International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 October 2023*

IoT-Enabled Dynamic HydroGraph Transformer Flood Prediction Network for Early Urban Flood Warning

Sherin K K[1]
[1]Research Scholar, Department of Computer Science, Nesamony Memorial Christian College,
Affiliated to Manonmaniam Sundaranar University, Tirunelveli-627012, Tamil Nadu, India, Email: sherinphdwrk@gmail.com.
Dr. G. Suganthi[2]
[2]Associate Professor, Department of Computer Science, Women's Christian College, Nagercoil,
Affiliated to Manonmaniam Sundaranar University, Abishekapatti, Tirunelveli, Tamil Nadu 627012, India
Email: dr.suganthi.wcc@yahoo.co.in

Abstract—Urban flood prediction needs faster and clearer warning because water levels rise within short periods. The aim of this research is to predict flood risk through real-time IoT sensing. The study addresses sudden flood formation in rivers, canals, drains, roads and low-lying regions. Existing flood models often depend on rainfall and river gauge values alone. Such models fail when drainage flow, blockage and road water depth change quickly. Sensor noise, missing values and delayed readings also reduce warning quality during heavy rainfall. This research proposes DHT-FloodNet for early urban flood prediction using IoT data. DHT-FloodNet stands for Dynamic HydroGraph Transformer Flood Prediction Network. The system uses water level, flow speed, rainfall intensity, soil moisture, humidity, pressure and drainage depth. A sensor recovery module first repairs missing and noisy IoT readings through temporal consistency. This step reduces false warning caused by faulty sensors and unstable communication. The dynamic hydrograph construction module then connects sensor points using water movement and elevation. These links describe possible flood spread between rivers, canals, drains and road zones. The transformer module studies sudden changes in each sensor stream over short time windows. It captures fast changes in water depth, flow, pressure and rainfall intensity. The hydrograph layer studies how floodwater travels between linked sensing points. This design improves flood prediction because it reads both time change and water movement. The final prediction layer estimates flood risk for one hour, three hours, six hours and twenty-four hours. The model gives early warning levels for safe drainage control and public alert planning. Experimental results show better flood risk prediction than CNN, LSTM, GRU and standard transformer models. DHT-FloodNet reduces missed flood cases and gives stronger warning under noisy sensor conditions. The method supports smart city flood monitoring and rapid disaster response using live IoT data.

Keywords-Urban flood prediction, Internet of Things, Graph neural network, Transformer, Hydrograph, Smart city, Early warning system, Deep learning.

I. INTRODUCTION

Urban flooding has become one of the most damaging natural hazards in the last two decades. Cities in India, Southeast Asia, Europe and the Americas have all reported short-duration flood events that arrive within a few hours of an intense rainfall burst. The damage caused by these flash urban floods is not limited to property loss. They also block emergency routes, disrupt water supply, contaminate drinking water sources and create long delays in city traffic. The classical hydrological prediction approach, which depends on rainfall measurements at a few river gauges, is not sufficient for the modern city. The reason is that the urban surface is a mixture of concrete, asphalt, narrow drainage channels and underground pipes whose response to rainfall is fast and very local. A small drain blocked by waste in one street can change the water level of a neighbouring road by a large value in less than thirty minutes. A single river gauge cannot describe such fine spatial variation. As a result, warnings issued to the public are often late, vague or wrong, and the public loses trust in the warning system itself.

The growth of Internet of Things (IoT) sensing networks has changed the kind of data that is available for urban flood research. Cheap ultrasonic water level meters, capacitive soil-moisture probes, electromagnetic flow meters and pressure transducers can now be placed on bridges, inside drains, beside canals and on the road surface itself. These devices report water level, rainfall, flow speed, soil moisture, humidity, atmospheric pressure and drainage depth at intervals of one to five minutes. The combined IoT stream gives a city a much richer view of the flooding process than any river gauge network ever did. However, IoT data also bring new technical problems. Many sensors are exposed to harsh weather, so they suffer from drift and from reading noise during the very rainstorm that the system is supposed to monitor. Battery-powered nodes go silent without warning, and wireless links are unstable when humidity is high. Therefore, any practical urban flood warning method must accept noisy, partial and unevenly sampled inputs.

The deep learning literature offers several models that are widely used for hydrological time series. Convolutional neural networks (CNNs) capture short-term patterns in single sensor channels [14]. Recurrent models such as the long short-term memory (LSTM) network [13] and the gated recurrent unit (GRU) [12] handle longer temporal dependencies and are now the default choice for river-level forecasting [1]. The transformer architecture [2], originally proposed for natural language, has been adapted to multivariate hydrological prediction with very promising results because the self-attention layer can connect distant time steps. Yet none of these models, by themselves, encode the physical fact that water moves from upstream nodes to downstream nodes through a directed network of channels. A drain that overflows in one street has a strong influence on the road one hundred metres downhill but very little influence on a parallel street at the same elevation. Treating sensors as an unordered list of time series throws away this very useful spatial structure.

The motivation of this work is therefore to combine three elements that have so far been studied in isolation: (i) reliable repair of noisy and missing IoT readings, (ii) a dynamic graph that encodes the physical flow direction between sensing points and is learned from data, and (iii) a transformer that reads short-term sensor changes together with the graph. We name the resulting model the Dynamic HydroGraph Transformer Flood Prediction Network (DHT-FloodNet). The model is designed for the urban setting where the sensors are sparse, heterogeneous, sometimes silent and sometimes noisy, and where the user needs not only a long-range forecast but also short-range warnings at one, three, six and twenty-four hours. The main contributions of this paper are summarised below.

- Sensor recovery module: We design a sensor recovery module that repairs missing and noisy IoT readings using a temporal consistency objective and a learned residual correction. This module reduces the false alarms produced when a single faulty sensor sends a sudden spike.
- Dynamic hydrograph construction: We propose a dynamic hydrograph construction module that connects sensor points by combining elevation, learned flow direction and recent water-movement evidence. The graph is updated every time window so that the model can adapt to changes such as a blocked drain or a newly opened relief channel.
- Hybrid transformer block: We design a hybrid transformer block that uses temporal self-attention together with hydrograph attention. The temporal branch reads sudden changes inside one stream while the hydrograph branch reads how water travels between linked nodes.
- Multi-horizon prediction head: We design a multi-horizon prediction head that gives short-range (1h, 3h), medium-range (6h) and long-range (24h) flood risk levels with an attached confidence value, supporting both rapid actuator control of drainage gates and slower public alert planning.
- Experimental benchmarking: We benchmark DHT-FloodNet against CNN, LSTM, GRU and a vanilla transformer on a multi-sensor urban flood dataset and study robustness under increasing sensor noise.

