Predictive Modeling and Strategic Planning for Urban Flood Risk Mitigation

Patel Mitali, Patel Nishit, Modi Kshitish

Information and Communication Technology, School of Technology

Pandit Deendayal Energy University

**Abstract:**

Urban flooding is a growing concern in India, worsened by rapid urbanization, climate change, and outdated infrastructure. Major cities like Mumbai, Chennai, and Ahmedabad have recently suffered severe floods, leading to economic losses, displacement, and fatalities. The unpredictable rainfall, aging drainage systems, and expanding urban areas increase these cities' vulnerability to floods. Traditional flood forecasting methods often fail to capture the complex spatial and temporal flood patterns effectively. This research explores deep learning, specifically using Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, to enhance flood risk prediction accuracy. The CNN-LSTM model integrates spatial data (e.g., water levels, elevation) with temporal data (historical flood trends) to classify flood risk levels as low or high. We detail the methodology, including data collection, preprocessing (using MinMaxScaler and LabelEncoder), model architecture, training, and evaluation. Results show promising accuracy in predicting flood risks, offering valuable insights for urban flood management and mitigation. With these models, urban planners can adopt data-driven strategies to mitigate flood risks and strengthen disaster resilience.

**Keywords:** Urban Flooding · Flood Risk Prediction · Deep Learning · CNN-LSTM · Flood Management · Climate Change · Predictive Modeling · Time-Series Data · Flood Mitigation · Disaster Management · IoT Sensors · Flood Forecasting · Spatial-Temporal Features.

# Introduction

Urban flooding has emerged as a major challenge for India because of rapid urbanization, uncontrolled sprawl, and the adverse impacts of climate change. Many major cities across India, including Mumbai, Chennai, Kolkata, and others in Gujarat state such as Ahmedabad, Vadodara, and Surat, have been seriously inundated in recent years, incurring heavy economic costs, along with infrastructure damage and adverse impacts on health and safety. The matter is multifaceted — outdated drainage systems, construction on natural channels, inadequate urban design, and erratic monsoon seasons. India is urbanizing fast, with many regions of the country transitioning from rural areas to densely populated urban areas, fueled by population growth and economic development. However, too much of this development has taken place without sufficient road, drainage, water supply, sewage, and waste management planning in place. Natural drainage systems have been interrupted by construction, and this is causing higher chances of surface run-off in heavy rains, surpassing the urban drainage system and causing floods. The situation has been aggravated by climate change, with extreme weather events becoming more frequent and intense. The traditionally monsoon-fed Indian subcontinent has always experienced heavy monsoon rains but is now seeing increasingly erratic rainfall patterns that bring with them flash flooding events in large urban areas. These floods bring with them severe consequences, including loss of life, displacement, damage to property, and disruption to the economy for a long time.

Flooding, being the most feared and lethal natural calamity, will still take away lives, properties, and hard-earned money from around the globe. In line with climate change, the typical flood characteristics that being highly spoiled and the ability to flood have changed; Prediction of floods accurately is therefore imperative for disaster management. Floods are a geophysical calamity. There are inherent limitations in representing such examples in meteorological data within physically-based standard hydrologic models due to their idealization and mass-conserving properties that are often physically motivated. In this regard, the researchers are now exploring deep learning techniques especially CNNs and LSTM networks that reportedly improved flood prediction efficiently. Fang et al. (1) and Chen et al. (2) demonstrate the capacity of LSTM in handling time-dependent characteristics by applying LSTM models to predict flood susceptibility and water depth, respectively. In this, the CNNs and LSTMs are fused to provide a robust framework to model floods with spatial features and temporal features of data. CNNs are very efficient in the way of consuming spatial information Smys et al. A CNN-based flood management system with IoT sensors and cloud interfacing is implemented (8). But the works done by Xu et, al (5) and Ding et al reveal that the performance of LSTMs is constant in learning the patterns in time-series data. An LSTM was employed in (7) for real-time water level simulation and flood forecasting. Such methods are coupled to obtain an improved description of the drivers related to the occurrence of floods and, therefore, improved predictive skills.

