FFM: REVOLUOOD Forecasting Using Decentralized FFNN and CNN2D ALGORITHMSTIONIZING FL

P Ramya[1], Dr. K Venkataramana[2], V Sridhar[3]
[1]PG Scholar, [2]Professor, [3]Assistant Professor
Dept. of Computer Science and Engineering,
VEMU Institute of Technology, P. Kothakota, Chittoor, Andhra Pradesh, India.

Abstract - Floods are one of the most common natural disasters that often occur and cause serious damage to property, agriculture, economy and life. Flood forecasting presents a major challenge for researchers who have been battling against forecasting floods for a long time. The flood prediction model was proposed using federal learning techniques which ensures data protection, guarantees data availability, promises data security, and predicts flooding by banning data transferred over the network for model training. Flood Forecasting Model (FFM) is the most advanced machine learning technology (ML) that conducts ding tests. Federal Learning technology seeks training local data models in the field instead of sending huge data records to central servers for local models aggregation and training, it focuses on transferring these local models within the network server. This proposed model integrates a local training models data segregated from eighteen clients investigation at which station flooding is about to happen and generates flood alarms at a 5-days lead time. Local models of Feed Forward Neural Networks (FFNN) are trained at client stations where tides were expected. The flood forecasting module of the local FFNN model predicts the expected water level by taking several regional parameters as inputs. Data records for five different rivers and barrels were collected between 2015 and 2021 and took into account which includes four aspects such as rainfall-runoff, snow melting, hydrodynamics and flow routing. The proposed flood forecasting model predicted that previous floods in selected zones occurred with an accuracy of 84% from 2010 to 2015.

Key Words*: *Feed Forward Neural Networks(FFNN), Federal learning, Flood Forecasting Model, Hydrodynamics, Machine Learning.

1. INTRODUCTION

The escalating frequency of natural and man-made disasters, including floods, has driven a global concern. Rising flood risks attributed to hydrological extremes, urbanization, and climate change pose severe threats to life, infrastructure, and economies. Developing countries are disproportionately affected, with floods causing casualties and economic crises. As climate change intensifies, floods become more frequent and intense. The need for accurate flood prediction systems has grown to mitigate impacts. Conventional methods, including statistical techniques, have struggled to provide precise predictions due to complex environmental factors. Machine learning (ML) offers promise, but data privacy and security concerns hinder its effectiveness. This article introduces a novel flood forecasting model using federated learning, addressing data privacy concerns while enhancing prediction accuracy.

2. LITERATURE SURVEY

In recent years, the proportion of humans caused by nature and humans has been increasing in the world [1]. In Hydrodynamic Modelling (Patro et al., 2009),This study focused on simulating flood behavior in large rivers using limited hydrological data. The authors used hydrodynamic models to predict river flows and flood patterns, which helped understand the physical process of water movement. Theoretical contribution: Accurate flood prediction requires modeling real-world conditions such as water flow, rainfall, and snowmelt. Global flood risk has raised due to hydrological extremities, increased urbanization and global warming [2].

Floods are devastating natural disasters that result in severe life losses, significant destruction of infrastructure, agriculture and downfall of overall socioeconomic system of a country. Floods are common in all parts ofthe world but their intensity vary from region to region. Flood Causes and Socio-Economic Impact (Rahman & Shaw, 2015) examined natural and social causes of floods in the Hindu Kush region. It emphasized the role of rapid urbanization, poor drainage, and climate change in increasing flood frequency. Theoretical contribution: Understanding flood causality is crucial for developing effective and timely flood warning systems[3].In developing countries, flood occurrences inflict countless casualties every year and cause cruel economic crises, rising pecuniary problems [4].

Global temperature escalation resulting in overall climate change cause an increased rate of snow melting and precipitation due to which floods are becoming more frequent and intense [5]. Figure 1 shows that frequency of flood occurrence in Pakistan is higher than other natural disasters [6]. Floods have been observed to outnumber at all other calamities happened in the South Asian countries during 2021 [7]. In the face of escalating threats posed by floods to both human life and economic infrastructure, governments are in critical need of reliable predictive systems to enable timely and effective interventions [8].

