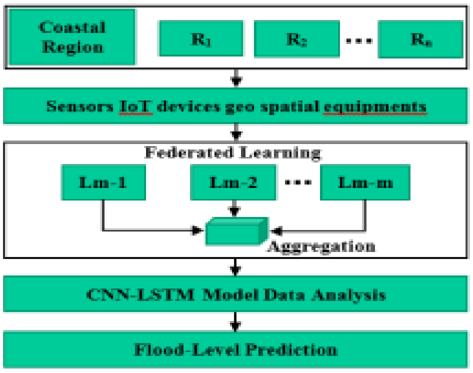
Global NEST Journal, Vol 27, No 6, 07511
Copyright© 2025 Global NEST
Printed in Greece. All rights reserved

A hybrid CNN-LSTM predictive model deployed federated learning model for advanced flood prediction systems to forecast coastal region of smart cities

Venkatesan A[1*] and Thiruvenkadam Krishnan[2]
[1]Research Scholar, Chennai Institute of Technology, Chennai-600069, Tamil Nadu, India
[2]Professor, Electronics Engineering (VLSI Design and Technology), Chennai Institute of Technology, Chennai-600069, Tamil Nadu, India
Received: 31/03/2025, Accepted: 17/04/2025, Available online: 05/05/2025
*to whom all correspondence should be addressed: e-mail: venkates1987@gmail.com
https://doi.org/10.30955/gnj.07511

Graphical abstract



Abstract

Flooding in coastal regions of smart cities poses significant challenges, including infrastructure damage, economic losses, and threats to public safety. Traditional flood prediction models often suffer from data privacy concerns, limited spatial-temporal generalisation, and computational inefficiencies. To address these challenges, this study proposes an advanced Federated Learning (FL) and CNN-LSTM-based predictive framework for flood forecasting in coastal urban regions. The FL paradigm enables decentralised model training across multiple locations while ensuring data privacy. Convolutional Neural Networks (CNNs) extract spatial flood-related features, while Long Short-Term Memory (LSTM) networks capture temporal dependencies in hydrometeorological data. Various sensors, IoT devices and geospatial equipment are deployed to monitor and record flood-related environmental factors in different coastal regions in smart cities. The generated data is analysed by CNN and LSTM models to predict the flood levels based on the flood-influencing factors estimated. The proposed FL-CNN-LSTM model is implemented and experimented with in Python, and the prediction efficiency is verified. It is also compared with the other earlier methods and evaluates performance. It shows that the FL-CNN-LSTM provides more accuracy and promising quality services like dependency reduction in centralised data storage, adaptiveness, and privacy preservation in flood forecasting systems. Most importantly, it provides a proactive natural disaster mitigation model, making it suitable for real-time coastal regions in smart cities.

Keywords: Smart cities, flood prediction, federated learning, CNN-LSTM, coastal resilience, machine learning, hydrometeorological forecasting

1. Introduction

Floods are natural geohazards that occur due to heavy and continuous rain. It is a natural phenomenon that causes a lot of damage to property and gross domestic product T. Ashizawa *et al*. (2022). One of the most destructive natural disasters is floods. With increasing urbanisation, climate changes, and extreme weather conditions, coastal regions are becoming more vulnerable to storms, sea level rise, and floods, K. A. Oladapo *et al*. (2020). Smart cities are equipped with advanced technologies, devices, and data analytics to provide novel pathways for improving flood resilience. In contrast, the earlier methods used for flood predictions were not equipped with data privacy, did not have scalability, and were inadequate in prediction accuracy. Also, they are not efficient and suitable for real-time applications. To solve this kind of challenge, this paper proposed a novel FL-CNN-LSTM model to improve the efficiency of flood prediction with privacy preservation in data transmission among various coastal regions of smart cities. The federated learning model helps to train the data sources available in multiple locations without sharing the original image, preventing privacy issues and problems. The deep learning algorithms CNN and LSTM involved in the proposed model efficiently extract the spatial and temporal features indicating the flood patterns to improve the prediction accuracy. The main objective of this research is to design and implement an efficient learning model for effective flood prediction in the coastal region of smart cities.

Recently, new datasets have been generated and used to detect natural disasters. Most of the datasets are generated from sensors, IoT devices, and geospatial devices that can generate numerical and alphanumerical data from the air, temperature, and other sources related to the environment and climate of a particular region. These datasets are either temporal or spatial, Gupta *et al*. (2019). Deep learning algorithms proved their efficiency in obtaining semantic information on the land cover and behaviours from these datasets. The temporal features are unavailable in the dataset, which cannot provide areas the cloud covers. They were moved to analyse the SAR images since they can provide cloud cover. The new version of satellite images obtained from Sentinel-1 offers a highly useful and large volume of data, including time-frequency and area information. Machine learning algorithms can be applied to these optical and radar images for disaster analysis.

There are various kinds of floods, such as coastal, flash, ponding, and river floods. The monsoon is also classified as the Northwest Monsoon and the Southeast Monsoon based on the season. Various kinds of datasets are used to predict different types of floods. Time series data are used in a wide range of longitudinal research modelling, Zerara (2021). It involves computing similar measures in a periodical interval over more amount of data. Time-series data forecasts the pattern based on the historical data in the analysis since it includes the time domain and other properties. Thus, time series data is used for flood prediction and showed an excellent output globally (W. Wu *et al*. (2020). Various earlier research works, like Shen *et al*. (2024), have used deep learning algorithms, like RNN, ANN, LSTM and other models for time-series analysis. The frequency analysis and rational and empirical methods are unsuitable for large-scale flood prediction but can only be used in small river flow prediction. Thus, an advanced deep learning-based approach is required to overcome these issues. Hence, this paper is motivated to implement a deep learning-based approach for flood prediction. Recent research works have shown excellent output using CNN and LSTM algorithms for forecasting applications using time-series data. In that sense, this paper aims to use the CNN-LSTM model to analyse satellite images for flood prediction obtained from different coastal regions. It used satellite images and sensor data to predict flooding more accurately. This paper contributes the following key points to improve the prediction accuracy and overall efficacy.

- A novel architecture is created to interconnect federated learning and deep learning models for data analytics.
- A federative learning model is created to interconnect the coastal regions of multiple smart cities, sharing their flood data with security and providing a decentralized storage and prediction model.
- Using spatial features, the CNN model is implemented to analyse and predict flood conditions in satellite images.
- The LSTM model uses temporal features to analyse and predict flood conditions in satellite images.
- An efficient hybrid deep learning model, CNN-LSTM, is used to analyse and predict flood conditions using spatiotemporal features.
- Integrating federated and deep learning models increases predictive accuracy and secures data privacy.

Following the introduction, Section 2 discusses the literature review; Section 3 explains the problem statement and the proposed methodology. Section 4 demonstrates the results and discussion with the dataset and experimental setup information. Finally, Section 5 provides the work's conclusion and future scope.

