flooding is notably absent. For instance, in Farahmand et al., a spatial-temporal graph-based model (ASTGCN) for nowcasting in Harris County, Texas is developed (Farahmand, Xu, and Mostafavi 2023). However, the examination is conducted on the singular flooding incident of Hurricane Harvey and its direct landfall onto the county, thereby effectively assessing the model's accuracy only in regards to coastal flooding. In contrast, the research of this paper delves into urban flooding over a 10-year duration, encompassing both large-scale events and persistent, frequent flash flood and pluvial occurrences. Furthermore, the primary objective of Farahmand et al. is to highlight the proficiencies of the new model, rather than to conduct an in-depth exploration of the specific advantages of its added elements. In contrast, this research traverses a range of models, inspecting the benefits of each advancement. In another existing study, Wang et al., a graph-structured model is also created to benefit urban flood insights (Z. Y. Wang et al. 2023). Nonetheless, the paper does not serve as a comparative analysis of varying models; also, the model developed identifies flood susceptibility, as opposed to producing forecasts. Finally, it is worth noting that Santos et al. developed a graph-based, deep learning model for flood prediction (Oliveira Santos et al. 2023). Yet, the forecasts are for riverine flooding instead of urban flooding. As urban flooding involves multiple factors distinct from riverine flood variables (i.e. lack of infiltration and complex drainage networks), a model analysis specifically tailored towards urban concerns has more utility to metropolitan stakeholders. Therefore, this paper is the first of its kind to pioneer the adaptation of graph-based neural networks for urban, pluvial flooding, while conducting an expansive exploration of the spatio-temporal aspects within the domain.

This paper follows a structured sequence. In 'Data pre-processing', the study initiates by discussing the study area of NYC and the urban characteristics which make it ideal for experimentation. Also, 'Data pre-processing' demonstrates the data collection and pre-processing steps. Next, 'Methodology' delves, with extensive detail, into the methodology employed for each machine learning and statistical technique, laying a solid foundation for the thorough analysis presented in 'Results and discussion'. Then, in 'Results and discussion', a detailed presentation and discussion of results unfold, exploring risk zones and model performances across diverse flood attributes and model features. Finally, in 'Conclusions', conclusions are drawn, as the findings from the study are synthesized, offering a

comprehensive overview of the outcomes and their implications for urban flood research.

2. Data pre-processing

2.1. Study area

Situated along the northeastern coast of the United States, NYC emerges as a metropolitan landscape – distinctly impervious and densely populated. Lacking efficient infiltration with approximately 72% impervious cover, and encompassing a mere 800 square kilometers, while boasting roughly eight million residents (City of New York 2022b; U.S. Census Bureau QuickFacts (n.d.), NYC embodies urbanization. Moreover, crucial details, such as the locations and widths of stormwater inlet drains and digitized maps of the sewer network are notably absent from public records. This lack of drainage data poses a challenge to hydro-hydraulic flood modeling and emphasizes the need for alternative methods to mitigate flooding issues within the city. Moreover, NYC holds economical influence, contributing approximately \$1.8 trillion annually to the US gross domestic product (Bureau of Economic Analysis 2021). Therefore, the intricate urban fabric, intertwined with the economic significance and challenges of minimally accessible data, poses NYC as the ideal study area for the application of data-driven techniques in urban flooding.

NYC has 59 localized, politically based districts, called Community Boards (DISs). DISs manage zoning and land-use policies and address general municipal concerns (City of New York 2023a). Figure 1 depicts the percent of impervious cover in NYC, with the outlines of each DIS. By borough, the breakdown of DISs is the following: 12 in Manhattan, 12 in the Bronx, 18 in Brooklyn, 14 in Queens and three in Staten Island. Due to the quantity and extent of the districts, they serve as an ideal starting point for variable aggregation and clustering analysis. In this study, shapefiles were downloaded from NYC Open Data (City of New York 2023b), and the processing of data was conducted by ArcGIS and Python. In the proceeding sections, the methods of aggregation are further detailed.

2.2. Dynamic variables

2.2.1. NYC 311 platform

NYC311, a crowdsourcing platform, provides gainful insights into sewer related conditions in NYC. Observations of city issues are reported by residents of the city, where the date, time and longitude and latitude coordinates of the incident are listed. Hence, the detailed temporal and locational information affords researchers the opportunity to employ data-driven techniques for analyses. For this study, NYC311 street flooding (SF), sewer backups (SB) and catch basin blockages

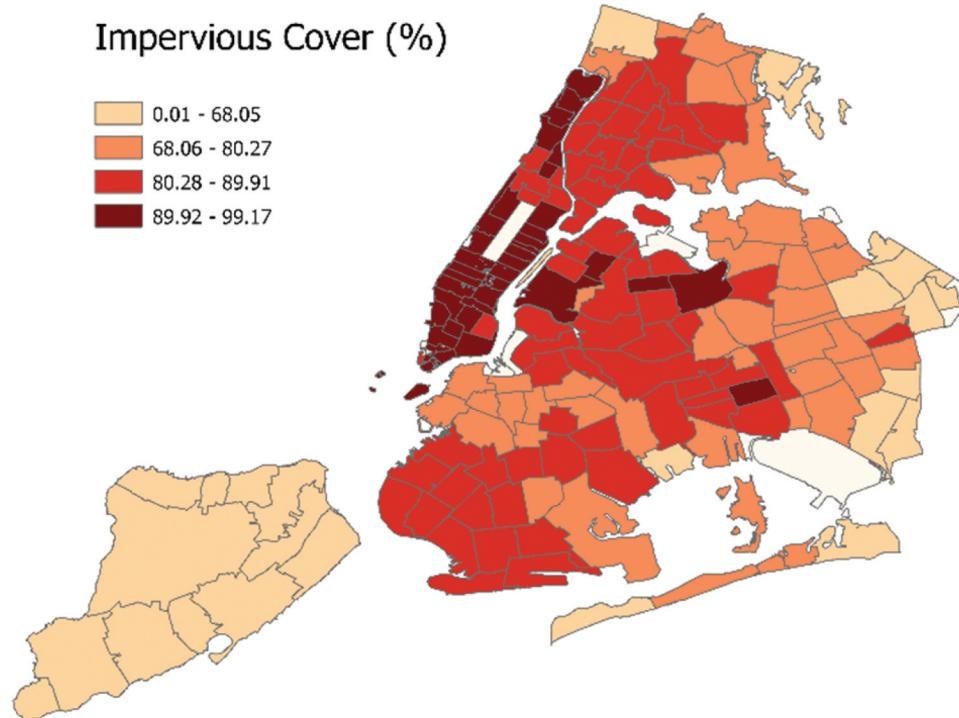


Figure 1. A map of the percentage of impervious cover for each NYC DIS.

(CB) reports, ranging from 1 January 2010 through 31 December 2019, were downloaded from the NYC Open Data website: <http://data.cityofnewyork.us> (Dates after 2019 were excluded, as the COVID pandemic is assumed to have an impact on reporting behavior, particularly in NYC, where residents and visitors relocated and returned at various intervals). For each report type, daily counts were aggregated to the DIS level. SF complaints, witnessed incidents of street flooding, served as the response variable. SB and CB reports were chosen for inclusion, as they are known influences towards SF (Agonafir et al. 2021; City of New York 2022a). Specifically, SB indicate an internal issue within the drainage network, where it is overtaxed (Schmitt, Thomas, and Ettrich 2004); moreover, Agonafir et al. has shown that SB reports are a significant predictor towards SF reports in almost half of NYC zip codes (Agonafir et al. 2021). Thus, the addition of SB may strengthen a model's ability to make predictions. Also, in the Agonafir et al. study, CB were found to have consequences on SF in roughly half the NYC zip codes (Agonafir et al. 2021). When a catch basin is blocked or clogged, runoff is not efficiently extracted into the stormwater drains, thereby allowing for ponding. In brief, from the NYC311 platform, the dynamic (values varying with time) infrastructural predictors, SB and CB, and the predictand, SF, were obtained. SF, SB and CB counts were totaled per day per DIS by using the timestamps and longitude and latitude coordinates.

2.2.2. Radar and gauge data

Precipitation, rain and snow, drive urban flooding occurrence. While it is apparent that rainfall is the primary contributor (Agonafir et al. 2023; Qin, Li, and Fu 2013; Schmitt, Thomas, and Ettrich 2004; Sharif et al. 2006; Valeo and Ho 2004), snowmelt also has influence, as when large amounts of snow liquifies, streets may be flooded (Semádeni-Davies and Bengtsson 1998; Valeo and Ho 2004). Concerning rainfall, there are intense rainstorms (large amounts of rain in a brief time interval), which contribute to flash floods, and there are prolonged rainy days, where the rainfall may not be intense, yet there is a sufficient amount of water over a longer duration. In both cases, the capacity of the stormwater drains may be exceeded (Agonafir et al. 2023). Therefore, this study used three predictor variables representing precipitation: Max hourly rainfall (MR), total daily rainfall (TR) and snowfall (SN).

Now, there are also varied methods of rainfall collection: in-situ (gauge) and remote sensing (radar and satellite). Some benefits of in-situ measurements include not being encumbered with cloud top reflectance, thermal radiance, retrieval algorithm and

overpass frequency issues (AghaKouchak, Nasrollahi, and Habib 2009); whereas radar data is advantageous in terms of spatial distribution (Thorndahl et al. 2017), capturing precipitation amounts at more locations within an area. Essentially, while both radar and gauge-based methods are widely adopted in hydrological studies, they each introduce unique sources of uncertainty that may influence model performance (Agonafir et al. 2023). Rain gauges, considered a direct measurement technique, may experience instrumental error, especially during light rainfall events (Gires et al. 2014). Also, the distance between rain gauges may result in underrepresentation of local storm variability, which can be especially problematic in small-scale urban catchments (Muthusamy et al. 2017). Radar data, on the other hand, offer enhanced spatial coverage, yet are subject to uncertainties related to retrieval algorithms, ground clutter and signal attenuation, particularly in complex urban terrain (AghaKouchak, Nasrollahi, and Habib 2009). Acknowledging these uncertainties, the selected data sources were chosen to balance spatial coverage and measurement accuracy across precipitation types. As both techniques are considered standard measurement methods, this study employed radar data for the MR and gauge data for the TR and SN variables. Each of the precipitation variables were determined at the DIS level.

The radar rain data was taken from the National Center for Atmospheric Research (NCAR)/Earth Observing Laboratory (EOL) website, and the gauge rain and snow data were retrieved from NOAA's Climate Data Online. For the radar, the resolution is 4 km by 4 km, and the gridded data is Stage IV, benefiting from manual quality control (EOL 2022). Hourly totals were gathered for the dates ranging from 1 January 2010 through 31 December 2019. Then, the maximum hourly value (MR) was taken for each radar point for each day. Ultimately, the MR values were assigned to each DIS based on the radar point's proximity to its centroid. Now, regarding the gauge data, the Global Historical Climatology Network (GHCN) by NOAA's National Centers for Environmental Information (NCEI) provides daily climatology details from land surface stations globally (National Centers for Environmental Information 2023). With respect to the daily rain totals (TR) and snow totals (SN), data was collected from the GHCN station, NY City Central Park, NY US, for the dates ranging from 1 January 2010 through 31 December 2019. These are direct measurements provided by the station. The TR and SN 24-hour amounts were assigned to each DIS. Therefore, by a combination of radar and gauge determinations, precipitation, representing rainfall intensity, total rainfall

and total snowfall, were assigned to each DIS at the daily level as predictors for the models.

2.3. Static features

There are multiple factors driving the occurrence of urban flooding. First, there are topographical variables, such as slope (SLP) and elevation (ELV). Regarding slope, the greater the incline of a surface, the greater the velocity and discharge of water; hence, at the bottom of the slope, the water will pond quickly (Bruwier et al. 2020). Concerning elevation, studies have also shown lower elevated areas to be at a higher risk of flooding. For one, lower elevated areas are more vulnerable to storm surges from coasts and rivers, and secondly, as mentioned prior, lower elevated areas may be located at the edge of a sloped surface (Ouma and Tateishi 2014; Woodruff, Irish, and Camargo 2013). In addition to the topographical, there are urban features, specifically the quantity of buildings (BLD) and the extent of building footprint (FTP), which affect flooding. Buildings are an impervious surface, such that water is unable to infiltrate through the ground. Moreover, multiple studies have found buildings to have a dominating influence on urban flooding, compared to other common flood factors (Agonafir et al. 2022; Bruwier et al. 2020; J. Lin et al. 2021). Another variable included represents percent impervious (IMP); it depicts the percentage of all impervious surfaces, such as buildings, sidewalks and streets, within a neighborhood. Next, a variable representing the area (SIZ) of the DIS was included in the study. The size of a region does not increase flooding occurrence, yet, larger areas have more opportunity for flooding occurrences, leading to higher flood counts; thus, the machine learning model will benefit from the information. Lastly, concerning location, latitude (LAT) and longitude (LNG) coordinates, the variables allow for a directionality indication of flood occurrence. For instance, Agonafir et al. found that street flooding in NYC had a southern and eastern locationality of increased flooding incidents (Agonafir et al. 2022). While the exact cause of flooding is not given by the location, the variable allows for a geographical pattern to be learned within machine learning models. Overall, physical features including slope, elevation, building extent, area and geographical coordinates are useful variables in understanding urban flooding via modeling.

For the physical features, shapefiles were downloaded from NYC Open Data and processed via ArcGIS analysis tools. For the SLP and ELV variables, a shapefile of elevation points was downloaded, and the mean elevation in meters (m) and mean degree of slope were determined per DIS. Also, a shapefile, providing the

number of buildings, was retrieved, and the total number of buildings per square kilometer (km^2) area of each DIS was tallied to represent the BLD variable. Regarding the FTP variable, a shapefile of building footprints was used, and via ArcGIS, for each building footprint, the area in km^2 was calculated. In each DIS, the sum of the footprint areas was determined. For the SIZ, LAT and LNG variables, with the DIS shapefile, the values were determined via geometry processing tools of each DIS polygon, where SIZ was calculated in km^2 , and the LAT and LNG represent the centroid points of each polygon. Therefore, by geoprocessing, the static features for the analysis were collected.