Despite numerous global and regional methodologies, models, and strategies proposed for flood prediction, the inherent complexity of this natural disaster has impeded substantial improvements in accuracy [9]. Flood Forecasting Using Deep Learning (Gude et al., 2020) This paper presented a deep learning framework for accurate flood prediction under uncertainty. It focused on combining multiple parameters and using past weather data to improve prediction. Theoretical contribution: Deep learning models can reduce error margins in complex environmental forecasting. Established statistical methods such as climatology average method (CLIM), flood frequency analysis (FFA), Bayesian forecasting models (BFM), and artificial neural networks (ANN) have utilized complex mathematical expressions to represent flood causing physical processes[10-11].

Federated Learning for Secure Model Training (Tehseen et al., 2021) introduced federated learning (FL) as a method for training ML models across distributed data sources without sharing raw data. It resolved concerns around privacy, latency, and data ownership in disaster prediction systems. Theoretical contribution: FL allows collaborative model training across decentralized systems while preserving data privacy[12-13]. Artificial Neural Network for Storm Surges (Kim et al., 2016) proposed an ANN-based model to forecast storm surge effects on coastal flooding in real-time. The model effectively captured short-term changes in water levels and demonstrated improved accuracy over traditional methods. Theoretical contribution: Neural networks can model nonlinear and complex environmental relationships better than rule-based systems[14].

The advent of machine learning (ML) has significantly advanced flood prediction systems by offering enhanced performance and cost-effective solutions. Hydrologists increasingly favor ML methods, seeking more accurate and efficient prediction models through novel ML techniques and hybridization of existing ones [15-16]. However, ML's dependency on extensive data for model training poses challenges, as concerns related to data privacy, security, and regulatory restrictions hinder data sharing among authorities [17-18].Traditionally, flood forecasting systems have employed centralized setups, concentrating both the prediction model and data in a single location for training before dissemination to all clients. Despite its convenience, this approach introduces latency, connectivity issues, and potential security and privacy risks [19-20].

Summary Table

Author	Year	Method	Dataset	Positives	Negatives
Patro et al.	2009	Hydrodynamic Modeling	Large river system data (India)	Accurate modeling of physical river flow	Requires physical parameters which may be unavailable
Rahman & Shaw	2015	Causal Flood Analysis	Hindu Kush flood event data	Highlights human and climate-related causes	Lacks predictive modeling approach
Kim et al.	2016	ANN (Artificial Neural Network)	Coastal storm surge data (Japan)	Effective real-time storm surge prediction	Limited to storm surges, not general flooding
Gude et al.	2020	Deep Learning	Historical flood records with weather data	Reduces uncertainty with high accuracy	Needs large training data and computation
Tehseen et al.	2021	Federated Learning	Decentralized environmental sensor data	Ensures data privacy and decentralized learning	Depends on network communication efficiency

3. PROBLEM STATEMENT

Nowadays, machine learning or deep learning algorithms are dependent on dataset for training a model and this dataset has to upload to centralized server from local machines through internet and this data uploading may take huge network delay or latency for upload and this data will get exposed to centralized server and data security will be breached and due to network latency we may see delay in response also. In some natural disaster scenarios like earthquake, floods, storm we need to have quick predicted response so government or peoples can take necessary action on time.

4. EXISTING SYSTEM

We know that machines or deep learning algorithms rely on data sets for teaching models, this data set should be loaded on the centralized server of local computers through the Internet, and this data download may require a huge download or delay, which is exposed to the centralized server and delayed network delayed. You can also see a response delay. In some scenarios of natural disasters, such as storms, such as storms that need to have a quick predictable answer for earthquakes, floods, governments and people to take necessary measures for the time.

Disadvantages:

- Network latency delays predictions.
- Data security risks arise.
- Delayed quick disaster responses.
- Dependency on quality data.
- Limited real-time response.

5. PROPOSED SYSTEM

In propose work author applying Federated Learning for flood forecasting which allow local machines to train a model on local data and then upload only trained model to centralized server for global training and this technique avoid dataset upload which remove all existing barriers such as Latency, data breached and security.

Local machine or centralized servers just have to take test data for prediction so network response and prediction will be quick.Federated learning (FL) is incorporated into flood forecasting in the proposed framework to moderate worries about information security and upgrade prediction accuracy. FL keeps up with information classification while empowering neighborhood associations to make models with their information by decentralizing model training.

