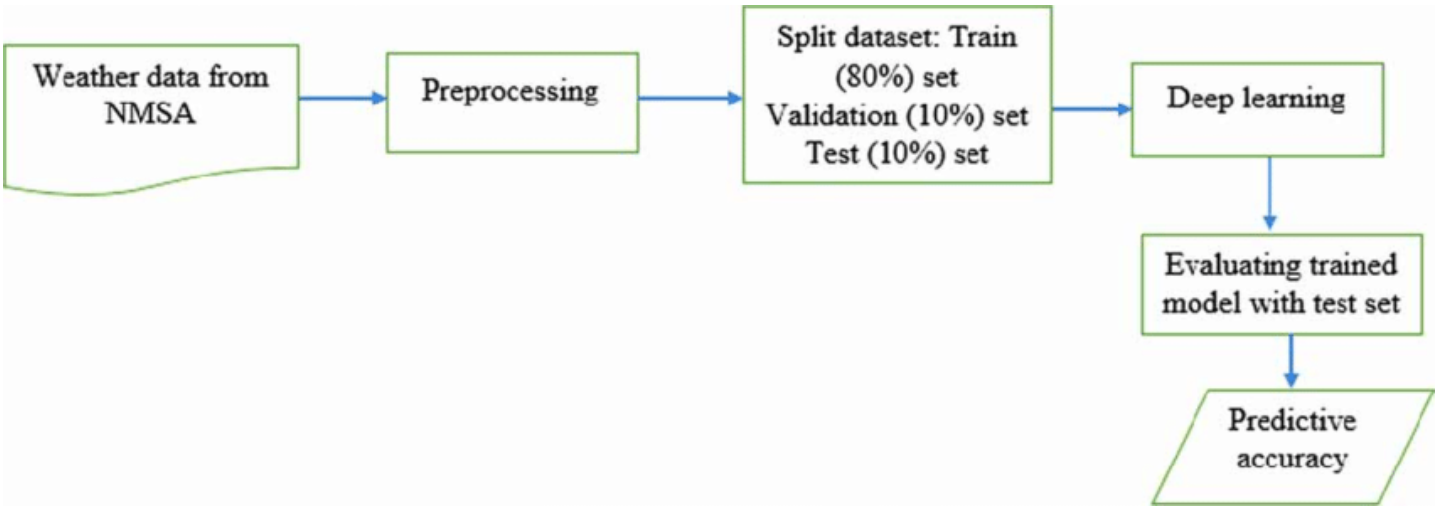
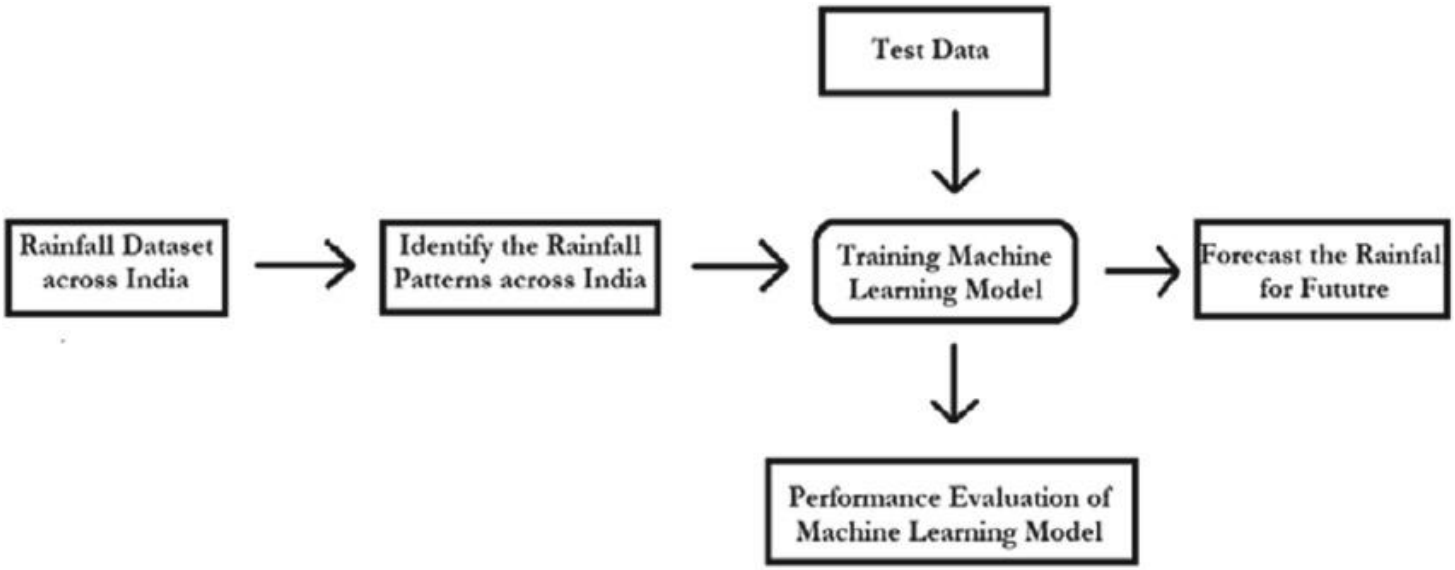
Project Design Phase-II Technology Stack (Architecture & Stack)