3. Methodology

With the overarching goal being the achievement of a model, which produces profound insights on urban flooding despite limited data, this study investigates the value of added components, ascending in complexity. The methodology is outlined as a flowchart in Figure 2. The preliminary step for this analysis was the delineation of meaningful zones. Borders of zip codes, DISs, or boroughs are not based on topographical or urban flood characteristics. Therefore, while the DIS serves as a sufficient starting point for data aggregation, a further outlining of risk zones based on flood factors, as opposed to political or postal bordering, has more utility in modeling endeavors. Accordingly, a spectral clustering technique was applied to identify areas of similar flood vulnerability characteristics. Specifically, the features used as inputs to the clustering algorithm were SLP, ELV, BLD, FTP and IMP. Then, six clusters (zones) based on likeness were created (Figure 3), and the dynamic and static variables were aggregated from the DIS to the cluster level. Thus, with zones of NYC regions aligned on related flood factors, predictive modeling was able to be performed.

The rationale for selecting the non-graph-based models – GLM, FNN, RNN and CNN – stems from both their individual strengths and their hierarchical relationship, especially within the neural network family, where each architecture incrementally enhances model capacity. Regarding the GLM, it is well suited for count data, particularly when the response variable represents the frequency of discrete events, such as daily flood reports (Agonafir et al. 2021). As highlighted in the broader statistical literature, Poisson regression is designed to handle non-negative integer responses and is effective when event occurrences are rare or overdispersed (Hardin and Hilbe 2007). Distinctly, within the field of flood modeling, Poisson GLMs have been successfully applied to assess the relationship

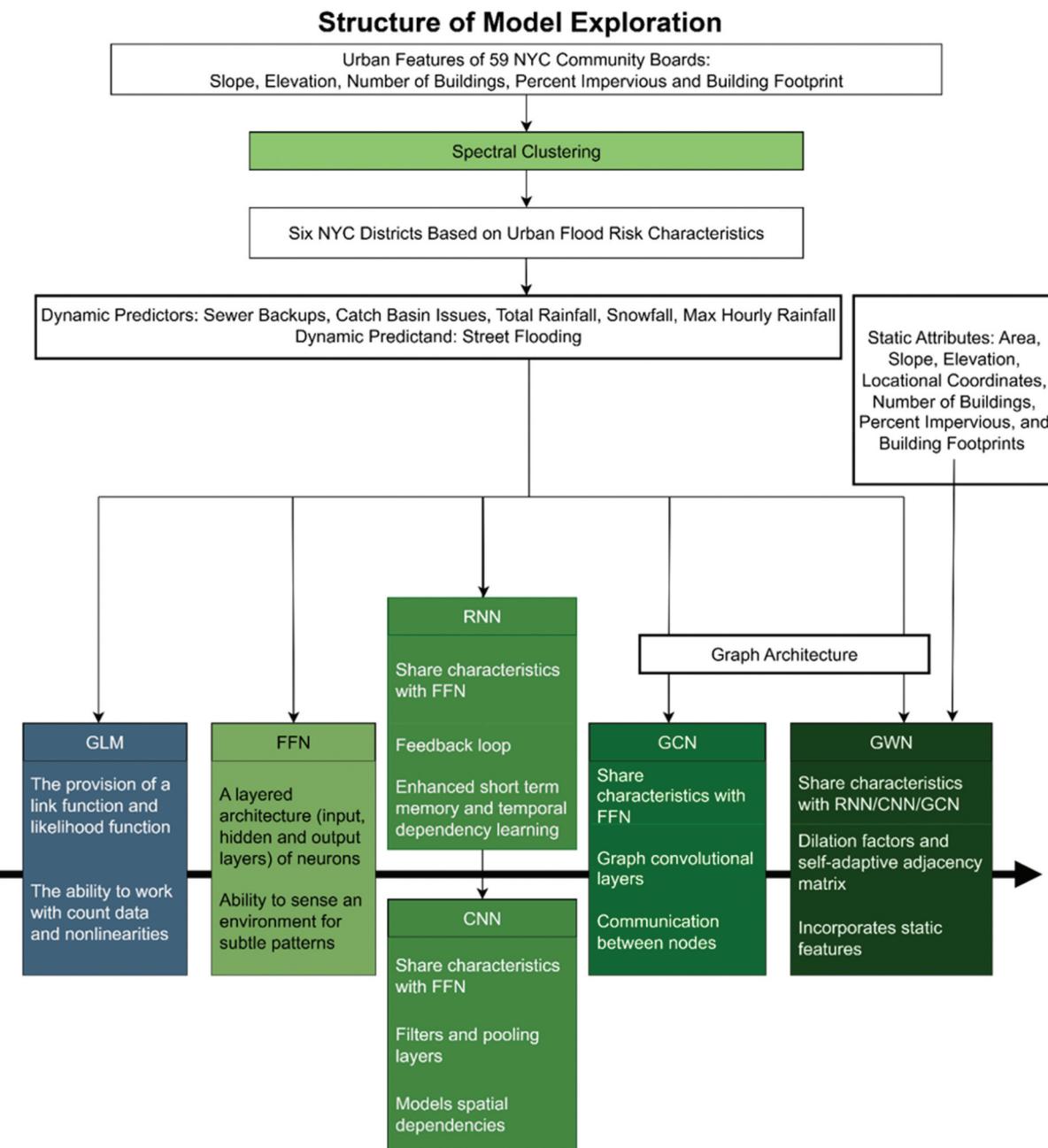


Figure 2. A flow chart outlining key processes.

between physical and environmental predictors and flood event frequencies. For example, Fang et al. (2019) used a Poisson GLM to identify significant flood drivers in the Yangtze River Basin, and Sadler et al. (2018) demonstrated its effectiveness in modeling flood patterns in Norfolk, Virginia, using crowdsourced data (a method directly aligned with this study's data approach). Hence, the Poisson GLM serves as an ideal statistical baseline model. As the study transitions to more complex machine learning models, the inclusion of the FNN, RNN and CNN is deliberate. The FNN represents the neural network baseline, while the RNN is

selected for its capacity to capture temporal dependencies through its feedback loop and short-term memory, and the CNN is chosen for its ability to detect spatial and sequential patterns in time-series data via convolutional filters. Moreover, the convolutional layers of the CNN facilitate a direct and equitable comparison with the graph-based models, which also incorporate convolutional components. Thus, these models offer complementary perspectives on how temporal and structural dynamics in the input data relate to flood incidence, thereby strengthening the overall model comparison framework.

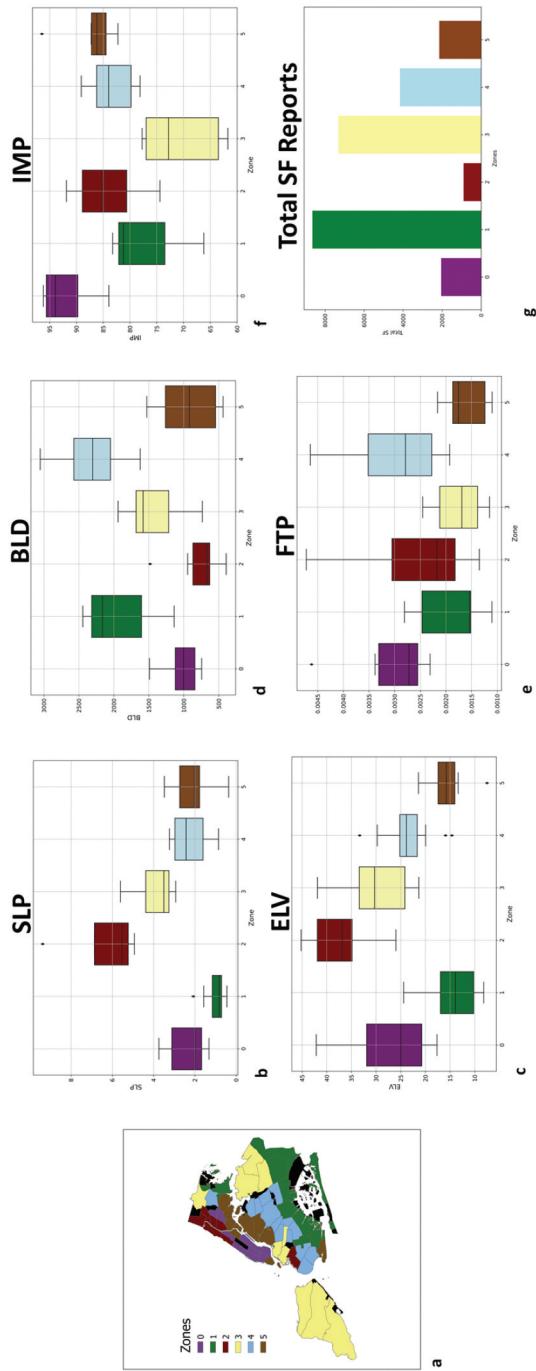


Figure 3. The six zones delineated by urban flood risk characteristics. Figure 3(a) depicts the mapping of the zonal spreads per DIS. Figure 3(b-f) are box plots depicting the range of flood characteristics of DISs within each zone. For each zone, the extent of the mean slope, mean elevation, number of buildings per DIS area, the sum of total building footprint area per DIS area, and mean percent of impervious cover of each DIS are shown. Figure 3(g) shows the bar plots illustrating the sum of the total SF reports in each zone over a 10-year period.

The first set of models were created to predict street flooding occurrence (SF) based on daily values of climatic (MR, TR, SN) and infrastructural influences (SB and CB) over a 10-year time span. The research begins with a standard statistical model, the GLM. Features of GLMs include the provision of a link function and likelihood function; in addition, GLMs possess the ability to work with count data and nonlinearities (Hardin and Hilbe 2007). Next, the research moves to the FNN, a more complex, yet traditional AI-based model. FFNs introduce a layered architecture (input, hidden and output layers) of neurons, which gives the models the ability to sense an environment for subtle patterns (Fine 2006; Setiono 2001). Also, FFNs have the benefit of adaptability, where elements may be added, allowing for varying capabilities. Particularly, the architecture of FFNs provides the basis for the RNN and CNN. For instance, with the RNN, there is the feed forward mechanism as found in the FFN; however, the RNN is also equipped with a feedback loop, thereby enhancing short term memory and temporal dependency learning (Fei and Lu 2018; Schmidt 2019). Regarding the CNN, the feed forward process comprises of convolutional and pooling layers, where spatial (patterns across neighboring time steps) aptitude is achieved via filtering kernels (Albawi, Mohammed, and Al-Zawi 2017; Durairaj and Mohan 2022; Koprinska, Wu, and Wang 2018; M. Sun et al. 2017). When working with time-series data, coordinates of observation values are created, and the spatial capability refers to the comprehension of these numerical relationships in time, which is processed in parallel, as opposed to in sequence as with the RNN (K. Wang et al. 2019). Hence, the RNN and CNN are both capturing temporal dependencies; however, the mechanism of temporal incorporation differs. This study investigates the RNN and CNN to explore the effects of each added component towards the forecasting of SF within this specific data set. Overall, the beginning models, varying in architecture and temporal capabilities, utilize the time-series predictors to make predictions on the response variable representing street flooding in the urban setting.

Graph-based neural networks consider locational aspects. Since there is a geographical component to urban flooding, as localized areas of susceptibility are to be discovered, this model study further explores the benefits of the graph neural network architecture. For the GCN, the predictors remain as MR, TR, SN, SB and CB, with the response variable as SF. As there are only six nodes in this study, all within a reasonably close geographical proximity, the model was created so that each cluster is connected to all other clusters. Each cluster represents a node, and via edges, external information

is communicated among the nodes, thereby bridging potential gaps within each node's incomplete internal data (Jiang, Wang, and Luo 2023; Piao et al. 2022; Scarselli et al. 2009; Y. Wang et al. 2023). Ensuingly, this study explores the GCN and the GWN. For the GCN, the inputs are as described above. A tensor creates a fully-connected graph, where features from neighboring nodes are aggregated to assess how conditions in one zone may affect another zone. The process is achieved via two graph convolutional layers. Lastly, the graph neural network architecture is transfigured to the GWN. Here, the GWN had been adapted from Sun et al. (A. Y. Sun et al. 2021). Considered the most complex model of this study, the GWN incorporates all the aspects of the previous neural network models – locational, spatial and recurrent elements – while also introducing novel features of its own, such as gated layers and the self-adaptive adjacency (SAA) matrix. Moreover, due to the SAA, the GWN is able to incorporate static features. For this study, the following static characteristics were fed into the GWN: SIZ, SLP, ELV, LAT, LNG, BLD, FTP and IMP.

For every model in the exploration, the coefficient of determination (R^2) was used as the main validation measure. The dataset of variables was partitioned into training and testing sets. The training data ranged from 1 January 2010 through 1 January 2018; the testing data ranged from 2 January 2018 through 31 December 2019. Each model made predictions spanning the testing date range. This fixed train-test split reflects a realistic deployment scenario in which models are trained on historical data and applied to future outcomes. Given the temporal stability of the dataset and the absence of structural breaks, this approach was deemed sufficient for evaluating model performance without the need for rolling validation. Additionally, since two models incorporate LSTM components, relying on the sequential nature of time series data, traditional k-fold cross-validation was not suitable due to the risk of temporal information leakage. After the training and testing, the predictions were compared to the observed, and the R^2 was ultimately computed. Additional metrics, mean-squared error (MSE) and mean arctangent absolute percentage error (MAAPE) were also determined. Hence by evaluating the goodness of fit, model comparisons were conducted. Details on preprocessing, hyperparameter settings, regularization mechanisms, and reproducibility controls can be found in Appendix A (Table A1).

3.1. Spectral clustering

Spectral clustering, an unsupervised machine learning technique, partitions groups based on similarities. For

this study, the SpectralClustering tool from the sklearn module is utilized in Python (Scikit-learn 2023). The data points of each DIS are SLP, ELV, BLD, FTP and IMP. Each feature was transformed independently via Standard Scaler, processing for a mean of zero and standard deviation of one, so as to prevent disproportional influence on the algorithm's computation. Here, the features were represented as x_1, \dots, x_n . For each vertex (DIS), edges were constructed from x_i to its k-nearest neighbor, x_j . The Euclidean distance $[t(x_i, x_j)]$ between each unique pair of x_i and x_j was calculated (Scikit-learn 2023). Then, a measure of similarity (s_{ij}) was determined as follows (Scikit-learn 2023):

$$s_{ij} = e^{-10 \times t(x_i, x_j)^2} \dots \quad (1)$$

Edges were created between each pair, and similarity is used as the edge weight. The purpose of similarity weights is for the edges between a pair of points in the same group to have greater weights than the edges between a pair of points that lie in separate groups (von Luxburg 2007). An unnormalized Laplacian graph was formed, with the matrix defined by Luxburg as follows (von Luxburg, Belkin, and Bousquet 2008):

$$L = D - S \dots \quad (2)$$

S was the similarity matrix: $S = (s_{ij})_{i,j=1,\dots,n}$ and D was the diagonal matrix with passes: $d_{ij} = s_{ij} \sum_{j=1}^n s_{ij}$. With the

computation of L , eigenvalue decomposition was initialized. The solver used was ARPACK, which computes k eigenvectors of L : v_1, \dots, v_k (von Luxburg 2007). Here, k is six (the number of desired flood zones). Let V be the matrix, where the eigenvectors were columns, and q_i represented the vector in the i -th row of V , then via the k-means algorithm, the points, $(q_i)_{i=1,\dots,n}$ were grouped into clusters (von Luxburg 2007). Hence, with the preceding machine learning technique, each of 59 DISs were grouped into six zones (labeled 0, 1, 2, 3, 4 and 5), based on the physical and urban traits.