2. Literature survey

In recent years, modern coastal cities have focused on flood prediction, leading to various developments, such as multiple approaches and models developed to improve flood prediction resilience and overall accuracy. The research shares information about some current studies that integrate advanced machine learning methods such as convolutional neural networks (CNNs), Federated Learning (FL), and long short-term memory (LSTM) networks. For example, Rao and Supraja (2024) have proposed an advanced flood prediction model. It uses the hybrid model, combining the CNN and federated learning models for remote-sensing applications. The main advantage of this approach is that it maintains the confidentiality and security of data since each local model is trained locally. To evaluate the model's efficiency in flood prediction, this proposed model was tested on the historical flood data. The result shows that this proposed hybrid model has achieved an accuracy rate of 84% while predicting historical floods. Kabir *et al*. (2020) have proposed another deep learning (DL) model: a deep convolutional neural network (DCNN) model for rapid fluvial flood prediction. The model was calibrated using data from a 2D hydraulic model and provided accurate results regarding the flooded areas, and it can be useful in real-time flood prediction. Giezendanner *et al*. (2023) have proposed a hybrid model that combines a CNN model with the LST model (CNN-LSTM) for Historical Inundation Mapping. To evaluate the efficiency of the proposed hybrid model, it was tested on various historical flood data. The result shows that this proposed hybrid CNN-LSTM model has outperformed the other traditional methods in flood prediction. It is also able to analyse and capture the temporal flood and spatial flood patterns more effectively.

Nasir & Atal (2023) have presented a federated learning model for analysing and predicting floods by analysing the data. It works on a novel flood forecasting framework of a five-day lead time based on the federated learning of locally trained models from different clients to provide a prediction. The model kept the data private and had high accuracy in forecasts. Nahak *et al*. (2024) have presented an investigation on the current challenges that occur while predicting floods and also provide an advanced flood prediction model by using an advanced federated learning model. Flood forecasting is a complex topic that has received attention from scholars in the past years. It integrated many models at the different stations to forecast future flood occurrences, providing alerts with five-day advance notice and help in precaution. A hybrid flood-predicting system using CNN and LSTM was proposed in an IEEE conference held in 2023 to predict floods in Kerala. The model incorporated spatial and temporal properties of the structural object and helped increase the prediction accuracy. One of the summarized articles from Reuters in 2024 was titled have presented an investigation on the various role of AI in improving weather forecasts, flood prediction and other natural disasters and it explained how AI helped the authorities to predict an actual urban flood. Such systems are good at processing large historical databases of various features, as well as at finding the patterns in them.

The article in The Times (2024) has presented an investigation on the concept of the 'Sponge city' idea standard as a natural-based practice that helps control the heavy rainfall in urban areas. Several other elements are also used for reducing the risk of floods, such as gardens, stormwater parks, and pavements with a permeable design to help facilitate the collection of excess rainwater and help absorb and redirect the rainwater, reducing the risk of floods. The article Guardian, published in 2024, conducted a report which investigates how artificial intelligence (AI) should be used to predict various flood impacts that climate change is expected to influence different communities in the United States over the next 75 years. Such images are vivid and generated by artificial intelligence, suggesting local risk and stressing the cunning of preparing for possible floods. Nasir and Atal (2023) have proposed a hybrid model for predicting floods. It combines the federated learning model with the feed-forward neural networks. To evaluate the overall efficiency of the model, it was compared to various traditional models. The result shows that this proposed hybrid model has achieved an accuracy rate of 84% in predicting previous floods and also provides data privacy for the collected data. In 2024, researcher architectures proposed a flood predictive system based on federated learning, a set of locally trained models by multiple clients to predict flood occurrence with 5-day ahead of time. The model ensured the privacy of the data collected and had a high predictive power. Giezendanner *et al*. (2023) have proposed a hybrid model for flood prediction. This model combines a CNN model with the LSTM model (CNN-LSTM) that fuses satellite data. It also improved the analyses and modelling of spatial and temporal variations of floods when compared to conventional methods. The result also shows that this proposed hybrid model outperformed all the traditional models in accuracy. Artificial intelligence was used in weather forecasting and provided accurate and enhanced prediction results of environmental or meteorological events such as urban flooding, as noted by Reuters in 2024. AI systems benefit from analysing large data sets with historical factors and can identify different patterns. All these studies have highlighted the advancement of various models in flood prediction. It comprises multiple techniques, such as DL and federated learning models, to improve overall efficiency and accuracy. At the same time, it also maintained data privacy and security. The application of the hybrid model for predicting flood combines the CNN model with the LSTM model, which was used to analyse and capture the temporal and complex spatial patterns in the historical data. Also, the uptake of nature-based solutions and/or AI-based forecasting has been used to provide more accurate and effective results in flood prediction.

Many of the earlier research works have proved that deep learning models are highly suitable for flood prediction and early flood risk prediction. For example, Karthick *et al*. (2025) have provided an accurate flood risk estimation using a deep learning algorithm and climatological data from Chennai regions. Preprocessing methods were used to increase the quality of the data. The MaxAbsScaler approach was used to eliminate missing values, and the Extended Elman Spiking Neural Network model has been implemented to analyze and predict the risk of the flood level. This model obtained the highest accuracy by eliminating the network problems during the training phase; the parameters were tuned by implementing the Chaotic Artificial Hummingbird Optimizer. Babu. T *et al*. (2024) have proposed a Multiagent Reinforced learning model integrated with deep neural RNN with VANET for predicting early flood by processing Sentinel-2 satellite images. The deep neural RNN model effectively extracts the intricate patterns from the dataset. The training model output is compared with the ground truth image outputs to help to increase the prediction accuracy. The experimental output shows that the proposed model obtained 94.8% accuracy in early flood detection. Venkatraman M. *et al*. (2025) have proposed optimization-driven deep differential RecurFlowNet (ODD-RecurFlowNet) for examining water quality to increase sustainability in agriculture and related environmental applications. The model uses a giant armadillo optimization algorithm for the feature selection process. It predicts and classifies water quality using a global attention-based RecurFlowNet model. A preprocessing model, including data cleaning and robust scalar normalization, was implemented to evaluate the input dataset used in the experiment and predict the water quality and quality index. A deep convolution neural network (DDiff-CNN) will be employed for the water quality classification. From the output, it is noticed that the obtained accuracy is 98.01% with 0.039% MSE, which proves its superiority level over the existing methods. Arun Mozhi Selvi Sundarapandi *et al*. (2024) have proposed modeling approaches such as a lightweight dense network, a tree structurally simple recurrent unit, and a tree structurally simple. Initially, the light-weighted dense network was implemented to convert the input meteorological data variables into grayscale images to predict the required patterns. The non-linear relationship between the input and output is obtained by implementing the TS-SRU model. Integrating the models increases flood prediction accuracy with good precision and recall values.