The total Flood Forecasting Model (FFM), which can expect flood events with expanded accuracy and lead time, is then made by conglomerating these confined models. The framework takes utilization of FL's ability to deal with an assortment of datasets from different geological regions, representing differences in provincial hydrological conditions and natural factors. This strategy works on the precision of flood gauges as well as permits proactive ways to deal with catastrophe the executives that are redone for specific locales.

Advantages:

- Local data training efficiency.
- Data privacy maintained.
- Reduced network latency.
- Enhanced data security.
- Faster prediction responses.

6. SYSTEM ARCHITECTURE

The flood forecasting system configuration utilizes an organized information and model evaluation process. Flood information is preprocessed first. The dataset contains training and test sets for model structure and validation. A Feed Forward Neural Network (FFNN) and 2D Convolutional Neural Network (CNN2D) are prepared to deal with fluctuated flood information spatial and worldly properties. Below mentioned diagram is an outline of the proposed architecture:

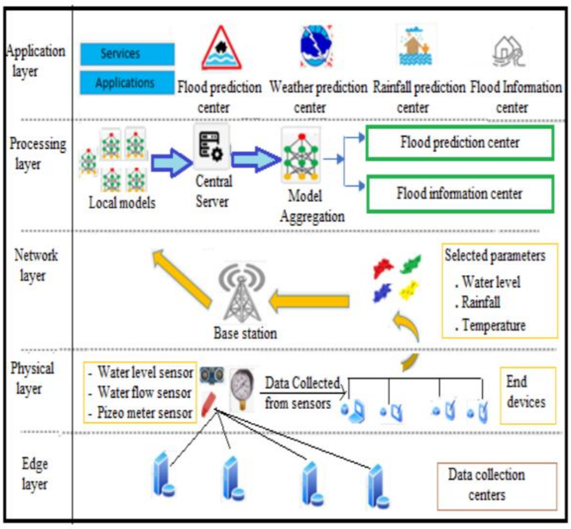


Fig -1: System Architecture

This architecture aims to enhance the accuracy of flood predictions by considering multiple hydrological factors and their complex interactions.

7. METHODOLOGY

Step 1: Dataset and Preprocessing
The project begins with the collection and use of a flood dataset. Although the original dataset used by the author was not available online, the Kerala flood dataset from Kaggle was used as a substitute. This dataset includes parameters such as monthly rainfall and corresponding water levels.
The preprocessing module handles missing values, normalizes data, and shuffles the entries to ensure that the model is trained on a balanced and clean dataset. Preprocessing is a vital step in preparing the data for better model performance.

Step 2: Training and Testing Split
Once preprocessed, the dataset is split into training (80%) and testing (20%) sets. The training data is used to build and train the machine learning models, while the test data is used to evaluate their performance in predicting future water levels.

Step 3: Feed Forward Neural Network (FFNN)
The core predictive model used in the proposed methodology is the Feed Forward Neural Network (FFNN). This model processes the input features through multiple layers and adjusts weights based on training performance. FFNN is selected for its simplicity and effectiveness in modeling time-based predictions such as water level trends. The model is trained using learning rate and multiple epochs, and it selects the best weights based on minimum error and highest prediction accuracy.

Step 4: Federated Learning Framework
In this setup, 18 local stations (representing different river locations) train their FFNN models independently using their local datasets. Instead of uploading the entire dataset to a central server, only the trained models are transmitted. The central server aggregates these models to create a stronger global model, reducing latency and enhancing privacy.
This federated approach allows predictions to be made locally and quickly. It also supports timely alerts to authorities with a lead time of 5 days, helping in early preparedness and response.

Step 5: Extension Using CNN2D
To improve prediction accuracy, the methodology also explores an extension model using Convolutional Neural Network 2D (CNN2D). CNN2D is known for capturing spatial patterns more effectively. When compared to FFNN, CNN2D showed higher accuracy and lower error rates (MSE and RMSE), proving to be a better alternative.

Step 6: Final Prediction and Deployment
After model training, the best-performing models (FFNN or CNN2D) are uploaded to the centralized server. Using test data, the models predict future water levels, which helps generate flood alerts. The complete application is designed as a Windows-based GUI system, enabling easy model upload and testing functionalities.

Step 7: Performance Evaluation
The system includes an accuracy comparison module that graphically displays the performance of both algorithms. The CNN2D model, due to its deep learning structure, demonstrates higher accuracy and lower error values compared to the traditional FFNN.