There are physical characteristics which influence a region's susceptibility to SF complaints. The primary intent of the spectral clustering application is informed delineation. Nevertheless, a further advantage is the depiction of regions sharing similar extents of known, physical, flood factors, as detected in the Random Forest model by Agonafir et al. (Agonafir et al. 2022) and discussed in previous urban flood literature. The Standard Scaler function was employed so that each attribute holds comparable influence. Hence, for this study, the purpose of the spectral clustering model is not to serve as a discovery or predictive model, as it is preparing the

data for the statistical model and supervised machine learning explorations.

By spectral clustering, six zones were designated based on the flood factors. The quantity of six was chosen to provide a higher degree of localization compared to county or borough levels. Alternative cluster counts, such as nine, were tested, but these resulted in overly sparse groupings that reduced the reliability of subsequent model evaluation. To illustrate the prevalence of each trait within a zone, box plots were created (Figure 3(b-f)). Each box plot is comprised of the values of the DISs within the specified zone. The plots illustrate the range of SLP, ELV, BLD, FTP and IMP. Also, the total SF complaints, over the 10-year timespan, for each zone is depicted in Figure 3(g). Since each DIS now belongs to a zone, the data must also be aggregated to the zonal level. For the dynamic variables, SF, SB and CB, the totals of each DIS with a zone were taken, and for MR, the mean of the DIS values within a zone were calculated. For TR and SN, the measurements were previously taken from a single source; hence, no aggregation was needed. For the static attributes (only used as an input to the GWN), SLP, ELV, LAT, LNG, BLD, FTP and IMP, mean values for each DIS within a zone were determined, and for SIZ, the sum of the areas of each DIS in a zone was calculated. Therefore, the dynamic predictors and response variables and the static characteristics for each zone were prepared for the performance of predictive modeling.

3.2. Poisson generalized linear regression model

The GLM used in this study is the Poisson GLM (hereafter referred to as GLM). Here, the GLM uses a log link function and a Poisson distribution of the exponential family. For each zone, the target variable and the explanatory variables were expressed at i -th observations as follows:

$$SF_i \sim \text{Poisson}(\lambda_i) \quad (3)$$

where,

$$\lambda_i = e^{\beta_0^0 + \beta_1^1 SB_i + \beta_2^2 CB_i + \beta_3^3 MR_i + \beta_4^4 TR_i + \beta_5^5 SN_i} \quad (4)$$

The β^k coefficients represent the strength of change in the log-relative rate of the SF for a one-unit change in the associated predictor variable, and β^0 is the intercept, which is the baseline rate when the predictors are zero.

3.3. Feed forward neural network

The FFN, applied to each zone independently, is composed of three layers: input, hidden and output. Figure 4 depicts the architecture. The input layer has the two infrastructural and the three climatic

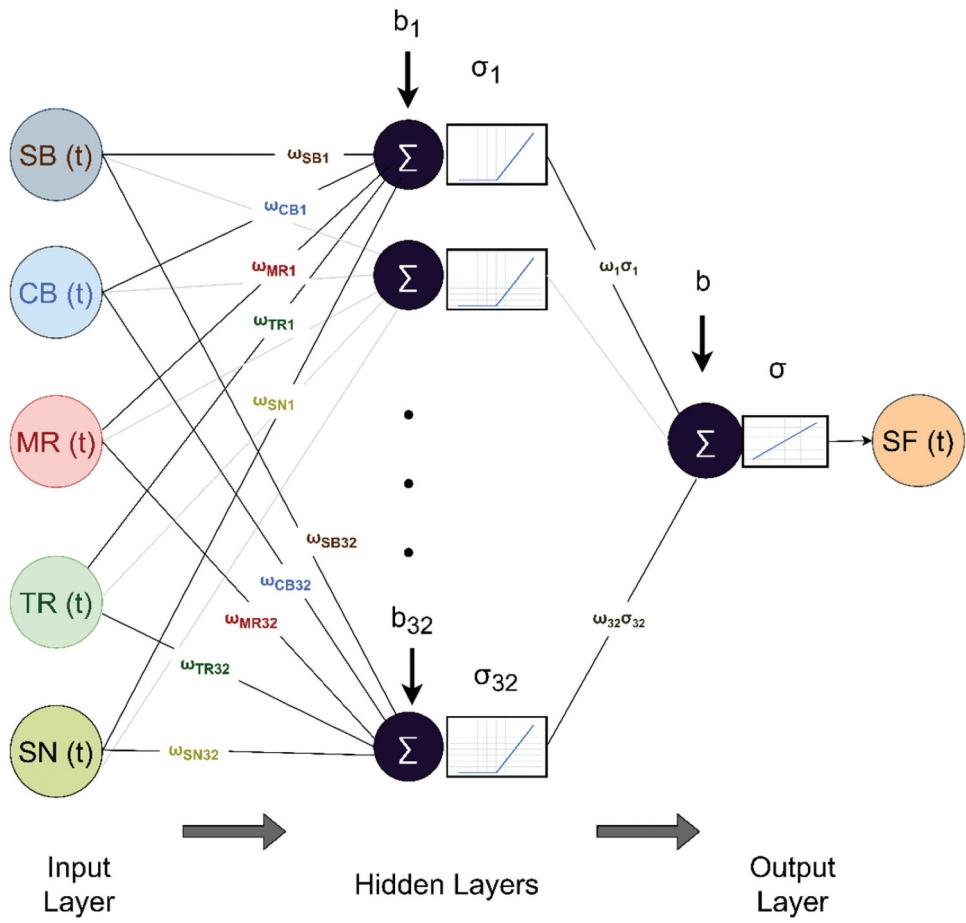


Figure 4. The FFN architecture.

predictors; each predictor is normalized using the Python sklearn’s Standard Scaler function. For this model, at each time step, there are 32 neurons within the hidden layer, and each predictor feeds forward to all neurons. For each connection, via the Adam optimizer gradient descent method, weights, ω , are initialized; moreover, for all neurons, a distinctive bias, b , is computed by random initialization. Then, for every neuron, i , there is a weighted sum calculation, as follows:

$$S_i = SB\omega_{SBi} + CB\omega_{CBi} + MR\omega_{MRi} + TR\omega_{TRi} + SN\omega_{SNi} + b_i \quad (5)$$

The sum then enters into the Rectified Linear Unit (ReLU) activation function, σ^R . Activation functions aid in understanding nonlinear relationships. The ReLU was chosen, as it is known for its accuracy and is widely used in deep learning modeling (Dubey and Jain 2019). The next step in the FFN is the forward movement of information from the last hidden layer to the output layer. Similar to the last weighted sum calculation, the weighted sum at the output neuron is computed:

$$S_{output} = \left(\sum \omega_i \sigma_i^R \right) + b \quad (6)$$

From the output layer, the data enters a linear function and produces the predicted counts of SF. The model was constructed via the PyTorch NN module (PyTorch 2023c). The model was run with a learning rate of 0.001, batch size of 32 and 100 epochs.

3.4. Recurrent neural network

The RNN architecture, built upon a network of neurons, is similar to the FFN structure. However, there is a difference within the hidden layer composition, where, as opposed to possessing only feed forward (FF) layers, the RNN includes a preceding long short-term memory (LSTM) layer (Figure 5). First, a concatenated input vector of the predictors at each time step enters each neuron. The vector is then concatenated with the hidden state vector at the previous time step. Then, via gradient descent optimization, unique (per neuron) input weight and bias are calculated. The input gate controls the extent of input information entering the

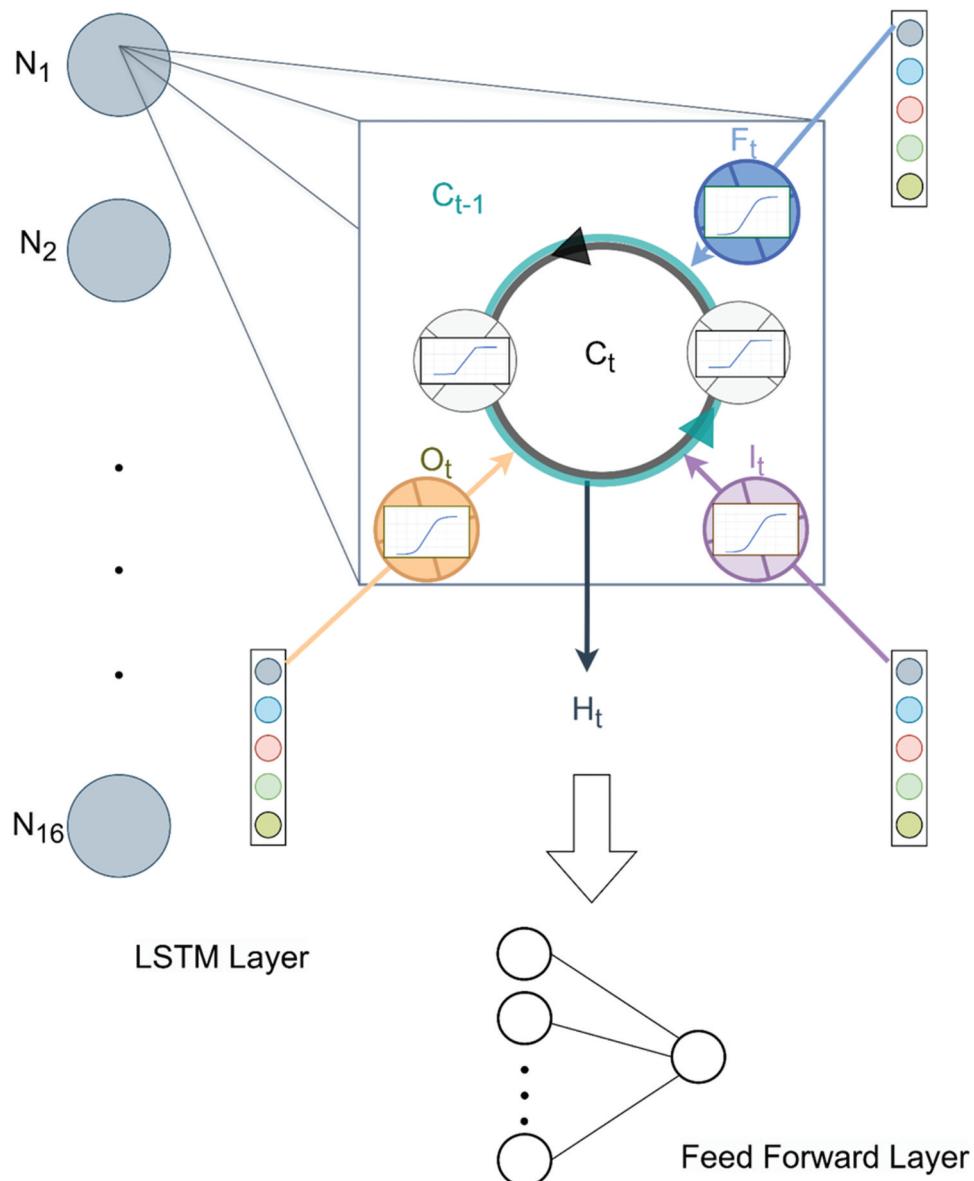


Figure 5. The hidden layer structure of the RNN.

cell state (Schmidt 2019; Tsantekidis, Passalis, and Tefas 2022), and the computations for each neuron of the 16 neurons are conducted. After the initial computation, the input enters the sigmoid activation function. Now, the forget gate, f , also receives the input vector of predictors; yet, its function is to filter out irrelevant information from the previous cell state (Schmidt 2019). Then, the sigmoid function is applied. With the productions of the input gate and the forget gate, the cell state, C , is computed. In the cell state computations, the tanh activation function is applied. Additionally, the input vector of predictors also pass through the output gate. The role of the output gate is to control the flow from the cell state to the hidden state and ultimately produce the output of the LSTM neuron (Chung et al. 2014). Finally,

with the product of the output gate, the hidden state, the information that passes to the next layer (the FF layer), is computed.

The process continues through the FF layer to predict SF counts. Here, the FF had 16 neurons. The RNN model underwent 1000 epochs, with a batch size of 32, a learning rate of 0.007, and a sequence length of six days. The model was constructed via the Pytorch LSTM module (PyTorch 2023b). The formulations [in alignment with (Farfán-Durán and Cea 2024; Kratzert et al. 2018)] for the LSTM are as follows:

$$f_t = \sigma(\omega_f x_t + u_f h_{t-1} + b_f) \quad (7)$$

$$i_t = \sigma(\omega_i x_t + u_i h_{t-1} + b_i) \quad (8)$$

$$\tilde{C}_t = \tanh(\omega_c x_t + u_c h_{t-1} + b_c) \quad (\text{Cell state candidate}) \quad (9)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (\text{cell state update}) \quad (10)$$

$$o_t = \sigma(\omega_o x_t + u_o h_{t-1} + b_o) \quad (\text{output gate}) \quad (11)$$

$$h_t = o_t \odot \tanh(C_t) \quad (\text{hidden state output}) \quad (12)$$

where,

σ = sigmoid activation

\tanh = hyperbolic tangent activation

\odot = Hadamard(element-wise)product

x_t = input at time t

h_{t-1} = previous hidden state

C_{t-1} = previous cell state

w = input to hidden weights

u = hidden to hidden(recurrent)weights

b = biases

3.5. Convolutional neural network

The CNN is also comprised of an input, hidden and output layer. The input layer and the output product are the same as the FFN; hence, the main difference from the FFN is the composition of the hidden layers. In particular, the CNN has two convolutional layers, a global average pooling layer and an FF layer. For the first convolutional layer, there are 16 kernels (or filters), f , of size one (each kernel is applied independently to each predictor at each time step). Each filter produces an output, c , by generating a unique bias and weight at each time step, t , and for each predictor. Therefore, in the CNN, the convolutional layers involve the weighted sum of predictors over time for each filter; this differs from the FNN, as in the FNN, the predictions are based on the weighted sum of predictors without considering the time dimension. Figure 6 illustrates the hidden layers of the CNN. The overview of computations are depicted here:

$$c_{f,t} = \omega_{f,t,SB} SB + \omega_{f,t,CB} CB + \omega_{f,t,MR} MR + \omega_{f,t,TR} TR + \omega_{f,t,SN} SN + b_{f,t} \quad (13)$$

The output of the first convolutional layer enters the second convolutional layer, c^* , via the ReLU activation function and computes the following:

$$C_{f,t} = \sigma(c_{f,t}) \dots \quad (14)$$

The second convolutional layer has 32 kernels, f^* . For each filter output, c^* , at a time step, the calculation is shown:

$$c_{f^*,t}^* = \sum_{f=1}^{16} \omega_{f^*,t,f} c_{f,t} + b_{f^*,t} \dots \quad (15)$$

The output of each kernel in the second convolutional layer also enters the ReLU function to gain enhanced pattern recognition:

$$C_{f^*,t}^* = \sigma(c_{f^*,t}^*) \quad (16)$$

After exiting the second convolutional process, there is input into the global average pooling layer. For each of the 32 channels, p , an average, g , is taken across all time steps (3652 days):

$$g_p = \frac{1}{3652} \sum_{t=1}^{3652} C_{f^*,t}^* \text{for } f^* = p \quad (17)$$

Now, the pooling output units are then processed by a feed forward layer of 32 neurons, producing predicted SF counts. The model was run with a learning rate of 0.001, batch size of 32 and 100 epochs. The model was constructed via the PyTorch CONV1D module (PyTorch 2023a).