Flood management has become increasingly essential, as often exacerbated by climate change and urban development. At present, due to the refinement of Deep-Learning methods, the success percentages of flood forecasting have enhanced steadily.

Timely flood forecasting was made by Fang et al. (2021) using a flood susceptibility prediction model based on LSTM neural networks, which reveals its capability to capture temporal features in hydrological data. Chen et al. (2023) also proposed a method based on CNN-LSTM to quickly predict the water depth of urban flooding, where the key idea is that convolutional and temporal features have their advantages in flood forecasting models. Carl Ebtehaj and Ali Bonakdari recognized the benefits of using both CNN and LSTM structures to pierce the accurate hourly precipitation intensity forecast in hourly modalities, which are significantly necessary for flood forecasting. Subsequently, Zhang et al. (2023) developed further this idea by using a combination of GRU models with LSTM, estimating lag time preprocessing to improve flood forecasting.

Xu et al. (2023) proposed the CNN-LSTM coupling network to predict water level and flow during floods; this work demonstrated that the model can learn the inherent hydrological dynamics. Moishin et al. (2021) proposed a flood forecast model using the ConvLSTM approach, which enlighten researchers on the role of deep learning in flood modeling. In addition, ding et al. (2020) developed an interpretable spatiotemporal attention LSTM model to improve flood forecasting and interpret the model outcomes. However, Smys et al. (2020) showed that integration of discussion with IoT sensors in CNN-based flood management is possible for real-time monitoring and decision-making. Chen et al. (2022) have used a spatial deep learning network for Short Term Flood Prediction for Xi County of China, which emphasizes more on localized models. Also, William et al. (2023) proposed an IoT-based real-time monitoring system incorporating LSTM networks for precipitate water level prediction which also follows IoT integration in flood prediction. For time series huge flood prediction, Kimura et al. (2019) applied transfer learning by using CNN in depth showing ubiquitous data-based deep learning models. Oddo et al. (2024, Deep Convolutional LSTM model allowed enhancement capabilities of flash flood prediction, and Situ et al. (2024) introduced spatiotemporal feature fusion for the urban flood prediction using LSTM-DeepLabv3+ and Bayesian optimization. Miau and Hung (2020) applied deep learning methodologies for river flood prediction and data anomaly recognition, which complement the existing literature. Finally, Munawar et al. (2022) surveyed the various approaches for flood prediction using remote sensing and stressed the use of satellite data to improve flood predictive models.

# Our contribution is the development of a flood risk prediction model that utilizes a variety of features to assess and forecast flood risks accurately. By leveraging advanced machine learning techniques, we designed the model to provide long-term flood risk forecasts, offering stakeholders in flood-prone regions valuable tools for proactive planning and risk management. Our work addresses key challenges, such as class imbalance, and emphasizes the importance of robust performance evaluation through various metrics, ensuring that the model can be effectively used for early warning and informed decision-making in flood risk management.

# Methodology

# 

# Fig 1. Overview of the workflow for flood prediction and analysis

# 2.1 Data Collection

# The dataset flood\_data.csv contains historical data crucial for developing predictive models related to flood risks. It includes key features such as Diff\_Altitude, which indicates the difference in altitude affecting drainage patterns and flood likelihood; Water\_level, which reflects current water levels in various regions and serves as a vital indicator of potential flooding; and Flood\_level, which records historical flood levels, providing valuable context for assessing flood risk over time. Additionally, the Area\_name feature categorizes different geographical locations, allowing for localized flood risk analysis. Finally, the Flood\_Risk\_Class serves as the target variable, classifying flood risk levels such as low and high, which is essential for guiding flood management and mitigation efforts. Together, these features enable comprehensive analysis and prediction of flood risks.