8. MODULES

Propose work consists of following modules

- The first step is onsite training and transmission of local data models using regional datasets towards central server for model aggregation.
- The next step, global model is trained based on local modes that calculates multiple parameters and predicts the client station where flood is about to happen with 5 days lead time.
- In the last step, local feed forward neural network (FFNN) model is trained on that specific client station to calculate expected water level and inform authorities for taking necessary actions regarding flood preparedness, mitigation and recovery.

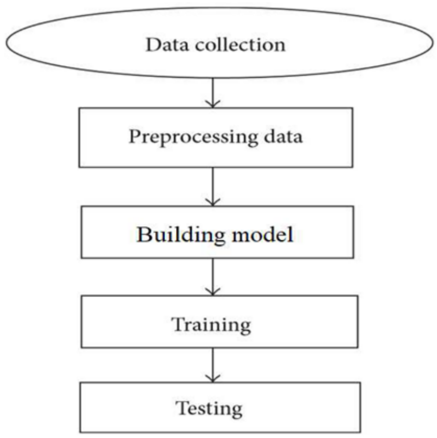


Fig -2: Module Implementation

In propose work author using 18 stations or rivers dataset to train FFNN algorithm locally and then report trained model to centralized server for global updates. Author has not published dataset on internet so we are using KERALA flood dataset from KAGGLE website. In below screen we are showing dataset details.

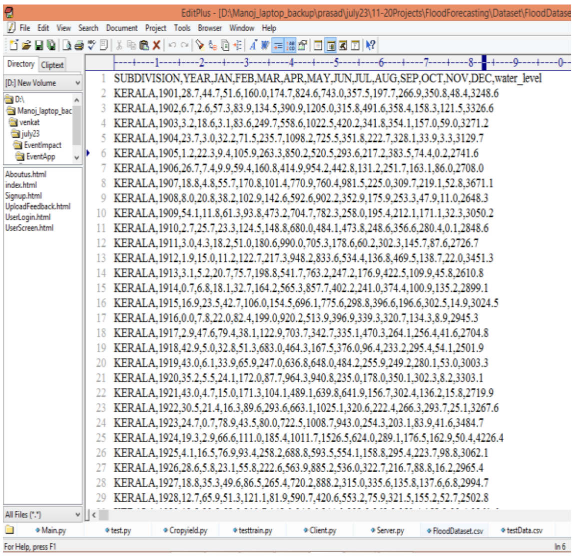


Fig -3: Data Set

In above dataset screen first row represents dataset column names and remaining rows represents dataset values where dataset has recordings of monthly rainfall and last column contains Water Level and based on predicted water level authorities will inform citizens about flood.

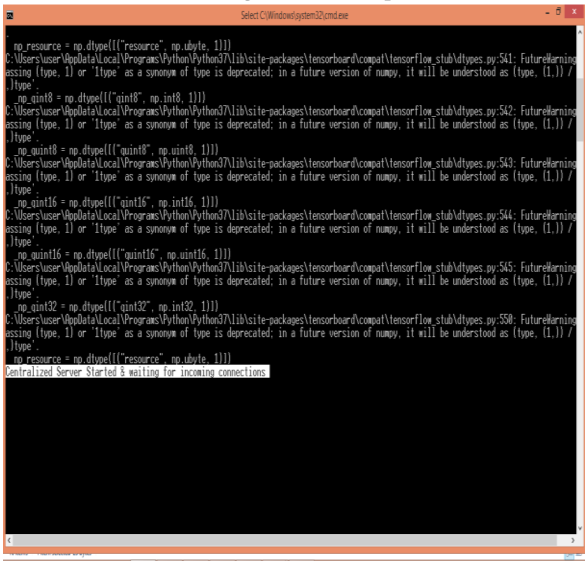
We have designed this application as Window based project as this project has to upload trained model to centralized and JUPYTER will not give flexibility of model upload to server so we designed as window based application.

To implement this project we have designed following modules

- Upload Flood Dataset: using this module we will upload, read and display dataset to application
- Pre-process Dataset: using this module we will remove missing values, normalized and shuffle the dataset values.
- Train & Test Split: used to split dataset into train and test where application using 80% dataset for training and 20% for testing.
- Run Feed Forward Neural Network: this module used to trained FFNN algorithm by using train data as input and this trained model can be applied on test data to calculate prediction accuracy.
- Run Extension CNN2D Algorithm: this module used to trained CNN2D algorithm by using train data as input and this trained model can be applied on test data to calculate prediction accuracy.
- Upload Federated Model to Server: using this module locally trained models can be upload to centralized servers for global updates.
- Accuracy Comparison Graph: can be used to plot comparison graph between propose FFNN and extension CNN2D.
- Flood Forecasting using Test Data: can be used to upload test data and then extension model will predict water level which help in knowing flood conditions.

