RAINFALL PREDICTION USING NEURAL NETWORK ENSEMBLE BASED APPROACH

A PROJECT REPORT

for

SOFT COMPUTING (ITE1011)

in

B.Tech (IT)

by

MOHIT GUPTA (19BIT0055)

SUYASH GUPTA (19BIT0073)

HRITIK DUBEY (19BIT0150)

4th Semester, 2nd Year

Under the Guidance of

Prof. L.AGILANDEESWARI

Associate Professor, SITE

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology and Engineering

MAY, 2021

<u>DECLARATION BY THE CANDIDATE</u>

We hereby declare that the project report entitled **"Rainfall Prediction using Neural Network"** submitted by us to Vellore Institute of Technology University, Vellore in partial fulfillment of the requirement for the award of the course **Soft Computing (ITE1011)** is a record of bonafide project work carried out by us under the guidance of **Prof. Agilandeeswari L.** We further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other course.

Place : Vellore   Signature

Date :

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology & Engineering [SITE]

CERTIFICATE

This is to certify that the project report entitled **"Rainfall Prediction using Neural Network"** submitted by **Mohit Gupta (19BIT0055) Suyash Gupta (19BIT0073) Hritik Dubey (19BIT0150)** to Vellore Institute of Technology University, Vellore in partial fulfillment of the requirement for the award of the course **Soft Computing (ITE1011)** is a record of bonafide work carried out by them under my guidance.

Prof. Agilandeeswari L
GUIDE
Asso. Professor, SITE

RAINFALL PREDICTION USING ENSEMBLE BASED APPROACH

Mohit Gupta[1], Suyash Rohit Gupta[2], Hritik Dubey[3]

Department of Information Technology, VIT University, Vellore, Tamil Nadu, India

Abstract

Rainfall is one of the major factors contributing to environmental changes, and can play a major role in influencing ecosystems, as well as the geography, and topology of an area. Additionally, rain is a significant factor in the economic well being of agricultural countries like India. Hence, it is important to understand the factors that come into play for influencing rainfall in different areas. Comprehending these factors are crucial to making counteractive measures to prevent any drastic changes in rainfall patterns so as to have optimal crop yields and prevent damage to the ecosystem. Using prediction methods involving techniques such as Artificial Neural Networks and Self Organizing Map can lead to the creation of robust rainfall prediction models which can guess the future rainfall patterns with a good amount of accuracy.

I. INTRODUCTION

Rainfall pattern means the distribution of rain geographically, temporally, and seasonally. The tropics receive more rainfall than deserts. Cooler places receive no rainfall, as it is converted to snow before it falls to the ground. Rainfall happens more in a particular time of a year, during a rainy season. In other seasons, rainfall is scant. Therefore, agriculture, worldwide, is planned based on rainfall's natural pattern. Water reservoirs, irrigation networks, and urban water supply systems are designed based on the average annual rainfall. If it rains a lot on a continuous basis for a longer time, there is a possibility of flood and subsequent disaster to the infrastructure. No rainfall or little rainfall for a longer period in an inhabited area could lead to drought and famine.

There are two approaches to predict rainfall. They are Empirical methods and dynamical methods. The empirical approach is based on analysis of historical data of rainfall and its relationship to a variety of atmospheric and oceanic variables over different parts of the world. The most widely used empirical approaches used for climate prediction are regression, ANN, fuzzy logic and group method of data handling.

II. BACKGROUND

DIFFERENT MODELS OF RAINFALL PREDICTION:

ANN- Artificial Neural Network

An artificial neural network is the piece of a computing system designed to simulate the way the human brain analyzes and processes information. It is the foundation of artificial intelligence (AI) and solves problems that would prove impossible or difficult by human or statistical standards.

SVM - Support Vector Machine

Support Vector Machine is a supervised machine learning algorithm which can be used for both classification or regression challenges.Support Vectors are simply the coordinates of individual observation.

BPN-Back Propagation Network

Backpropagation is the central mechanism by which neural networks learn. It is the messenger telling the network whether or not the net made a mistake when it made a prediction.

SOM- Self-organizing map

A self-organizing map or self-organizing feature map (SOFM) is a type of artificial neural network (ANN) that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map, and is therefore a method to do dimensionality reduction.

Fuzzy Logic

Fuzzy Logic is a method of reasoning that resembles human reasoning. The approach of FL imitates the way of decision making in humans that involves all intermediate possibilities between digital values YES and NO.

CNN-Convolutional Neural Network

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other.

RBF-Radial basis function

A radial basis function is a function that assigns a real value to each input from its domain (it is a real-value function), and the value produced by the RBF is always an absolute value; i.e. it is a measure of distance and cannot be negative.

MLP-Multilayer perceptron

A multilayer perceptron is a class of feedforward artificial neural network (ANN).MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron.

ELM-Extreme learning machines

Extreme learning machines are feedforward neural networks for classification, regression, clustering, sparse approximation, compression and feature learning with a single layer or multiple layers of hidden nodes, where the parameters of hidden nodes need not be tuned

FFN- Feedforward neural network

A feedforward neural network is an artificial neural network wherein connections between the nodes do not form a cycle. As such, it is different from its descendant: recurrent neural networks.

ARIMA-AutoRegressive Integrated Moving Average

ARIMA is an acronym that stands for AutoRegressive Integrated Moving Average. This is one of the easiest and effective machine learning algorithm to performing time series forecasting

KNN- K-nearest neighbors

The k-nearest neighbors algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems. It's easy to implement and understand, but has a major drawback of becoming significantly slower as the size of that data in use grows.

MLR-Multiple linear regression

Multiple linear regression is a statistical technique. It can use several variables to predict the outcome of a different variable. The goal of multiple regression is to model the linear relationship between your independent variables and your dependent variable

Categorised based on Model used in combination with other models,Methodologies & techniques-

- **Artificial Neural Network**

  Secondary models used:

  - Multi Regression model
  - Autoregressive integrated moving average(ARIMA)
  - K-Nearest Neighbour
  - Extreme learning model
  - Convolutional Neural Networks
  - Support Vector Machines

- **Support Vector Machine**

- **Back Propagation Network**

  Secondary models used:

  - Feed Forward Network
  - Self Organizing Map
  - Support Vector Machines

- **Fuzzy Logic Network**

  Secondary models used:

  - Support Vector Machines
  - Self Organizing Map
  - Artificial Neural Network

- **Convolutional Neural Networks**
- **Multi layer Perceptron**

  Secondary models used:

  - Fuzzy
  - RBF
  - Artificial Neural Network

- **Extreme learning model**

III. Literature Survey

❖ Artificial Neural Network

Article Number	Authors	Methodology or Techniques used	Merits	Issues	Metrics
[12]	Neelam Mishra, Hemant Kumar Soni, Sanjiv Sharma, A K Upadhyay	Artificial Neural Network (ANN), Feed Forward Neural Network (FFNN)	141 years of rainfall data were analyzed in the survey using data from the Indian Meteorological department. Two models were developed based on prediction for one month and two months ahead in time respectively. The model M1 for one month showed better results, especially for ANN.	The study here is limited to the Northern part of India, and so it may not be applied to the whole nation. Some factors like topography, latitude, elevation, seasons, distance from sea etc. were not taken into consideration.	Mean Square Error **(MSE)** and Magnitude of Relative Error (**MRE**)
[16]	Kumar Abhishek, Abhay Kumar, Rajeev Ranjan, Sarthak Kumar	ANN (Artificial Neural Network) a) Feed Forward with Back –Propagation b) Layer Recurrent c) Cascaded feed Forward back Propagation, Back-propagation	• As the number of neurons increases in an ANN, the MSE decreases. • BPA is the best algorithm out of the three tested. • LEARNGDM is the best learning function to train your data with. • LEARNGD is a bit time	The final results on the actual rainfall in the selected area were not very well highlighted.	MSE(Mean Squared Error), SD(Standard Deviation)

		Algorithm(BPA),	consuming. • TRAINLM is the best training function. • A Multi-layer Algorithm is better than a Single-layer algorithm in terms of performance.		

[21]	Nikhil kumar B. Shardoor , Mandapati Venkateswar Rao	ANN,ML,data mining and satellitetechniquesOnce after preprocessing the data, the dataset is divided into 80% of dataset is used for training purpose and 20% of data of dataset is used for the testing the predictive build model	Once after successful validation of the build model i.e the model working efficiently with correct output then the model is deployed for the future application.The predictive model is build using the available rainfall dataset, mathematical equations and algorithms of data mining, machine learning and so on	As rainfall is a nonlinear in nature, it values are not constant, so statistical model yield poor inaccuracy in result	If (Evaluated rainfall ≥ (Avg. rainfall + 10% Avg. Rainfall)) then Status ("Above Normal")
					
					If (Evaluated Rainfall ≤ (Avg. rainfall - 10% Avg. Rainfall)) then Status ("Below Normal")

★ ANN + Multi Regression model

Article Number	Authors	Methodology or Techniques used	Merits	Issues	Metrics
[10]	Ahmed El-Shafie, Abdrabbo A. A. Shehata AbouKheira, Mohd Raihan Taha	ANN(Artificial Neural Networks) model, Multi Regression MLR model, Feed Forward Neural Network FFNN model	The problem was modeled differently based on Monthly and Yearly predictions each. The analyzed study cases suggest that, ANN provide better results than the MLR model regarding the statistical criteria used to make the comparison between ANN and MLR models	The results showed that the ANN model can be used to help with forecasters, however it can only supplement their task, however it cannot replace and take over the task.	Root Mean Square Error **RMSE**, Mean Absolute Error **MAE**, Coefficient Of Correlation CC and BIAS.

★ ANN+ MLR +ARIMA

[8]	Meysam Ghamariadyan , Monzur Alam Imteaz	WANN(Wavelet-Artificial Neural Network), ANN(Artificial Neural Network), ARIMA(Autoregressive integrated moving average), MLR(Multiple linear regression)	Abundant amount of data was used to train and verify the model. Data from 1908-1999 and 2000-2016 from 10 weather stations was used. The WANN was used to predict accuracy in terms of RMSE(root Mean	No visible issues could be identified.	**RMSE**

			square error). The model was used to predict upto 1,3,6 and 12 months in future.		