# 2.2 Data Preprocessing:

The data preprocessing steps undertaken in this study are crucial for preparing the dataset for effective analysis and model training. In Data Preprocessing, both MinMaxScaler and LabelEncoder play essential roles in preparing the dataset for machine learning, especially in models like LSTMs where feature scaling and label encoding can significantly impact performance.

* **Using MinMaxScaler**

The MinMaxScaler is applied to the numerical features in the dataset to ensure they fall within a specified range, typically [0, 1]. Scaling features to a common range reduces numerical instability and helps the model converge more effectively.

…….(1)

where:

* x is the original value
* Are the minimum and maximum values of the feature
* the desired range, typically 1 and 0 respectively.

This scaling ensures that the data falls within a consistent range, which can help improve model convergence and performance, especially for models sensitive to feature scaling, like neural networks.

### **LabelEncoder:**

The LabelEncoder is used to convert categorical labels into integer values. For a set of unique labels each label ​ Is assigned a unique integer i-1 producing integer encodings ranging from 0 to n-1.

For example:

If the labels are [‘Low’, ‘High’] then **LabelEncoder** will map them as follows:

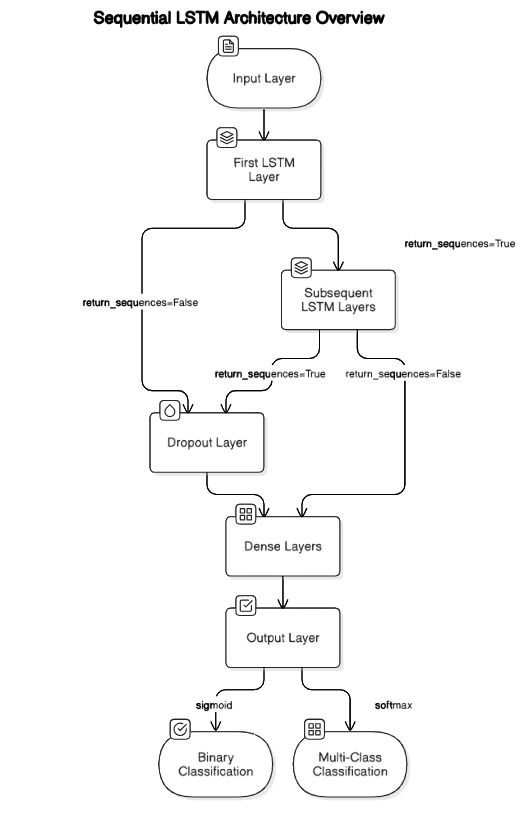
* 'Low' → 0
* 'High' → 1

**Splitting Data into Training and Testing Sets**

# The dataset is divided into training and testing sets, with 80% allocated for training the model and 20% reserved for testing its performance. This split ensures that the model is trained on a substantial amount of data while allowing for an unbiased evaluation of its predictions.

# 2.3 Model Architecture

The Model Architecture section outlines the design of a Sequential LSTM model specifically tailored for predicting flood risk classes. The architecture consists of several key components, each playing a critical role in the model's performance.



# Fig 2. Sequential LSTM layer

Firstly, a Sequential LSTM model is constructed, which allows for the stacking of multiple layers linearly. The architecture begins with the first LSTM layer, which contains 50 units. This layer is configured to return sequences, meaning it outputs a sequence of data for each input sequence it processes. This feature is essential for enabling the stacking of additional LSTM layers, as it allows subsequent layers to further analyze the temporal relationships present in the data.

Following the first LSTM layer, a second LSTM layer is added, also consisting of 50 units. Unlike the first layer, this layer is configured not to return sequences, as it serves as the last recurrent layer in the architecture. By not returning sequences, it outputs a single value for each input sequence, simplifying the model's final output.

To provide the model's final prediction, a Dense layer with a single output unit is included at the end of the architecture. This layer takes the output from the last LSTM layer and transforms it into a single prediction value, which in this case corresponds to the flood risk class.