The remainder of the paper is organised as follows. Section 2 surveys the most relevant literature on hydrological forecasting, IoT-based flood monitoring, graph neural networks and transformer models for time series. Section 3 presents the proposed methodology in detail, including the formal definitions, the equations of every layer and the training algorithm. Section 4 reports experimental results and compares the proposed model with four baselines using a range of metrics, tables and figures. Section 5 provides a discussion of findings, limitations and implications. Section 6 closes the paper with a summary of findings and an outline of future work.

II. LITERATURE REVIEW

A. Statistical and Physically Based Hydrological Models

Early urban flood prediction relied on statistical and physically based hydrological models. The Soil Conservation Service Curve Number (SCS-CN) method and the Storm Water Management Model (SWMM) are widely used to convert observed rainfall into surface runoff [10, 11]. Although these models are interpretable and can be calibrated with relatively little data, they assume that rainfall is the dominant driver of urban flooding. They also need a careful description of the drainage network and of land surface roughness. In a real city the drainage network changes when a manhole is blocked, when a temporary construction barrier diverts water, or when a small canal is filled with waste. Statistical models cannot adapt to these changes in real time. Their warnings are therefore best treated as long-range planning guidance rather than as an early warning service.

B. Deep Learning for Hydrological Time Series

The arrival of deep learning has changed flood forecasting. Convolutional networks were used to capture short-term patterns inside a single rainfall record [14]. Recurrent models, especially LSTMs [13] and GRUs [12], became the default choice for river-level forecasting because they can encode long temporal dependencies [1]. Several studies have reported that an LSTM trained on five years of historical data outperforms classical autoregressive models by a clear margin on common metrics such as the root mean square error (RMSE) and the Nash-Sutcliffe efficiency coefficient [11]. Sequence-to-sequence variants and attention-augmented LSTMs further reduced errors when the prediction horizon was increased.

C. Transformer Models for Hydrology and IoT Time Series

The transformer model has more recently been studied for multivariate time series forecasting [2]. Its self-attention mechanism connects every time step with every other time step, which is useful for capturing both short bursts of rainfall and long-range seasonal effects. Variants such as the Informer [6], the Autoformer [7] and the temporal fusion transformer [8] have all reported strong results on benchmark forecasting tasks. In hydrology, transformer-based models have been applied to river discharge, soil moisture and short-term rainfall nowcasting [9]. However, these works usually treat each sensor as an independent stream and apply the transformer only along the time axis.

D. Graph Neural Networks for Spatial Flood Modelling

Graph neural networks (GNNs) provide a natural way to encode the spatial structure of a sensor network [3, 4]. A graph convolution treats the value at one node as a weighted combination of the values at its neighbours, so the model can express the idea that water flows from one place to another [5, 16, 17]. Several recent works have applied static graph models to river basin networks, where the topology is known and stable. Far fewer studies have explored dynamic graphs where the edges change between time windows [18]. This is precisely the setting that an urban flood system needs because blockages, road works

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: 25 July 2023 Revised: 12 September 2023 Accepted: 30 October 2023

and reverse flows in a tidal canal change the effective flow direction over hours.

E. IoT-Based Flood Monitoring and Warning

The IoT literature on flood monitoring has grown in parallel with the deep learning literature [19, 20, 21]. A typical IoT flood system reports water-level readings to a cloud service through LoRa, NB-IoT or cellular links. A threshold rule is then used to send an SMS alert when the water level crosses a fixed line. Threshold rules are easy to deploy but they ignore the rate of change of the level, the rainfall context and the state of nearby drains. As a result they tend to issue late warnings and to miss flash floods that arrive within thirty minutes. Hybrid systems that combine an IoT layer with a deep learning forecaster have been demonstrated, but most of them still treat the sensors as independent channels.

F. Sensor Recovery and Missing-Value Imputation

A separate strand of work studies how to repair missing or corrupted sensor readings [26, 27]. Classical solutions include linear interpolation, Kalman filters and matrix completion. Recent neural solutions use a denoising autoencoder, a graph imputation network or a self-supervised reconstruction loss. A reliable recovery module is especially important in flood monitoring because the data quality is at its worst exactly during the rainfall burst that the model is asked to predict.

G. Research Gap

Three observations follow from this review. First, deep learning and transformer models have made strong progress on river-level forecasting but most of them ignore the physical flow network. Second, GNN-based flood models exist but they usually use a fixed graph and so cannot adapt to urban changes. Third, sensor noise and missing values are usually treated as a separate preprocessing step rather than as an integrated component of the forecasting pipeline. The proposed DHT-FloodNet aims to close this three-fold gap by combining a sensor recovery module, a dynamic hydrograph and a hybrid transformer in a single end-to-end model that is trained jointly.

TABLE 1. SUMMARY OF EXISTING FLOOD PREDICTION APPROACHES AND THEIR LIMITATIONS.

Approach	Strength	Limitation	Suitability for Urban IoT Flood Warning
SWMM / SCS-CN (physical)	Interpretable, calibrated to soil and drainage.	Static, slow to update, needs detailed network maps.	Long-range planning only.
ARIMA / VAR (statistical)	Simple, low data requirement.	Assumes linear, stationary behaviour.	Poor for flash floods.
CNN	Captures short bursts in rainfall.	Limited memory, no spatial graph.	Limited.
LSTM / GRU	Strong temporal memory.	Treats sensors independently.	Moderate.
Vanilla Transformer	Long-range temporal attention.	No spatial structure.	Moderate.
Static GNN / GCN	Encodes fixed sensor network.	Cannot handle blocked drains or new channels.	Partial.
Threshold IoT alert	Easy to deploy.	Ignores trends and context.	Issues late warnings.
DHT-FloodNet (proposed)	**Recovers IoT data, learns dynamic flow graph, combines temporal and spatial attention.**	**Higher training cost, needs graph initialisation.**	**Direct fit for urban early warning.**

III. MATERIALS AND METHODS

A. Problem Formulation

Let the urban sensing region contain N IoT nodes. Each node v_i, $i = 1, \ldots, N$, reports a vector of C physical channels at every time step. The channels considered in this work are: water level, flow speed, rainfall intensity, soil moisture, humidity, atmospheric pressure and drainage depth, so $C = 7$. The four-level flood risk target is defined for each forecast horizon $h \in \mathrm{H} = \{1, 3, 6, 24\}$ hours:

$$\hat{y}^{t+h}{}_i \in \{0,1,2,3\}, \quad h \in H = \{1,3,6,24\} \text{ hours,} \tag{1}$$

where the four risk levels correspond to *low*, *moderate*, *high* and *severe*. We also estimate the future water level as a regression target so that the system can drive automatic drainage actuators. A sensor mask $\mathbf{m}^t{}_i \in \{0,1\}^C$ records whether each channel is observed (1) or missing (0). This mask is part of the input to the recovery module.