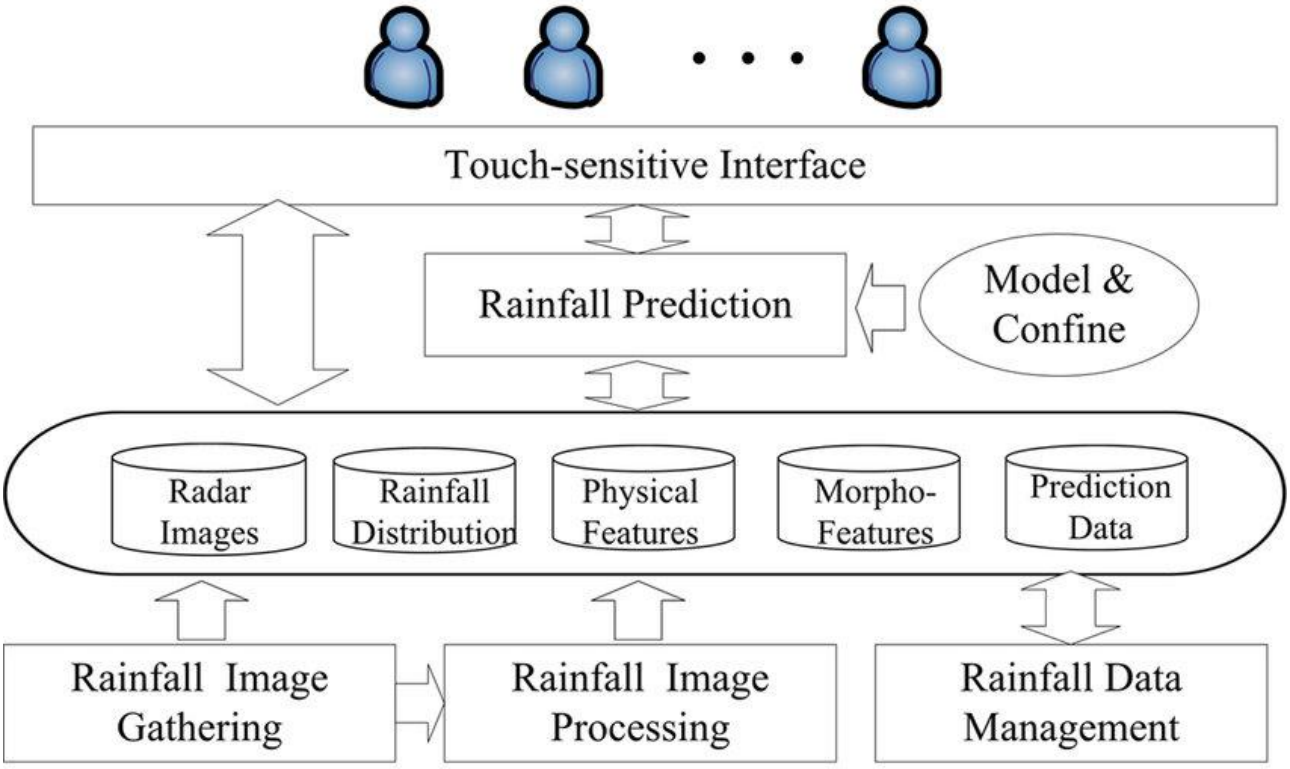
Date	1 November 2023
Team ID	592670
Project Name	Machine Learning Approach for Predicting the Rainfall
Maximum Marks	4 Marks