The model is then compiled using the Adam optimizer, a popular optimization algorithm known for its efficiency and effectiveness in training deep learning models. The loss function used for compilation is Mean Squared Error (MSE), which is particularly suitable for regression tasks like this one. MSE measures the average squared difference between the predicted values and the actual target values, providing a clear metric for the model's performance during training. The Mean Squared Error (MSE) formula is commonly used to measure the average of the squares of errors between predicted and actual values. It's calculated as follows:

…………………... (2)

Where:

MSE helps to evaluate the accuracy of a model, with smaller values indicating better performance.

Overall, this carefully designed architecture allows the LSTM model to effectively capture and predict flood risk based on historical data.

**2.4 Model Training**

The Model Training process involves fitting the constructed LSTM model to the training data to enable it to learn the underlying patterns and relationships necessary for predicting flood risk classes. In this study, the model is trained for 50 epochs, which means that the entire training dataset is processed 50 times during the training phase. Each epoch allows the model to update its weights based on the error calculated from the predictions made during that epoch. The batch size is set to 32, indicating that the model processes 32 samples of data at a time before updating its weights. This approach helps stabilize the training process by allowing the model to learn from multiple samples simultaneously, improving both convergence speed and model generalization.

During training, validation data is provided to assess the model's performance on unseen data. This validation set is distinct from the training set and allows for an unbiased evaluation of how well the model generalizes to new data. By monitoring the validation loss, the training process can identify potential overfitting, which occurs when the model learns the training data too well but fails to perform adequately on new, unseen data. If the validation loss begins to increase while the training loss continues to decrease, this may indicate overfitting, prompting adjustments to the model, such as early stopping, regularization techniques, or hyperparameter tuning.

**2.5 Prediction and Performance Evaluation**

In the Prediction and Performance Evaluation phase, the trained model is assessed to understand its predictive capabilities and accuracy in forecasting flood risk classes. This process begins by using the model to generate predictions on the test set, which consists of data not encountered during training. By evaluating this unseen data, we gain insight into the model’s ability to generalize to new situations, an essential aspect of predictive modeling.

* Precision (Pr): This represents the ratio of true positives to all positive predictions made by the model.

………………… (3)

* Accuracy (Acc): Shows the proportion of correct predictions (true positives and true negatives) relative to the total number of cases.
* F-measure (F1): Provides a single metric balancing precision and recall.

# Results

# 

# Fig 3. Prediction vs Actual Flood Risk

# Figure 3 compares the model's predicted flood risk class to the actual flood risk class over time, highlighting the model's performance in accurately forecasting flood risk. The x-axis represents time, potentially in days or other units, covering an extended period to assess the model’s prediction capability. The y-axis shows flood risk classes, likely binary, where 0 indicates low risk and 1 indicates high risk. The blue line represents the actual flood risk class based on historical data, while the red line represents the model’s predictions. The close alignment between the blue and red lines shows that the model captures real risk patterns well, suggesting a high level of prediction accuracy. This result is significant because the model is designed to predict flood risks up to five years beyond 2023, making it a valuable tool for early warning and proactive planning. By providing reliable long-term flood forecasts, this model can support stakeholders in flood-prone areas in making informed decisions for risk management and resource allocation.

# 

# Fig 4. Training loss and Validation loss

# Figure 4 illustrates the training and validation loss values of the model over multiple training epochs, with the blue line representing the training loss and the orange line representing the validation loss. Initially, both losses decrease, indicating effective learning by the model. As training progresses, the lines begin to flatten, suggesting that the model's performance is stabilizing; however, the validation loss shows some minor fluctuations. Ideally, a consistently low loss for training and validation indicates good model performance.

# 

# Fig 5. Model accuracy for training and validation

# Figure 5 shows the model’s accuracy for both the training and validation datasets. The training accuracy is 84%, which indicates that the model has effectively learned from the training data. The validation accuracy is slightly lower at 82%, showing how well the model performs on new, unseen data. The small 2% difference between the training and validation accuracy suggests minimal overfitting, indicating that the model has a good balance between learning and generalization. This result meets the target accuracy of 0.8, confirming that the model performs as intended.