2.1. Limitation and motivation

Due to advancements in flood prediction and forecasting, earlier methods face many critical limitations compared to the current situation and industry needs. No centralised flood prediction models are available to collect, process, aggregate, and forecast multiple areas. This also leads to potential data privacy breaches. Many unauthorised third parties access the sensitive hydrometeorological data obtained from various sensors, IoT devices, and satellite imagery, causing security problems, Nasir and Atal, (2023). Earlier methods have focused on extracting spatial or temporal features for flood prediction, where the prediction accuracy is low for global data. A potential hybrid model is needed to extract spatial and temporal features to improve prediction accuracy in satellite images. Most models analysed small-size static data obtained from a single location and predicted flood conditions. The static prediction outcomes cannot provide accurate predictions on flood conditions, which needs to analyse time-series data received from sensors or IoT devices generated continuously. Time series data includes flood data obtained over a long period and helps to create knowledge-discovered information to test the current situation. Some models are trained at localised datasets and provide less accuracy when applied in various coastal regions under different geographical and environmental conditions, where an additional model is needed for aggregating the locally trained data to train global data. Also, their efficiency is not applied and verified in the real-time flood data prediction. Thus, this paper has been motivated to design and implement a novel framework by integrating the Federated learning model for global data processing, the CNN model for predicting flood conditions using spatial features, and the LSTM model for predicting flood conditions using temporal features. This framework can overcome the above-said limitations and provides improved prediction accuracy for global time-series data

2.2. Problem statement

In coastal regions of smart cities, flooding makes land, properties, money, and life more dangerous. The earlier methods and approaches used for flood prediction face many challenges because of limited privacy and inefficiency in accurately extracting all the spatiotemporal features and flood patterns. This problem is considered a significant problem and is understood clearly. Thus, it is explained mathematically as follows: Let

$$X_t \cdot \in \boldsymbol{R}^{m\times n}$$

be the feature matrix obtained from the input data at time ***t***, where ***m*** denotes the spatial information and ***n*** denotes the hydrological and meteorological variables (e.g., temperature, precipitation, river discharge, sea level). Estimate the flood levels at the same time ***t*** is represented as

$$\boldsymbol{y}_t \in \boldsymbol{R}^m$$

The proposed method CNN-LSTM is used as the predictive function $\boldsymbol{f_0}$. Since flood data is collected from different geolocations, a federated learning model is used to aggregate the trained data, which are trained at the local server where they are generated. The data aggregation is obtained using the following expression:

$$\boldsymbol{\theta}^{t+1} = \sum_{i=1}^{K} \frac{\boldsymbol{n}_i}{\boldsymbol{N}} \boldsymbol{\theta}_i^t$$

The number of clients ***K***, generates and trains the data size of $\boldsymbol{n_i}$ at client ***I*** is aggregated. The entire data obtained from all the clients is

$$\boldsymbol{N} = \sum_{i=1}^{K} n_i$$

The model parameters of client I at iteration t are represented by θ_i^t The CNN model used the convolution layers to extract spatial features from flood data using

$$\boldsymbol{F}_s = \boldsymbol{\sigma}\left(\boldsymbol{W}_c * \boldsymbol{X} + \boldsymbol{b}_c\right)$$

The weight $\boldsymbol{W_c}$ is dynamically and with the bias b_c in each convolution layer to learn and extract the spatial features. The convolution operation (*) performs filtering operations activated by the function σ, to g*et al*l the features $\boldsymbol{F_s}$. The sequential data is processed by the LSTM model using the following equations:

$$\boldsymbol{h}_t = \boldsymbol{\sigma}\left(\boldsymbol{W}_h \boldsymbol{F}_s + \boldsymbol{U}_h \boldsymbol{h}_{t-1} + \boldsymbol{b}_h\right)$$

$$\boldsymbol{y}_t = \boldsymbol{\sigma}\left(\boldsymbol{W}_o \boldsymbol{h}_t + \boldsymbol{b}_o\right)$$

Like the CNN model, LSTM also uses weight values $\boldsymbol{W_h}$ and $\boldsymbol{U_h}$ with the bias value $\boldsymbol{b_h}$. The hidden state $\boldsymbol{h_t}$ during the time t, provides the hidden information and the obtained output layer parameters, such as $\boldsymbol{W_o}$, $\boldsymbol{b_o}$. The error and loss values are calculated to confirm the prediction and forecasting accuracy rate. For example, mean square error (MSE) is used to train the model:

$$\boldsymbol{L} = \frac{1}{\boldsymbol{N}}\left(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t\right)^2$$

The predicted output indicating the flood severity at time ***t*** is denoted as $\hat{y}_t$. Since the input data is collected from many coastal regions of the smart city, it is aggregated by deploying a Federated Learning model, which uses federated-average (FedAvg) for the global model, and it is expressed as:

$$\boldsymbol{\theta}_{global} = \sum_{i=1}^{K} \frac{\boldsymbol{n}_i}{\boldsymbol{N}} \boldsymbol{\theta}_i$$

Where the raw data is processed at the local servers and updated in the global server. This makes the research people design and implement an efficient flood prediction model with privacy preservation and improved prediction accuracy and forecasting of flood events. To provide a better solution, this research proposes a predictive framework based on Federated Learning (FL) and CNN-LSTM that allows for distributed learning at various locations while safeguarding data privacy. The model incorporates Convolutional Neural Networks (CNN) to extract spatial features and Long-Short-Term Memory (LSTM) networks to capture temporal dependencies in flood-related data.

2.3. Proposed model

Flooding significantly threatens coastal smart cities, damaging property, fatalities, and economic disturbances.

Conventional models for predicting floods frequently face challenges due to limited data availability, privacy issues, and their ineffectiveness in accurately capturing intricate spatial and temporal flood patterns. The challenge is to create a sophisticated flood prediction system that preserves privacy and is efficient in computation while providing accurate real-time forecasts of flood events. This paper implements CNN, LSTM, and CNN-LSTM for flood prediction in small-scale, large-scale, and global satellite images to save people by providing prior information. CNN model provides promising output in satellite image processing and classification, and LSTM provides promising output in time-series, series, and continuous data with memory and time management. Thus, this paper integrated CNN and LSTM models for efficient flood prediction for the global satellite dataset.

2.4. Convolutional neural network

In digital image processing, CNN recognises the objects in the input image. It comprises various layers, such as convolution, pooling, and fully connected layers. This convolution layer ensures that the incoming two-dimensional data is filtered appropriately to create an appropriate feature map. Different feature maps need different combinations to get a final result. To generate a non-linear signal, CNN requires an activation function. The rectified linear unit (ReLU) is frequently retained for an activation function. The convolution operation takes a weighted sum of local patches from the input image and then applies an activation function. This operation can be expressed mathematically as:

$$Z_{i,j}^{l} = f\left(\sum_{m}\sum_{n} W_{m,n}^{l} X_{(i+m),(j+n)}^{l-1} + b^{l}\right)$$

Where the activation at the output feature map *l*, position at (*i*, *j*) denoted as Z_i, j^l. weights of the filter (kernel) in the layer *l*, denoted as $W_{m,n}{}^{l}$, values from the prior layer are denoted as $X_{(i+m),(j+n)}^{l-1}$, bias in layer *l*, denoted as b^l, and the activation function, typically ReLU ($f(x)$) = max (o, *x*) denoted as *f.* The pooling layer typically extracts the invariant features by removing the non-maximal values to do non-linear down-sampling. Pooling layers are applied post-convolution to diminish the spatial dimensions of the feature maps. Max-pooling, which chooses the maximum value within a region, is often utilized:

$$P_{i.j} = \max\left(Z_{(i.j)}, Z_{(i+1.j)}, Z_{(i.j+1)}, Z_{(i+1.j+1)}\right)$$

Where the pooled value for the region for the region in the feature map Z, denoted as $P_{i,j}$

The output is transformed into a 1D vector following several convolutional and pooling layers:

$$Flatten(Z) = vector\ of\ all\ elements\ of\ Z$$

The flattened feature vector is received by the fully connected layers, which apply weighted sums and then use an activation function (e.g., softmax for classification):

$$y = f(W.x + b)$$

Where: weights is denoted as *W,* input is denoted as *x,* Bias term is denoted as *b,* and Softmax function for multi-class classification is denotes as *f.*