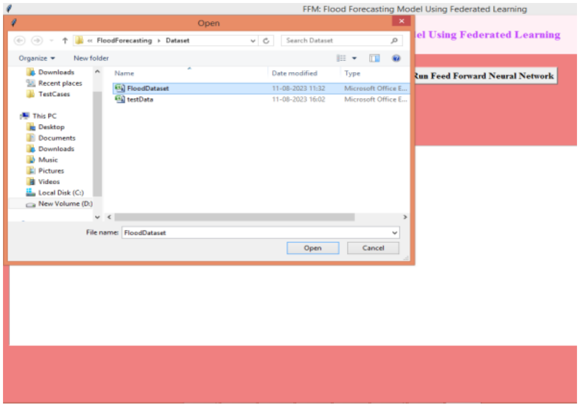
9. RESULTS

1. First double click on 'runServer.bat' file to start centralized server and get below output.

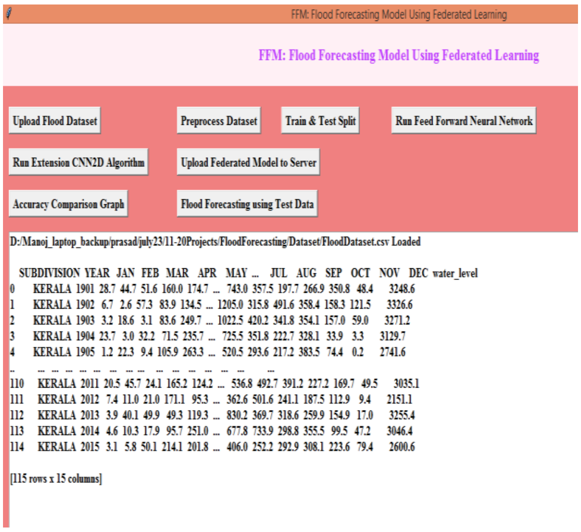


2. In above screen Centralized server started and now let it run and then double click on 'run.bat' file to start client which will train model locally by uploading local dataset and get below output.

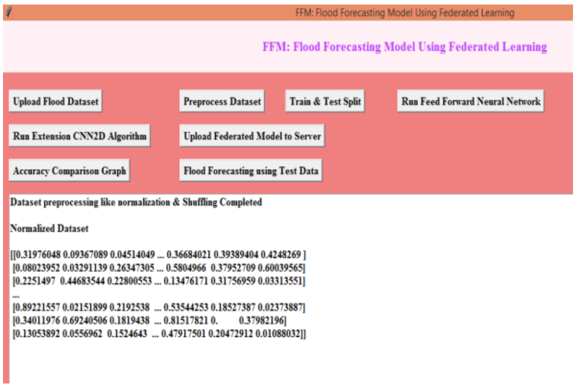
3. In above screen click on ‘Upload Flood Dataset’ button to load dataset and get below screen.



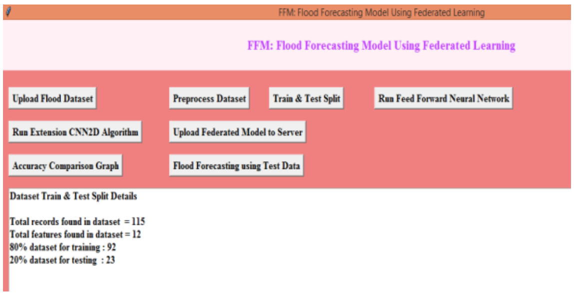
4. In above screen selecting and uploading ‘Flood Dataset’ and then click on ‘Open’ button to load dataset.



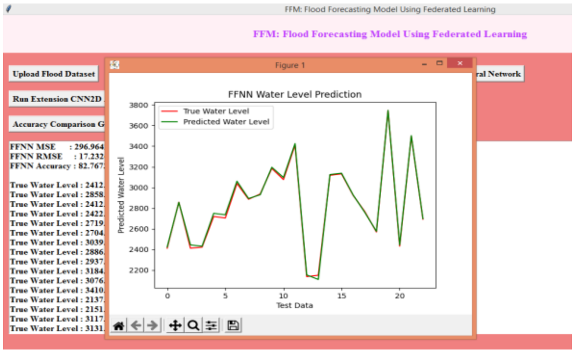
5. In above screen dataset loaded and now click on ‘Pre-process Dataset’ button to process dataset and get below output.