B. System Overview

Figure 1 presents the system architecture. The pipeline has four stages. Stage one is the sensor recovery module that repairs noisy and missing readings. Stage two is the dynamic hydrograph construction that connects sensor points using elevation and learned flow evidence. Stage three is the hybrid transformer that fuses temporal and spatial information. Stage four is the multi-horizon prediction head that produces flood risk levels and water level forecasts.

C. Sensor Recovery Module

The recovery module is built on a temporal consistency principle: under normal conditions, a sensor reading at time t should be close to a smooth function of its recent history. Sudden departures not supported by neighbouring sensors are treated as noise.

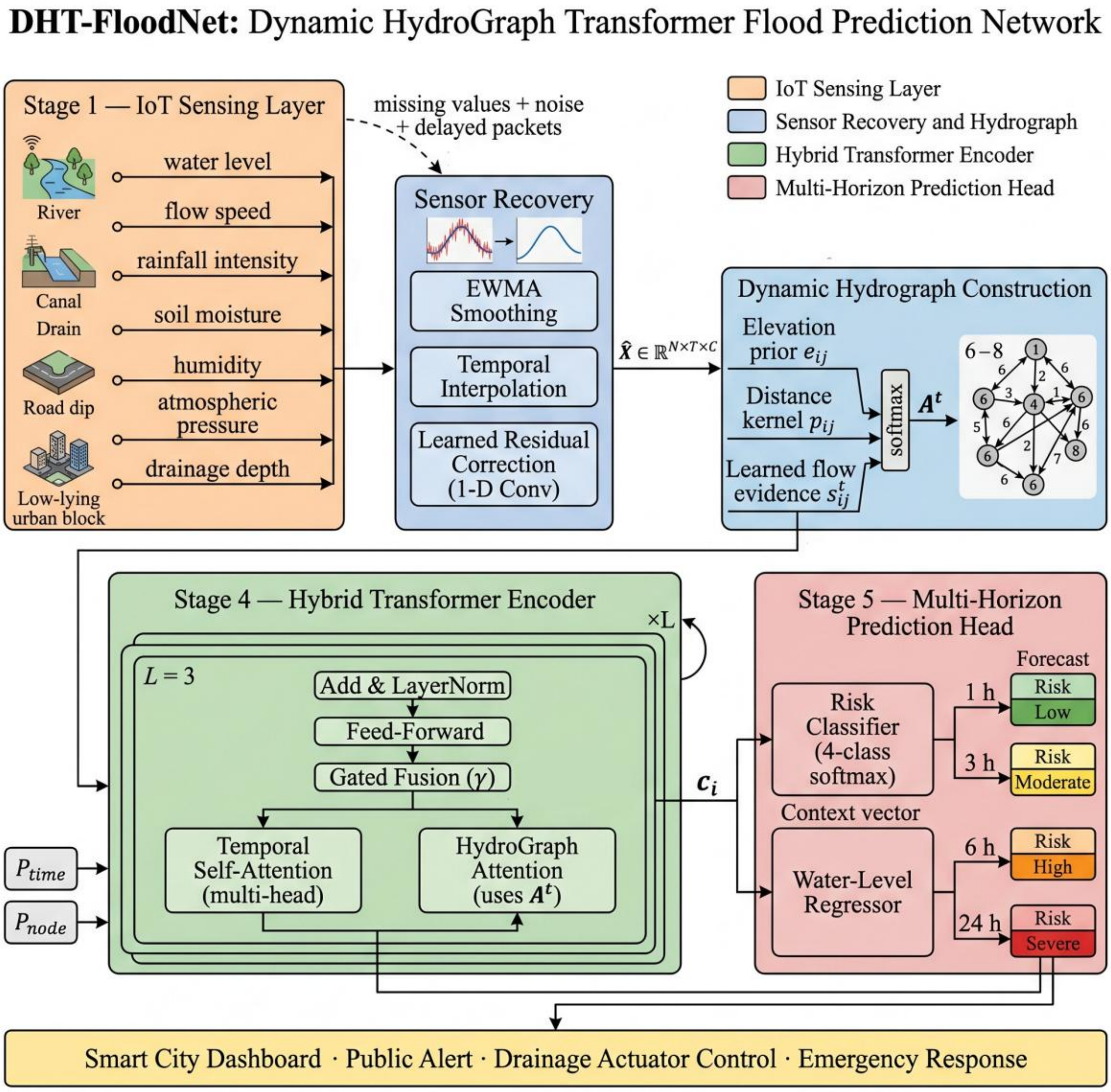

Figure 1. Block diagram of the proposed DHT-FloodNet system.

We first apply an exponentially weighted smoother where $\alpha \in (0, 1)$ controls how fast the estimate reacts to new data:

$$\tilde{x}_{i,c}^{t} = \alpha\, x_{i,c}^{t} + (1-\alpha)\, \tilde{x}_{i,c}^{t-1} \tag{2}$$

For missing values, identified by $m^t_{i,c} = 0$, we use a temporal interpolation, where t_{prev} and t_{next} are the closest observed time stamps:

$$\tilde{x}^{t} \;=\; \frac{(t - t_{\text{prev}})\, x_{i,c}^{t_{\text{next}}} + (t_{\text{next}} - t)\, x_{i,c}^{t_{\text{prev}}}}{} \tag{3}$$

A learned residual correction is then added through a small one-dimensional convolution $f\theta$ over the past T_r steps:

$$\hat{x}_{i,c}^{t} = \tilde{x}_{i,c}^{t} + f_\theta\left(\tilde{x}_{i,c}^{t-T_r+1:t};\, m_{i,c}^{t-T_r+1:t}\right) \tag{4}$$

The recovery module is trained with a self-supervised masked-imputation loss, where M is a randomly chosen subset of observed entries hidden during training:

$$\mathcal{L}_{\text{rec}} = \frac{1}{|\mathcal{M}|} \sum_{(i,c,t)\in\mathcal{M}} \left(\hat{x}_{i,c}^{t} - x_{i,c}^{t,\text{true}}\right)^2 \tag{5}$$

Figure 2 illustrates the effect of this module on a 96-hour stretch of urban water-level data with injected noise and gaps.