3.6. Graph convolutional neural network

The main contribution of graph-based methods is the sharing of information via neighbors. The input predictors are the same as the previous models; however, as opposed to assessing each node's predictors singularly and running separate models for each, the input vector includes all the nodes and their respective predictor values at each time step. Within the model, individual assessments occur, and SF counts are produced for each node. At the hidden layer, the GCN begins the process by the creation of edges, the advanced communication channels between nodes of locational proximity. In this study, each cluster represents a node, and the edges are constructed such that each node is fully connected to every other node. Figure 7 illustrates the edges connecting the clusters (nodes) of this study. An input vector of the dynamic predictors, X at each time step is entered into each node, c . Additionally, X is also messaged into neighboring nodes, N (those nodes connected to the node by edges), via aggregation by edge weights. The messaging, m , calculation, for a node is as follows:

$$m_c = \sum_{i \in N_c} \omega_{c,i} X_i \quad (18)$$

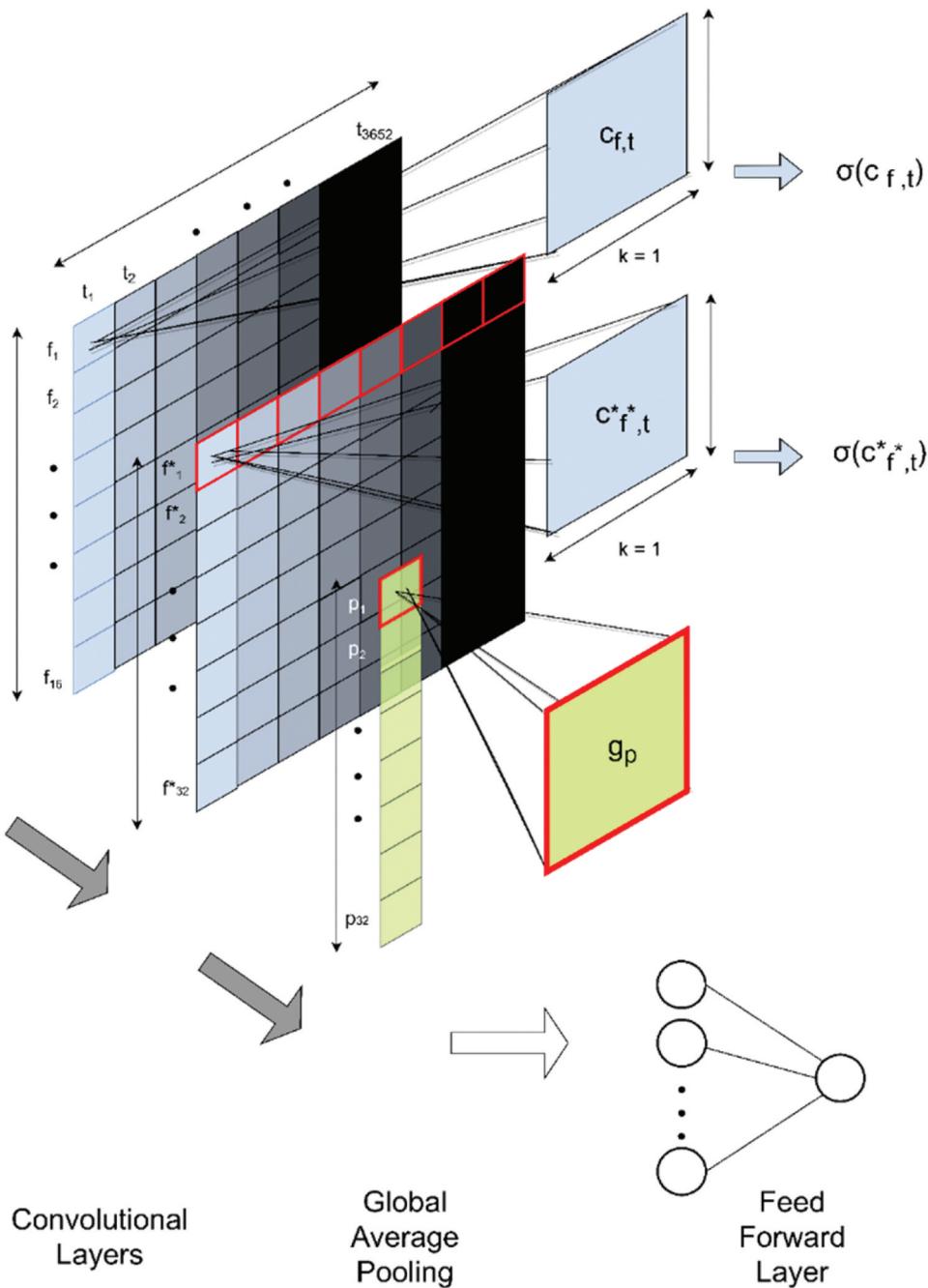


Figure 6. The hidden layer structure of the CNN.

After messages are aggregated, feature representation is updated at the next time step:

$$X_{c,t+1} = f(X_{c,t}m_c) \dots \quad (19)$$

The aggregated updated node representations occur within the graph convolutional layers. The model is constructed via the PyTorch Geometric GCN module (PyTorch Geometric 2023). There are two convolutional layers, of which the model acquires knowledge of the effect each neighbor's feature may have on another. The

ReLU activation function was employed. The learning rate was 0.001, and the GCN underwent 100 epochs.

3.7. Graph wavenet

The GWN is graph-based, with sophisticated recurrent and convolutional aspects. The key advances from the GCN, RNN and CNN are the additions of the adjacency matrix, A , dilation factors, l , and skip connections. The input set of dynamic predictors and the output of predicted SF counts are similar to the other graph-based

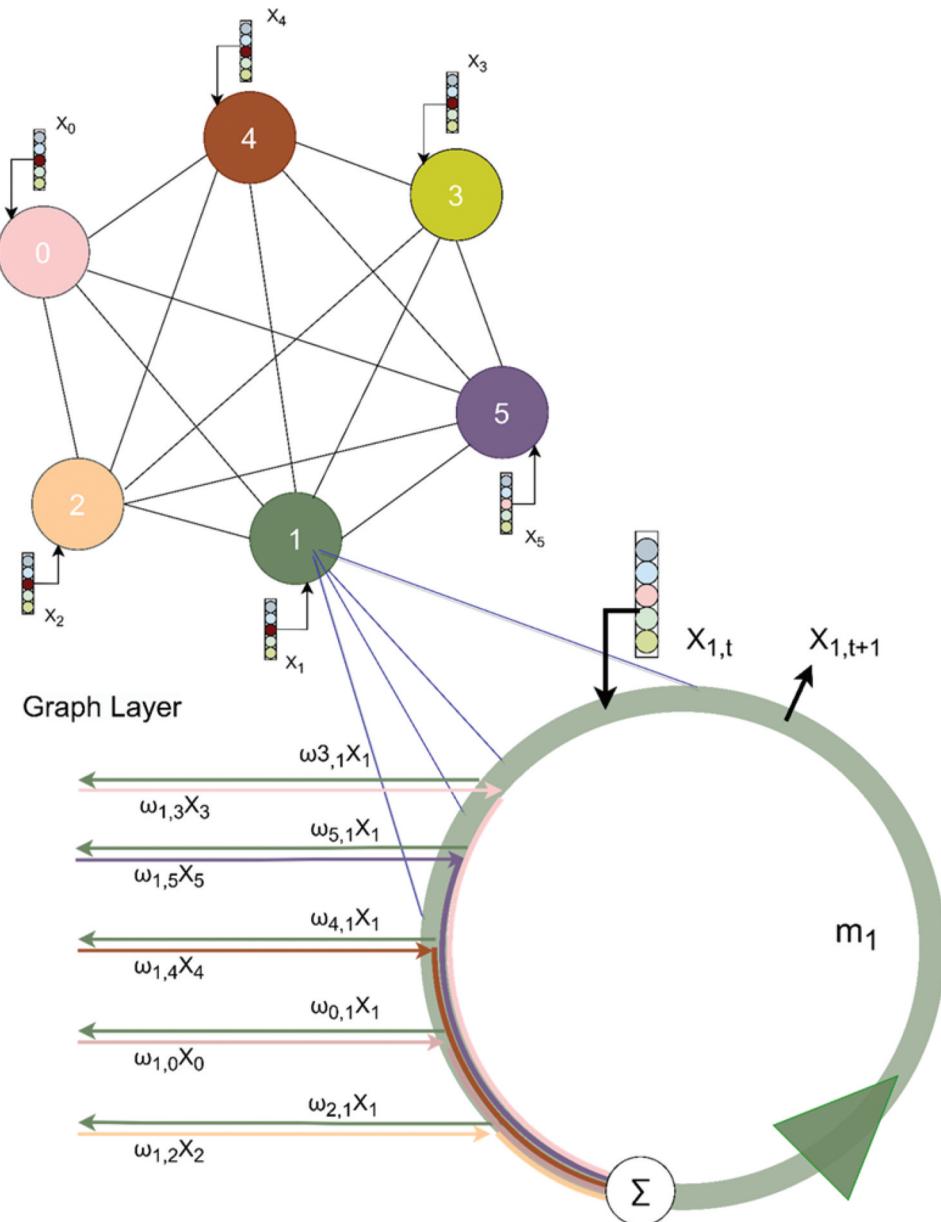


Figure 7. The graph structure of the GCN.

models. Although, for the GWN, the arrangement of the input differs. The input is a vector, D , which includes the nodes and the dynamic predictors per node for each sequence, q . Here, the sequence is six; thus, the vector includes information from t , $t-1$ and so forth until $t-5$. D then enters the A , along with the static attributes (SIZ, SLP, ELV, LAT, LNG and BLD), T . With the data, A makes informed decisions towards graph construction and the determination of node neighbors, via edges. Moreover, for each unique filter, A creates a unique transformed input vector for a particular node. Now, for each node, n , there is a transformed input vector, described as follows:

$$X_{n,f} = f(D, T, q) \dots \quad (20)$$

The transformed vectors for each node then enter convolutions. Recall from the CNN section, that the convolution operation, c , is a function of the predictors, weights and biases, where the weights and biases are unique for each time step and filter. It is similar here; however, for the GWN, it is also a function of the dilation factor. For the GWN, the output of the convolution operation will be referred to g :

$$g_{n,f} = f(X_{n,f}, \omega_f, b_f, l) \dots \quad (21)$$

Next, g is split, where one enters into the tanh activation function, and the other is processed by the sigmoid activation function. While information passes through the sigmoid and tanh activation functions in the RNN,

this occurs at sequential steps. Whereas, for the GWN, the transformations by the sigmoid and tanh activation functions are brought together in element-wise multiplication. The determinations of the output tensors of the sigmoid activation function, G^S , and the output of the tanh activation function, G^T , are shown here:

$$G_{n,f}^S = \sigma_f^S(g_{n,f}) \dots \quad (22)$$

$$G_{n,f}^T = \sigma_f^T(g_{n,f}) \quad (23)$$

An element-wise multiplication, M is then performed on both tensors.

$$M_{n,f} = G_n^S \odot G_n^T \dots \quad (24)$$

The input tensor, M, then passes through a 1×1 convolution, where a point-wise convolution operation takes place, reducing the hidden dimensions from 11 to one. The output, P, is described here:

$$P_{n,f} = f(M_{n,f}, \omega_{n,f}, b_{n,f}) \dots \quad (25)$$

Now, during this process, the original input tensor, X, is also preserved. It is added, elementwise, to P, to produce a residual connection, r. In this way, the output not only learns from the transformed input, via convolutions and gating mechanisms, but it also learns from its original input.

$$r_{n,f} = P_{n,f} \oplus X_{n,f} \quad (26)$$

Meanwhile, there is another part of the output of the 1×1 convolution with a different utility. This output, known as a skip connection, will be designated as s:

$$s_{n,f} = f(M_{n,f}, \omega_{n,f}, b_{n,f}) \dots \quad (27)$$

At this point, the calculations have been shown at the filter level. As explained in the CNN section, the calculations of each filter are aggregated to the layer level. Hence, describing the calculations at the layer level, the residual connection exiting the layer, y, will be denoted as R, and the skip connection exiting y will be denoted as S. For this model, R exits the filter and is utilized in subsequent skip connections, S exits the filter, and residual connections from previous layers are incorporated to form a feedforward output, K. Let z represent the quantity of layer skips, then the calculation of K is as follows:

$$K_y = s_{n,f,y} \oplus \sum_{z=0}^z R_{n,f,y-z} \quad (28)$$

The final output then undergoes a ReLU activation function, and the model prepares predictions for SF counts on the testing dates. The process is shown in Figure 8. For this study, the hyperparameters for the

GWN included a 0.01 learning rate, batch size of 15, and 50 epochs.