★ ANN+ K-Nearest Neighbour+Extreme learning machine

Article Number	Authors	Methodology or Techniques used	Merits	Issues	Metrics
[4]	Yajnaseni Dash, Saroj K.Mishra, Bijay K.Panigrahi	KNN(K-Nearest Neighbours), ANN(Artificial Neural Network), ELM(Extreme Learning machine)	Has potential for predicting both summer monsoon and as well as post-monsoon of Kerala with minimum prediction error scores.	Restricted to a particular area(Kerala), ELM provided better results with less error scores and hence KNN and ANN may be said useless.	Prediction and precision **Percentage error** **MSE(Mean squared error)** **W-weights vector**
[23]	K C Gouda, Libujashree R, Priyanka Kumari, Manisha Sharma, Ambili D Nair	Conventional Regression model and Artificial Neural Network, Back propagation algorithm using multilayered neural network, Linear regression and K-fold, KNN(K-Nearest Neighbour) Wavelet Decomposition	It was found from the survey that most of the researchers used back propagation algorithm and had substantial results.The output of the neural network trained using back propagation algorithm gives less error when compared to other methods.	Some limitations of those methods have been found in forecasting techniques that use Linear regression, K-fold, KNearest Neighbour, and Wavelet Decomposition	

★ ANN+ KNN+ELM+ Convolutional Neural Networks + Support Vector Machines

Article Number	Authors	Methodology or Techniques used	Merits	Issues	Metrics
[13]	Ali Haidar, Brijesh Verma	Convolutional Neural Networks (CNNs), Support Vector Machines (SVMs), Adaptive Network-Based Inference Systems (ANFISs), Extreme Learning Machine	The study found that the RMSE prop algorithm is the best for optimization. A deep CNN was developed to predict monthly rainfall. The developed model was compared to the first version of the Australian Community Climate and	The CNN model performed better specifically when months had higher average rainfall. Locations and datasets included were incredibly specific to one	Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Pearson correlation (r) and Nash Suttcliff coefficient of efficiency (NSE)

		(ELM), Regression Trees (RT) and K-Nearest Neighbors (KNNs)	Earth-System Simulator (ACCESS-S1) and a Multi-Layered Perceptron (MLP), where better performance was revealed with the proposed CNN model.	location.	

❖ Support Vector Machines

Article Number	Authors	Methodology or Techniques used	Merits	Issues	Metrics
[9]	Changhyun Choi, Jungwook Kim,Heechan Han,Daegun Han and Hung Soo Kim	ANN(Artificial Neural Networks), SVM(Support Vector Machines), DT(Decision Tree)	Significant training and testing periods were given for the model, between 2009-2013 and 2013 to 2015 each. Additionally, the water levels were measured every 20 minutes between 2011 and 2015. Wide array of ML techniques as well as testing metrics.	Backflow in the downstream of the Nakdong river was not taken into account. Scatter plot had difficulties in predicting peak values of rainfall.	correlation coefficient (**CC**), Nash–Sutcliffe efficiency (**NSE**), and root mean square error (**RMSE**)
[15]	Xiaobo Xhang, Sachi Nandan Mohanty, Ajaya Kumar Parida, Subhendu Kumar Pani, Bin Dong.	Regression, Support Vector Machines (SVM)	Tested the flexibility of the SVM in time series forecasting and compared it with a multi-layer back-propagation (BP) neural network. SVM outperformed the multi-layer back-propagation neural network	In future multiple or multivariate time series prediction can be used for more clarity of prediction, which was not implemented in this study.	MSE (mean squared error), correlation coefficient, coefficient of efficiency and MAE (mean absolute error)
[20]	Qinghua Miao , Baoxiang Pan, Hao Wang, Kuolin Hsu and Soroosh Sorooshian	CNN(Convolutional Neural Networks), SVM(Support Vector Machines), quantile mapping (QM)	proposed a new method for radar precipitation nowcasting by combining the CNNs with Long Short Term Memory Networks (LSTM). Among all the meteorological variables, the geopotential height might be the most important one. Considering both the accuracy and complexity of the model, the paper suggests that the combination of geopotential height and total water vapor might be reasonable.	No significant drawbacks.	RMSE, Nash and Sutcliffe coefficient (NSE) and relative bias (RB)

❖ Back propagation Network

Article Number	Authors	Methodology or Techniques used	Merits	Issues	Metrics
[18]	Akash D Dubey	ANN, Feed forward backpropagation.	There were a total 12 ANN models which were trained using the 800 data samples of the region which were collected over 100 years. After the training, the validation and testing of the neural networks were done using 200 data samples each.	Topography, Geography, and other such factors based on the location were not taken into account.	MSE(Mean Square Error)
[23]	K C Gouda, Libujashree R, Priyanka Kumari, Manisha Sharma, Ambili D Nair	Conventional Regression model and Artificial Neural Network, Back propagation algorithm using multilayered neural network, Linear regression and K-fold, KNN(K-Nearest Neighbour) Wavelet Decomposition	It was found from the survey that most of the researchers used back propagation algorithm and had substantial results.The output of the neural network trained using back propagation algorithm gives less error when compared to other methods.	Some limitations of those methods have been found in forecasting techniques that use Linear regression, K-fold, KNearest Neighbour, and Wavelet Decomposition	
[24]	S.renuga Devi, P.Arulmozhivarman, C.Venkatesh, Pranay Agarwal	BPN(Back Propagation Neural Network), CBPN(Cascade forward back propagation neural network), DTDNN (Distributed time delay neural network), NARX(Non linear autoregressive exogenous network).	These models were tested to the actual data set resulting in very accurate forecasting details. The study deals with two data sets (one containing daily rainfall, temperature, humidity data and the other containing daily rainfall data from nearby weather stations).	The model was used only to predict the rainfall data of the next day (only 1 day ahead). Some better neural network models could have been used to increase its efficiency.	**Levenberg Marquardt** - weight updating technique. **Sensitivity analysis** - identifying the key predictors.

★ BPN+Feed Forward Network

Article Number	Authors	Methodology or Techniques used	Merits	Issues	Metrics
[3]	Ankita Sharma, Geeta Nijhawan	Back Propagation algorithm, ANN(Artificial Neural Network), FFN(Feed Forward Network), Cascaded Back-propagation, Layer Recurrent network.	FFN connected in a cascade manner for a multilayer network. Actual and predicted value matches. Various algorithms tested and corresponding graphs were plotted which showed MSE (Mean squared error) and epochs. Increase in the number of neurons showed a decrease in MSE.	In FFN, information travels in only one direction(forward). Different networks resulted in different results which shows inconsistency.	NFTOOL (Neural network fitting tool), NNTOOL (Open network/Data manager) **M = Sum of entries/number of entries.** **Normalized value = (x-M)/SD**
[26]	Kumar Abhishek, Abhay Kumar, Sarthak Kumar, Rajeev Ranjan	Back-propagation algorithm, Back-propagation-feed forward neural network.	The result gathered from the model and the actual result matches . The error rate is very low. Built test cases to find the number of hidden neurons in the layers for best performance.	The models use a complex and complicated approach by adding three layers in the neural network. Different models used for different number of neurons.	**W-weight**, **B-bias** **MSE (Mean Squared error).**

★ BPN+FFN +RBF

Article Number	Authors	Methodology or Techniques used	Merits	Issues	Metrics
[10]	Ahmed El-Shafie, Abdrabbo A. A. Shehata AbouKheira, Mohd Raihan Taha	ANN(Artificial Neural Networks) model, Multi Regression MLR model, Feed Forward Neural Network FFNN model	The problem was modeled differently based on Monthly and Yearly predictions each. The analyzed study cases suggest that, ANN provide better results than the MLR model regarding the statistical criteria used to make the comparison between ANN and MLR models	The results showed that the ANN model can be used to help with forecasters, however it can only supplement their task, however it cannot replace and take over the task.	Root Mean Square Error **RMSE**, Mean Absolute Error **MAE**, Coefficient Of Correlation CC and BIAS.

★ BPN+Self Organizing Map

Article Number	Authors	Methodology or Techniques used	Merits	Issues	Metrics
[11]	Deepak Ranjan Nayak,Pranati Mishra,Amitav Mahapatra	ANN(Artificial Neural Networks), BPN(Back Propagation Networks), RBFN(Radial Basis Function Networks), SOM(Self Organizing Map)	The paper reports and summarizes a survey of different rainfall predictions using neural network architectures over 25 years. Back propagation networks for rainfall prediction got the most significant results. The forecasting techniques that use MLP, BPN, RBFN, SOM and SVM are suitable to predict rainfall.	Broad range of cases for literature survey with not enough focus put on any specific cases.	**CC, RMSE**
[14]	Deepak Ranjan Nayak, Amitav Mahapatra, Pranati Mishra	The proposed preprocessing techniques included moving average (MA) and singular spectrum analysis (SSA). The modular models were composed of local support vectors regression (SVR) models or/and local artificial neural networks (ANN) models	This paper reports a detailed survey on rainfall predictions using different neural network architectures over twenty-five years. The survey took into account and factored in back propagation for the prediction of rainfall.	Some limitations in methods such as SOM(Self Organizing Map) and SVM(Support Vector Machine) were found.	Root Mean Square Error, Mean Absolute Error, Coefficient Of Correlation and BIAS
[29]	R.Arya and Maya L Pai	Optimised Genetic-Artificial Neural Network Model, SOM Clustering Approach, BPNN model with hybrid BPNN- GA.	A hybrid combination of neuro-genetic model has been used for the weight optimization of network layers of ANN. By using the five meteorological parameters such as SST, SLP, Humidity, U-wind, V-wind of IO region, gives better average prediction accuracy for BPNN-GA approach than the FF-BPNN alon.	Despite the fact that GA and ANN have their own methodologies and their own benefits and drawbacks, now a day there are certain advanced methodologies to consolidate and to build a new Hybrid Frameworks called Neuro-Genetic Models	Root Mean Square Error **RMSE**,

★ BPN+SOM+RBF+SVM

Article Number	Authors	Methodology or Techniques used	Merits	Issues	Metrics
[17]	Mohini P. Darji, Vipul K. Dabhi, Harshadkumar B.Prajapati	MultiLayer Perceptron Neural Network (MLPNN), Back Propagation Algorithm (BPA), Radial Basis Function Network (RBFN), SOM (Self Organization Map) and SVM (Support Vector Machine)	Through this research, it was concluded that ANN is the best method as it can predict patterns that were not fed in the input. Back propagation is also most commonly used in most research. A comparison between methods is given: Multilayer Feed Forward Neural Network (MLFN), Recurrent Neural Network (RNN), Time Delay Neural Network (TDNN). TDNN performs better for forecasting of yearly rainfall data whereas time lag FFNN gives better performance for forecasting of weekly rainfall data	TDNN performs better for forecasting of yearly rainfall data whereas time lag FFNN gives better performance for forecasting of weekly rainfall data.	MSE, RMSE.