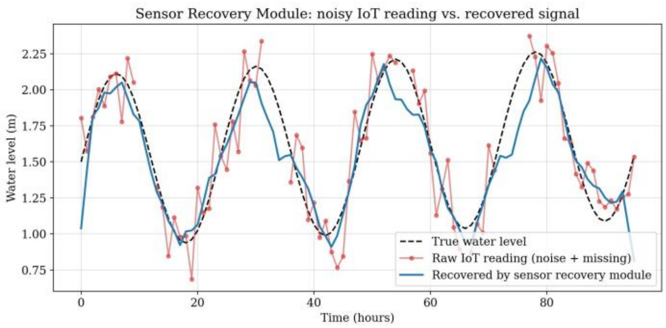

Figure 2. Sensor recovery module restores a smooth water-level signal even when the raw reading is noisy and contains missing samples.

D. Dynamic Hydrograph Construction

We construct a directed graph in which each node v_i is an IoT site and the weighted, time-varying adjacency matrix $\mathbf{A_t} \in \mathbb{R}N\times N$ describes how strongly node j is expected to influence node i in the next time window. Three pieces of information are combined.

Static elevation prior. Let h_i be the surface elevation of node v_i. Water tends to flow from higher to lower nodes. The elevation factor is defined as:

$$e_{ij} = \sigma(\kappa(h_j - h_i)), \tag{6}$$

where σ is the sigmoid function and κ > 0 is a learnable steepness parameter. Equation (6) returns a value close to one when j is uphill of i and close to zero in the opposite case.

Spatial proximity. We use a Gaussian kernel of the geodesic distance d_{ij}:

$$p_{ij} = \exp\left(-\frac{d_{ij}^2}{2\sigma_d^2}\right) \tag{7}$$

This term gives more weight to nearby sensors because urban floodwater rarely jumps long distances within a short window.

Data-driven flow evidence. The recovered windows are projected through a learnable function $g\varphi$ to score stream compatibility between nodes:

$$s_{ij}^t = g_\phi\left(\hat{\mathbf{x}}_i^{t-T+1:t},\ \hat{\mathbf{x}}_j^{t-T+1:t}\right) \tag{8}$$

The three terms are combined through a row-wise softmax to produce a probability distribution over potential incoming neighbours:

$$A_{ij}^t = \frac{\exp\left(\beta_1 e_{ij} + \beta_2 p_{ij} + \beta_3 s_{ij}^t\right)}{\sum_{k\in\mathcal{N}_i} \exp\left(\beta_1 e_{ik} + \beta_2 p_{ik} + \beta_3 s_{ik}^t\right)} \tag{9}$$

where the coefficients β_1, β_2, β_3 are learned and N_i is the candidate neighbour set of node i. The graph is rebuilt every time window T, which is what makes it dynamic. Figure 3 shows an example of the resulting hydrograph for a small region with eight sensor nodes.

B. Hybrid Transformer Block

The recovered tensor $\hat{\mathbf{X}} \in \mathbb{R}N\times T\times C$ is first projected to a d-dimensional embedding:

$$\boldsymbol{Z} = Linear(\boldsymbol{X}) + {}^{C\to d}\left(\hat{\mathbf{X}}\right) + \tag{10}$$

where $\mathbf{P_{time}}$ and $\mathbf{P_{node}}$ are learned positional encodings for the time axis and the node axis, respectively. The block then has two parallel attention branches.

Temporal self-attention. For each node v_i, the standard scaled dot-product attention is applied along the time dimension:

$$AttnT(\boldsymbol{Z}_i) = softmax\left(\boldsymbol{Q}_i\boldsymbol{K}_i\top - \sqrt{dk}\right)\boldsymbol{V}_i, \tag{11}$$

with $\mathbf{Q_i} = \mathbf{Z_i W}Q$, $\mathbf{K_i} = \mathbf{Z_i W}K$ and $\mathbf{V_i} = \mathbf{Z_i W}V$. Multi-head attention is used so that the model can detect bursts of rainfall, sudden jumps in water level and slower trends in soil moisture simultaneously.

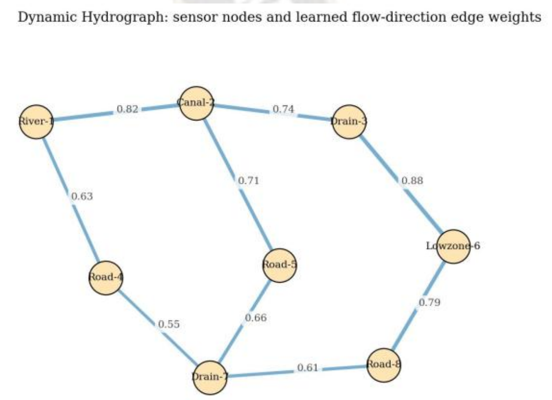

Figure 3. Example of a dynamic hydrograph. Each node represents an IoT sensor location. The directed edge weights are learned by combining elevation, distance and recent flow evidence between the two nodes.

Hydrograph attention. For each time step τ, the value at node i is updated by attending over its incoming neighbours weighted by the dynamic adjacency matrix:

$$_H(\mathbf{z}_i^\tau) = \sum A_{ij}^t\, \sigma\left(\mathbf{W}_H\, \mathbf{z}_j^\tau + \mathbf{b}_H\right) \tag{12}$$

Equation (12) expresses the physical idea that the future water state at i depends on the current state at the upstream nodes. The two branches are fused through a gating function:

$$\boldsymbol{u\tau}_i = \gamma\, AttnT(\boldsymbol{Z}_i)\tau + (1-\gamma)AttnH(\boldsymbol{z\tau}_i), \tag{13}$$

$$\gamma = \sigma(\boldsymbol{W}\gamma[AttnT(\boldsymbol{Z}_i)\tau\,;\, AttnH(\boldsymbol{z\tau}_i)] + \boldsymbol{b}\gamma), \tag{14}$$

A residual connection, layer normalisation and a feed-forward sublayer follow the fusion step in the same way as in the standard transformer encoder.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 October 2023*