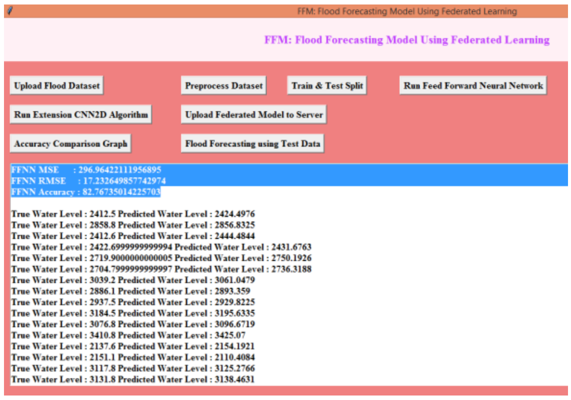
6. In above screen dataset pre-processing such as normalization and shuffling completed and now click on ‘Train & Test Split’ button to split dataset and get below output.



7. In above screen displaying dataset size and then displaying train and test size and now click on ‘Run Feed Forward Neural Network’ button to train propose FFNN algorithm and get below output.

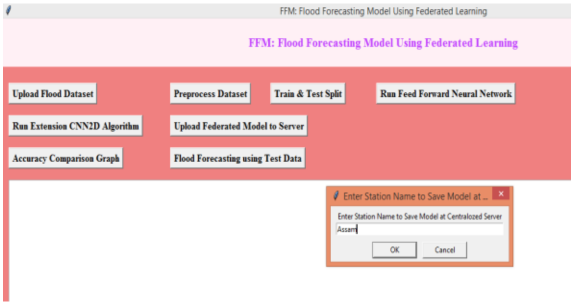


8. In above screen FFNN training completed and in above graph x-axis represents Number of Days and y-axis represents Water level where red line represents True water level and green line represents Predicted water level and we can see both lines are fully overlapping with little gap so we can say predicted and true values are very close and FFNN giving best prediction and now close above graph to get below output.



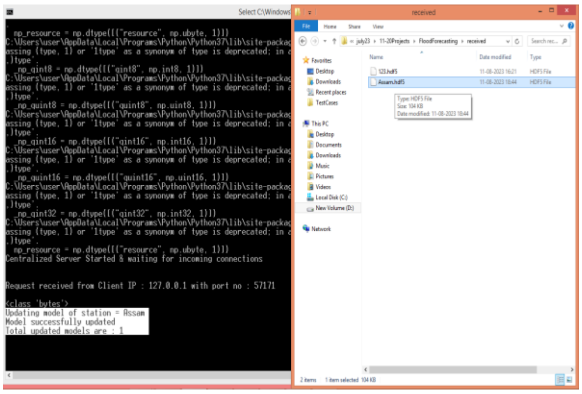
9. In above screen in first 3 lines we can see FFNN algorithm MSE, RMSE and accuracy values and then we can see true and predicted water levels for future days and now click on ‘Run Extension CNN2D Algorithm’ button to train extension algorithm.

10. In above screen with extension we can see both predicted and true which means reads and green lines are fully overlapping so we can say extension model is better than propose and we can see MSE and RMSE also lower compare to propose and accuracy is high for extension algorithm and now close above graph and then click on 'Upload Federated Model to Server' button to upload trained model to server and get below output.



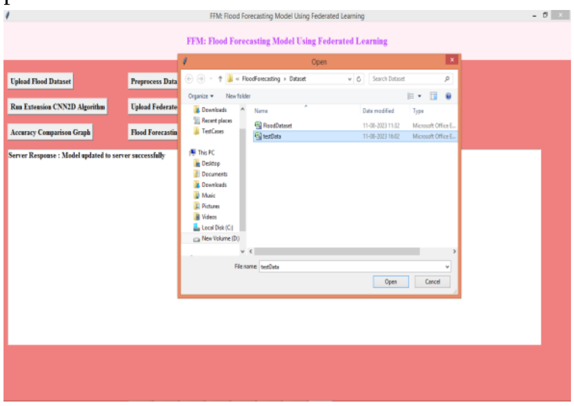
11. In above screen just enter some station name and then click OK button to upload model to server and get below output.