❖ Fuzzy Logic

Article Number	Authors	Methodology or Techniques used	Merits	Issues	Metrics
[22]	Ajith Abraham, Ninan Sajeeth Philip and P.K. Mahanti	Neuro-fuzzy system - a combination of ANN and Fuzzy Inference System (FIS)in such a way that neural network learning algorithms are used to determine the parameters of FIS	The considered connectionist models are very robust, capable of handling the noisy and approximate data that are typical in weather data, and therefore should be more reliable in worst situations	Network performance could have been further improved by providing more training data.	

★ Fuzzy + SVM + SOM

[5]	Dhawal Hirani, DR. Nitin Mishra	Self organizing map(SOM),Support Vector Machine (SVM) Fuzzy Logic, Weather Research & forecast model,General Data mining Rainfall prediction model,	Support Vector Machines (SVMs) Better generalization performance than other NN models. SOM reduces dimensions & displays similarities.	Very few researchers of this field used this technique for rainfall prediction and got satisfactory results.	**SVM Hyperplane-** f(x) = w x+ b(1) x-training pattern, **w-weight vector & b-bias term**

★ Fuzzy +SVM +ANN

[30]	Suvendra Kumar Jayasingh, Jibendu Kumar Mantri	The hybrid climate modeling systems will combine the features of Fuzzy logic, Decision Tree, Artificial Neural Network(ANN), Support Vector machine(SVM) etc	Hybrid of soft computing climate models is a new attempt to predict the weather with better accuracy. In this paper we have used the fuzzy rule based Decision Tree hybridized model of soft computing techniques which could give better result than the existing models	Some limitations in methods such as SVM(Support Vector Machine) were found.	

❖ Convolutional Neural Network

Article Number	Authors	Methodology or Techniques used	Merits	Issues	Metrics
[20]	Qinghua Miao , Baoxiang Pan, Hao Wang, Kuolin Hsu and Soroosh Sorooshian	CNN(Convolutional Neural Networks), SVM(Support Vector Machines), quantile mapping (QM)	Proposed a new method for radar precipitation nowcasting by combining the CNNs with Long Short Term Memory Networks (LSTM)Among all the meteorological variables, the geopotential height might be the most important one. Considering both the accuracy and complexity of the model, the paper suggests that the combination of geopotential height and	No significant drawbacks.	RMSE, Nash and Sutcliffe coefficient (NSE) and relative bias (RB)

			total water vapor is to be used.		
[27]	Kqi Lun Chong, Sai Hin Lai, Yu Yao, Ali Najah Ahmed, Wan Zurina Wan Jaafar, Ahmed El-Shafie	WT(Wavelet Transform), CNN(Convolutional Neural Network)	WT is used to process the raw rainfall dataset into a set of decomposed wavelet components for the CNN model using discrete wavelet transform(DWT). Daily datasets from Jan 2002 to December 2017 have been used. The predicted result and the actual result match accurately. The model can predict for any period of time range.	The statistical indices used have a range from 0 to a large finite value, and can hence generate a very big number which would be very hard to understand and process. It is dependent on the scale of the number used.	**RMSE**(Root Mean Square error), **RSR**(Sparse Regularization), **MAE**(Mean absolute Error)
[28]	Pencheng Zhang,Wennan Cao, Wenrui Li	ACRF(Altitude Combined Rainfall Forecasting) model, K-Means algorithm, CNN(Convolutional Neural Network).	This model considers the surface and altitude to make the prediction. Surface and altitudes are generally the key factors which result in fluctuation of result. K-Mean algorithm is used to select the meteorological data. ACRF is used to test 92 meteorological stations in China.	The data used as input in the model is only from a specific region (China). A better algorithm could be used to select data from the stations.	**TS**(Threat rating), **MSE**(Mean square error) **Surface and altitude.** **Principal component analysis method** .

❖ Multi layer Perceptron

Article Number	Authors	Methodology or Techniques used	Merits	Issues	Metrics
[25]	Sankhadeep Chatterjee, Bimal Datta, Soumya Sen, Nilanjan dey, Narayan C.Debnath	Greedy forward selection algorithm, K-Means algorithm, HNN(Hybrid Neural Network), MLP(Multi layer Perceptron) , FNN(Forward Neural Network).	Using the k-means algorithm each cluster of neural networks is trained. The proposed hybrid neural network is then compared with MLP-FNN in terms of several statistical performance measuring metrics.	The model is restricted to a specific region (West - Bengal). The proposed accuracy of HNN is 84.26% which could be improved.	**W- weight** , **B-bias**, Percentage, Statistical metrics.

★ MLP+ANN+RBF

[1]	Sunyoung Lee, Sungzoon Cho, Patrick M. Wong	RBF (Radial basis function) , ANN (Artificial Neural Networks), MLP (Multi-layer perceptron) These models are universal function approximators, providing results with high accuracy. Divide and Conquer Approach, Orographic effect.	Estimated and observed Rainfall values were almost equal. Used the Divide and Conquer approach.	Orographic effect was used to predict rainfall in two small areas which resulted in much higher error, RBF approach would have been better.	

★ MLP+FUZZY

[7]	Afolayan Abimbola Helen, Ojokoh Bolanle A., Falaki Samuel O.	Comparative Analysis of Rainfall Prediction Models Using Neural Network and Fuzzy Logic The ANN and FL techniques could be improved upon by combining it with genetic algorithms for its optimization purpose.	The performance analysis of the two models is done using mean square error, root mean square error; mean absolute percentage error, and prediction accuracy.	The prediction accuracy of neural network is 77.17% and that of the fuzzy logic is 68.92% The results show that the neural network model is better than the fuzzy logic model.	**MSE(Mean squared error)** **Mean Absolute Error (MAE)** **Prediction Error *PE*** **RRoot Mean Square Error (RMSE)**

❖ ELM +ARIMA

Article Number	Authors	Methodology or Techniques used	Merits	Issues	Metrics
[2]	Kaushik Dutta, Gouthaman. P	LSTM & ARIMA model,LASSO regression and Neural Network approach. Air temperature, Air humidity,Wind speed, Sunshine duration taken as four input layers. Rainfall, Medium rainfall, High rainfall as Output layer	Ability to increase the quality of algorithm & dataset. The more the data used for prediction will be efficient, the more will be the accuracy of the prediction. Same goes with the accuracy of the algorithm.	Error finding technique-There are many more errors which can have negative effect of algorithm's accuracy such as MSME (Mean squared error), MAE (Mean absolute error) etc	
[19]	S. Poornima and M. Pushpalatha	RNN(Recurrent Neural Networks)	This technique is used for creating a prediction	The model does not increase the accuracy,	LTSM(Long Term Short Memory)

			model over a long period of time. This approach is compared with other methods, namely, Holt–Winter, ARIMA, ELM, RNN and LSTM, in order to demonstrate the improvement of rainfall prediction in the proposed approach. For attaining good accuracy, it is necessary to consider previous datasets.	as compared to the existing LTSM models, as there is no significant change.	$Ht = g(Wh\ [Ht-1, Xt\ ] + bh\ )$ $Ot = f(Wo * Ht + bo)$ where Ht is the hidden state, Ot is the predicted output, Wh and Wo are the weights assigned in hidden layer and output layer, bh and bo are the bias for hidden and output layer, g is the activation function used in hidden layer and f is the output function for prediction

❖ RBF+HPSOGA

Article Number	Authors	Methodology or Techniques used	Merits	Issues	Metrics
[6]	Jiansheng Wu, Jin Long, Mingzhe Liu	HPSOGA(Hybrid of Particle Swarm Optimization into genetic algorithm), RBF-NN(Radial Basis Function Neural Network).	Individuals are categorised into divisions such as elites(best performing individuals), swarm(worst performing individuals). Hybrid optimization strategy showed promising results as an alternative to forecasting tools.	No real proof as to the hybrid optimization strategy will act as an alternative to the forecasting tools because only certain amount of data (upper-half of elites) is used as input in RBF-NN	**w-weight vector, b-bias**

Key Highlights from Literature Survey

- 80% of all the neural network projects in development use back-propagation.
- Almost every neural network seems to be using Root mean square error) to calculate the error between actual and predicted results.
- Mean square error(MSE) also seems to be in abundant use while verifying the result obtained from the neural network model with the actual result.
- ACRF could be used as it is superior to the existing methods in treat rating(TS) and mean square error(MSE).

Reason to use ANN:

- ANNs are data driven models and do not require restrictive assumptions about the form of the basic model.
- ANN can also predict the pattern which is not provided during training (generalization).
- ANN is efficient at training large-size samples due to its parallel processing capability.
- ANN has the ability to implicitly detect complex nonlinear relationships between dependent and independent variables.

Issues in Existing Systems

1. The existing models do not consider the surface and altitudes as parameters for predicting the rainfall in a region. If the altitude and surface is not considered in the models then the results keep fluctuating.
2. Topography, Geography, and other such factors based on the location are not taken into account.
3. Often the systems created are not very effective in predicting peak values of rainfall.

Motivation & Objective

Rainfall is one of the major factors in the balance of the ecosystem, and as such it is imperative to be able to see what factors contribute to the change in rainfall, negatively or positively.

This will help us to predict how rainfall patterns will be observed from certain given changes, and how it will influence the environment around it.