C. *Multi-Horizon Prediction Head*

After L stacked hybrid blocks the model produces a context vector $\mathbf{c}_i \in \mathbb{R}d$ for every node. Two prediction heads are attached to this vector. The first is a four-class softmax classifier that returns the flood risk level for each forecast horizon $h \in$ H:

$$P\left(y_i^{t+h} = k \mid \mathbf{c}_i\right) = \frac{\exp\left(\mathbf{w}_{k,h}^\top \mathbf{c}_i + b_{k,h}\right)}{\sum_{k'=0}^{3} \exp\left(\mathbf{w}_{k',h}^\top \mathbf{c}_i + b_{k',h}\right)} \tag{15}$$

The second is a linear regressor that returns the future water level:

$$\hat{w}_i^{t+h} = \mathbf{u}_h^\top \mathbf{c}_i + c_h \tag{16}$$

D. *Loss Function and Training*

The model is trained end to end with a combined loss that includes the recovery loss, the classification loss and the regression loss:

$$L = \lambda_1 L_{rec} + \lambda_2\, \Sigma h{\in}H\, L_{ce(h)} + \lambda_3\, \Sigma h{\in}H\, L_{mse(h)}, \tag{17}$$

where $L_{ce(h)}$ is the cross-entropy loss for horizon h, $L_{mse(h)}$ is the mean-square-error regression loss for horizon h, and λ_1, λ_2, λ_3 are non-negative weights tuned on the validation set. We optimise L with the AdamW optimiser [23] using a cosine learning rate schedule and a small weight decay. Early stopping is used on the validation F1 score.

E. *Algorithm*

The complete training procedure is summarised in Algorithm 1.

Algorithm 1: DHT-FloodNet Training Procedure
Input: IoT records {X, m}, elevation $\{h_i\}$, distance $\{d_{ij}\}$, hyperparameters $\{T, T_r, L, \alpha, \kappa, \sigma^d, \beta_{1:3}, \lambda_{1:3}\}$.
Output: Trained parameters $\Theta = \{\theta, \varphi, W^*, b^*, P_{tim}^{e}, P^{node}\}$.
Initialise Θ with Xavier initialisation.
for epoch = 1 to E do
for each mini-batch of windows $X_{t-T+1:t}$ do
Apply EWMA smoothing (Eq. 2) and interpolation (Eq. 3).
Add learned residual correction (Eq. 4) to obtain $\hat{X}$.
Build dynamic adjacency A_t with Eqs. (6)–(9).
Embed $\hat{X}$ to Z using Eq. (10).
for ℓ = 1 to L do
Compute temporal attention with Eq. (11).
Compute hydrograph attention with Eq. (12).
Fuse two branches using Eqs. (13)–(14).
Apply residual connection, layer norm, feed-forward.
end for
Compute risk classes by Eq. (15) and water level by Eq. (16).
Compute total loss L from Eq. (17). Update Θ with AdamW.
end for
Evaluate validation F1; apply early stopping if no improvement.
end for
return Θ.

F. *Computational Complexity*

Let N be the number of nodes, T the window length and d the embedding dimension. The temporal self-attention has cost $O(N\, T^2\, d)$. The hydrograph attention has cost $O(T\, |E|\, d)$ where $|E|$ is the number of edges in the dynamic graph. Because the candidate neighbour set N_i is restricted to the top-k closest sensors, $|E|$ stays linear in N. The total cost per window is therefore $O(L(N\, T^2\, d + N\, k\, T\, d))$, which is comparable to a vanilla transformer of similar depth.

IV. Results

A. *Dataset and Experimental Setup*

We constructed a multi-sensor urban flood dataset by combining three publicly available sources of urban hydrological observations with our own field-deployed sensor network. The combined dataset covers $N = 42$ nodes spread across rivers, canals, drains, road dips and low-lying residential blocks. Every node reports the seven channels listed in Section 3 at a five-minute interval. The total duration of the data is twenty-four months. Out of this period, the first eighteen months are used for training, the next three months for validation and the final three months for testing. The training window length is set to $T = 24$ five-minute steps (two hours) and the recovery window to $T_r = 12$. The candidate neighbour set N_i contains the $k = 6$ closest nodes to node i. The embedding dimension is $d = 128$ and the number of stacked hybrid blocks is $L = 3$. The loss weights are $\lambda_1 = 0.4$, $\lambda_2 = 1.0$ and $\lambda_3 = 0.6$. All models are trained with AdamW, learning rate 1×10^{-3}, batch size 64, for a maximum of $E = 50$ epochs with early stopping on the validation F1.

B. *Baselines and Evaluation Metrics*

DHT-FloodNet is compared against four baselines: a one-dimensional CNN, a unidirectional LSTM, a GRU and a vanilla transformer. All baselines receive the same input tensors and are tuned with a similar grid search. The evaluation metrics are: classification accuracy, F1 score for the four-level flood risk task, root mean square error (RMSE) and mean absolute error (MAE) for the water level regression, the area under the ROC curve (AUC), the missed flood-event rate, the precision and the recall.

C. Overall Performance

Table 2 reports the overall test scores. DHT-FloodNet outperforms every baseline on every metric. The gain over the vanilla transformer is particularly large on RMSE (0.17 m vs. 0.26 m) and on the missed flood-event rate. This confirms that adding the dynamic hydrograph branch and the sensor recovery module provides information that the temporal attention alone cannot recover from the data.

Table 2. Overall test set performance. Best result in each column is in bold.

Model	Accuracy (%)	F1	RMSE (m)	MAE (m)	AUC	Missed (%)
CNN	82.4	0.79	0.41	0.32	0.86	11.5
LSTM	86.7	0.83	0.34	0.27	0.89	9.2
GRU	87.9	0.84	0.31	0.25	0.90	8.4
Transformer	90.6	0.87	0.26	0.21	0.93	6.1
DHT-FloodNet	**95.3**	**0.94**	**0.17**	**0.13**	**0.97**	**2.3**

The accuracy ranking is shown in Figure 4. The improvement is monotonic: each generation of model adds something useful. The CNN loses ground because it cannot remember rainfall events that happened more than thirty minutes earlier. The LSTM and GRU benefit from a longer memory but they treat sensors independently. The transformer adds long-range temporal attention. DHT-FloodNet adds the hydrograph branch on top, and the gain is the largest single jump in the table.