The loss function can compute the water errors between the observed and model outputs through the fully connected layer's connection to the max-pooling layer. Binary Cross-Entropy is often employed for binary classification tasks:

$$L = -\frac{1}{N}\sum_{i=1}^{N} y\log\left(\hat{y_i}\right) + (1-y_i)\log\left(1-\hat{y_i}\right)]$$

Where: True label (1=flood, 0 = no flood) is denoted as y_i Predicted probability of flood is denoted as $\hat{y}_i$, and the number of samples is denoted as *N.*

For multiclass classification, Categorical Cross-Entropy is applied:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log\left(\hat{y}_{i,c}\right)$$

The number of classes (e.g., minor, moderate, major flood) is denoted as *C,* True class label for sample *I* is denoted as $y_{i,c}$, and Predicted probability for class *c* is denoted as ($\hat{y}_{i,c}$). The latest research indicates that Mean Square Error (MSE) is used as a loss function. This establishes the model for flood detection. This facilitates the extraction of significant features and empowers them to provide the patterns and structures necessary for flood detection. Through the application of CNNs, the flood detection model can proficiently examine satellite images and pinpoint areas affected by floods. This model has a rich feature derived from a large-scale image analysis task. This pretrained model enables flood detection to obtain information from representations derived from extensive image datasets. This makes it possible to process the performance even when dealing with minimum flood-specific data. The satellite images highlight areas affected by floods and other disasters that should be observed. This divides affected and non-affected regions into separate folders (e.g., flood images and non-flood images). This plays as a foundation for the analysis of flood detection.

2.5. Model architecture

Some convolutional neural network models, such as ResNet50, VGG16, and InceptionV3, are used to create flood detection models. The structure of the CNN model is shown in **figure-1**. The CNN model performs by receiving the source from the areas detected by flood as the input. The input source is converted by cov2D, the fundamental operation dealing with image data. The source then reduces its spatial dimension of image data by maxpool2D, which is repeated twice. After the con2D and maxpooling2D process, the image is flattened, which converts the 2d array into a 1D array by the dense layer, which connects the previous layers. This process modifies the source and is used for the flood detection model. The modified data is used in flood and non-flood areas for flood detection, which is none other than the output of the CNN.

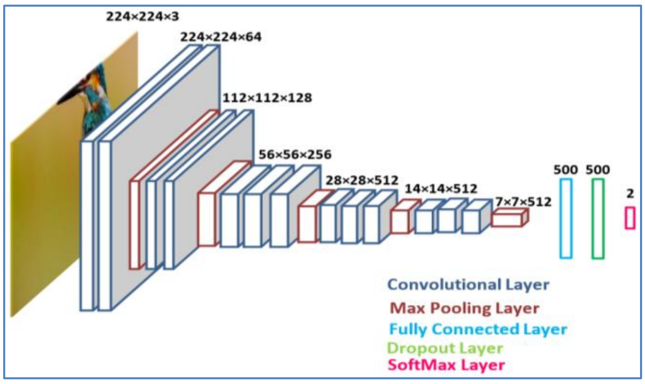


Figure 1. CNN model Architecture

2.6. LSTM

LSTM finds temporal relationships in satellite images because a flood is a time-dependent event. CNN transfers sequential features to LSTM to learn all the patterns regarding time. The cells in the LSTM are mathematically expressed as:

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right) \quad (3)$$

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right) \quad (3)$$

$$\tilde{C}_t = tanh\left(W_C \cdot \left[h_{t-1}, x_t\right] + b_C\right) \quad (3)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (3)$$

$$o_t = \sigma\left(W_o \cdot \left[h_{t-1}, x_t\right] + b_o\right) \quad (3)$$

$$h_t = o_t \cdot \tanh\left(C_t\right) \quad (3)$$

In the above equations, at the time t, the forget gate is represented as f_t, input gate is represented as i_t, and the output gate is represented as o_t. The state of each cell is represented as C_t, and the hidden state is h_t. The activation function activates all the processes of each cell σ, called the sigmoid function. Finally, the fully connected layer, called the dense layer, predicts the results of flood or non-flood in the given image. It activates the process using a sigmoid function and Adam optimiser. the LSTM model architecture is shown in **figure-2**.

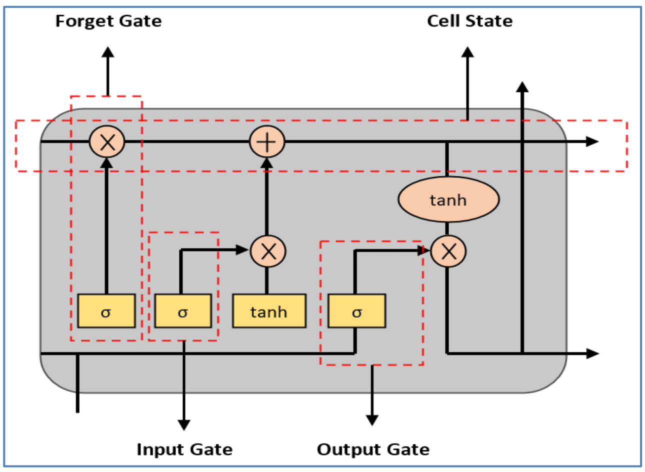


Figure 2. LSTM Model

2.7. Federated learning used CNN-LSTM framework

The flood prediction and forecasting model is developed to predict the severity of floods and warn people living in coastal areas. The entire architecture of the FL-CNN-LSTM framework comprises multiple layers, such as application, data analysis, network, and data generation layers, as shown in **Figure-3**. This framework predicts flood conditions from multi-dimensional data. Initially, the raw data is generated by deploying various sensors, IoT devices, and related equipment for monitoring and recording floods, weather, and rainfalls [edge layer]. The data generated at the edge layer is collected and preprocessed. The missing data, redundant data, overfitting and underlying data are eliminated to improve the data quality. After preprocessing, the data concerning various parameters is transmitted separately to the base station through multiple channels.

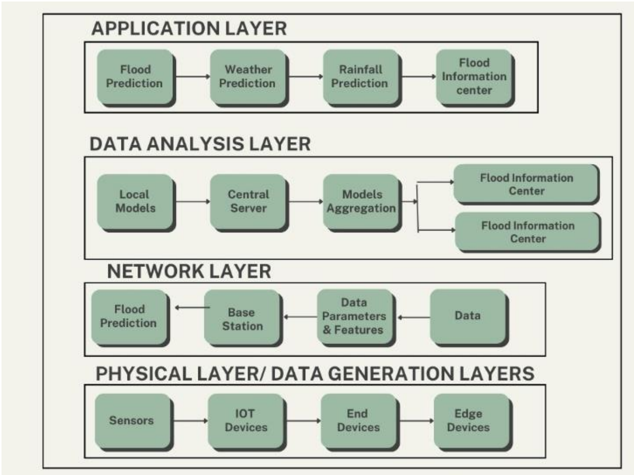
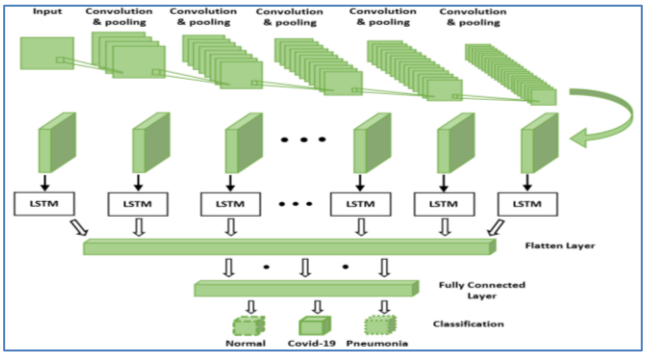