4. Results and discussion

4.1. Risk zones

Spectral clustering, an unsupervised machine learning technique, created six zones based on the flood attributes of each DIS. Figure 3 shows the predominance of each of the five characteristics within a zone and plots total SF occurrence in each zone. By a visual analysis of the plots, elements of risk are conveyed. For instance, zone 2 is shown to have the highest mean incline (Figure 3(b)), allowing for water to flow, as opposed to ponding. Moreover, zone 2 has the greatest elevation (Figure 3(c)) and the least number of buildings per unit area (Figure 3(d)). A higher elevation and lower quantity of buildings are known to reduce urban flood susceptibility. Subsequently, the physical qualities of zone 2 may serve as a plausible explanation for the zone having the lowest total SF complaints (Figure 3(g)). Similarly, zonal characteristics may also explain SF occurrence in zone 1. When viewing Figure 3(b-d,g), it is shown that zone 1 has the opposite extent, with the flattest surface (low slope), lowest elevation, and the second highest number of buildings per square unit; notwithstanding, zone 1 has the most SF reports. Thus, the extensive flood risk characteristics of zone 1 may be the antecedent for its high complaints. Concerning the remaining zones, the prevalence of a combination of flood attributes is not as strongly skewed. For example, zone 0, which has a low SF total, ranges mediumly in SLP, ELV and BLD; although, it prevails on the higher end for FTP, and it has the greatest IMP. Regarding zone 3, there are no extremes in flood characteristics; yet, zone 3 has the second highest total SF. With zone 4, it has the highest BLD and FTP; yet the values of the other flood attributes extend moderately. Lastly, concerning the box plot of zone 5, it does not depict many extremes in flood characteristics, except a relatively low FTP and high IMP. Regarding total SF, zone 5 retains low SF reports (slightly higher than that of zone 0). Overall, via visual inspection, it appears that some flood characteristics, particularly BLD, SLP and ELV, maintain stronger effects on total SF reports. This was also found in Agonafir et al. (2022), where the random forest algorithm detected the relative importance of BLD, SLP and ELV to be greater than IMP and FTP. For the zones with moderate flood characteristics, modeling forecasting techniques, specifically machine learning methods, have utility, as the algorithms possess the ability to detect intricacies within a learned environment.

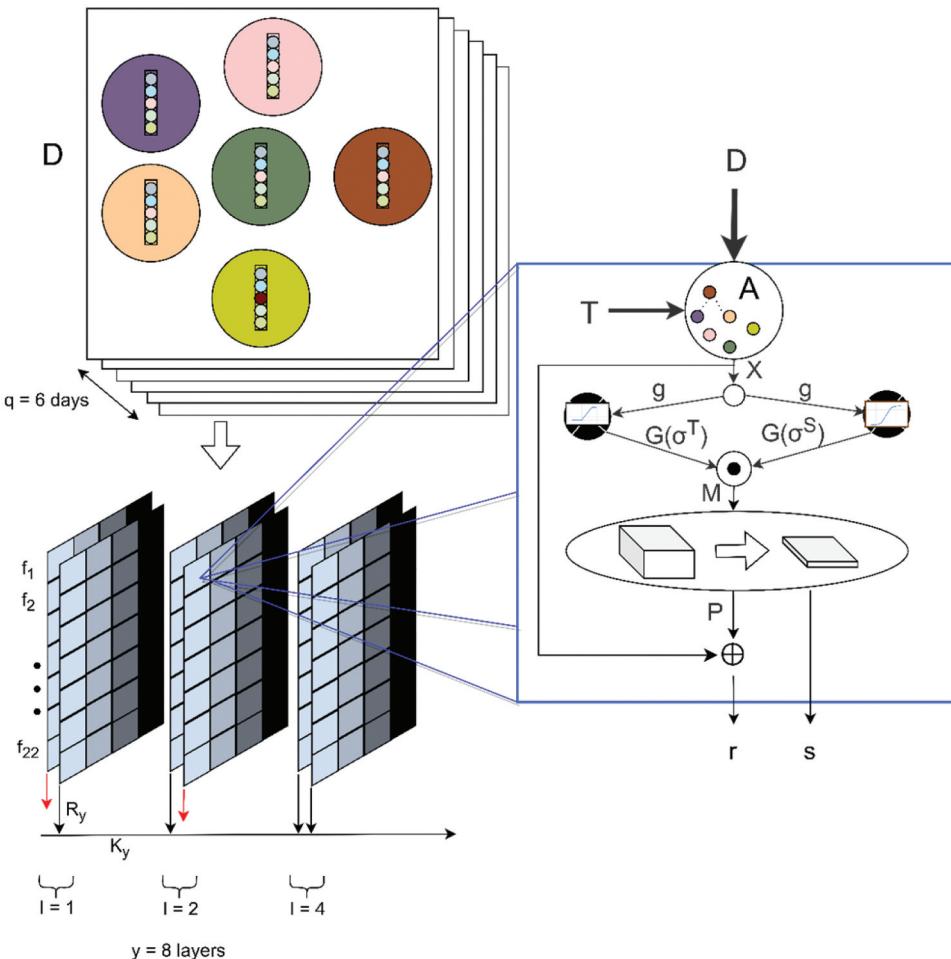


Figure 8. A diagram depicting GWN configuration.

4.2. Model results

4.2.1. Validation results of models

The evaluation of the models with varying complexities is conducted via the assessment of R^2 values. Each model utilized a training set between 2010 to 2018 to produce daily predictions for the testing range from 2018 through 2019, and goodness-of-fit was determined. Each model, aside from the GLM, underwent 50 runs, and the average of the mean and median R^2 values of all nodes were tabulated in [Table 1](#). The model results are mapped, at the zonal level, in [Figure 9](#). Ordered from least to greatest, the results, depicting the mean R^2 , for

the models are the following: GLM (0.26), FFN (0.35), RNN (0.36), CNN (0.36), GCN (0.43) and GWN (0.51). Thus, the GWN is the model with the best performance. Moreover, the delta between the model with the lowest R^2 (GLM) and the highest (GWN) is found to be significant at a value of 0.25. Furthermore, when only comparing ANNs, there is a 0.16 delta between the GWN's R^2 (highest) and FFN's R^2 (lowest). Lastly, inspecting the difference in R^2 between the simplest introduction of the feed forward (FFN) to the statistical model (GLM), there is a 0.09 delta, and between the simplest graph neural network (GCN) to the GLM, there exists a 0.08 delta.

Table 1. A listing of the R^2 values for each model at each zone and the mean and median determinations.

Zones	0	1	2	3	4	5	Median	Mean
GLM	0.20	0.20	0.24	0.38	0.25	0.27	0.25	0.26
FFN	0.26	0.39	0.17	0.57	0.39	0.34	0.35	0.35
RNN	0.25	0.39	0.23	0.56	0.40	0.33	0.36	0.36
CNN	0.26	0.39	0.22	0.57	0.40	0.34	0.36	0.36
GCN	0.37	0.45	0.28	0.58	0.44	0.44	0.43	0.43
GWN	0.35	0.59	0.31	0.72	0.55	0.52	0.54	0.51
Median	0.26	0.39	0.24	0.57	0.40	0.34		
Mean	0.28	0.40	0.24	0.56	0.41	0.37		

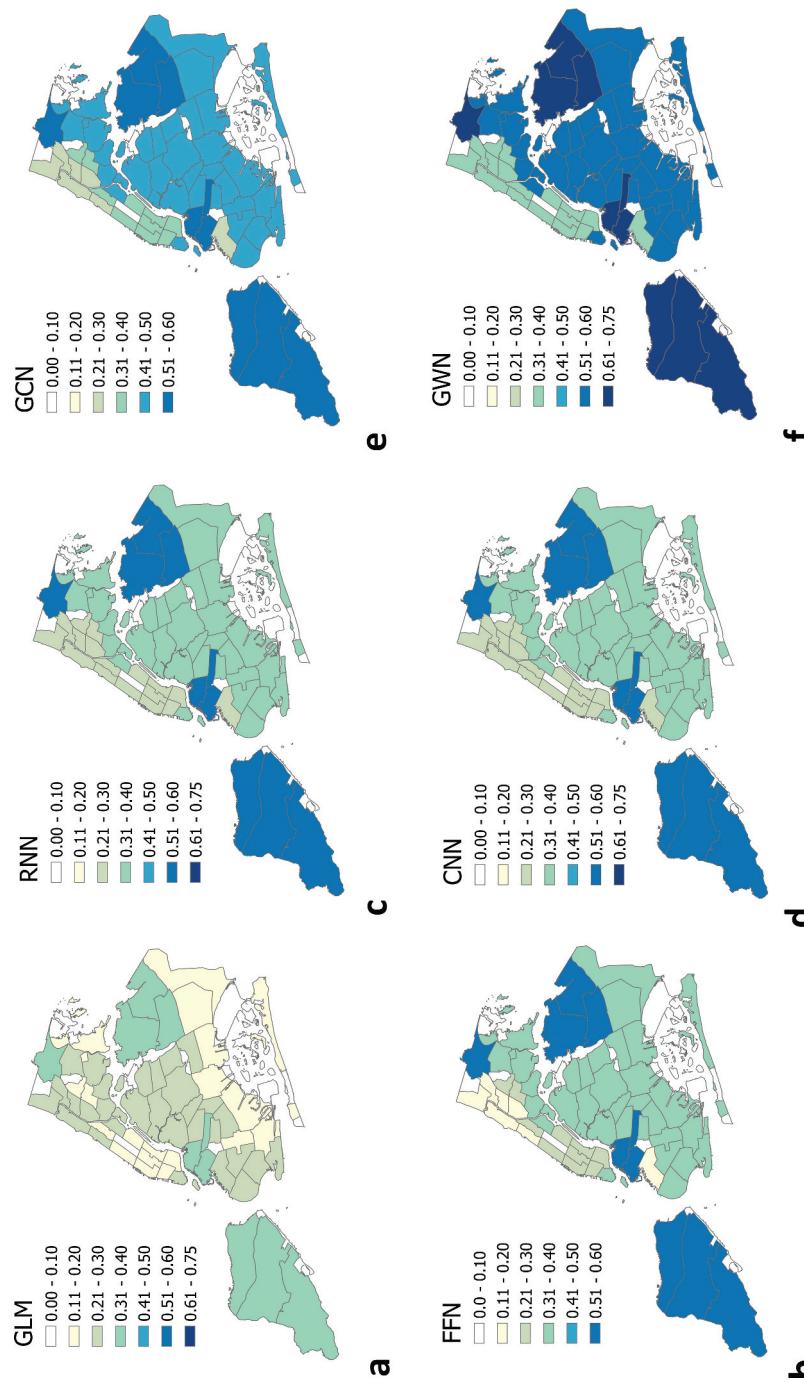


Figure 9. Maps of the mean R^2 values for each of the models in the exploration.

Contiguously, the results bring forth apparent connections. Firstly, there is notable performance improvement from the feed forward models (FFN, RNN and CNN) to the GWN, while minimal (in some zones) to no improvement exists from the FFN to the RNN or to the CNN. This indicates that LSTM and convolutional layers benefit from the extensive structural detail, such as gating mechanisms, dilations and skip connections, of the GWN. Secondly, as illustrated by the increase in R^2 achieved by the FFN when compared to the GLM, a simple machine learning model outperforms the generalized linear regression model. Finally, there is an overall improvement from the introduction of the graph architecture to the feed forward process, as highlighted by the mean performance of the GCN being greater (by 0.07) than the RNN, and CNN. This suggests that the graph-based structure assists in environmental learning for certain datasets. Principally, due to the substantial differences in model performance, the results highlight the importance of careful consideration towards deciding an appropriate forecasting technique.

Flash floods induce rapid, large-scale fluctuations in urban floods, thereby posing greater dangers (reducing the time to undertake preventive action) than sustained flood conditions of prolonged uniform rainfall. Consequently, R^2 is adopted as the principal evaluation metric, as it quantifies the fraction of variance explained by the model, capturing performance across extremes. MSE and MAAPE statistics will also be reported for completeness; however, these error-based measures uniformly weigh all residuals and do not preferentially emphasize extreme high-variance events. The first additional metric is MSE, which is calculated as the average of the squared residuals. Secondly, MAAPE is employed, as there may be days on which there are no reports of flooding, and to accommodate the zero values, MAAPE applies slope as an angle, as opposed to slope as a ratio (Kim and Kim 2016). Specifically, MAAPE is computed by taking the arctangent of each absolute percentage error and then averaging (and scaling) across all time steps. Therefore, while it is essential to foresee high flooding events, particularly for human safety, MSE and MAAPE allow for a more holistic view of the models' capabilities.

MSE and MAAPE results are shown in Table 2. A notable outcome is the GCN having the lowest MSE (7.33) and the second lowest MAAPE (53.70%). Considering that the GCN also has a strong performance in the R^2 metrics, the model proves to be stable in SF predictions. Next, it is observed that the GWN has the lowest MAAPE at 33.81%, which is 37% lower than that of the GCN. Yet, it has the second highest MSE (9.11). As the GWN has the highest R^2 , the metrics indicate that the GWN has relative accuracy; however, it has large absolute errors, at times. Lastly, the GLM, interestingly, has the second lowest MSE. An explanation may be that the GLM exhibits a conservative bias by regularizing its predictions toward the overall mean flood count. As a result, it may underestimate extreme flooding. This implication is supported by the plots of observed and predicted flood counts for the testing period (Figure 10). Visually, it is seen that the GLM often underpredicts large flood count days; whereas, the neural network models acquire the peaks more efficiently. Lastly, it is important to note that the graph neural networks surpass all models in every metric. The connection of the nodes by edges, which the graph neural networks create, allow for better urban flood prediction capabilities.

4.2.2. Zonal analysis

As spectral delineation has created each zone with varying physical traits, analyzing the results at the zonal level provides additional insights into model strengths. First, it is essential to inspect the relationship between R^2 (Figure 11) and total SF per zone (Figure 3(g)). There appears to be a general trend, where R^2 values are greater in zones with a larger quantity of total complaints. For instance, zones 0 and 2 have lower total SF reports, and the R^2 values for these zones are also the lowest. Moreover, zone 3 has the second highest quantity of complaints, and it has the best performance of all models. Thus, more available response data appears to be a benefit to the models. Moreover, the graph-based models significantly boost the quality of predictions, especially in the cases of zones 0 and 5. This may be attributed to the message-passage capability of separate nodes (zones), such that a zone is not only learning its specific environment, but also gaining a sense of the surroundings. Nonetheless, while there appears to be a positive link between quantity of complaints and validation values, there are other factors influencing model performance. For instance, zone 1 has the highest number of SF reports; yet, the mean R^2 values are lower (0.40) than those of zone 4 (0.41), which has less than half the complaints. In addition, zone 5 outperforms zone 0 in all models (by a difference of 0.09 mean R^2), despite having a similar quantity of SF reports. These observations

Table 2. A listing of mean MSE and MAAPE values, across all zones, for each model.