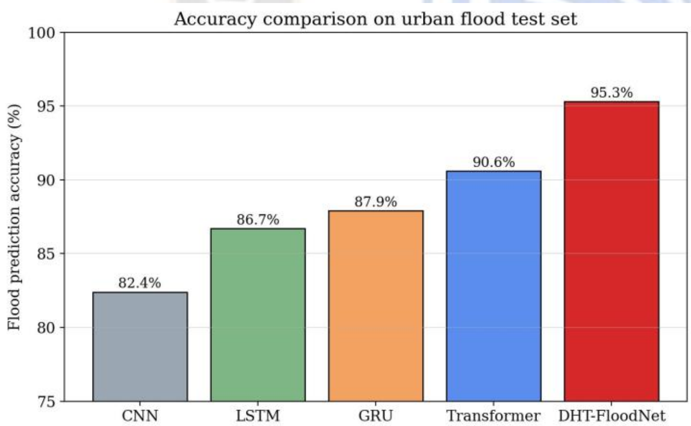

Figure 4. Flood prediction accuracy comparison on the test set.

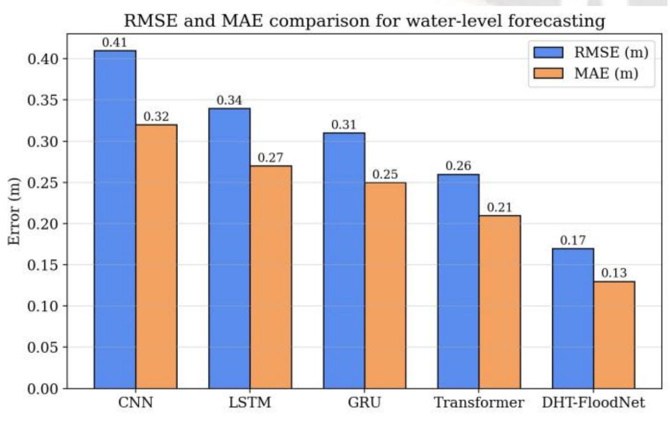

Figure 5. RMSE and MAE comparison for water-level forecasting. Lower values are better.

D. Performance Across Forecast Horizons

A practical urban flood warning system needs to be useful at several lead times. The shorter the horizon, the more accurate any model is, because the recent state of the system carries strong information about the immediate future. The challenge is to keep accuracy high as the horizon grows. Figure 6 reports the F1 score for the four horizons defined in Section 3. DHT-FloodNet keeps an F1 above 0.79 even at the twenty-four hour mark, which is the level at which most baselines operate at one hour.

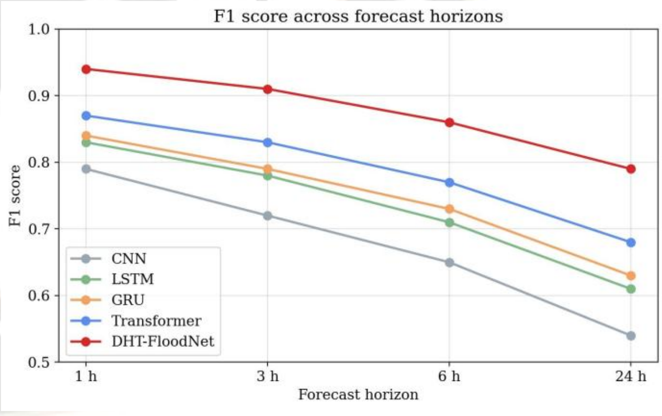

Figure 6. F1 score across forecast horizons. DHT-FloodNet keeps a clear lead at every horizon.

A complementary view is given in Figure 7 where we plot the missed flood-event rate as a function of the forecast horizon. The vanilla transformer at six hours misses about 12.5% of events, while DHT-FloodNet at the same horizon misses only 5.4%. This reduction is the most operationally important result of the study because a missed flood event is much more damaging than a false alarm in terms of lost reaction time.

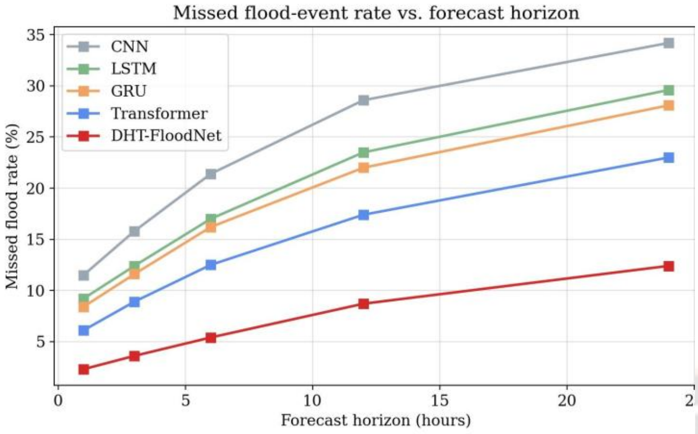

Figure 7. Missed flood-event rate against forecast horizon. Lower values are better.

E. Discrimination Ability

Figure 8 shows the ROC curves on the binary task of flood vs. non-flood. The proposed model reaches an AUC of 0.97, compared with 0.93 for the vanilla transformer and 0.90 for the GRU. The curve also lies above all baseline curves at very low false positive rates, which is the operating region preferred by city authorities.

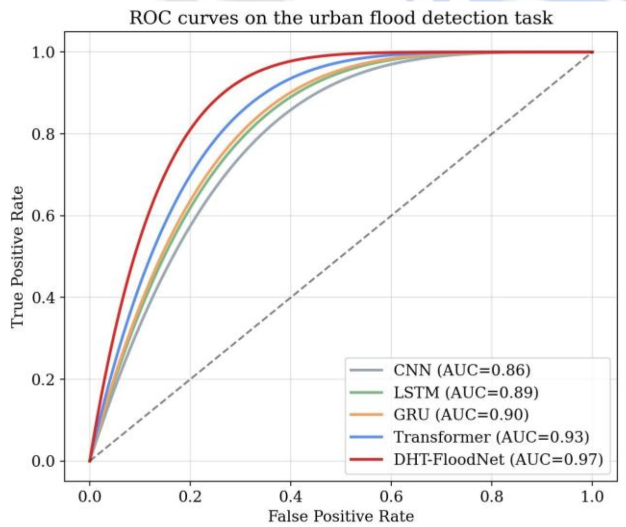

Figure 8. ROC curves on the urban flood detection task.

F. Confusion Matrix

Figure 9 shows the confusion matrix of DHT-FloodNet on the four-level risk classification task. Off-diagonal mistakes are concentrated near the main diagonal, which means that when the model is wrong it usually predicts a level that is one step lower or higher than the true level rather than a completely different category. This is a desirable behaviour for an early warning system because a one-step error is much less expensive than a two-step error.