Figure 3. Federated Learning Model

The flood prediction and forecasting model is developed to predict the severity of floods and warn people living in coastal areas. The entire architecture of the FL-CNN-LSTM framework comprises multiple layers, such as application, data analysis, network, and data generation layers, as shown in **Figure-1**. This framework predicts flood conditions from multi-dimensional data. Initially, the raw data is generated by deploying various sensors, IoT devices, and related equipment for monitoring and recording floods, weather, and rainfalls [edge layer]. The data generated at the edge layer is collected and preprocessed. The missing data, redundant data, overfitting and underlying data are eliminated to improve the data quality. After preprocessing, the data concerning various parameters is transmitted separately to the base station through multiple channels. The physical layer contains edge-based data collection centres. These sensors will continuously gather and transmit data to the local client station for local model training. These sensors encompass rain gauge sensors, water flow sensors, water flow calculating sensors and water level sensors, each playing a vital role in monitoring the various aspects of the water system. The proposed model involves processing hydrological and meteorological datasets from a targeted region in central Asia. In Table 1, the data gathered for this research is displayed. All tracked data parameters on the physical layer undergo rigorous regulation during pre-processing before local model training. These pre-processing datasets will be executed for dataset normalisation on local stations where the corrupt records and missing values will be removed. The network layer in **Figure-3** ensures the secure transmission of local data models to a central server. To assure the privacy preservation of these local models, the TCP/IP protocols have been utilized in this layer. By obtaining local data models from a client, this layer can provide asynchronous communication, and for training global models which can predict floods, these local data models will be transmitted to the server. The processing layer in **Figure-3** facilitates the execution of the FL cycle by training local models at client stations, transferring local models to the client-server, aggregating the local model, global data model learning at the central server and initiating predicted flood alerts.

```
Federated_Learning_Algortihm( )
{
Server Module
    Time = 0
    Local module = 0
    For i = 1 to n  // n is the number of training epochs
        Z = maximum number of sensors
        K = maximum number of parameters
        z.activate = true
        For j = 1 to m
                Local module = (local module + 1)^k
        Client-updation= local − module_t(k)
        End j
                local − module_{t+1}k
        End i
Client Updation
        For i= 1 to e // number of epochs
            For each local module
                Set batch size b ∈ β  // maximum batch size
                local module_i = local module_k − Δ(local module)
            End for
        End i
    Return local − module_k
End
```

The above algorithm explains the federated learning-based flood prediction process's overall process. Initially, the server sends a message to all the client's local modules 0. Then, the server gets the updates from all active clients in fixed intervals. All the active clients connected with a server calculate their local calculations and update the server. Each client calculates their weighted matrix with a learning η and trains the module. After the training process, the updated module (local $module_k$) is transmitted to the server. Finally, the weighted average matrix obtained from various local modules trains the global data and triggers flood alerts accordingly.



CNN-LSTM Model

2.8. Experimental setup

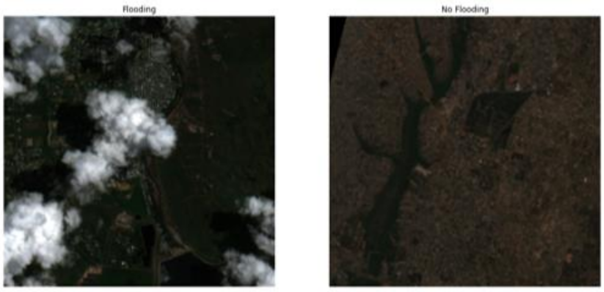
In this paper, the input datasets are experimented with using the simulation software installed in the system with an Intel i7 processor, 1TB HDD, 64GB RAM, NVIDIA GPU, 3.0GHZ processor, and Windows OS. The deep learning model is built using Python software, and the data processing is processed using the GPU backend. The proposed model is trained using real-time and meteorological satellite images. The CNN model is applied to extract the essential features from the input data. The LSTM model is applied to detect the temporal feature dependencies among the input data. The federated CNN-LSTM approach ensures data privacy and reduces computational complexity in forecasting floods from satellite images. The model's performance is evaluated using accuracy, loss, and F1-score values.

2.9. Dataset

The dataset has 412 time series. Each series contains between 4 and 20 optical figures and 10 and 58 SAR figures (https://www.kaggle.com/datasets/virajkadam/sen12flood). On average, each series has about 9 optical figures and 14 SAR figures. The statistics were taken between December 2018 and May 2019. A flood is visible in 40% of the optical figures from the Sentinel-2 satellite and 47% of the SAR figures from the Sentinel-1 satellite. Like in the MediaEval dataset, when a flood happens in a sequence, all the figures after it are marked as flooded. This is based on the idea that the ground still looks different even after the flood has passed.

3. Results and discussion

Due to the impact on livelihoods, the climate-changing rate causes frequent flooding worldwide, affecting people's survival rate. The flooding is analysed using geospatial data, which is used for mapping, disaster management, and navigation to reduce the impact of risks on flooding. The geospatial data is used as maps by high-resolution optical and radar imagery (HRORI), which is used for the detailed image of the earth's surface and forecasting weather parameters. The maps using sentinel radar are used in HRORI to identify the flood-affected areas and predict water-related geohazards to acquaintance water bodies in affected regions with the help of learning algorithms in HRORI. Thus, this paper applies various learning algorithms such as CNN, LSTM, CNN-LSTM, and federating learning models to predict floods and safeguard people's livelihoods. Using the simulation software, the various flood prediction results are analysed, and the obtained results are discussed in this section, along with numerical and graphical results.



(a) (b)

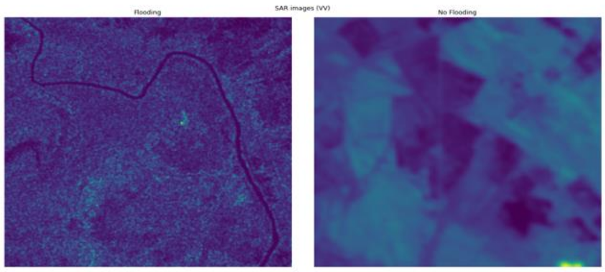
Figure 4. Sample input satellite Image with and without flood sign

Figure-4 (A) and (b) show two satellite figures in optical RGB format, comparing the same area before and after a flood, respectively. The figure-(b) on the right shows the region under normal conditions, with no flooding. Everything is clear, including roads, buildings, and natural features. The land is dry, and no sign of water is covering any area. Figure-(a) on the left shows the same region during a flood. Large areas are covered with water, making seeing roads and other structures difficult. Some parts of the figure look darker because of water saturation. Some areas also have clouds, which is common during extreme weather. This comparison helps us understand how flooding changes the landscape and shows how satellite figures help monitor natural disasters and their effects on people and places.