Problem Statement

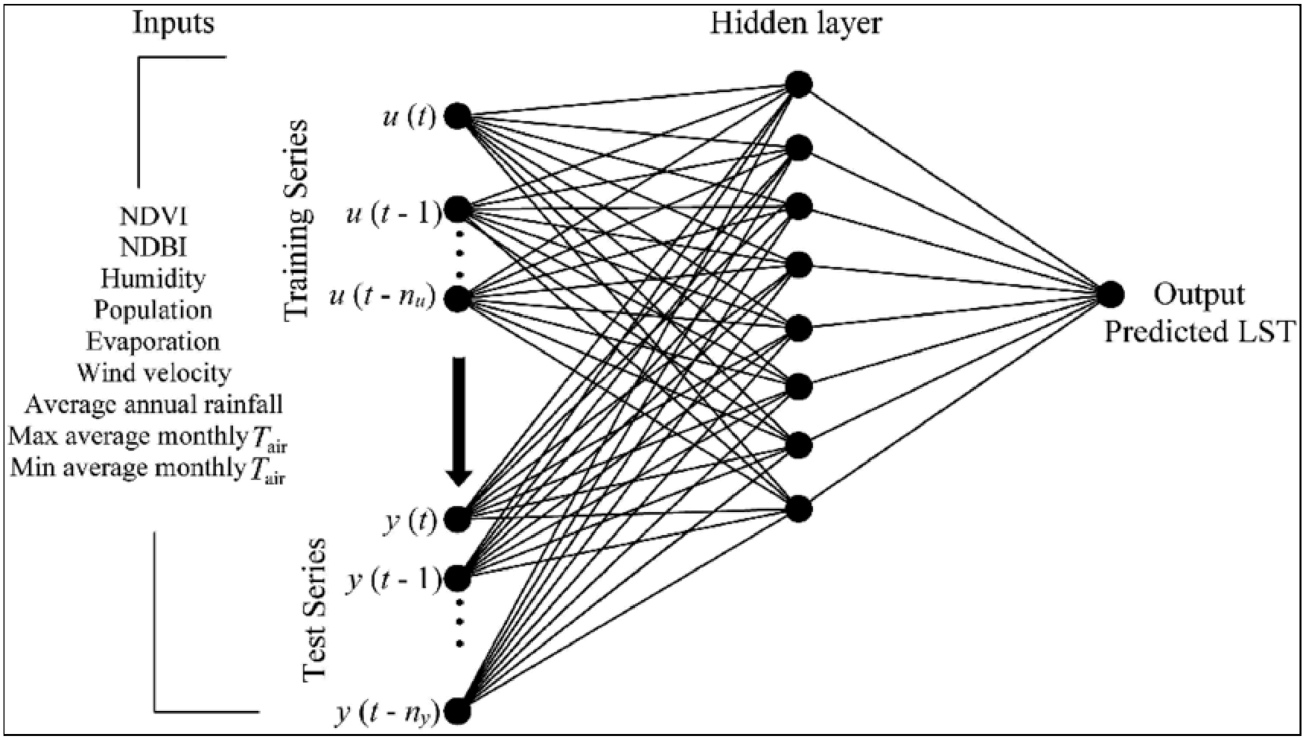
Rainfall, despite being such an important natural resource, is difficult to predict when even the slightest change in factors affecting it comes into play.

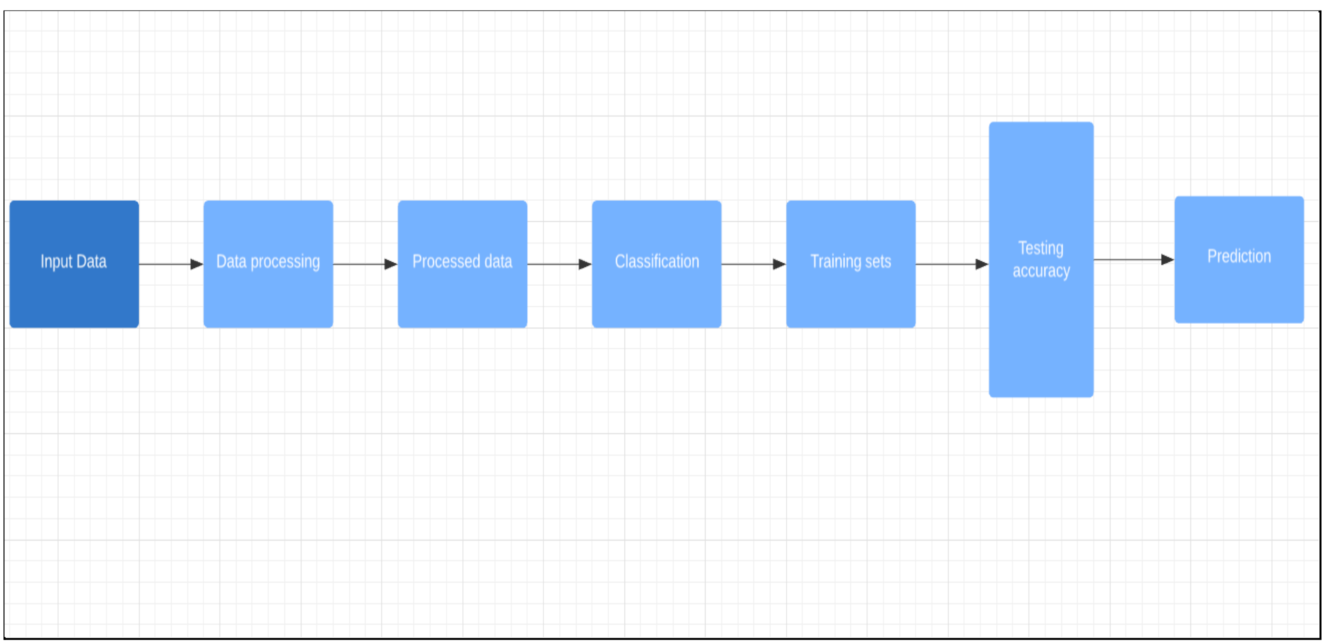
Thus, we are aiming to create a system which can accurately predict the rainfall, for such a multitude of factors.

Evaluation Metrics

- Percentage error
- MSE(Mean squared error)
- W-weights vector
- Mean Absolute Error (MAE)
- Prediction Error ***PE***
- Root Mean Square Error (RMSE)
- Magnitude of Relative Error (MRE)
- Pearson correlation ®

GENERAL ARCHITECTURE





Data Processing:

Here, in the first step of the process we convert the data given to us from kaggle and government data repositories into a form more suitable to be accepted and trained by our system and Neural Networks.

Classification:

Here, the processed data is classified or grouped based on various different factors depending on the application using models like Back propagation network and Radial Basis Function.

Training Sets:

Here the classified sets along with other data sets are used for training the Neural Network Model, here the machine keeps on learning with the desired data sets provided.

Testing Accuracy:

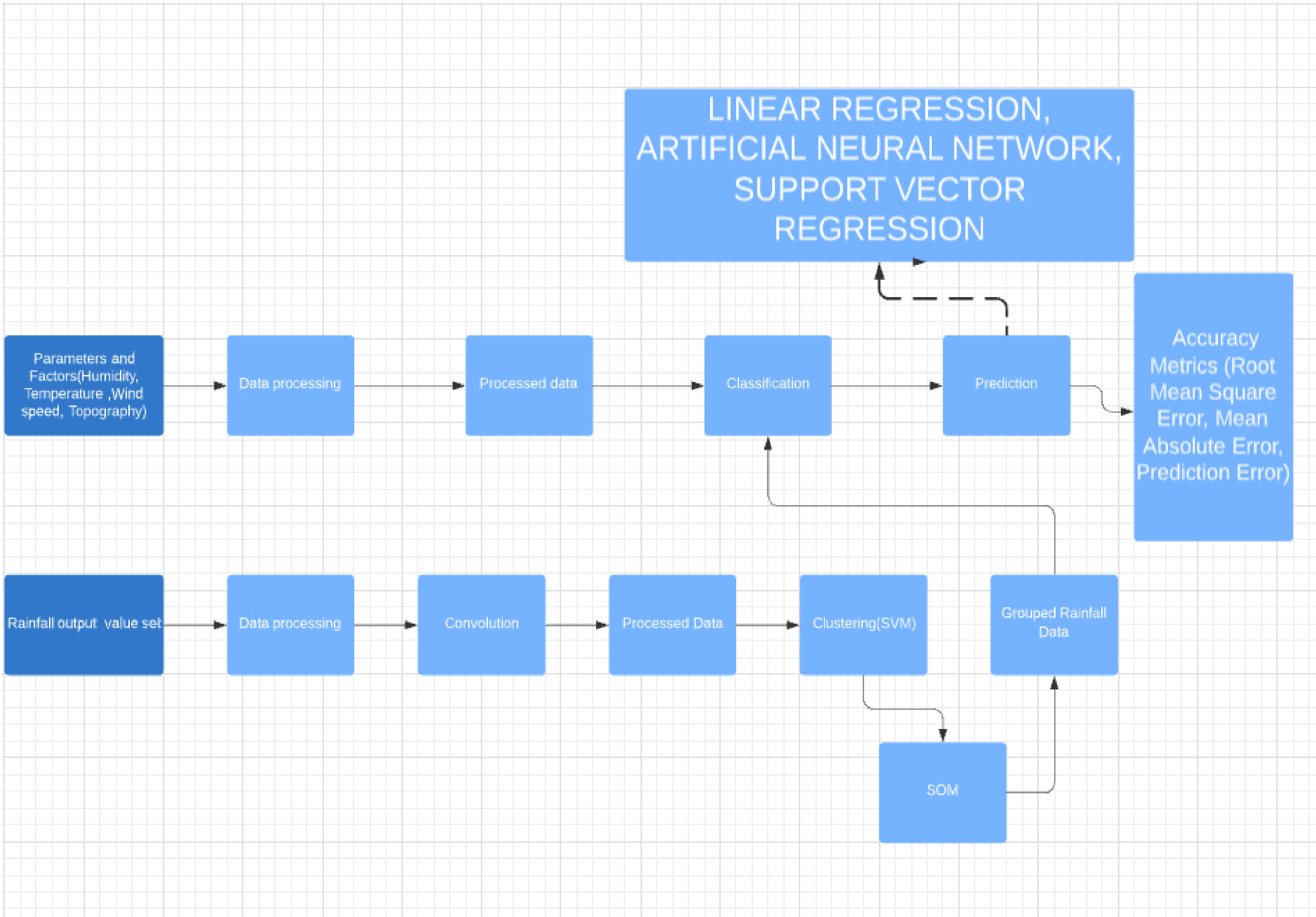
Using the output from the training sets i.e. comparing with the actual future values of rainfall and and the predicted values of rainfall we test the accuracy with different types of Neural Networks and other models used for prediction.

Prediction:

In the final step, the collective data is fed into the Neural Network which provides the results that act as a prediction of the rainfall in short term as well long term basis as per the given inputs.