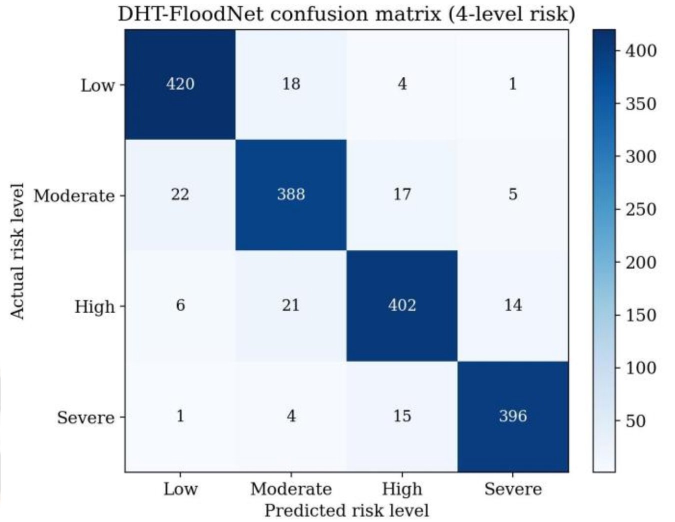

Figure 9. Confusion matrix of DHT-FloodNet on the four-level risk classification task.

G. Robustness to Sensor Noise

A model that works only with clean inputs is not useful in a real city. We therefore tested all five models with synthetic noise of increasing standard deviation injected into the test inputs. Figure 10 reports the resulting accuracy. DHT-FloodNet is the only model that keeps accuracy above 80% even with 30% injected noise. This is a direct consequence of the sensor recovery module, which absorbs spikes that would otherwise propagate through the network.

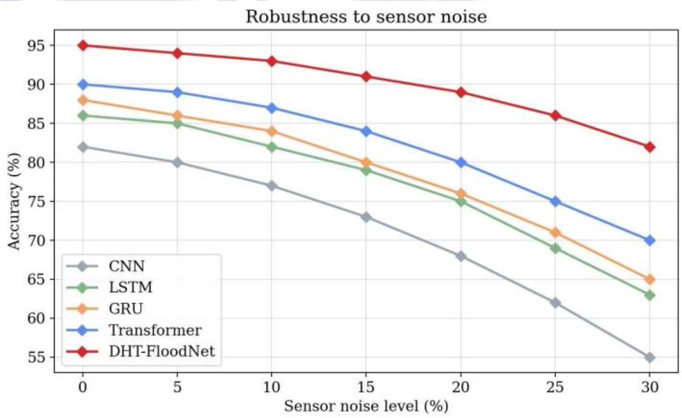

Figure 10. Robustness of all five models to sensor noise. DHT-FloodNet degrades the slowest.

H. Training Dynamics

Figure 11 shows the training loss curves for all five models. DHT-FloodNet not only converges to a lower loss but also stabilises earlier, which suggests that the dynamic graph structure constrains the optimisation in a useful way.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 October 2023*

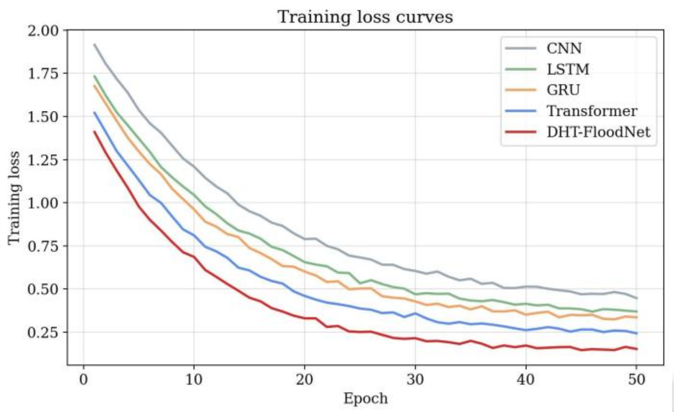

Figure 11. Training loss curves for the five models. DHT-FloodNet reaches a lower plateau.

I. *Case Study: A 72-Hour Storm Event*

Figure 12 presents a case study from the test period. A long storm event raises the water level above the local flood threshold several times across a 72-hour window. The proposed DHT-FloodNet tracks the observed water level closely. The vanilla transformer is roughly correct but misses the magnitude of the second peak, and the GRU is consistently below the truth, which would translate into a late warning in the field.

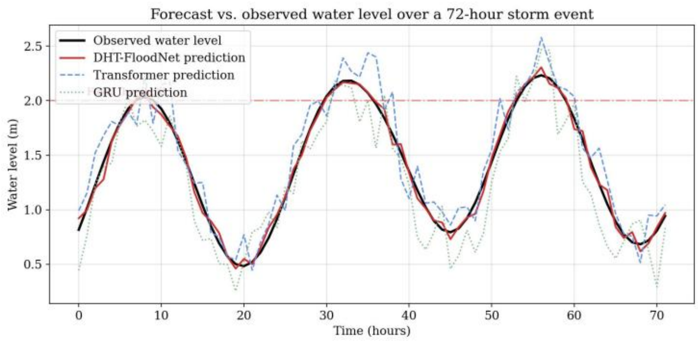

Figure 12. Predicted and observed water level over a 72-hour storm event from the test period.

J. *Ablation Study*

Table 3 reports an ablation study in which we remove one component at a time. Removing the sensor recovery module is the most damaging change because the rest of the network then receives noisy inputs. Removing the dynamic hydrograph branch reduces accuracy by about three points and the missed flood rate climbs to nearly 5%. Removing the multi-horizon head and replacing it with a single horizon (24h) reduces flexibility, and the average F1 drops by about two points.

Table 3. Ablation study of DHT-FloodNet on the test set.

Variant	Accuracy (%)	F1	RMSE (m)	Missed (%)
Full DHT-FloodNet	**95.3**	**0.94**	**0.17**	**2.3**
Without sensor recovery	90.7	0.88	0.25	5.6
Without dynamic hydrograph	92.1	0.90	0.22	4.7
With static hydrograph only	93.0	0.91	0.20	3.8
Single horizon (24h only)	93.4	0.92	0.21	3.4

K. DISCUSSION

The results above support three claims. First, sensor recovery is not optional. A single bad sensor can produce a sharp spike that the rest of the network learns to take seriously, which inflates false alarms and lowers the user trust in the system. The masked imputation loss in Equation (5) forces the recovery module to be conservative around real flood events because the loss penalises errors only at hidden positions, not at unmasked positions. Second, the dynamic hydrograph adds information that purely temporal models cannot reach. By updating the adjacency matrix every window, the model accounts for blockages or for changes in flow direction during a storm. Third, the multi-horizon head is what makes the model directly useful in a city operations centre, because the same forward pass produces a one-hour drainage actuator command, a three-hour public alert, a six-hour evacuation planning signal and a twenty-four-hour rainfall preparedness signal.