Models	MSE	MAAPE (%)
GLM	10.52	52.66
FFN	8.18	67.34
RNN	8.07	67.61
CNN	8.05	67.29
GCN	7.33	53.70
GWN	9.11	33.81

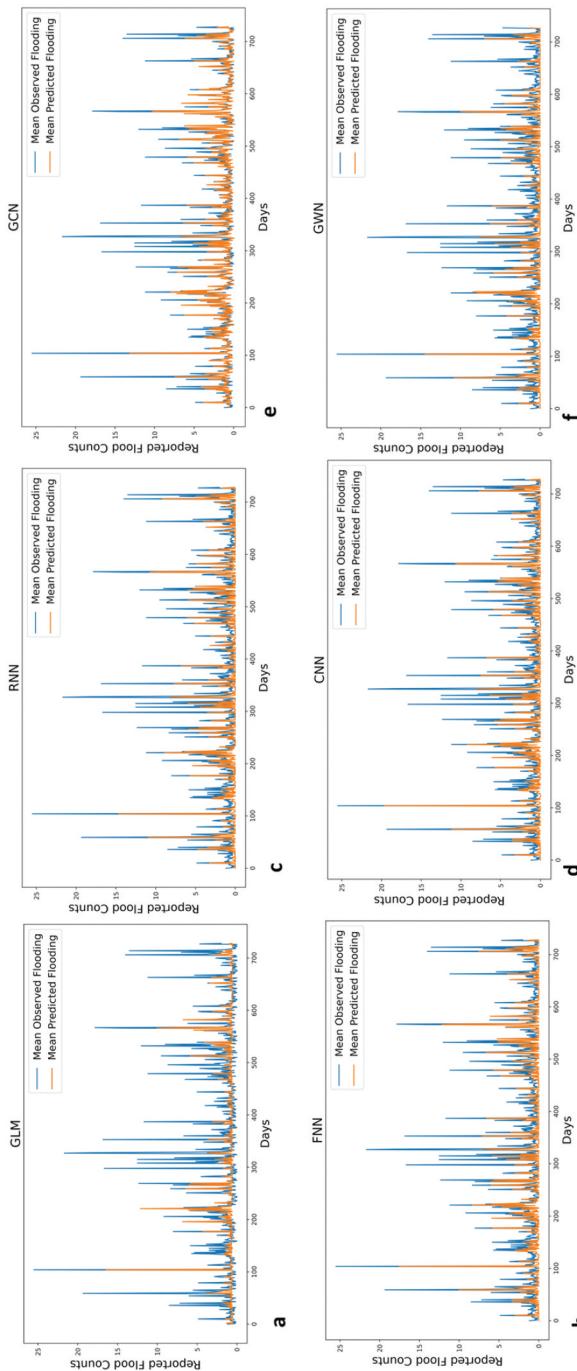


Figure 10. Daily plots for the observed and predicted flood counts for each model for the testing period. Figure 10(a-f)

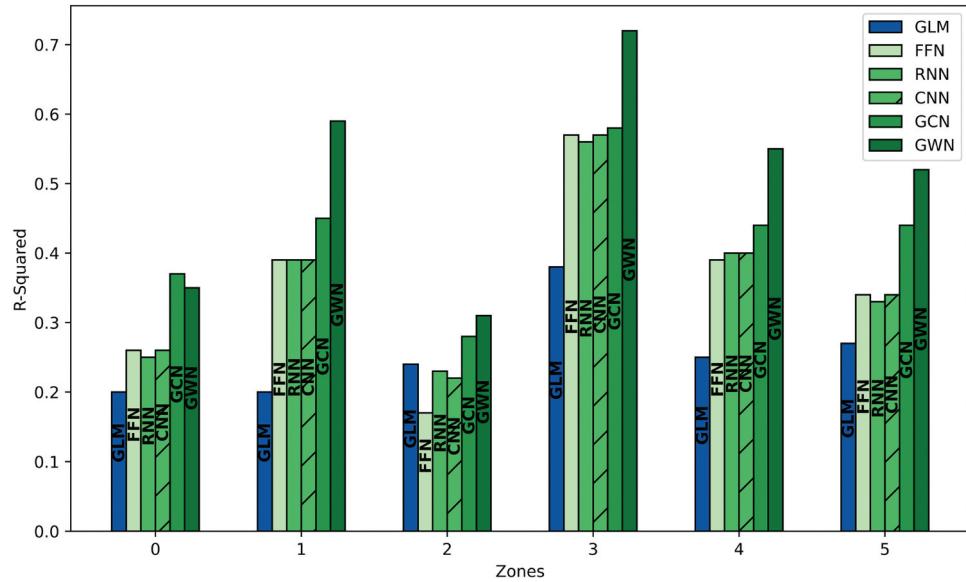


Figure 11. The bar plots of the mean R^2 of each model at each zone.

indicate the presence of other factors influencing zonal differences in modeling prediction aptitude. For example, concerning zone 1, additional variables may prove useful as model inputs. Specifically, zone 1 is comprised of regions mostly along the waterbodies of the Long Island Sound, Lower New York Bay, Jamaica Bay and the Atlantic Ocean; furthermore, this sector also has the lowest elevation. Thus, the zone's vulnerability to sea level rise could be heightened due to a combination of low elevation and proximity to water bodies. Accordingly, an additional variable expressing sea level rise may benefit modeling endeavors for the zone. Regarding zone 0, the lower performance within the cluster compared to zone 5 may be attributed to potential bias within the crowdsourced platform. Agonafir et al. found that commuters who drive are more likely to report SF (Agonafir et al. 2022). Given that zone 0 encompasses various Manhattan neighborhoods, and Manhattan is the borough with the highest influx of commuters (City of New York 2019), employing both public transportation and vehicles, the crowdsourced response data may exhibit subtle inconsistencies, posing challenges in detecting flooding patterns. Hence, the model performance in a zone may be affected by the amount of SF complaints, an insufficient set of variables or bias in the crowdsourced platform.

Next, the zonal analysis will gear towards the examination of individual model performance. An evident observation of the model results is the GWN exhibiting the highest R^2 [oftentimes, at a great margin] across all the zones, except for one. Now, as the sole deviation from this trend of GWN dominance occurs at zone 0, the continued examination at the zonal level will begin at

this curious exception. In the case of zone 0, the GCN outperforms the GWN by a marginal delta of 0.02. The difference is slight, as thus, considering the low volume of response data in the sector, the variation likely does not hold significance in model comparison. Nevertheless, within zone 0, there exist remarkable takeaways. The graph neural networks (GCN and GWN), when compared to the FFN, RNN and CNN, demonstrate stronger prediction accuracies. As the differences range from 0.09 to 0.12 in R^2 values at zone 0, the results pronounce the benefits of the graph structure. The graph neural networks add value by not only including locational elements, but by also allowing various areas to be connected and communicate with each other, and for a location, such as zone 0, with limited data, feeding a model with added information assists in the learning of environmental patterns. This merit of the graph-based architecture is additionally seen when inspecting the results of zones 1, 2, 4 and 5. Specifically, zone 5, also with low response data (the total SF complaints of zone 5 are less than a third of zone 3), obtains GCN and GWN R^2 values, ranging from deltas of 0.10 to 0.19 greater than those of the non-graph-based ANN models. The region where there is a balance in model performance between the GCN and the FFN, RNN and CNN is zone 3. Since zone 3 has a high number of SF complaints, the non-graph-based ANNs are not as encumbered by low data volume; hence, their performance is competitive with the GCN. Nonetheless, for all zones, the GWN transcends, and, due to key elements in the graph-based neural network structure, such as neighboring nodal data gains, the GCN also attains strong prediction results.

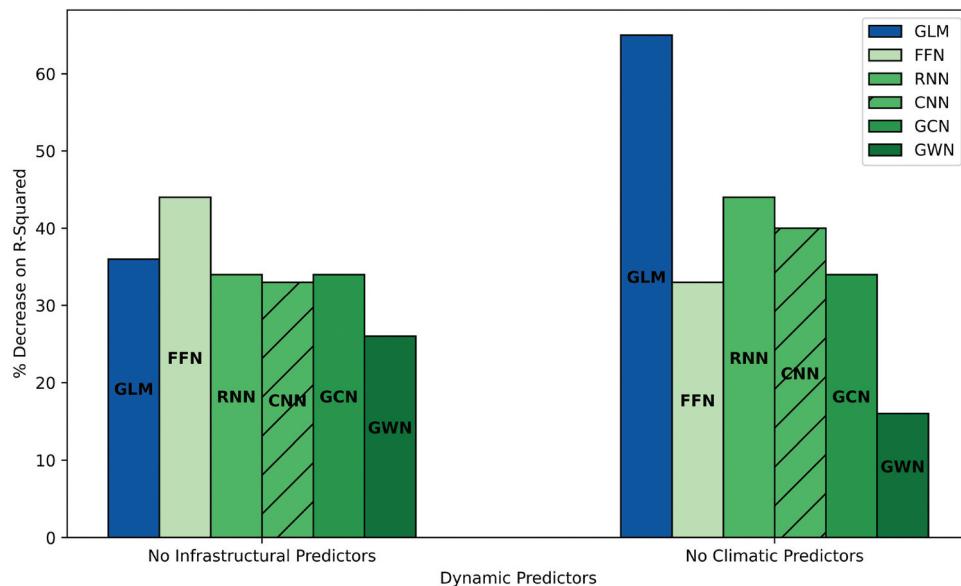


Figure 12. Plots of R^2 decrease with variable exclusion.

Also, at a zonal inspection, it is observed that the ANNs overwhelmingly outperform the GLM. The difference in R^2 values from GLM to an ANN range from 0.05 to 0.17 for zone 0, 0.19 to 0.39 for zone 1, 0.18 to 0.34 for zone 3, 0.14 to 0.30 for zone 4 and 0.06 to 0.25 for zone 5. Hence, it is seen that the employment of even a simple neural network may have substantial benefits to data-driven urban flood modeling. The only exception to the observation lies in zone 2. In zone 2, the GLM achieves a higher R^2 than the FFN, RNN and CNN by a difference of 0.07, 0.01 and 0.02, respectively. Zone 2 has the lowest SF reports, at roughly half of the response data as that of zone 0 (the zone with the second fewest) and less than one-eighth the SF complaints as zone 1 (the zone with the highest SF complaints). Therefore, an inference is that the GLM is not necessarily outperforming the other ANNs; however, due to very limited response data, the other ANNs are not performing at their fullest aptitudes.

Finally, while the zonal analysis has illustrated the advantages of ANN modeling for urban flooding and the benefits of applying a graph-based structure, the examination principally underscores the aptitude of the GWN. In Figure 11, the comparative analysis demonstrates that under conditions where the R^2 values of the FFN, CNN or RNN edge closely to those of the GCN, the GWN maintains a dominant performance. Conversely, when the validation determinations of the GCN ascend the other ANNs, the GWN upholds its position. The observed outcomes stem from the comprehensive nature of the GWN, incorporating key elements from the preceding ANN models. These advanced features include the integration of convolutional and LSTM layers

and the incorporation of the self-adaptive adjacency matrix within the graph architecture. Moreover, the structures within each component of the GWN are multiplexed, as opposed to simple additions. In a Section 4.3, Graph wavenet deconstruction, the ancillary complexities and their influence on model results will be explored in more detail.

4.2.3. Feature importance

To illustrate how flood-related factors affect a model's performance, a feature importance analysis is conducted. The infrastructural dynamic variables (SB and CB) are removed from each model, and the results are compared against the original R^2 values of each model (when all the variables are present). Likewise, the climatic dynamic variables (MR, TR and SN) are also removed and compared against the original R^2 values. This approach, which assumes a linear relationship between input variables and the target variable, provides insights into the contributions of the variables in isolation. The difference in R^2 values for each set of variable removal from the original R^2 is divided by the original R^2 to see the extent of effect. The results are shown in Figure 12. While acknowledging the simplicity of this method and its reliance on linear relationships, it offers an interpretable way to rank the importance of dynamic predictors within the context of the models.

There are a few notable observations of the feature importance results. First, the most perceptible effect is that of the climatic variables on the GLM. The GLM experiences a significant decrease when the precipitation predictors are absent. As precipitation is the fundamental cause of urban flooding, without its

representation, the simplistic calculations of the GLM do not suffice for strong prediction capabilities. Next, it is seen that the performance of the FFN is more reliant on the infrastructural variables than that of the CNN and RNN. An implication of this finding may be that the CNN and RNN models have more adept utilization of the seasonality and temporal nature of precipitation occurrence due to their enhanced spatial and sequential pattern recognition time-series data. Lastly, an observation is that the GWN appears to be less dependent on either set of variables. Of all the models, the R^2 of the GWN decreases by the least relative extent for both exclusions. This strengthens the assertion that the GWN is more robust, as it possesses an improved capability of acquiring the environment, despite being given a lean set of variables. Nonetheless, there are differences in performance decline, as the GWN performs better with the set of infrastructural-only variables. This observation may be attributed to its graph-based structure. Particularly, the graph-based models are aided by locational information, and the infrastructural predictors allow the models to sense the presence of chaos. If there are sewer backups and catch basin issues reported throughout the city, the models are alerted towards a more probable occurrence of street flooding. On an ending note, the feature importance plots identify model aptitude within a particular set of variables, further dissecting key model strengths and limitations.