IV. PROPOSED ALGORITHM

Architecture of the proposed system



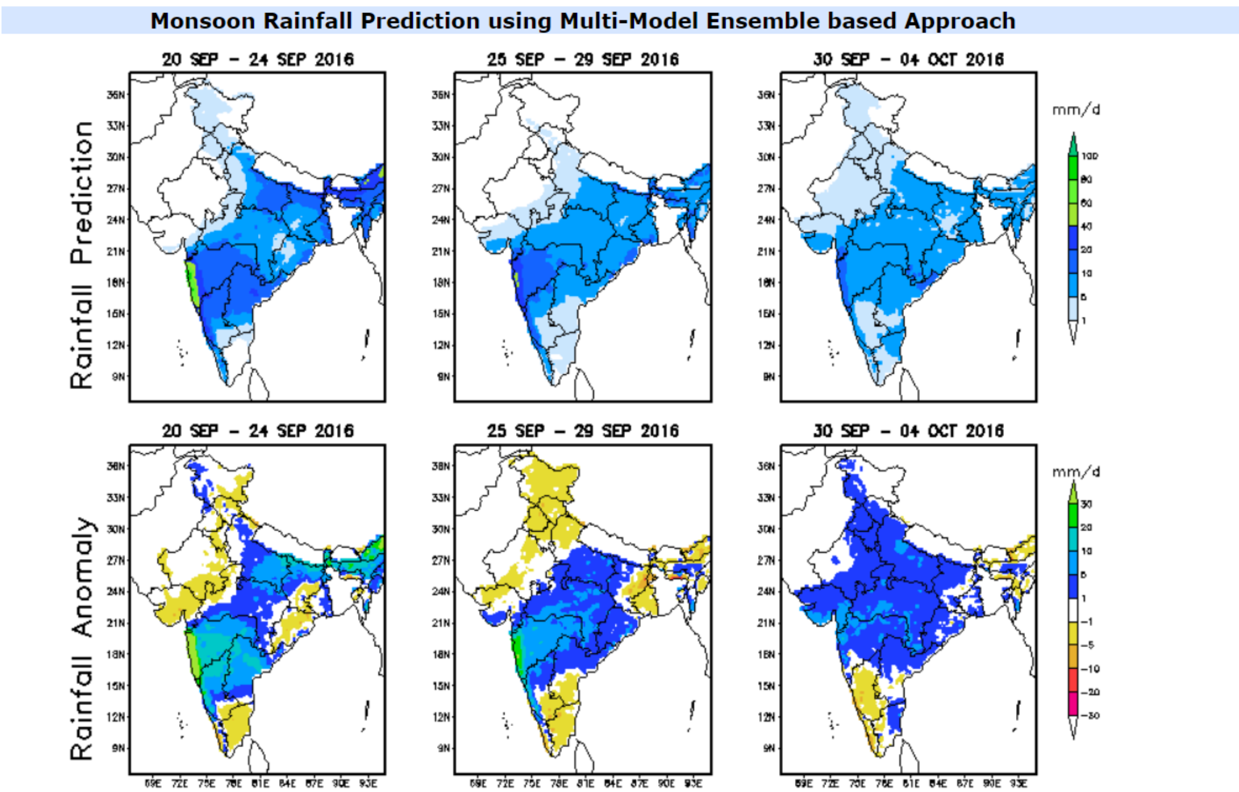
Input Data:

Our system predicts the Rainfall based on different features like humidity, temperature, pressure etc. with reasonable accuracy. Dependent variable here is Rainfall (Precipitation) & the independent variables are Humidity, Temperature, Pressure, Wind speed, Topography etc.

This is different from previous approaches to predicting rainfall data, as it incorporates several factors mentioned above, out of which most models either use none or only a few of the mentioned factors.

Features and Novelty

➢ In the new architecture, we have considered surface and altitude as parameters for better results. The extra input layer also considers topography & geographical parameters like humidity, temperature, wind into account.
➢ We have separated the data in the ratio of 80:20 for training and testing. We have also used the data of the previous three months to predict the rainfall in the next consecutive month.
➢ In the next layers, we proceed with the data input from the first layer to process it
➢ In the following layer we classify using models such as BPNN, RBFN which have been ensembled to provide the most accurate results in the proposed model..
➢ For the clustering part we are using SVM and SOM, again ensembled, similar to classification to group the data for further classification and prediction.
➢ This input is then combined and ensembled with the factors included and prediction of output is done.
➢ The accuracy is then tested using means such as Mean absolute error, and root mean squared error.



Datasets

The dataset to be used in the system will contain the rainfall of several regions in and across the country. It contains rainfall from 1901 – 2015 for the same. Along with that annual rainfall is also used and the rainfall between the transition of two months. There are in total 4116 rows present in the dataset.

- The dataset has been collected from - Data Gov Category – Rainfall in India Released under – NSDAP Contributor – Ministry of Earth Sciences, IMD Group – Rainfall Sectors – Atmosphere science, earth sciences, science & technology.
- https://www.kaggle.com/rajanand/rainfall-in-india

- https://www.kaggle.com/nasirmeh/prediction-of-rainfall

WORKSPACE SCREENSHOTS

WorkSpace

The code below perform the functionality as followed :

Input Data (Dataset) --> Data Processing --> Processed Data --> Classification --> Training sets --> Testing the accuracy --> Prediction

In [ ]:
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
```

UPLOADING DATASET- The dataset has average rainfall for every year from 1901-2015 for each state.

In [ ]:
```
from google.colab import files
uploaded = files.upload()
import io
data = pd.read_csv(io.BytesIO(uploaded['Rainfall.csv']))
```