(a) (b)

Figure 5. Region (a) Before (b) after flood

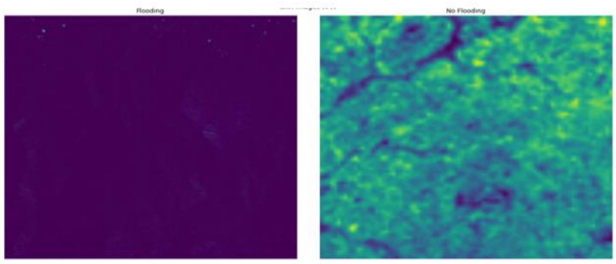
The two satellite **figures-5** (a) and (b) show a region before and after a flood. Figure-(b) shows everything as usual, taken before the flood. Roads, land, and plants are easy to see, and the water stays in its usual place. Figure-(b) shows the region after a flood. Much of the land is underwater, making it hard to see things. The satellite image looks darker because of all the water, and some clouds cover part of the view. These figures help show how destructive floods can be and how satellites help in disaster response.



(a) (b)

Figure 6. SAR image (a) During and (b) after the flood

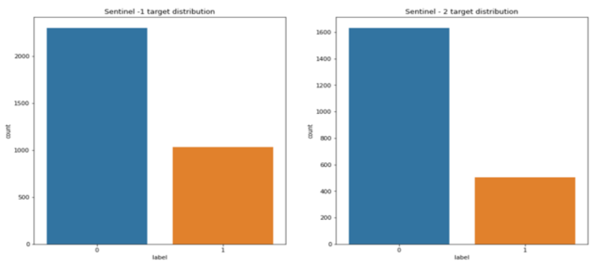
The SAR images compare a landscape before and after flooding. In the flooded image (**Figure-6 (a)**), dark areas show water coverage, mainly along rivers and floodplains, while bright areas indicate land and vegetation. In the dry image (**Figure-6 (b)**), water is less widespread, and the land appears more colourful due to urban structures and vegetation. This highlights how SAR images help in tracking floods.



(a) (b)

Figure 7. SAR image-Region (a) Before and (b) after the flood

The image compares SAR pictures of a landscape before and after flooding. The "Flooding" image- (**figure-7 (a)**) shows dark areas indicating water coverage, which smooths the surface. The "No Flooding" image-**(figure-7 (a))** shows land with precise features that strongly reflect radar signals. This difference highlights how flooding affects surface reflectivity and demonstrates SAR's role in detecting flood-affected areas.



(a) (b)

Figure 8. Target Label Distributions For (a) Sentinel-1 And (b) Sentinel-2 Datasets

Figure-8 (a) and (b) compare the target label distributions for Sentinel-1 and Sentinel-2 datasets. In both charts, most data points are labelled 0, meaning most input images show no flooding area. The Sentinel-1 dataset has 2500 no-flooding and 1000 flooding satellite images. The

Sentinel-2 dataset has 1600 no-flooding and 500 flooding satellite images. This indicates that the dataset is imbalanced, with much more non-flooded data than flooded data, making it harder to detect flooded areas using machine learning models. The input dataset is classified into three phases: training, validation, and testing to balance it.

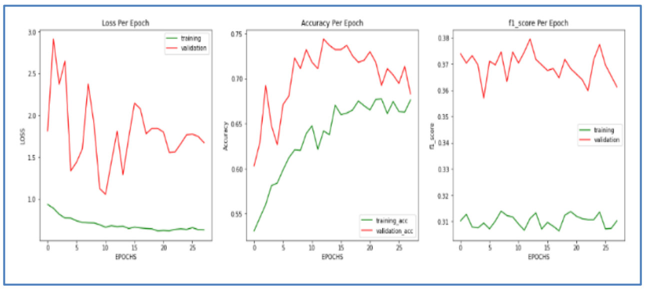


Figure 9. Training and validation (a) Loss, (b) Accuracy, and (c) F1-score of the CNN model

The Loss Per Epoch of the CNN model graph in **Figure-9 (A)** shows the model's prediction loss rate over 25 epochs. The training loss (green line) decreases steadily and reaches 0.3. And the validation loss (red line) keeps changing significantly and stays higher than the training loss of 1.8 loss rate. This might mean the model is overfitting or having trouble generalizing. The Accuracy Per Epoch graph in **Figure-9 (b)** shows the LSTM model's training and validation accuracy. The result indicates that the validation accuracy (red line) is higher than the training accuracy (green line), which is 68% on detecting the flood from the satellite images. This might mean the validation data is more straightforward for the model to classify or that its learning patterns are specific to the validation data. The F1-score Per Epoch graph in **Figure-9(c)** shows how balanced the model is between precision and recall. The validation F1 Score (red line) is higher than the training F1 Score (green line), suggesting the model performs better on the validation data. However, the validation F1 Score fluctuates, indicating the model's performance changes across different validation samples.

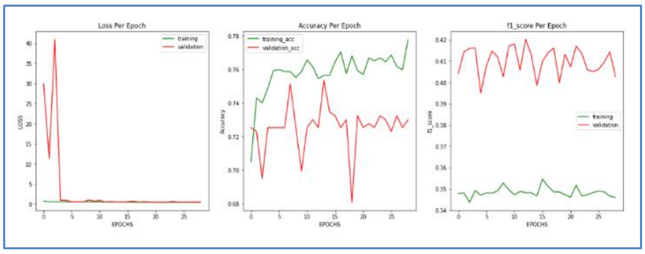


Figure-10. Training and validation (a) Loss, (b) Accuracy, and (c) F1-score of the LSTM model

The Loss Per Epoch of the LSTM model graph in **Figure-10 (A)** shows how the model's error changes over time. The training loss decreases steadily, so the model predicts the flooded and non-flooded regions with a loss rate of 0.1. The validation loss is initially high but drops quickly, indicating the model improves quickly. The Accuracy Per Epoch of the LSTM model graph in **Figure-10 (b)** shows how well the model predicts. The training accuracy keeps improving, which means the model is learning the patterns in the data with an accuracy of 78%. The validation accuracy increases but generally improves to 71%, showing that the model works well, though not always perfectly. The F1-score Per Epoch of the LSTM model graph in **Figure-10 (c)** shows the balance between precision and recall. The validation F1 Score is usually higher than the training F1 Score of (0.40), meaning the model performs well on new data. But the model

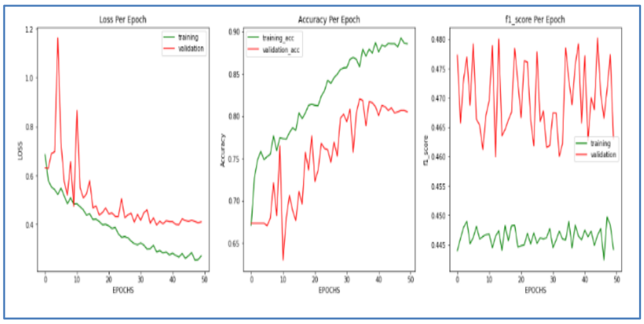


Figure 11. Training and validation (a) Loss (b) Accuracy, and (c) F1-score of the CNN-LSTM model