Technical Architecture

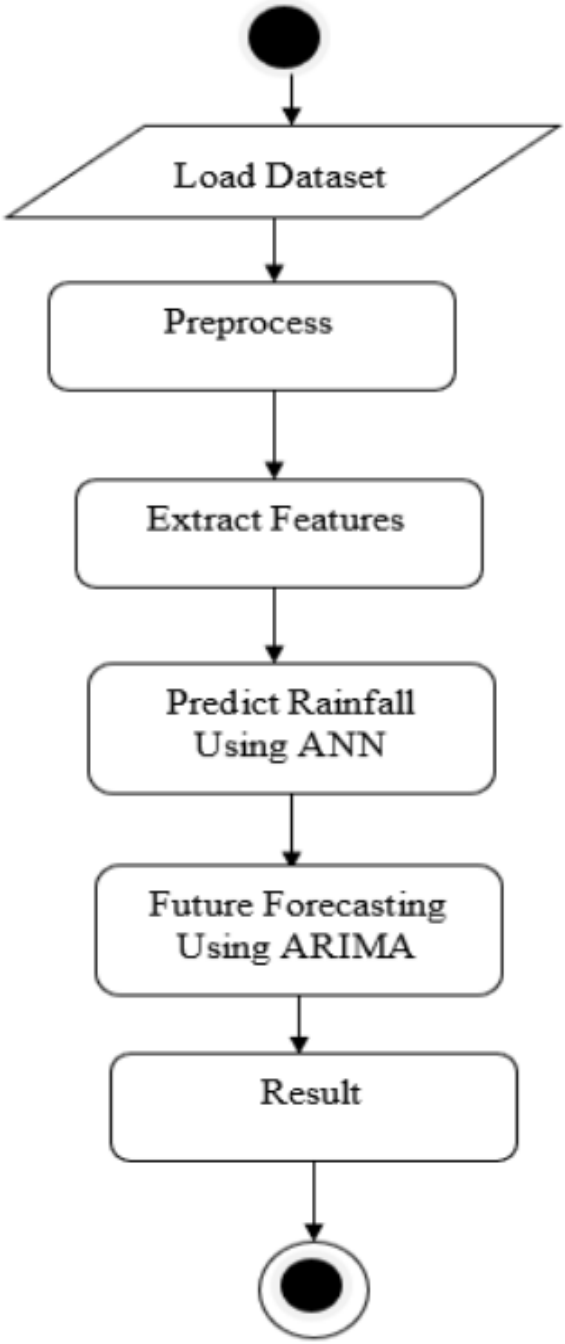
- Architecture of the proposed model of rainfall prediction.