```
Saving Rainfall.csv to Rainfall.csv
```

In [ ]:
```
data = data.fillna(data.mean())
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4116 entries, 0 to 4115
Data columns (total 19 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   SUBDIVISION  4116 non-null   object
 1   YEAR         4116 non-null   int64
 2   JAN          4116 non-null   float64
 3   FEB          4116 non-null   float64
 4   MAR          4116 non-null   float64
 5   APR          4116 non-null   float64
 6   MAY          4116 non-null   float64
 7   JUN          4116 non-null   float64
 8   JUL          4116 non-null   float64
 9   AUG          4116 non-null   float64
 10  SEP          4116 non-null   float64
 11  OCT          4116 non-null   float64
 12  NOV          4116 non-null   float64
 13  DEC          4116 non-null   float64
 14  ANNUAL       4116 non-null   float64
 15  Jan-Feb      4116 non-null   float64
 16  Mar-May      4116 non-null   float64
 17  Jun-Sep      4116 non-null   float64
 18  Oct-Dec      4116 non-null   float64
dtypes: float64(17), int64(1), object(1)
memory usage: 611.1+ KB
```

OBSERVATION, PROCESSING & CLASSIFICATION OF DATASET THROUGH DIFFERENT FIGURES

OBSERVATION, PROCESSING & CLASSIFICATION OF DATASET THROUGH DIFFERENT FIGURES

In [ ]:
```
data.head()
```

Out[ ]:

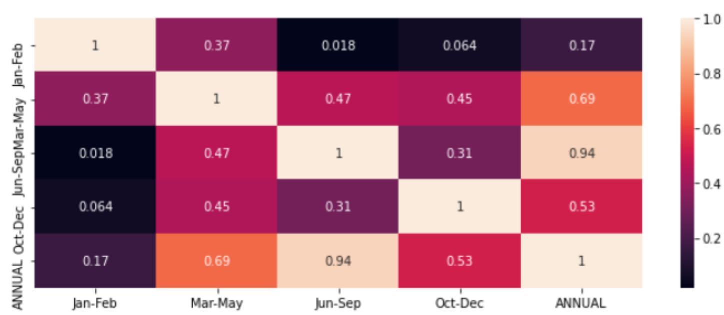
	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May	Jun-Sep	Oct-Dec
0	ANDAMAN & NICOBAR ISLANDS	1901	49.2	87.1	29.2	2.3	528.8	517.5	365.1	481.1	332.6	388.5	558.2	33.6	3373.2	136.3	560.3	1696.3	980.3
1	ANDAMAN & NICOBAR ISLANDS	1902	0.0	159.8	12.2	0.0	446.1	537.1	228.9	753.7	666.2	197.2	359.0	160.5	3520.7	159.8	458.3	2185.9	716.7
2	ANDAMAN & NICOBAR ISLANDS	1903	12.7	144.0	0.0	1.0	235.1	479.9	728.4	326.7	339.0	181.2	284.4	225.0	2957.4	156.7	236.1	1874.0	690.6
3	ANDAMAN & NICOBAR ISLANDS	1904	9.4	14.7	0.0	202.4	304.5	495.1	502.0	160.1	820.4	222.2	308.7	40.1	3079.6	24.1	506.9	1977.6	571.0
4	ANDAMAN & NICOBAR ISLANDS	1905	1.3	0.0	3.3	26.9	279.5	628.7	368.7	330.5	297.0	260.7	25.4	344.7	2566.7	1.3	309.7	1624.9	630.8

In [ ]:
```
data.describe()
```

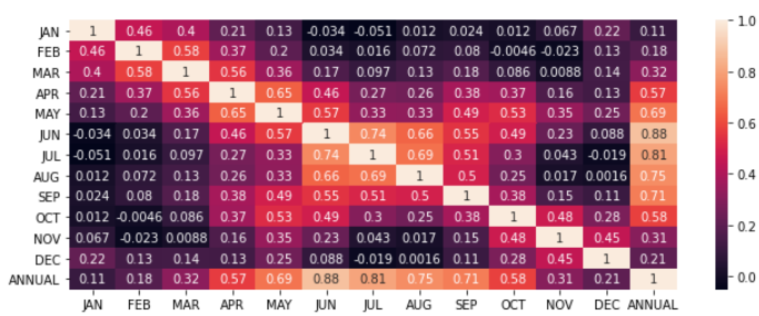
Out[ ]:

	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	N
count	4116.000000	4116.000000	4116.000000	4116.000000	4116.000000	4116.000000	4116.000000	4116.000000	4116.000000	4116.000000	4116.000000	4
mean	1958.218659	18.957320	21.805325	27.359197	43.127432	85.745417	230.234444	347.214334	290.263497	197.361922	95.507009	3
std	33.140898	33.569044	35.896396	46.925176	67.798192	123.189974	234.568120	269.310313	188.678707	135.309591	99.434452	6
min	1901.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.400000	0.000000	0.000000	0.100000	0.000000	0
25%	1930.000000	0.600000	0.600000	1.000000	3.000000	8.600000	70.475000	175.900000	156.150000	100.600000	14.600000	0
50%	1958.000000	6.000000	6.700000	7.900000	15.700000	36.700000	138.900000	284.900000	259.500000	174.100000	65.750000	9
75%	1987.000000	22.125000	26.800000	31.225000	49.825000	96.825000	304.950000	418.225000	377.725000	265.725000	148.300000	4
max	2015.000000	583.700000	403.500000	605.600000	595.100000	1168.600000	1609.900000	2362.800000	1664.600000	1222.000000	948.300000	6

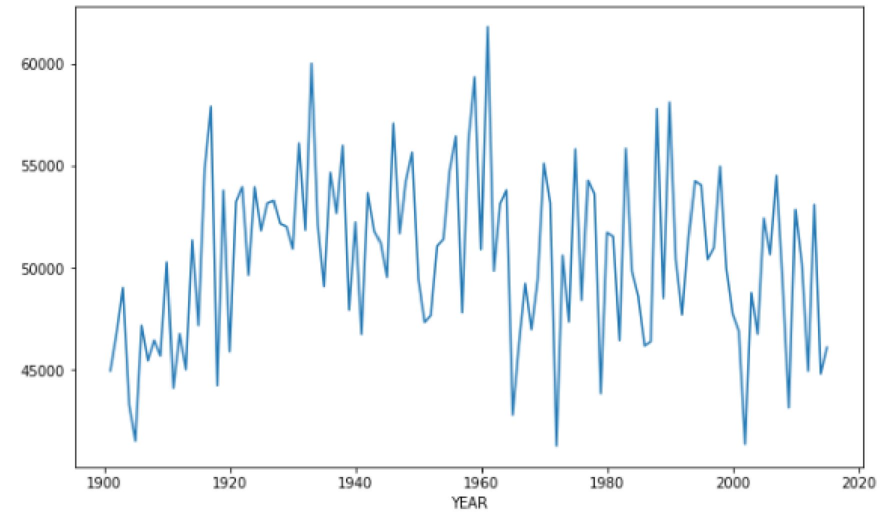
In [ ]:
```
plt.figure(figsize=(11,4))
sns.heatmap(data[['Jan-Feb','Mar-May','Jun-Sep','Oct-Dec','ANNUAL']].corr(),annot=True)
plt.show()
```



In [ ]:
```
plt.figure(figsize=(11,4))
sns.heatmap(data[['JAN','FEB','MAR','APR','MAY','JUN','JUL','AUG','SEP','OCT','NOV','DEC','ANNUAL']].corr(),annot=True)
plt.show()
```

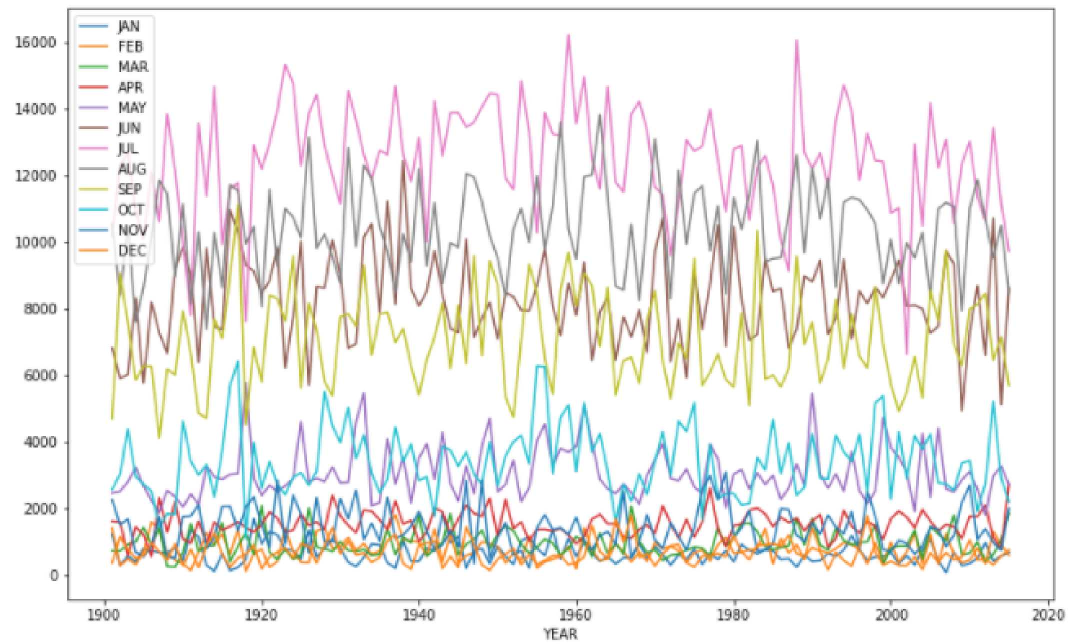


```
In [ ]: data.groupby("YEAR").sum()['ANNUAL'].plot(figsize=(10,6));
```

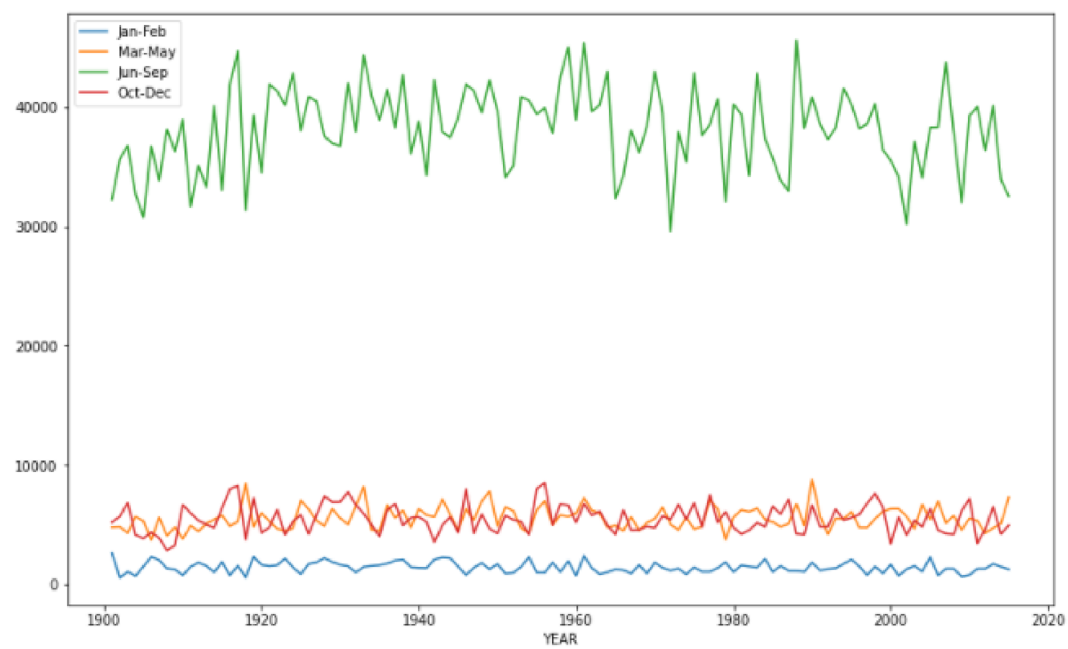