4.3. Graph wavenet deconstruction

The GWN, extending beyond a basic encompassment of the characteristics of the ANN models in this study, elevates each fusion with intricate compositions and pathways. For instance, in its graph-based structure, although the foundational nodes and messaging through vertices (edges) are shared by the GCN, the GWN transcends with the incorporation of the advanced adjacency matrix. The matrix, skillfully integrating static attributes and facilitating early temporal review through sequenced data during the node-edge creation process, provides tailored inputs to each filter. Furthermore, in regards to the recurrent and convolutional aspects, the GWN builds upon the passage of information via dilation factors and residual and skip connections, exceeding the simplified structures of the RNN and CNN. Specifically, the RNN features a feedback mechanism via the LSTM layer, which learns information via a lookback period of six days. While the GWN also includes a lookback of six days, the temporal learning onsets at the adjacency matrix; additionally, the inputs proceed through gated activation functions and merge by element wise multiplication, as opposed to sequential summation

processes. This careful procedure allows for improved biases and weights; moreover, it reduces the occurrences of vanishing gradients or exploding activations, which are known risks of RNN modeling (H. Lin et al. 2022; Rangapuram et al. 2018). Now, transitioning to the comparison of the convolutional structure between the GWN and the CNN, the layers of the GWN exhibit a higher level of knowledge transfer. First, the dilation factor of the GWN allows for different convolutional layers to capture varying time ranges (Rathore et al. 2021); therefore, each layer brings a distinctive evaluation of pattern recognition, allowing for a holistic perception of the temporal environment. Second, the residual and skip connections enable a direct extraction from layers with pertinent information. To prevent distortion of the vital information as it passes sequentially layer to layer until reaching the current layer, the skip connections allow the current layer to retrieve the information of a previous (not immediately preceding) layer before it undergoes subsequent convolutions. As thus, the sophistication of the GWN convolutions allow for a more evolved and exclusive learning progression. Due to the complexity of the graph, recurrent and convolutional fundaments, the GWN not only incorporates, but ascends.

To highlight the improvements facilitated by the added components of the GWN, an additional model run is performed. In this GWN simulation (hereinafter referred to as GCR), the adjacency matrix is excluded, while preserving spatial and temporal aspects. The mean and median R^2 values are determined to be 0.46 and 0.47, respectively. The results may be found in Appendix B (Table B1). When contrasting the GCR with the GWN, which achieves mean and median values of 0.51 and 0.54, respectively, the impact of excluding the adjacency matrix becomes evident. The GCR experiences a decrease of 0.05 in the mean and 0.07 in the median. Additionally, at the zonal level, every cluster demonstrates a reduction in performance. With the absence of the GWN's adjacency matrix, the GCR lacks the integration of static features into its structure. This omission may limit its ability to capture key spatial relationships informed by physical characteristics such as elevation or impervious surface coverage. Therefore, the improved performance of the GWN relative to the GCR indicates the meaningful contribution of static features towards localized spatial learning.

Furthermore, the GCR, an adaptation of the GWN (GWN but without the adjacency matrix), allows for a comparison of the GWN's advanced convolutional and recurrent features (by which the GCR maintains) to the basic convolutional (temporal) features of the GCN. Notably, the primary distinction between the GCN and

GCR lies in the temporal and convolutional aspects. The results, wherein the GCR outperforms the GCN with a mean and median R^2 difference of 0.03 and 0.04, respectively, emphasize the strengths of temporal and spatial learning introduced by the GWN. Therefore, as theorized, the success of the GWN may be attributed to the multiple traits within the algorithm.

4.4. Advantages of graph-based models over conventional neural network models

The employment of graph-based models offers advantages over conventional neural network approaches such as RNNs and CNNs when modeling urban flood events. While RNNs and CNNs are effective in capturing temporal sequences and patterns, both architectures are inherently limited to fixed, independent learning. In contrast, flooding events often occur across irregular spatial units, such as neighborhood blocks or even underground, connected, subway lines. Inherently, flooding is an interdependently connected process. Water traverses land, neighborhood to neighborhood (node to node), often influenced by topography, land use and built infrastructure. Similarly, atmospheric precipitation does not fall randomly; it moves in spatially coherent patterns influenced by local weather systems. Thus, these physical and geospatial realities purport the requirement of a model that dynamically accounts for both physical proximity and shared environmental characteristics across non-contiguous areas.

Graph-based models directly address relations via edge communications, allowing the model to connect nodes that share flood-relevant traits, while acknowledging geographical contiguities. In this study, the GWN, by combining graph convolution with temporal gating, produced the highest performance across most metrics, effectively learning spatial-temporal dependencies with greater nuance. In cases where the GWN did not produce the top metrics, the GCN typically did, further demonstrating the strength of graph-based models in capturing flooding dynamics. These results support the conclusion that the flexible spatial representation enabled by graph neural networks is well-suited for real-world flood forecasting. Unlike conventional artificial neural network models, graph-based architectures reflect the underlying connectivity and rippling nature of flooding, thereby providing effective capabilities.

4.5. Limitations and future considerations

4.5.1. Computational efficiency

An evaluation of computational efficiency across the models provides important context for their practical

implementation, particularly in real-time forecasting scenarios. In terms of processing requirements, the GWN is the most resource-intensive model due to its layered architecture and reliance on graph-based convolutions and gated temporal operations. However, when applied to count-based datasets of moderate size, such as the one used in this study, the computational burden remains manageable. Concerning the GLM, FFN, RNN, CNN and GCN models, the models exhibited relatively similar runtimes and lower computational demands. Nevertheless, given the appreciable predictive performance of the GWN, the additional computational cost may be justified, particularly in applications where accuracy and spatial-temporal nuance are prioritized, such as real-time urban flood forecasting.

4.5.2. The FFN, RNN and CNN

The added complexities by the RNN and CNN yield minimal model performance. The FFN obtains a mean R^2 value of 0.35, and the RNN and CNN each obtain mean R^2 values of 0.36. Only in zone 2, there exists a substantive improvement by the RNN and CNN, where the R^2 values are 0.06 and 0.05 greater than the FFN, respectively. A plausible explanation for the limited benefits is the simplicity of the model layers, which lack the detail and mechanisms needed to produce discernable results. This assertion gains credibility when considering the performance of the GWN, where an interleaved system of data flow, employing skip connections and advanced gating mechanisms, achieves superior prediction accuracy. Another contributing factor may be the absence of significant temporal dependency within the dataset. Given the nature of urban flooding, often ensuing in the form of a flash flood, where the onset and finality of the disaster occurs within a brief timespan of 6 hours (NWS 2022), flooding on one day, oftentimes, does not exert influence on the following day. Simplified models designed to capture spatial and temporal dependencies may overlook these subtle patterns. Lastly, many of the variables, including the response variable, are retrieved from the NYC 311 dataset, and crowdsourced data is not as accurate in illustrating the environment as physical measurements. Thus, a simplified feedback loop or spatial assessment may not suffice. The limitations of the crowdsourced platform are further discussed in the next section. In summary, while the neural network architecture attains noticeable improvements in model performance, the basic LSTM or convolutional layers are not as advantageous; this may be attributed to the temporal nature

of urban flooding and the limitations of crowdsourced data.

4.5.3. The crowdsourced platform

Crowdsourcing has been applied in previous urban flood modeling initiatives, particularly in cities like NYC, where flood data is scarce. The incorporation of residential reports provides insights into flooding occurrences, which, otherwise, would not be obtained. However, while this method is valuable, the leveraging of eyewitness accounts is not as exact as physical measurements. Also, another issue of crowdsourced data is the potential for bias – a greater inclination of certain types of people to report issues. For example, it has been indicated that certain socio-demographical attributes may be factors in SF complaints, thereby possibly indicating that a particular set of residents are more likely to utilize the platform (Agonafir et al. 2022). Hence, there are bias concerns. Nonetheless, there exists strength in the reports, as they are taken by individuals observing an event. For instance, the bias attributed to socio-demographics is a *potential* consideration, since it may also be inferred that specific neighborhoods, prone to flood occurrences, may be comprised of a certain set of socio-demographics. Moreover, the validity of the crowdsourced data's depiction of flooding occurrence is reinforced by the climatic, topographical and infrastructural predictors holding substantial significance in the crowdsourced response variable, SF complaints (Agonafir et al. 2021, 2022). Furthermore, the 311 NYC street flooding reports, at the very least, capture the concerns of residents inclined to report, potentially identifying those most at personal risk. Indeed, the attribute of commuters who drive holds a sizable relative importance in Agonafir et al. (Agonafir et al. 2022), and as the leading cause of death from flooding is vehicular, accurate predictions in these regions may prove life-saving. In essence, in metropolitans with limited data, crowdsourcing, despite some drawbacks, enables the continuity of predictive modeling, sustaining efforts that would otherwise cease.

Moreover, in metropolitans, particularly in the specific study area of NYC, flood sensors (measuring water levels) are being installed, enabling proximate applicability of this explorative analysis. The findings presented here accentuate the models best suited for the local landscape, with the GWN delivering promising results. Despite the constraints of the crowd-sourced platform, the GWN attains an R^2 of 0.72 for zone 3, demonstrating its potential. Anticipating even greater predictive accuracy with actual measurements from sensors, this study outlines techniques

applicable to the urban city. Once physical data becomes accessible, this model exploration provides policymakers and stakeholders with an outline of the strengths and weaknesses of models, ascending in complexity, while also pinpointing the overall, most effective model for forecasting floods during a predicted rain event.

5. Conclusions

By a diverse, novel pooling of machine learning techniques, this study advances our understanding of urban flooding, offering detailed insights into risk zones, comparing the performance of various models, and emphasizing the effectiveness of graph-based neural networks, particularly the graph wavenet. Listed below are the key conclusions of the research:

- Spectral clustering has utilization in risk zone identification and border delineation. The analysis of these zones reveals relationships between specific physical characteristics (such as slope, elevation and building density) and the occurrence of street flooding. Notably, zones with higher elevation and lower building density exhibited lower susceptibility to flooding, emphasizing the importance of urban characteristics in flood risk assessment.
- Machine learning models demonstrate superior performance to the GLM. Unlike the GLM, which assumes linearity in the parameters, machine learning models offer greater flexibility by adapting to complex, nonlinear patterns present in the data.
- By a systematic evaluation of the performance of varying flood prediction models, ranging from traditional statistical models to advanced neural networks, the GWN emerges as the most suitable model for urban flood forecasting in NYC, outperforming other models, including the GCN, CNN and RNN. Hence, there is significance in incorporating advanced spatio-temporal aspects and dynamic graph creation for accurate flood forecasting.
- Via zonal analysis, graph-based structures are shown to be particularly beneficial in areas with sparse data, where traditional models may struggle.

The discoveries of this research accord practical value onto urban stakeholders, especially in cities such as NYC, where water measurement sensors are currently being deployed. This synergy between advanced modeling techniques, particularly spatio-temporal graph neural networks, and emerging sensor technologies ensures informed decision-making, enabling urban planners and emergency responders to

safeguard communities, reduce economic losses and enhance overall resilience to the challenges posed by urban flooding.

Acknowledgements

The authors would like to acknowledge funding from NSF grant 2019625. We also thank Dr Upmanu Lall for providing valuable feedback on this work.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The work was supported by the National Science Foundation [2019625].

Author contribution

Candace Agonafir: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – Original Draft and Preparation. **Tian Zheng:** Conceptualization, Formal Analysis, Funding Acquisition, Methodology, Project Administration, Resources, Supervision.

Code availability

Codes used in this study are available from the following links: <https://github.com/A-Candace/ANN-Exploration>; <https://github.com/A-Candace/GraphWavenet>.

Data availability statement

Physical, topographical, and crowdsourced data from this study are publicly available at NYC Open Data and NYC311. Radar and gauge data are available from EOL/UCAR and NOAA databases, respectively. The DOI for the dataset of this study is 10.17632/z2syvptbx4.1.

References

- AghaKouchak, A., N. Nasrollahi, and E. Habib. 2009. "Accounting for Uncertainties of the TRMM Satellite Estimates." *Remote Sensing* 1 (3): 606–619. <https://doi.org/10.3390/rs1030606>.
- Agonafir, C., T. Lakhankar, R. Khanbilvardi, N. Krakauer, D. Radell, and N. Devineni. 2022. "A Machine Learning Approach to Evaluate the Spatial Variability of New York City's 311 Street Flooding Complaints." *Computers, Environment and Urban Systems* 97:101854. <https://doi.org/10.1016/j.compenvurbsys.2022.101854>.
- Agonafir, C., T. Lakhankar, R. Khanbilvardi, N. Krakauer, D. Radell, and N. Devineni. 2023. *Water Security* 19 August (2023): 100141. <https://doi.org/10.1016/j.wasec.2023.100141>.
- Agonafir, C., A. Ramirez Pabon, T. Lakhankar, R. Khanbilvardi, and N. Devineni. 2021. "Understanding New York City Street Flooding Through 311 Complaints." *Journal of Hydrology* 605 (March 2021): 127300. <https://doi.org/10.1016/j.jhydrol.2021.127300>.
- Albawi, S., T. A. Mohammed, and S. Al-Zawi. 2017. "Understanding of a Convolutional Neural Network." 2017 International Conference on Engineering and Technology (ICET), Akdeniz University, Antalya, Turkey, 1–6. <https://doi.org/10.1109/ICEngTechnol.2017.8308186>.
- Al-Suhili, R., C. Cullen, and R. Khanbilvardi. 2019. *An Urban Flash Flood Alert Tool for Megacities-Application for Manhattan*. New York City, USA: Hydrology. <https://doi.org/10.3390/HYDROLOGY6020056>.
- Bruwier, M., C. Maravat, A. Mustafa, J. Teller, M. Pirotton, S. Erpicum, P. Archambeau, and B. Dewals. 2020. "Influence of Urban Forms on Surface Flow in Urban Pluvial Flooding." *Journal of Hydrology* 582 (December 2019): 124493. <https://doi.org/10.1016/j.jhydrol.2019.124493>.
- Bureau of Economic Analysis. 2021. "Metropolitan Statistical Areas."
- Chung, J., C. Gulcehre, K. Cho, and Y. Bengio. 2014. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. <http://arxiv.org/abs/1412.3555>.
- City of New York. 2019. *The Ins and Outs of NYC Commuting*. September. <https://www.nyc.gov/assets/planning/download/pdf/planning-level/housing-economy/nyc-ins-and-out-of-commuting.pdf>.
- City of New York. 2022a. *Flood Prevention*. <https://www1.nyc.gov/site/dep/environment/flood-prevention.page>.
- City of New York. 2022b. *Impact of NYW Bonds*. <https://www1.nyc.gov/site/nyw/investing-in-nyw-bonds/the-impact-of-investing.page>.
- City of New York. 2023a. *About Community Boards*. <https://www.nyc.gov/site/cau/community-boards/about-community-boards.page>.
- City of New York. 2023b. *NYC Open Data*. <https://data.cityofnewyork.us/City-Government/Community-Districts/yfnk-k7r4>.
- Dubey, A. K., and V. Jain. 2019. "Comparative Study of Convolution Neural Network's ReLU and Leaky-ReLU Activation Functions." In *Applications of Computing, Automation and Wireless Systems in Electrical Engineering*, edited by S. Mishra, Y. R. Sood, and A. Tomar, pp. 873–880. Springer Singapore.
- Durairaj, D. M., and B. H. K. Mohan. 2022. "A Convolutional Neural Network Based Approach to Financial Time Series Prediction." *Neural Computing & Applications* 34 (16): 13319–13337. <https://doi.org/10.1007/s00521-022-07143-2>.
- EOL. 2022. *NCEP/EMC 4km Gridded Data (GRIB) Stage IV Data*. https://data.eol.ucar.edu/cgi-bin/codiac/fgr_form?id=21.093.
- Falconer, R. 2021. "Hurricane Ida Death Toll Rises Past 60." *Axios*. <https://wwwaxios.com/2021/09/06/hurricane-ida-death-toll-rises-power-outage>.
- Fang, T., Y. Chen, H. Tan, J. Cao, J. Liao, and L. Huang. 2019. "Flood Risk Evaluation in the Middle Reaches of the Yangtze River Based on Eigenvector Spatial Filtering Poisson Regression." *Water (Switzerland)* 11 (10): 1969. <https://doi.org/10.3390/w11101969>.