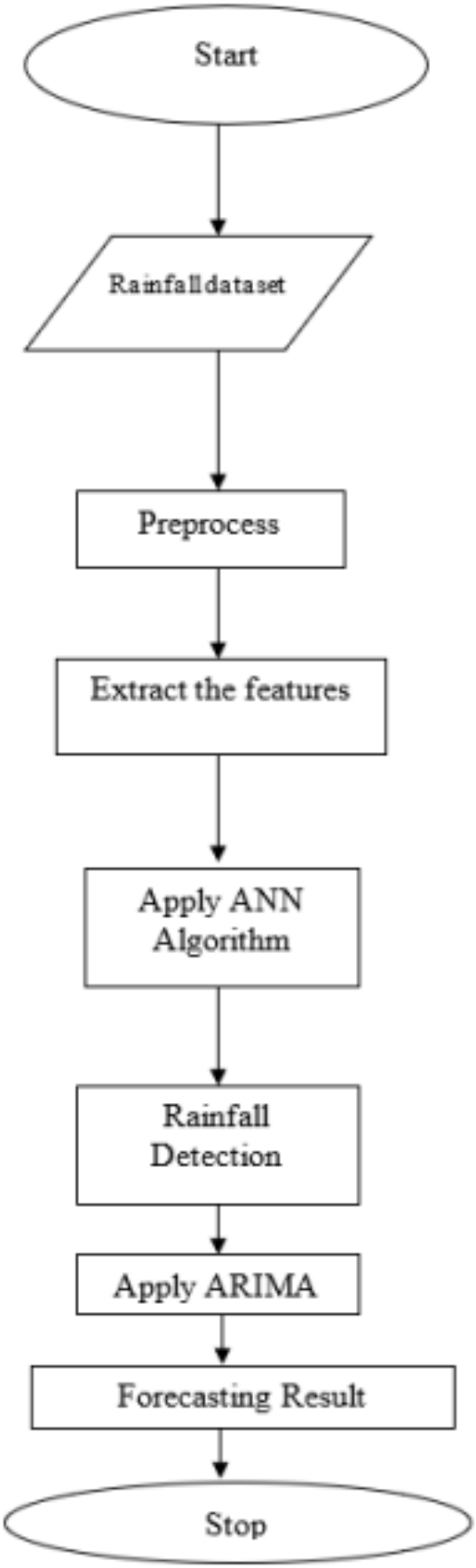



Touch-sensitive Interface
Rainfall Prediction
Model & Confine
Radar Images
Rainfall Distribution
Physical Features
Morpho-Features
Prediction Data
Rainfall Image Gathering
Rainfall Image Processing
Rainfall Data Management

- Activity Diagram



- Flow Chart


Start
Rainfall dataset
Preprocess
Extract the features
Apply ANN Algorithm
Rainfall Detection
Apply ARIMA
Forecasting Result
Stop

- Architecture outline of the classification of rain fall prediction

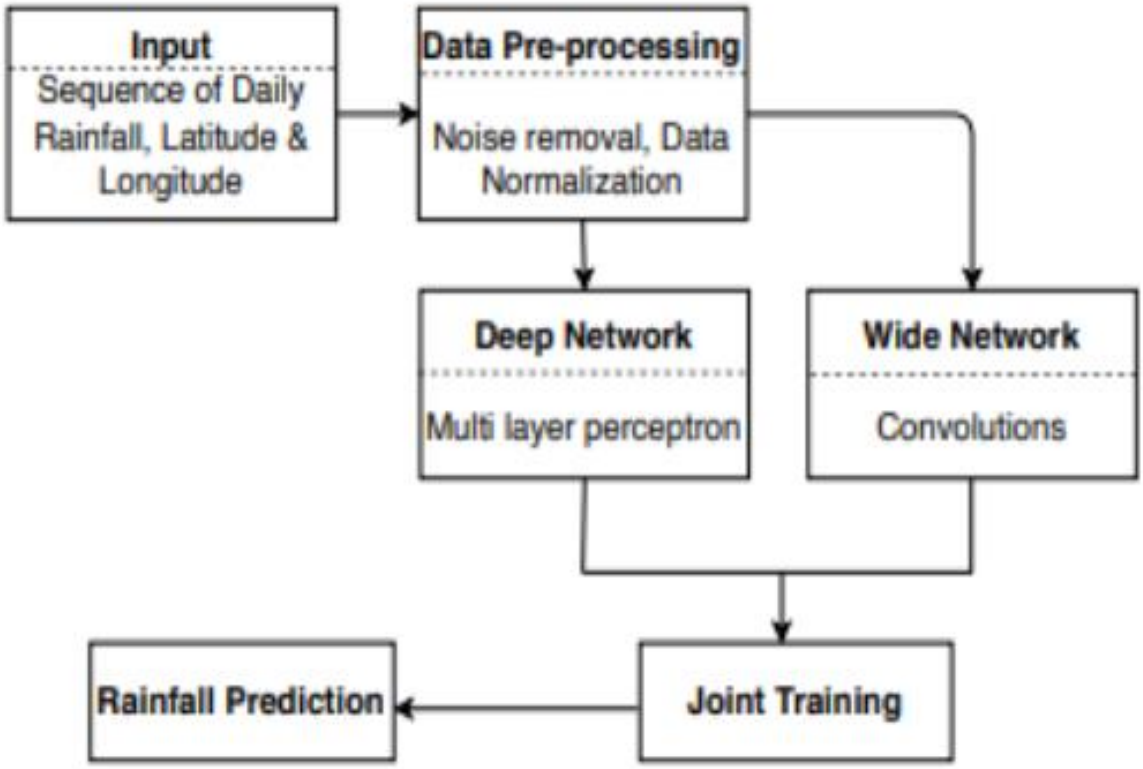


Table-1 : Components & Technologies:

Sr.No.	Component	Description	Technology
1.	User Interface:	• Used to create interactive web-based dashboards for users to access weather forecasts. • Utilize technologies to develop native apps with user-friendly interfaces.	• Web Application: HTML, CSS, JavaScript, and front-end frameworks (e.g., React, Angular, Vue.js) • Mobile App Development: Swift (for iOS) or Kotlin (for Android)
2.	Application Logic-1:	• Used to implement the application logic, including data processing, APIs, and business rules.	• Backend Frameworks: • Node.js, Django, Ruby on Rails, or Flask
3.	Database:	• Can store structured data, such as historical weather data and user profiles. • For unstructured or semi-structured data	• Relational Databases: Technologies like PostgreSQL, MySQL, or Microsoft SQL Server • NoSQL Databases: MongoDB, Cassandra, or Redis.
4.	File Storage/Data:	• Ideal for storing large datasets, model checkpoints, and other files. • Handle vast amounts of data.	• Cloud Storage Services: Technologies like Amazon S3, Google Cloud Storage, or Azure Blob Storage. • Distributed File Systems: Hadoop HDFS or distributed file systems like GlusterFS
5.	Framework:	• Used to develop and train machine learning models, including deep learning models. • Can help in creating APIs and integrating machine learning models with the application.	• Machine Learning Frameworks: Tools like TensorFlow, PyTorch, or scikit-learn • Web Frameworks: Backend web frameworks such as Express.js, Flask, or Ruby on Rails
6.	Deep Learning Model:	• TensorFlow and Keras: These libraries are often used to build, train, and deploy deep learning models for tasks like image recognition, which can be applied to cloud imagery. • PyTorch: PyTorch is another popular deep learning framework known for its flexibility and dynamic computation graph capabilities.	• CNN, Transfer Learning
7.	Infrastructure (Server/Cloud):	• Scalable cloud infrastructure for hosting applications, databases, and machine learning workloads. • Used to containerize and manage machine learning models and applications for efficient scaling and deployment.	• Cloud Computing Platforms: Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure • Containerization and Orchestration: Docker and Kubernetes

Table-2: Application Characteristics:

Sr. No.	Component	Description	Technology
1.	Open-Source Frameworks	• Used for building deep learning models for rainfall prediction. • Flexibility and dynamic computation graph capabilities. • Library for Python that provides tools for data preprocessing, feature selection, and model evaluation.	• TensorFlow, PyTorch, Scikit-learn
2.	Security Implementations:	• used to secure data transmission between clients and servers, ensuring data privacy and integrity. • to control access to your application and data. • Use encryption algorithms and libraries to protect sensitive data at rest and in transit.	• Secure Sockets Layer (SSL) and Transport Layer Security (TLS) • OAuth 2.0, JSON Web Tokens (JWT), or API keys • Encryption
3.	Scalable Architecture:	• Design your application using a microservices architecture, dividing it into smaller, independent services that can be deployed and scaled separately. • Used to create lightweight, portable containers for applications, making it easier to scale and deploy them across various environments. • Allows for the automated deployment, scaling, and management of containerized applications.	• Microservices, • Containerization like Docker, • Kubernetes
4.	Availability:	• Use load balancers to distribute traffic evenly across multiple server instances, ensuring high availability. • Implement redundancy by replicating your application components and use failover mechanisms to ensure continuous service availability in case of server failures. • used to cache and deliver content from edge locations, reducing latency and improving availability.	• AWS Elastic Load Balancing, • NGINX, Cloudflare, Akamai
5.	Performance:	• Implement caching mechanisms using technologies to store frequently accessed data, reducing the load on your database and improving response times. • Compression techniques to reduce the size of data sent over the network, improving data transfer and application load times. • Content delivery technologies can help distribute content closer to end-users, reducing latency and enhancing performance.	• Redis, Memcached, • GZIP, • CDNs and edge computing