```
In [ ]: data[['YEAR', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
          'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].groupby("YEAR").sum().plot(figsize=(13,8));
```

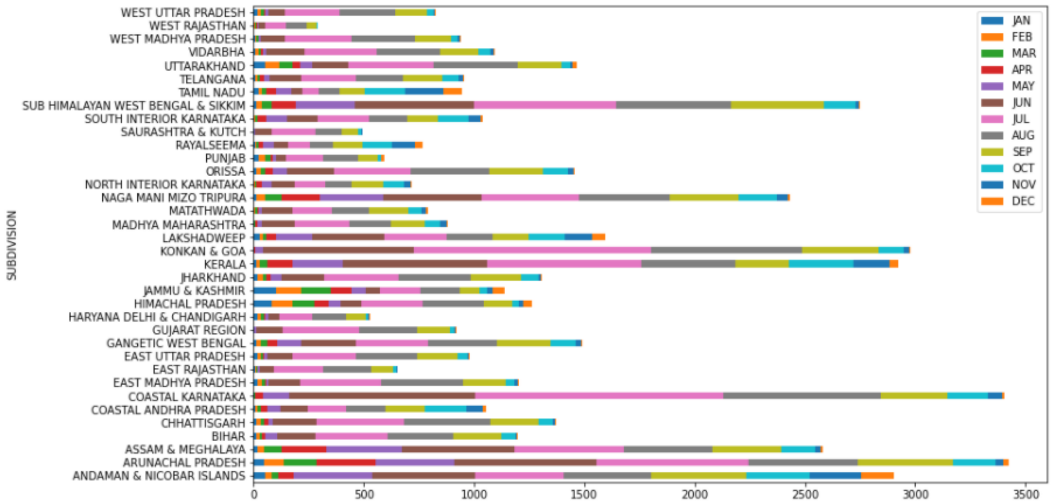
In [ ]:
```
data[['YEAR', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
        'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].groupby("YEAR").sum().plot(figsize=(13,8));
```



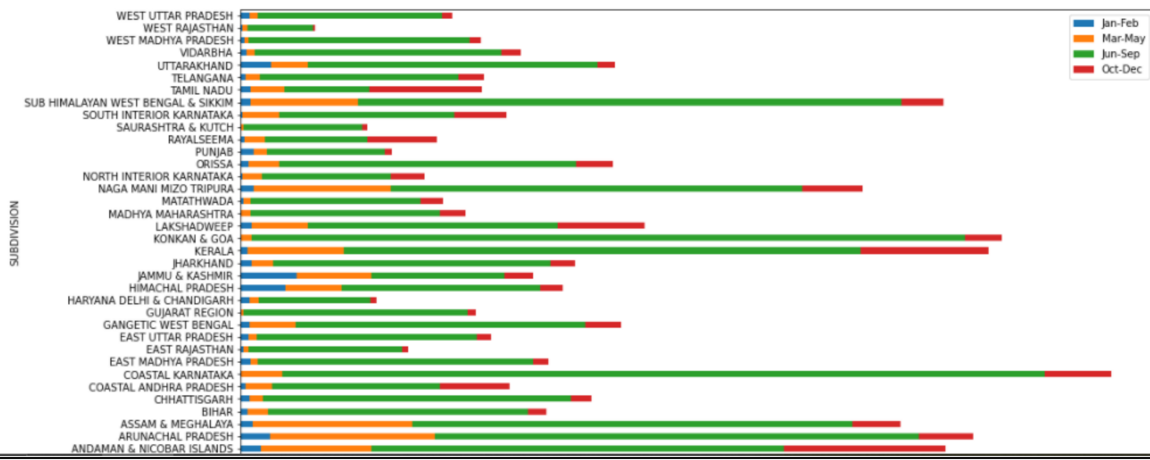
In [ ]:
```
data[['YEAR','Jan-Feb', 'Mar-May',
        'Jun-Sep', 'Oct-Dec']].groupby("YEAR").sum().plot(figsize=(13,8));
```



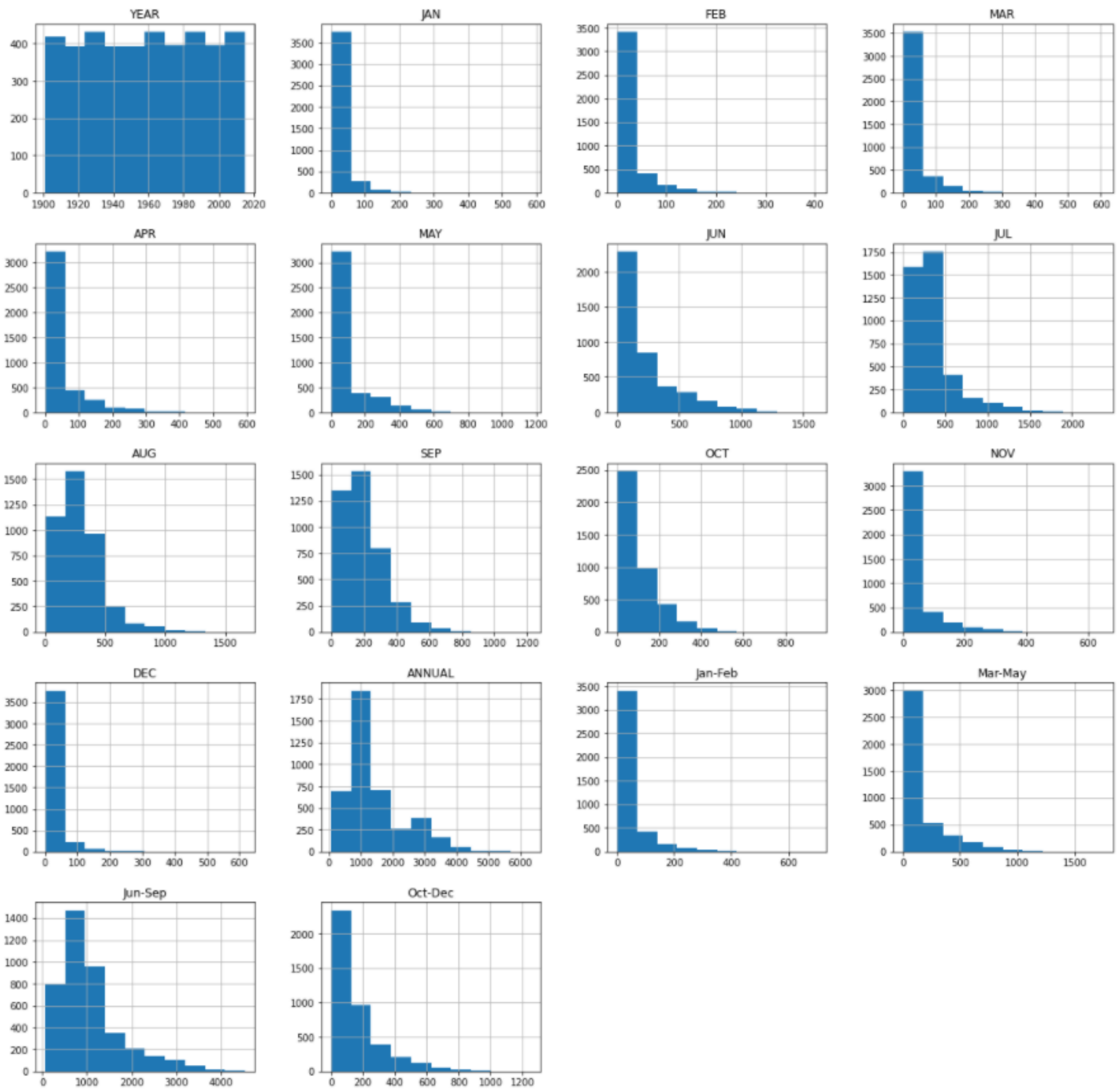
```
In [ ]: data[['SUBDIVISION', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
               'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].groupby("SUBDIVISION").mean().plot.barh(stacked=True,figsize=(13,8));
```




```
In [ ]: data[['SUBDIVISION', 'Jan-Feb', 'Mar-May',
               'Jun-Sep', 'Oct-Dec']].groupby("SUBDIVISION").sum().plot.barh(stacked=True,figsize=(16,8));
```



```
In [ ]: data.hist(figsize=(20,20));
```



```python
In [ ]: #Function to plot the graphs
        def plot_graphs(groundtruth,prediction,title):
            N = 9
            ind = np.arange(N)  # the x locations for the groups
            width = 0.27       # the width of the bars

            fig = plt.figure()
            fig.suptitle(title, fontsize=12)
            ax = fig.add_subplot(111)
            rects1 = ax.bar(ind, groundtruth, width, color='r')
            rects2 = ax.bar(ind+width, prediction, width, color='g')

            ax.set_ylabel("Amount of rainfall")
            ax.set_xticks(ind+width)
            ax.set_xticklabels( ('APR', 'MAY', 'JUN', 'JUL','AUG', 'SEP', 'OCT', 'NOV', 'DEC') )
            ax.legend( (rects1[0], rects2[0]), ('Ground truth', 'Prediction') )

        #     autolabel(rects1)
            for rect in rects1:
                h = rect.get_height()
                ax.text(rect.get_x()+rect.get_width()/2., 1.05*h, '%d'%int(h),
                        ha='center', va='bottom')
            for rect in rects2:
                h = rect.get_height()
                ax.text(rect.get_x()+rect.get_width()/2., 1.05*h, '%d'%int(h),
                        ha='center', va='bottom')
        #     autolabel(rects2)

            plt.show()
```

PREDICTION

For prediction we formatted data in this way, given the rainfall in the last three months we try to predict the rainfall in the next consecutive month. For all the experiments we used 80:20 training and test ratio.

The following prediction algorithms are used-

- Linear regression
- SVR
- Artificial neural nets

In [ ]:

```
# seperation of training and testing data
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error

division_data = np.asarray(data[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
       'AUG', 'SEP', 'OCT', 'NOV', 'DEC']])

X = None; y = None
for i in range(division_data.shape[1]-3):
    if X is None:
        X = division_data[:, i:i+3]
        y = division_data[:, i+3]
    else:
        X = np.concatenate((X, division_data[:, i:i+3]), axis=0)
        y = np.concatenate((y, division_data[:, i+3]), axis=0)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)
```

In [ ]:

```
#test 2010
temp = data[['SUBDIVISION','JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
       'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].loc[data['YEAR'] == 2010]

data_2010 = np.asarray(temp[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
       'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].loc[temp['SUBDIVISION'] == 'PUNJAB'])

X_year_2010 = None; y_year_2010 = None
for i in range(data_2010.shape[1]-3):
    if X_year_2010 is None:
        X_year_2010 = data_2010[:, i:i+3]
        y_year_2010 = data_2010[:, i+3]
    else:
        X_year_2010 = np.concatenate((X_year_2010, data_2010[:, i:i+3]), axis=0)
        y_year_2010 = np.concatenate((y_year_2010, data_2010[:, i+3]), axis=0)
```

In [ ]:
```
#test 2010
temp = data[['SUBDIVISION','JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
       'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].loc[data['YEAR'] == 2010]

data_2010 = np.asarray(temp[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
       'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].loc[temp['SUBDIVISION'] == 'PUNJAB'])

X_year_2010 = None; y_year_2010 = None
for i in range(data_2010.shape[1]-3):
    if X_year_2010 is None:
        X_year_2010 = data_2010[:, i:i+3]
        y_year_2010 = data_2010[:, i+3]
    else:
        X_year_2010 = np.concatenate((X_year_2010, data_2010[:, i:i+3]), axis=0)
        y_year_2010 = np.concatenate((y_year_2010, data_2010[:, i+3]), axis=0)
```

In [ ]:
```
#test 2005
temp = data[['SUBDIVISION','JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
       'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].loc[data['YEAR'] == 2005]

data_2005 = np.asarray(temp[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
       'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].loc[temp['SUBDIVISION'] == 'PUNJAB'])

X_year_2005 = None; y_year_2005 = None
for i in range(data_2005.shape[1]-3):
    if X_year_2005 is None:
        X_year_2005 = data_2005[:, i:i+3]
        y_year_2005 = data_2005[:, i+3]
    else:
        X_year_2005 = np.concatenate((X_year_2005, data_2005[:, i:i+3]), axis=0)
        y_year_2005 = np.concatenate((y_year_2005, data_2005[:, i+3]), axis=0)
```

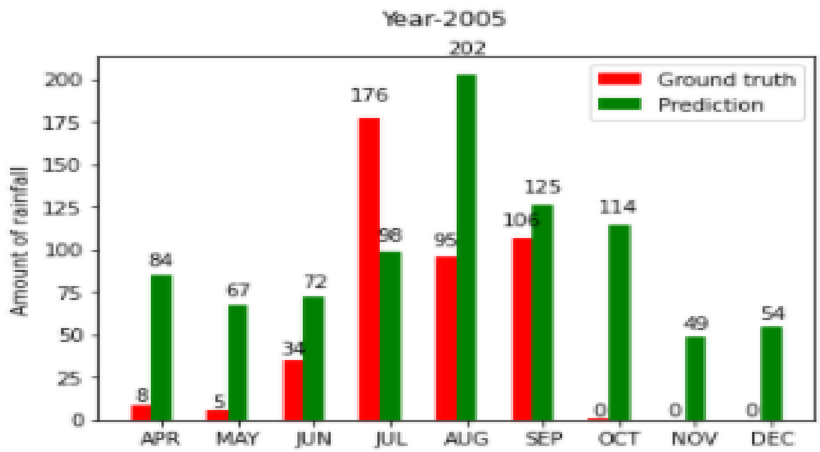
In [ ]:
```
#test 2015
temp = data[['SUBDIVISION','JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
       'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].loc[data['YEAR'] == 2015]

data_2015 = np.asarray(temp[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
       'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].loc[temp['SUBDIVISION'] == 'PUNJAB'])