- Farahmand, H., Y. Xu, and A. Mostafavi. 2023. "A Spatial-Temporal Graph Deep Learning Model for Urban Flood Nowcasting Leveraging Heterogeneous Community Features." *Scientific Reports* 13 (1). <https://doi.org/10.1038/s41598-023-32548-x>.
- Farfán-Durán, J. F., and L. Cea. 2024. "Streamflow Forecasting with Deep Learning Models: A Side-by-Side Comparison in Northwest Spain." *Earth Science Informatics* 17 (6): 5289–5315. <https://doi.org/10.1007/s12145-024-01454-9>.
- Fei, J., and C. Lu. 2018. "Adaptive Sliding Mode Control of Dynamic Systems Using Double Loop Recurrent Neural Network Structure." *IEEE Transactions on Neural Networks and Learning Systems* 29 (4): 1275–1286. <https://doi.org/10.1109/TNNLS.2017.2672998>.
- Fine, T. L. 2006. *Feedforward Neural Network Methodology*. Springer New York. <https://books.google.com/books?id=s-PIBwAAQBAJ>.
- Gires, A., I. Tchiguirinskaia, D. Schertzer, A. Schellart, A. Berne, and S. Lovejoy. 2014. "Influence of Small Scale Rainfall Variability on Standard Comparison Tools Between Radar and Rain Gauge Data." *Atmospheric Research* 138:125–138. <https://doi.org/10.1016/j.atmosres.2013.11.008>.
- Hardin, J. W., and J. M. Hilbe. 2007. *Generalized Linear Models and Extensions Second*. Stata Press.
- Jiang, B., B. Wang, and B. Luo. 2023. "Sparse Norm Regularized Attribute Selection for Graph Neural Networks." *Pattern Recognition* 137:109265. <https://doi.org/10.1016/j.patcog.2022.109265>.
- Kim, S., and H. Kim. 2016. "A New Metric of Absolute Percentage Error for Intermittent Demand Forecasts." *International Journal of Forecasting* 32 (3): 669–679. <https://doi.org/10.1016/j.ijforecast.2015.12.003>.
- Koprinska, I., D. Wu, and Z. Wang. 2018. "Convolutional Neural Networks for Energy Time Series Forecasting." 2018 International Joint Conference on Neural Networks (IJCNN), 1–8. <https://doi.org/10.1109/IJCNN.2018.8489399>.
- Kratzert, F., D. Klotz, C. Brenner, K. Schulz, and M. Herrnegger. 2018. "Rainfall–Runoff Modelling Using Long Short-Term Memory (LSTM) Networks." *Hydrology and Earth System Sciences* 22 (11): 6005–6022. <https://doi.org/10.5194/hess-22-6005-2018>.
- Lin, H., A. Gharehbaghi, Q. Zhang, S. S. Band, H. T. Pai, K.-W. Chau, and A. Mosavi. 2022. "Time Series-Based Groundwater Level Forecasting Using Gated Recurrent Unit Deep Neural Networks." *Engineering Applications of Computational Fluid Mechanics* 16 (1): 1655–1672. <https://doi.org/10.1080/19942060.2022.2104928>.
- Lin, J., X. He, S. Lu, D. Liu, and P. He. 2021. "Investigating the Influence of Three-Dimensional Building Configuration on Urban Pluvial Flooding Using Random Forest Algorithm." *Environmental Research* 196:110438. <https://doi.org/10.1016/J.ENVRES.2020.110438>.
- Mosavi, A., P. Ozturk, and K. Chau. 2018. "Flood Prediction Using Machine Learning Models: Literature Review." *Water* 10 (11): 1536. <https://doi.org/10.3390/w10111536>.
- Muthusamy, M., A. Schellart, S. Tait, and G. B. M. Heuvelink. 2017. "Geostatistical Upscaling of Rain Gauge Data to Support Uncertainty Analysis of Lumped Urban Hydrological Models." *Hydrology and Earth System Sciences* 21 (2): 1077–1091. <https://doi.org/10.5194/hess-21-1077-2017>.
- National Centers for Environmental Information. 2023. *Global Historical Climatology Network Daily (GHCNd)*. <https://www.ncei.noaa.gov/products/land-based-station/global-historical-climatology-network-daily>.
- NWS. 2022. *Flash Flooding Definition*. <https://www.weather.gov/phi/FlashFloodingDefinition>.
- Oliveira Santos, V., P. A. Costa Rocha, J. Scott, J. V. G. Thé, and B. Gharabaghi. 2023. "A New Graph-Based Deep Learning Model to Predict Flooding with Validation on a Case Study on the Humber River." *Water (Switzerland)* 15 (10): 1827. <https://doi.org/10.3390/w15101827>.
- Ouma, Y. O., and R. Tateishi. 2014. "Urban Flood Vulnerability and Risk Mapping Using Integrated Multi-Parametric AHP and GIS: Methodological Overview and Case Study Assessment." *Water* 6 (6): 1515–1545. <https://doi.org/10.3390/w6061515>.
- Piao, Y., S. Lee, D. Lee, and S. Kim. 2022. *Sparse Structure Learning via Graph Neural Networks for Inductive Document Classification*. <https://github.com/qkrdmsghk/TextSSL>.
- Plumer, B. 2021. "Flooding from Ida Kills Dozens of People in Four States." *The New York Times*. <https://www.nytimes.com/live/2021/09/02/nyregion/nyc-storm>.
- PyTorch. 2023a. *Conv1d*. <https://pytorch.org/docs/stable/generated/torch.nn.Conv1d.html>.
- PyTorch. 2023b. *LSTM*. <https://pytorch.org/docs/stable/generated/torch.nn.LSTM.html>.
- PyTorch. 2023c. *Module*. <https://pytorch.org/docs/stable/generated/torch.nn.Module.html>.
- PyTorch Geometric. 2023. *Torch_Gemoteric.Nn*. <https://pytorch-geometric.readthedocs.io/en/latest/modules/nn.html>.
- Qin, H., Peng, Z. Xi Li, and G. Fu. 2013. "The Effects of Low Impact Development on Urban Flooding Under Different Rainfall Characteristics." *Journal of Environmental Management* 129:577–585. <https://doi.org/10.1016/J.JENVMAN.2013.08.026>.
- Rangapuram, S. S., M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski. 2018. In *Advances in Neural Information Processing Systems*, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman and N. Cesa-Bianchi, 31.
- Rathore, N., P. Rathore, A. Basak, S. H. Nistala, and V. Runkana. 2021. "Multi Scale Graph Wavenet for Wind Speed Forecasting." 2021 IEEE International Conference on Big Data (Big Data), 4047–4053. <https://doi.org/10.1109/BigData52589.2021.9671624>.
- Sadler, J. M., J. L. Goodall, M. M. Morsy, and K. Spencer. 2018. "Modeling Urban Coastal Flood Severity from Crowd-Sourced Flood Reports Using Poisson Regression and Random Forest." *Journal of Hydrology* 559:43–55. <https://doi.org/10.1016/j.jhydrol.2018.01.044>.
- Scarselli, F., M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. 2009. "The Graph Neural Network Model." *IEEE Transactions on Neural Networks* 20 (1): 61–80. <https://doi.org/10.1109/TNN.2008.2005605>.
- Schmidt, R. M. 2019. *Recurrent Neural Networks (RNNs): A Gentle Introduction and Overview*. <http://arxiv.org/abs/1912.05911>.
- Schmitt, T. G., M. Thomas, and N. Ettrich. 2004. "Analysis and Modeling of Flooding in Urban Drainage Systems." *Journal of Hydrology* 299 (3–4): 300–311. [https://doi.org/10.1016/S0022-1694\(04\)00374-9](https://doi.org/10.1016/S0022-1694(04)00374-9).
- Scikit-learn. 2023. *Sklearn.Cluster.SpectralClustering*. <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html>.
- Semadeni-Davies, A., and L. Bengtsson. 1998. "Sensibilité de la Fonte des Neiges au Rayonnement en Milieu Urbain."

- Hydrological Sciences Journal* 43 (1): 67–89. <https://doi.org/10.1080/02626669809492103>.
- Setiono, R. 2001. "Feedforward Neural Network Construction Using Cross Validation." *Neural Computation* 13 (12): 2865–2877. <https://doi.org/10.1162/089976601317098565>.
- Sharif, H. O., D. Yates, R. Roberts, and C. Mueller. 2006. "The Use of an Automated Nowcasting System to Forecast Flash Floods in an Urban Watershed." *Journal of Hydrometeorology* 7 (1): 190–202. <https://doi.org/10.1175/JHM482.1>.
- Sun, A. Y., P. Jiang, M. K. Mudunuru, and X. Chen. 2021. "Explore Spatio-Temporal Learning of Large Sample Hydrology Using Graph Neural Networks." *Water Resources Research* 57 (12): 57(12). <https://doi.org/10.1029/2021WR030394>.
- Sun, M., Z. Song, X. Jiang, J. Pan, and Y. Pang. 2017. "Learning Pooling for Convolutional Neural Network." *Neurocomputing* 224:96–104. <https://doi.org/10.1016/j.neucom.2016.10.049>.
- Thorndahl, S., T. Einfalt, P. Willems, J. Ellerbaek Nielsen, M. C. Ten Veldhuis, K. Arnbjerg-Nielsen, M. R. Rasmussen, and P. Molnar. 2017. "Weather Radar Rainfall Data in Urban Hydrology." *Hydrology and Earth System Sciences* 21 (3): 1359–1380. <https://doi.org/10.5194/hess-21-1359-2017>.
- Tsantekidis, A., N. Passalis, and A. Tefas. 2022. "Chapter 5 - Recurrent Neural Networks." In *Deep Learning for Robot Perception and Cognition*, edited by A. Iosifidis and A. Tefas, pp. 101–115. Academic Press. <https://doi.org/10.1016/B978-0-32-385787-1.00010-5>.
- U.S. Census Bureau QuickFacts: New York city, New York. n.d. "Bronx County (Bronx Borough), New York; Kings County (Brooklyn Borough), New York; New York County (Manhattan Borough), New York; Queens County (Queens Borough), New York; Richmond County (Staten Isl)." <https://www.census.gov/quickfacts/fact/table/newyorkcitynewyork,bronxcountybronxboroughnewyork,kingscountybrooklynboroughnewyork,newyorkcountymanhattenboroughnewyork,queenscountyqueensboroughnewyork,richmondcountystatenislandboroughnewyork/PST045219>.
- Valeo, C., and C. L. I. Ho. 2004. "Modelling Urban Snowmelt Runoff." *Journal of Hydrology* 299 (3–4): 237–251. [https://doi.org/10.1016/S0022-1694\(04\)00368-3](https://doi.org/10.1016/S0022-1694(04)00368-3).
- von Luxburg, U. 2007. "A Tutorial on Spectral Clustering." *Statistics and Computing* 17 (4): 395–416. <https://doi.org/10.1007/s11222-007-9033-z>.
- von Luxburg, U., M. Belkin, and O. Bousquet. 2008. "Consistency of Spectral Clustering." *Annals of Statistics* 36 (2): 555–586. <http://www.jstor.org/stable/25464638>.
- Wang, K., K. Li, L. Zhou, Y. Hu, Z. Cheng, J. Liu, and C. Chen. 2019. "Multiple Convolutional Neural Networks for Multivariate Time Series Prediction." *Neurocomputing* 360:107–119. <https://doi.org/10.1016/j.neucom.2019.05.023>.
- Wang, Y., F. Bu, X. Lv, Z. Hou, L. Bu, F. Meng, and Z. Wang. 2023. "Attention-Based Message Passing and Dynamic Graph Convolution for Spatiotemporal Data Imputation." *Scientific Reports* 13 (1): 6887. <https://doi.org/10.1038/s41598-023-34077-z>.
- Wang, Z., H. Lyu, and C. Zhang. 2023. "Pluvial Flood Susceptibility Mapping for Data-Scarce Urban Areas Using Graph Attention Network and Basic Flood Conditioning Factors." *Geocarto International* 38 (1). <https://doi.org/10.1080/10106049.2023.2275692>.
- Woodruff, J. D., J. L. Irish, and S. J. Camargo. 2013. "Coastal Flooding by Tropical Cyclones and Sea-Level Rise." *Nature* 504 (7478): 44–52. <https://doi.org/10.1038/nature12855>.

Appendices

Appendix A

Table A1. 'Training hyperparameters and preprocessing'.

Model	Input Normalization	Architecture & Hyperparameters	Regularization	Optimizer & LR	Loss	Epochs	Batch Size	Random Seed
FNN	Z-score (StandardScaler on predictors)	1 hidden layer, 16 units → ReLU → 1 output	None	Adam, lr = 0.007	L1Loss	1000	entire series	fixed
CNN	Z-score (Standard-Scaler on predictors)	Conv1d 16 → Conv1d 32 (kernel = 1) → global mean → FC→1	None	Adam, lr = 0.001	L1Loss	1000	entire series	fixed
RNN	Z-score (Standard-Scaler on predictors)	LSTM(hidden = 16) → FC→1	None	Adam, lr = 0.007	L1Loss	1000	entire series	fixed
GCN	None on dynamic inputs	2× GCNConv(hidden = 64 → out = 1)	None	Adam, lr = 0.001	MSELoss	100	full graph	fixed
GWN	None on dynamic inputs; static attributes z-scored via StandardScaler	WaveNet blocks	Dropout 0.3; L2 weight_decay = 1e-4	Adam via Optim, lr = 0.01	MSELoss	50	30	fixed

Appendix B

Table B1.

Zones	0	1	2	3	4	5	Median	Mean
GCR	0.29	0.56	0.28	0.68	0.52	0.42	0.47	0.46