12. In above screen we got response from server as 'model uploaded' and in below server screen we can see received model details.

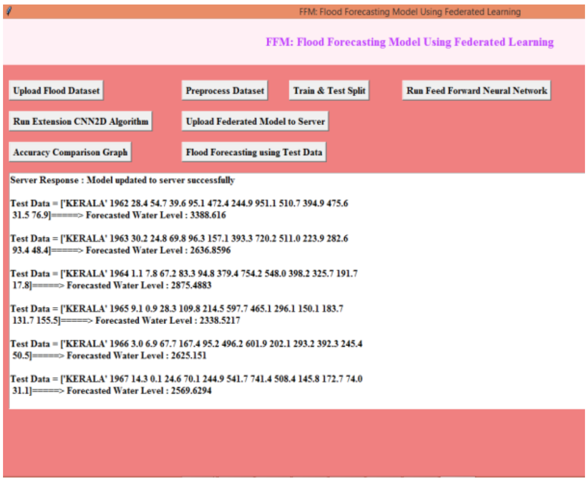


13. In above screen in white colour text we can see server output about model saving and in server 'received' folder we can see 'Assam' model is saved and similarly for all given station server will saved model.

14. In above comparison graph x-axis represents algorithm names and y-axis represents accuracy and MSE values and we can see for extension algorithm accuracy is high and MSE, RMSE error is lower compare to propose FFNN algorithm and now close above graph and then click on 'Flood Forecasting using Test Data' button to upload test and then predict water level.



In above screen uploading test data and then click on 'Open' button to get below output



In above screen before arrow symbol we can see test data and after arrow symbol =➔ we can see predicted water level.

10 . CONCLUSIONS

This proposed Flood Forecasting Model (FFM) successfully integrates Federated Learning with machine learning techniques to provide accurate and timely flood predictions. By training models locally at 18 stations and sharing only trained models with a central server, the system ensures data privacy, reduces network latency, and improves scalability. The use of Feed Forward Neural Network (FFNN) provides reliable predictions, while the extended CNN2D model significantly enhances accuracy. The system can forecast floods with up to 5 days lead time, giving authorities ample time to act and minimize disaster impact. This decentralized, privacy-preserving approach offers a powerful solution for modern flood management and can be extended for other environmental monitoring applications.

11. FUTURE ENHANCEMENT

In propose work author has used traditional Feed Forward neural network algorithms and did not used any advanced algorithms like Convolution 2D Neural Network which gain popularity in all domains for its accurate and successful prediction accuracy of more than 90%. So to enhance accuracy we have used CNN2D as extension for flood forecasting.

In our project the following things can be implemented in future.

- ➢ To implement Federated Learning for advanced algorithms like Convolution 2D Neural Network.
- ➢ To integrate Adaptive Power Control with AI.

12. REFERENCES

[1] Floods World Health Organization. Accessed: Feb. 3, 2022.[Online]. Available: https://www.who.int/news-room/questions-andanswers/item/how-do-i-protect-my-health-in-a-flood

[2] S. Patro, C. Chatterjee, R. Singh, and N. S. Raghuwanshi, ''Hydrodynamic modelling of a large flood-prone river system in India with limited data,'' Hydrol. Processes, vol. 23, no. 19, pp. 2774–2791, 2009.

[3] A. Rahman and R. Shaw, ''Floods in the Hindu Kush region: Causes and socio-economic aspects,'' in Mountain Hazards and Disaster Risk Reduction. Tokyo, Japan: Springer, 2015, pp. 33–52.

[4] A. Rahman and A. N. Khan, ''Analysis of 2010-flood causes, nature and magnitude in the Khyber Pakhtunkhwa, Pakistan,'' Natural Hazards, vol. 66, no. 2, pp. 887–904, 2013.

[5] G. Terti, I. Ruin, S. Anquetin, and J. J. Gourley, ''Dynamic vulnerability factors for impact-based flash flood prediction,'' Natural Hazards, vol. 79, no. 3, pp. 1481–1497, Dec. 2015.