The Loss Per Epoch of the CNN-LSTM model graph in **Figure-11 (a)** shows how the model's loss decreases over time. The green line represents training loss, which steadily declines and reaches a 0.1 loss rate when processing the 50th epochs, indicating that the model is learning from the training data. The red line represents validation loss, which fluctuates significantly, suggesting that the model might not generalize well to unseen data and could be overfitting. **Figure-11(b)** shows the model's training and validation accuracy. The green line (training accuracy) increases consistently, showing that the model is improving at making correct predictions on training data and achieved 90% accuracy on the 50th epochs. The red line (validation accuracy) also improves but is unstable, meaning the model's performance varies on new data, which could indicate overfitting or noisy validation data. The F1 Score measures shown in **Figure-11 (c)** illustrate the balance between precision and recall. The green line (training F1 Score) remains relatively stable but does not improve much, suggesting the model is not significantly enhancing the precision-recall balance on training data. The red line (validation F1 Score) fluctuates heavily, indicating inconsistent performance on unseen data, which may require better tuning or more balanced data. On the 50th epoch, the model achieved a 0.463 F1 score. **Figure-12** shows actual Vs predicted images that explain a model's prediction about flooding. The image on the first column is the original satellite image with "No Flooding" labelled. The second image is a saliency map highlighting the necessary pixels the model used to make the "No Flooding" prediction. The third image shows a Grad-CAM visualisation, highlighting the areas the model considered most important using a heatmap. The fourth image is an improved version of the Grad-CAM, which refines the highlighted areas. These images help us understand how the model made its "No Flooding" prediction. The overall result illustrates that the proposed model is more suitable for flood prediction.

We compare the proposed FL-CNN-LSTM model with other standard flood prediction models to assess how well it works. These include ARIMA, a statistical forecasting method based on time-series data; Random Forest, a machine learning model for analyzing hydrological data; and ANN, a deep learning method for predicting floods. We also compare it with a CNN-LSTM model that combines CNN and LSTM but is trained on a centralized dataset. Finally, we compare it with the proposed FL-CNN-LSTM, which uses federated learning to ensure decentralized training and better privacy protection. The FL-CNN-LSTM model achieves an accuracy of 94.5%, which is better than traditional machine learning and deep learning models. Although CNN-LSTM shows competitive performance with 89.7% accuracy, it doesn't preserve privacy. The FL-based model allows decentralized training, which means it works well across different coastal environments without needing retraining for each region. However, other models need retraining for various areas, limiting their scalability. Regarding training time, FL-CNN-LSTM takes 10.7 seconds, slightly faster than CNN-LSTM at 12.3 seconds but slower than simpler models like ARIMA (3.2 seconds) and RF (5.6 seconds). The trade-off between accuracy and training time is worth it because of the better flood prediction results. Lastly, FL-CNN-LSTM is the only model that ensures privacy by not collecting data in a central server, unlike other models that raise security concerns due to data aggregation. The graph compares the performance of five models across Accuracy, Precision, Recall, and F1-Score. The proposed FL-CNN-LSTM model outperforms all others with the highest scores in every metric, demonstrating its superior predictive capabilities. CNN-LSTM follows closely, while ARIMA shows the lowest performance. This highlights FL-CNN-LSTM as the most efficient and reliable model among the evaluated options.

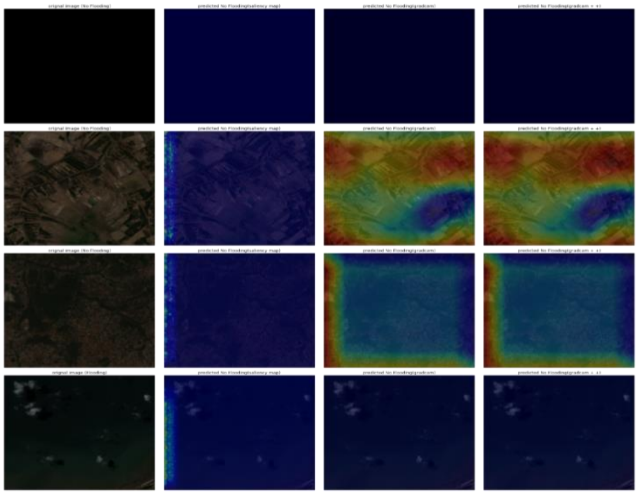


Figure 12. Actual vs. Predicted Satellite Flood Image

Table 1. Performance Comaprison

Model	Training Time(s)	Privacy Preservation	Generalization Across Regions
ARIMA (Boulton *et al*., 2022)	3.2	No	Low
Random Forest (Zhang *et al*., 2023)	5.6	No	Moderate
ANN (Shen *et al*. 2022)	8.1	No	Moderate
CNN-LSTM(Karthik *et al*., 2025)	12.3	No	Moderate
Fl-CNN-LSTM(Proposed)	10.7	Yes	High

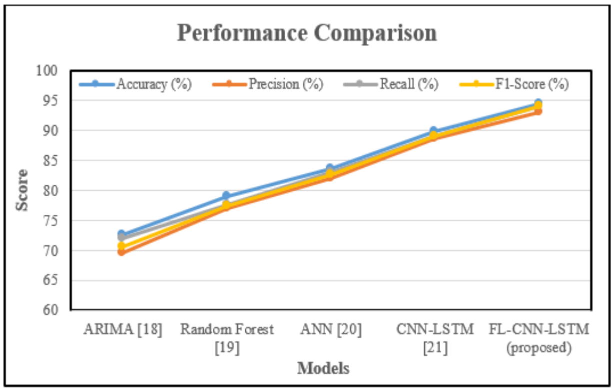


Figure 13. Performance Comparison

Table 1 compares the training time, privacy preservation, and generalization of different machine learning models across regions. ARIMA trains the fastest, taking only 3.2 seconds, but it doesn't preserve privacy and has low generalization. Random Forest takes 5.6 seconds to train, doesn't preserve privacy, but has moderate generalization. The Artificial Neural Network (ANN) takes 8.1 seconds, doesn't preserve privacy, and has moderate generalization. CNN-LSTM takes the longest time, 12.3 seconds, does not preserve privacy, and has moderate generalization. The proposed FL-CNN-LSTM model offers a good balance, taking 10.7 seconds to train, maintaining privacy, and providing high generalization across regions.

4. Conclusion

Flood prevention in shoreline areas of smart cities is a major concern of accurate, extensible, and anonymity-promoting approaches. Conventional flood prediction systems frequently face challenges connected with data privacy, virtual inadequacy, and restricted abstraction throughout various geographic zones. This paper presents a novel federated learning (FL) derived from a composite CNN-LSTM model developed for reinforcing flood defence precision to ensure information security and resilience. Extraction of Federated Learning (FL), this model authorizes localised training throughout various locations, prohibiting data breaches during the process of upgradation in the robustness of the forecasting model. CNN portion productivity holds the location features derived from the radar image and sensor signal, during which the LSTM network executes the temporary addictions in the meteorological sequential data, resulting in enhanced flood prediction accuracy. To prove the efficiency of the proposed model, it is compared with existing models in terms of accuracy, precision, recall, and F1-score. The result of the comparison states that, compared to other models, the proposed CNN-LSTM model has achieved high performance metrics results with more than 95% accuracy. The overall observational outputs illustrate that the FL-CNN-LSTM model exceeds the conventional approach by providing better forecasting accuracy, cost-effective computing, upgraded data privacy, and increased adaptability for implementing actual-time flood prediction in smart cities. Though this model performed more efficiently, it has some limitations when performing with diverse environmental and geographical conditions of data. The proposed model entirely depends on continuous time-series sensor data, which may have limitations on the region without a sensor.