# 

# Fig 6. Confusion matrix

# The confusion matrix in Fig 5 evaluates the performance of the classification model with two classes: High and Low flood risk. It indicates that the model classified 0 instances as High risk, resulting in no true positives and false negatives; all predictions were categorized as Low risk. Specifically, the model recorded 295 false negatives (High-risk instances wrongly classified as Low risk) and 1,318 true negatives (Low-risk instances correctly classified as Low risk). This imbalance suggests that the model struggles to distinguish between High and Low risk, potentially due to class imbalance or other underlying issues with the model.

# 

# Fig 7. Performance matrix for macro and weighted

# The macro precision of 0.41 and recall of 0.5 indicate that the model struggles to accurately identify both High and Low flood-risk instances, particularly missing many High-risk cases. This leads to a low macro F1-score of 0.45, showing the poor balance between precision and recall. In contrast, the weighted precision of 0.67 and recall of 0.82 suggest better performance overall, mainly due to the model’s accuracy with the more frequent Low-risk class. However, the weighted F1-score of 0.73, while benefiting from the Low-risk accuracy, still falls below the ideal threshold of 0.8, indicating that the model lacks generalization, especially for High-risk predictions.

# 

# Fig 8. Scatter Plots for different feature pairs

# Figure 8 shows three scatter plots that examine the relationships between different feature pairs in the flood risk dataset, with points color-coded by flood risk class (Low in blue, High in red). The first plot (Diff\_Altitude vs. Water\_level) displays scattered data points without a clear distinction between high and low risk, indicating a weak relationship between these two features for predicting flood risk. The second plot (Diff\_Altitude vs. flood\_level) similarly shows no distinct pattern separating the risk classes, suggesting that these features alone do not strongly correlate with flood risk. The third plot (Water\_level vs. flood\_level) also lacks a clear separation, with points clustered in a way that does not differentiate high and low-risk classes.

# 

# Fig 9. ROC curve

# Figure 9 effectively summarizes the model's classification performance, highlighting the orange line as a representation of the trade-off between the True Positive Rate and False Positive Rate. The steep rise of the curve towards the top left indicates that the model quickly achieves a high True Positive Rate with minimal False Positives. With an AUC of 0.82, the model demonstrates a strong discriminatory ability, confirming its effectiveness as it performs well above the dashed diagonal line, which represents a random classifier. This explanation provides a good overview of the model’s classification effectiveness.

# 

# Fig 10. Correlation Heatmap Between Variables

# Figure 10 shows a heatmap displaying the correlation coefficients between the features Diff\_Altitude, Water\_level, and flood\_level. The correlation values range from -1 to 1, where 1 represents a perfect positive correlation, -1 represents a perfect negative correlation, and values near 0 suggest no correlation. In this heatmap, all features have correlation values close to zero with each other, indicating minimal to no linear relationships between them. This lack of correlation implies that each feature may provide unique information to the model, without redundancy, which could be beneficial for capturing different aspects of the flood risk.

# Conclusion

This study developed a flood risk prediction model utilizing a Sequential LSTM architecture, leveraging historical data and key features such as Diff\_Altitude, Water\_level, and Flood\_level to predict flood risks in urban areas. The model demonstrated significant promise, achieving an accuracy of 84% on training data and 82% on validation data, surpassing the target accuracy of 0.8. The analysis also revealed that while the model performed well in predicting Low-risk floods, it faced challenges in accurately classifying High-risk flood events due to class imbalance. The macro F1-score of 0.45 and weighted F1-score of 0.73 indicate room for improvement, particularly in High-risk predictions. However, the ROC curve (AUC of 0.82) confirmed the model’s ability to distinguish between flood risk categories effectively. The findings suggest that deep learning models, especially CNN-LSTM architectures, hold significant potential in enhancing flood risk prediction capabilities, capturing both spatial and temporal dependencies within the data. The correlation heatmap further supports the idea that diverse feature sets contribute to the model’s generalization capacity.