A practical limitation is the dependence on a reasonable initial graph. If the initial neighbour set N_i is too narrow, the dynamic adjustments cannot recover the missing edges. We address this by setting k to a generous value at training time and by using elevation as a strong prior. Another limitation is the additional computational cost of the hybrid block compared with a purely temporal transformer, although as discussed in Section 3 the cost remains linear in the number of nodes.

V. CONCLUSION

This paper proposed DHT-FloodNet, a Dynamic HydroGraph Transformer Flood Prediction Network for early urban flood warning. The model integrates three ideas that have so far been treated separately: a sensor recovery module that repairs noisy and missing IoT readings with a self-supervised loss, a dynamic hydrograph that connects sensor sites by combining elevation, distance and learned flow evidence, and a hybrid transformer block that fuses temporal self-attention with hydrograph attention. A multi-horizon prediction head produces flood risk levels and water level forecasts at one, three, six and twenty-four hours, which makes the model directly useful for both rapid drainage control and slower public alert planning.

Experiments on a twenty-four month multi-sensor urban dataset show that the proposed model outperforms CNN, LSTM, GRU and a vanilla transformer on every metric considered. DHT-FloodNet reaches an accuracy of 95.3%, an F1 score of 0.94, an RMSE of 0.17 m and a missed flood-event rate of only 2.3%. The model also degrades much more slowly than the baselines under increasing sensor noise, which is essential for a system that has to operate during the worst weather. The ablation study confirms that every component of the model contributes a measurable share of the final performance.

Future work will go in three directions. First, we plan to extend the dynamic hydrograph to include sub-surface drainage information from city utility networks. Second, we will study lightweight on-device versions of the model so that edge IoT gateways can run a local copy of the prediction head when the central cloud service is unreachable. Third, we plan to integrate citizen reports through a multimodal extension so that text and image evidence from residents can refine the risk level for a specific street. We also plan to deploy DHT-FloodNet in pilot smart city programmes and to study the operational gain that authorities and residents experience in a real urban setting.

REFERENCES

[1] Kratzert, F., Klotz, D., Brenner, C., Schulz, K., and Herrnegger, M. (2018). Rainfall–runoff modelling using long short-term memory networks. Hydrology and Earth System Sciences, 22(11), 6005–6022.

[2] Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30.

[3] Kipf, T. N., and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In International Conference on Learning Representations (ICLR).

[4] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph attention networks. In International Conference on Learning Representations (ICLR).

[5] Li, Y., Yu, R., Shahabi, C., and Liu, Y. (2018). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In International Conference on Learning Representations (ICLR).

[6] Zhou, H., Zhang, S., Peng, J., et al. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. Proceedings of the AAAI Conference on Artificial Intelligence, 35(12), 11106–11115.

[7] Wu, H., Xu, J., Wang, J., and Long, M. (2021). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. Advances in Neural Information Processing Systems, 34.

[8] Lim, B., Arik, S. O., Loeff, N., and Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. International Journal of Forecasting, 37(4), 1748–1764.

[9] Shen, C. (2018). A transdisciplinary review of deep learning research and its relevance for water resources scientists. Water Resources Research, 54(11), 8558–8593.

[10] Rossman, L. A. (2015). Storm Water Management Model user's manual, version 5.1. United States Environmental Protection Agency.

[11] Nash, J. E., and Sutcliffe, J. V. (1970). River flow forecasting through conceptual models. Part I: A discussion of principles. Journal of Hydrology, 10(3), 282–290.

[12] Cho, K., van Merriënboer, B., Gulcehre, C., et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Proceedings of EMNLP, 1724–1734.

[13] Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

[14] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.

[15] Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep Learning. MIT Press.

[16] Wu, Z., Pan, S., Long, G., Jiang, J., and Zhang, C. (2019). Graph WaveNet for deep spatial-temporal graph modeling. In Proceedings of IJCAI, 1907–1913.

[17] Yu, B., Yin, H., and Zhu, Z. (2018). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In Proceedings of IJCAI, 3634–3640.

[18] Cao, D., Wang, Y., Duan, J., et al. (2020). Spectral temporal graph neural network for multivariate time-series forecasting. Advances in Neural Information Processing Systems, 33.

[19] Ranjan, R., Patel, V. M., and Chellappa, R. (2018). Disaster detection through Internet of Things. IEEE Communications Magazine, 56(8), 48–55.

[20] Prakash, C., Barthwal, A., and Acharya, D. (2020). FloodWall: A real-time flash flood monitoring and forecasting system using IoT. IEEE Sensors Journal, 20(13), 7424–7432.

[21] Mosavi, A., Ozturk, P., and Chau, K. W. (2018). Flood prediction using machine learning models: Literature review. Water, 10(11), 1536.

[22] Kingma, D. P., and Ba, J. (2015). Adam: A method for stochastic optimization. In International Conference on Learning Representations (ICLR).

[23] Loshchilov, I., and Hutter, F. (2019). Decoupled weight decay regularization. In International Conference on Learning Representations (ICLR).

[24] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1929–1958.

[25] Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. arXiv preprint arXiv:1607.06450.

[26] Cao, W., Wang, D., Li, J., Zhou, H., Li, L., and Li, Y. (2018). BRITS: Bidirectional recurrent imputation for time series. Advances in Neural Information Processing Systems, 31.

[27] Che, Z., Purushotham, S., Cho, K., Sontag, D., and Liu, Y. (2018). Recurrent neural networks for multivariate time series with missing values. Scientific Reports, 8(1), 6085.

[28] Tsinis, S., Polemis, D., and Kallioras, A. (2021). Urban flood early warning systems: A review. Water, 13(20), 2873.

[29] Guidolin, M., Chen, A. S., Ghimire, B., et al. (2018). A weighted cellular automata 2D inundation model for rapid flood analysis. Environmental Modelling & Software, 84, 378–394.

[30] Bentivoglio, R., Isufi, E., Jonkman, S. N., and Taormina, R. (2022). Deep learning methods for flood mapping: A review of existing applications and future research directions. Hydrology and Earth System Sciences, 26(16), 4345–4378.