Future work

Apart from the favourable completion, there is a scope for the upcoming analysis and advancement. The present model focuses on the systematic satellite and IoT sensor signal. However, incorporating diverse digital sources like live social media insights, aerial mapping and community input helps to improve forecasting abilities. Maximizing FL architecture through integrating approaches like distinct secrecy and protecting all-party computing will reinforce security regulations during the maintenance of data usage. Analysing advanced DL systems like GNNs (Graph Neural Networks) and Modifier-related models will clarify the geospatial pattern recognition and update the prediction accuracy. The upcoming investigation will involve implementing the proposed method in real-time flood indication systems and calculating the efficiency in various climatic conditions and hazardous areas. Incorporating this technological development, the FL-CNN-LSTM model is highly potent towards upgrading flexible, discerning and worldwide adaptable flood warning systems, contributing to updated hazard prevention and climate stability in smart cities. In the future, researchers can integrate drone-based systems and real-time data for effective flood prediction. In addition, GNN-based systems enhance geospatial reasoning in dynamic environmental.

References

Aijaz, J., and Sultan, A. (2024). Artificial Intelligence and Climate Change. *CISS Insight Journal*, **12**(2), 62–82.

Armano, P. G. and Marchesi, M. (2021). Time Series Prediction Using ARIMA and Machine Learning Models for Hydrological Data Analysis, *Journal of Hydrology*, **596**, 125987.

Ashizawa T., Sudo N., and Yamamoto H. (2022). How Do Floods Affect the Economy? An *Empirical Analysis using Japanese Flood Data.*

Babu T, Raveena Selvanarayanan, Tamilvizhi Thanarajan and Surendran Rajendran (2024) Integrated Early Flood Prediction using Sentinel-2 Imagery with VANET-MARL-based Deep Neural RNN, *Global NEST Journal*, **26**(10). https://doi.org/10.30955/gnj.06554.

Bischke, B., Helber, P., Schulze, C., Srinivasan, V., Dengel, A., and Borth, D. (2017). The Multimedia Satellite Task at MediaEval 2017. In *MediaEval*.

Boulton, J. B., Wilson, A. G. and Cooper, K. S. (2022). Flood Prediction Using Machine Learning: A Comparative Study of Random Forest, Gradient Boosting, and SVM, *IEEE Transactions on Geoscience and Remote Sensing*, **60**(5), 4502–4517.

Farooq, M. S., Tehseen, R., Qureshi, J. N., Omer, U., Yaqoob, R., Tanweer, H. A. and Atal, Z. (2023). FFM: Flood forecasting model using federated learning. *IEEE Access*, **11**, 24472–24483.

Giezendanner, J., Mukherjee, R., Purri, M., Thomas, M., Mauerman, M., Islam, A. K. M., and Tellman, B. (2023). Inferring the past: a combined CNN-LSTM deep learning framework to fuse satellites for historical inundation mapping. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2155–2165).

Gupta, R., Hosfelt, R., Sajeev, S., Patel, N., Goodman, B., Doshi, J., Heim, E., Choset, H. and Gaston, M., (2019). xBD: A Dataset for Assessing Building Damage from Satellite Imagery. *arXiv preprint arXiv:1911.09296*.

Kabir, S., Patidar, S., Xia, X., Liang, Q., Neal, J. and Pender, G. (2020). A deep convolutional neural network model for rapid prediction of fluvial flood inundation. *Journal of Hydrology*, **590**, 125481.

Karthik.S, Surendran.R, Sam Kumar G.V. and Senduru Srinivasulu (2025). Flood Prediction in Chennai based on Extended Elman Spiking Neural Network using a Robust Chaotic Artificial Hummingbird optimizer, *Global NEST Journal*, **27**(4). Available at: https://doi.org/10.30955/gnj.07113.

Mohanty, S. N., Nanjundan, P., and Kar, T. Artificial Intelligence in Forecasting. https://www.kaggle.com/datasets/virajkadam/sen12flood

Nahak, S. K., Acharya, S. K., and Padhy, D. (2024). Enhanced Flood Forecasting: Revolutionizing Prediction with Federated Learning. *In International Conference on Smart Computing and Communication* (457–467). Singapore: Springer Nature Singapore.

Oladapo, K.A., Idowu, S.A., Adekunle Y., and Ayankoya F. (2020). Categorization of Conditioning Variables for Pluvial Flood Risk Assessment, *International Journal of Scientific Engineering and Research*., **11**(8), 355–368.

Pareek, P. K., Srinivas, C., Nayana, S., and Manasa, D. S. (2023). Prediction of Floods in Kerala using Hybrid Model of CNN and LSTM. In 2023 *IEEE International Conference on Integrated Circuits and Communication Systems* (ICICACS) (01–07). IEEE.

Siva, N., Sivaiah, B. V., Dharani, K., Srilekha, K. V., Yaswanth, E., and Venkateshwarlu, G. (2024). Federated Learning-Based Flood Forecasting Model Enhancing Predictive Accuracy and Privacy in Flood Prediction. In 2024 5t*h International Conference for Emerging Technology* (INCET) (1–7). IEEE.

Song, C. (2022). Application of nature-based measures in China's sponge city initiative: Current trends and perspectives. *Nature-Based Solutions*, **2**, 100010.

Sundarapandi, A.M.S. *et al*. (2024) A Light weighted Dense and Tree structured simple recurrent unit (LDTSRU) for flood prediction using meteorological variables, *Global NEST Journal*, **26**(8). Available at: https://doi.org/10.30955/gnj.06242.

Supraja, A., Rao, S. K., Vamsi, T., Prasanthi, P., Parvez, S. and Jhansi, K. (2024). Flood Routing in BVRIT Narsapur campus using Software Applications on Sustainable water resources management. *In Journal of Physics: Conference Series*, **2779**(1), 012027. IOP Publishing.2024.

Venkatraman, M., Surendran R., Senduru Srinivasulu, Vijayakumar K. (2025) Water quality prediction and classification using Attention based Deep Differential RecurFlowNet with Logistic Giant Armadillo Optimization, *Global NEST Journal*. https://doi.org/10.30955/gnj.0679 & nbsp

W. Shen, F. Li, and H. Tang. (2022). Hybrid CNN-LSTM Model for Flood Forecasting Using Satellite and Sensor Data, *IEEE Access*, **10**, 120458–120471.

Wu, W. Emerton R., Duan Q., Wood A. W., Wetterhall F. and Robertson, D. E. (2020). Ensemble flood forecasting: Current status and future opportunities, *Wiley Interdisciplinary Reviews: Water*, **7**(3), doi: 10.1002/wat2.1432.

Zerara M. (2021). Machine Learning and Statistical Methods for Time Series Forecasting : A Case Machine Learning and Statistical Methods for Time Series Forecasting: a Case Study with Water Demand, Authorea Preprints, doi: 10.36227/techrxiv.15019698.v1.

Zhang, H., Chen, R. and Liu, S. (2023). Deep Learning-Based Flood Prediction with Artificial Neural Networks: Case Studies from Coastal Regions, *Environmental Modelling & Software*, **147**, 105196.