[6] Annu. Flood Rep., 2022. Accessed: Jun. 13, 2022. [Online].Available:https://ffc.gov.pk/wp-content/ uploads/2021/04/2020-Annual-Report-ofOo-CEA-CFFC.pdf

[7] L. Martinez, ''Innovative techniques in the context of actions for flood risk management: A review,'' Engineering, vol. 2, no. 1, pp. 1–11, 2020.

[8] N. W. Chan and D. J. Parker, ''Response to dynamic flood hazard factors in peninsular Malaysia,'' Geograph. J., vol. 162, no. 3, pp. 313–325, 1996.

[9] S. Niknam, H. S. Dhillon, and J. H. Reed, ''Federated learning for wireless communications: Motivation, opportunities, and challenges,'' IEEE Commun. Mag., vol. 58, no. 6, pp. 46–51, Jun. 2020.

[10] R. Sadiq, M. A. Al-Zahrani, A. K. Sheikh, T. Husain, and S. Farooq, ''Performance evaluation of slow sand filters using fuzzy rule-based modelling,'' Environ. Model. Softw., vol. 19, no. 5, pp. 507–515, May 2004.

[11] S. Sankaranarayanan, M. Prabhakar, S. Satish, P. Jain, A. Ramprasad, and A. Krishnan, ''Flood prediction based on weather parameters using deep learning,'' J. Water

Climate Change, vol. 11, no. 4, pp. 1766–1783, Dec. 2020.
[12] K. Aziz, A. Rahman, G. Fang, and S. Shrestha, ‘‘Application of artificial neural networks in regional flood frequency analysis: A case study for Australia,’’ Stochastic Environ. Res. Risk Assessment, vol. 28, no. 3, pp. 541–554, Mar. 2014.
[13] K. Haddad and A. Rahman, ‘‘Regional flood frequency analysis in eastern Australia: Bayesian GLS regression-based methods within fixed region and ROI framework—Quantile regression vs. parameter regression technique,’’ J. Hydrol., vol. 430, pp. 142–161, Apr. 2012.
[14] S. Kim, Y. Matsumi, S. Pan, and H. Mase, ‘‘A real-time forecast model using artificial neural network for after-runner storm surges on the Tottori coast, Japan,’’ Ocean Eng., vol. 122, pp. 44–53, Aug. 2016.
[15] H. R. Pourghasemi, S. Pouyan, M. Bordbar, F. Golkar, and J. J. Clague, ‘‘Flood, landslides, forest fire, and earthquake susceptibility maps using machine learning techniques and their combination,’’ Natural Hazards, vol. 115, no. 3, pp. 1–20, Feb. 2023.
[16] B. Cai and Y. Yu, ‘‘Flood forecasting in urban reservoir using hybrid recurrent neural network,’’ Urban Climate, vol. 42, Mar. 2022, Art. no. 101086.
[17] R. P. Romansky and I. S. Noninska, ‘‘Challenges of the digital age for privacy and personal data protection,’’ Math. Biosci. Eng., vol. 17, no. 5, pp. 5288–5303, 2020.
[18] B. Dash, P. Sharma, and A. Ali, ‘‘Federated learning for privacypreserving: A review of PII data analysis in fintech,’’ Int. J. Softw. Eng. Appl., vol. 13, no. 4, pp. 1–13, Jul. 2022.
[19] M. Ma, C. Liu, G. Zhao, H. Xie, P. Jia, D. Wang, H. Wang, and Y. Hong, ‘‘Flash flood risk analysis based on machine learning techniques in the Yunnan province, China,’’ Remote Sens., vol. 11, no. 2, p. 170, 2019.
[20] F. Lai, Y. Dai, S. Singapuram, J. Liu, X. Zhu, H. Madhyastha, and M. Chowdhury, ‘‘FedScale: Benchmarking model and system performance of federated learning at scale,’’ in Proc. Int. Conf. Mach. Learn., Jun. 2022, pp. 11814–11827.

BIOGRAPHIES

P Ramya,
234M1D5812,
Dept. of CSE,VEMU IT College,
P Kothakota, Andhra Pradesh.
ramyatheja2001@gmail.com.

Dr.K Venkataramana,
Professor,
Dept. of CSE,VEMU IT College,
P Kothakota, Andhra Pradesh.
Venkatkathi84@gmail.com.

V Sridhar,
Assistant Professor,
Dept. of CSE,VEMU IT College,
P Kothakota, Andhra Pradesh.
Sreedharv1234@gmail.com.