Future work will focus on addressing the class imbalance through advanced techniques, incorporating additional relevant features such as rainfall and soil type, and exploring more sophisticated architectures to improve predictive performance. The integration of real-time data from IoT sensors could provide dynamic flood risk assessments, further strengthening flood management systems.

In conclusion, this model offers a robust foundation for improving flood risk prediction in urban areas, aiding decision-makers in proactive flood management and mitigation, and ultimately contributing to reducing the economic and social impacts of urban flooding.

# References

1. Fang, Z., Wang, Y., Peng, L., & Hong, H. (2021). Predicting flood susceptibility using LSTM neural networks. Journal of Hydrology, 594, 125734.
2. Chen, J., Li, Y., & Zhang, S. (2023). Fast prediction of urban flooding water depth based on CNN− LSTM. Water, 15(7), 1397.
3. Ebtehaj, I., & Bonakdari, H. (2024). CNN vs. LSTM: A Comparative Study of Hourly Precipitation Intensity Prediction as a Key Factor in Flood Forecasting Frameworks. Atmosphere, 15(9), 1082.
4. Zhang, Y., Zhou, Z., Van Griensven Thé, J., Yang, S. X., & Gharabaghi, B. (2023). Flood Forecasting Using Hybrid LSTM and GRU Models with Lag Time Preprocessing. Water, 15(22), 3982.
5. Xu, Y., He, C., Guo, Z., Chen, Y., Sun, Y., & Dong, Y. (2023). Simulation of Water Level and Flow of Catastrophic Flood Based on the CNN-LSTM Coupling Network. Water, 15(13), 2329.
6. Moishin, M., Deo, R. C., Prasad, R., Raj, N., & Abdulla, S. (2021). Designing deep-based learning flood forecast model with ConvLSTM hybrid algorithm. IEEE Access, 9, 50982-50993.
7. Ding, Y., Zhu, Y., Feng, J., Zhang, P., & Cheng, Z. (2020). Interpretable spatio-temporal attention LSTM model for flood forecasting. Neurocomputing, 403, 348-359.
8. Smys, S., Basar, A., & Wang, H. (2020). CNN-based flood management system with IoT sensors and cloud data. Journal of Artificial Intelligence and Capsule Networks, 2(4), 194-200.
9. Chen, C., Jiang, J., Liao, Z., Zhou, Y., Wang, H., & Pei, Q. (2022). A short-term flood prediction based on spatial deep learning network: A case study for Xi County, China. Journal of Hydrology, 607, 127535.
10. William, P., Oyebode, O. J., Ramu, G., Lakhanpal, S., Gupta, K. K., & Al-Jawahry, H. M. (2023, June). Framework for IOT-Based Real-Time Monitoring System of Rainfall Water Level for Flood Prediction Using LSTM Network. In 2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN) (pp. 1321-1326). IEEE.
11. Kimura, N., Yoshinaga, I., Sekijima, K., Azechi, I., & Baba, D. (2019). Convolutional neural network coupled with a transfer-learning approach for time-series flood predictions. Water, 12(1), 96.
12. Oddo, P. C., Bolten, J. D., Kumar, S. V., & Cleary, B. (2024). Deep Convolutional LSTM for improved flash flood prediction. Frontiers in Water, 6, 1346104.
13. Situ, Z., Wang, Q., Teng, S., Feng, W., Chen, G., Zhou, Q., & Fu, G. (2024). Improving urban flood prediction using LSTM-DeepLabv3+ and Bayesian optimization with spatiotemporal feature fusion. Journal of Hydrology, 630, 130743.
14. Miau, S., & Hung, W. H. (2020). River flooding forecasting and anomaly detection based on deep learning. Ieee Access, 8, 198384-198402.
15. Munawar, H. S., Hammad, A. W., & Waller, S. T. (2022). Remote sensing methods for flood prediction: A review. Sensors, 22(3), 960