X_year_2015 = None; y_year_2015 = None
for i in range(data_2015.shape[1]-3):
    if X_year_2015 is None:
        X_year_2015 = data_2015[:, i:i+3]
        y_year_2015 = data_2015[:, i+3]
    else:
        X_year_2015 = np.concatenate((X_year_2015, data_2015[:, i:i+3]), axis=0)
        y_year_2015 = np.concatenate((y_year_2015, data_2015[:, i+3]), axis=0)
```

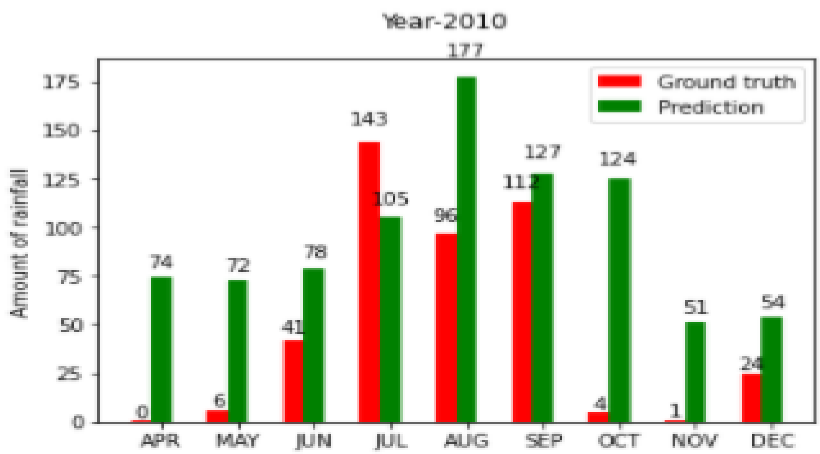
In [ ]:
```python
from sklearn import linear_model

# linear model
reg = linear_model.ElasticNet(alpha=0.5)
reg.fit(X_train, y_train)
y_pred = reg.predict(X_test)
print(mean_absolute_error(y_test, y_pred))
```

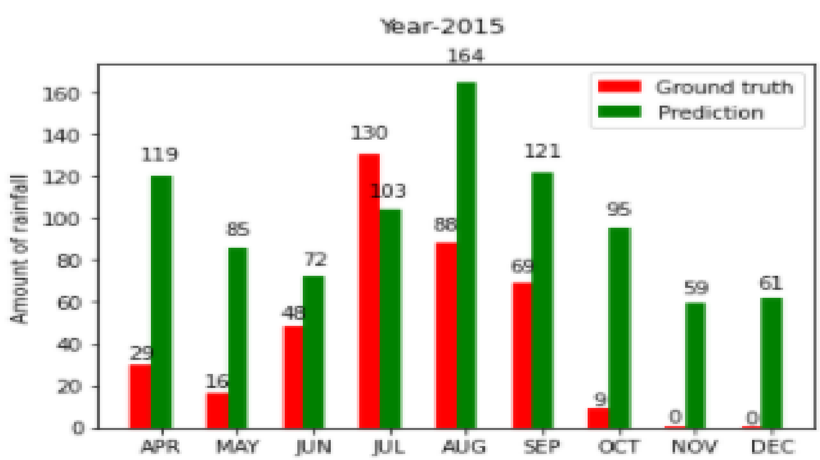
```
96.32435229744083
```

In [ ]:
```python
#2005
y_year_pred_2005 = reg.predict(X_year_2005)


#2010
y_year_pred_2010 = reg.predict(X_year_2010)

y_year_pred_2015 = reg.predict(X_year_2015)

print("MEAN 2005")
print(np.mean(y_year_2005),np.mean(y_year_pred_2005))
print("Standard deviation 2005")
print(np.sqrt(np.var(y_year_2005)),np.sqrt(np.var(y_year_pred_2005)))


print("MEAN 2010")
print(np.mean(y_year_2010),np.mean(y_year_pred_2010))
print("Standard deviation 2010")
print(np.sqrt(np.var(y_year_2010)),np.sqrt(np.var(y_year_pred_2010)))


print("MEAN 2015")
print(np.mean(y_year_2015),np.mean(y_year_pred_2015))
print("Standard deviation 2015")
print(np.sqrt(np.var(y_year_2015)),np.sqrt(np.var(y_year_pred_2015)))


plot_graphs(y_year_2005,y_year_pred_2005,"Year-2005")
plot_graphs(y_year_2010,y_year_pred_2010,"Year-2010")
plot_graphs(y_year_2015,y_year_pred_2015,"Year-2015")
```

```
MEAN 2005
47.477777777777774 96.60640201623632
Standard deviation 2005
60.27486628913282 44.75556760630732
MEAN 2010
48.0 96.23320118559711
Standard deviation 2010
52.08919700326013 38.94352508698204
MEAN 2015
43.7 98.30013917047046
Standard deviation 2015
42.14443419163832 31.951731018336222
```







In [ ]:

```
from keras.models import Model
from keras.layers import Dense, Input, Conv1D, Flatten

# NN model
inputs = Input(shape=(3,1))
x = Conv1D(64, 2, padding='same', activation='elu')(inputs)
x = Conv1D(128, 2, padding='same', activation='elu')(x)
x = Flatten()(x)
x = Dense(128, activation='elu')(x)
x = Dense(64, activation='elu')(x)
x = Dense(32, activation='elu')(x)
x = Dense(1, activation='linear')(x)
model = Model(inputs=[inputs], outputs=[x])
model.compile(loss='mean_squared_error', optimizer='adamax', metrics=['mae'])
model.summary()
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 3, 1)]            0
_________________________________________________________________
conv1d (Conv1D)              (None, 3, 64)             192
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 3, 128)            16512
_________________________________________________________________
flatten (Flatten)            (None, 384)               0
_________________________________________________________________
dense (Dense)                (None, 128)               49280
_________________________________________________________________
dense_1 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_2 (Dense)              (None, 32)                2080
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 33
=================================================================
Total params: 76,353
Trainable params: 76,353
Non-trainable params: 0
_________________________________________________________________
```

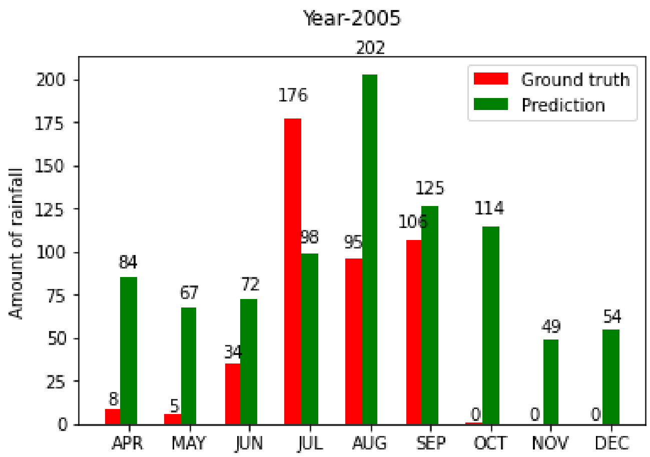
In [ ]:

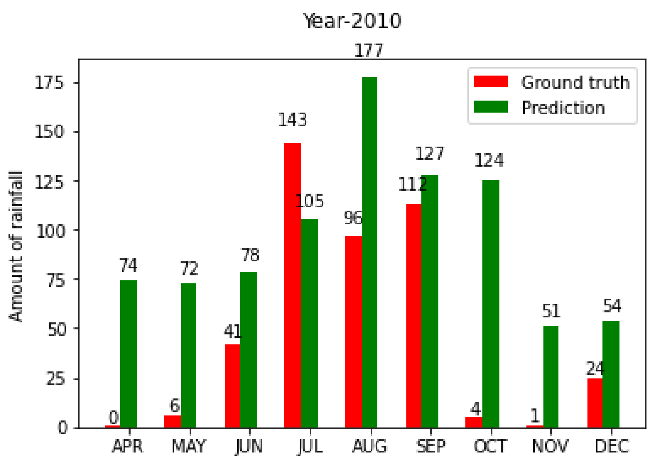
```
model.fit(x=np.expand_dims(X_train, axis=2), y=y_train, batch_size=64, epochs=10, verbose=1, validation_split=0.1, shuffle=True)
y_pred = model.predict(np.expand_dims(X_test, axis=2))
print(mean_absolute_error(y_test, y_pred))
```

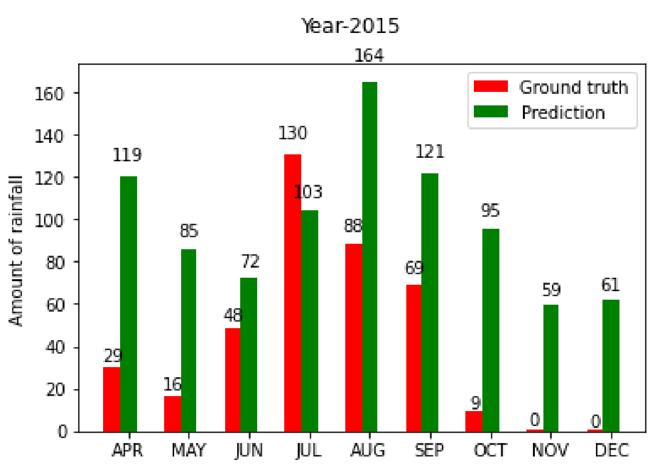
```
Epoch 1/10
469/469 [==============================] - 3s 5ms/step - loss: 22687.3552 - mae: 91.5300 - val_loss: 17666.3320 - val_mae: 85.8736
Epoch 2/10
469/469 [==============================] - 2s 4ms/step - loss: 18733.6728 - mae: 86.8882 - val_loss: 17513.2109 - val_mae: 87.2290
Epoch 3/10
469/469 [==============================] - 2s 4ms/step - loss: 18898.1248 - mae: 87.0914 - val_loss: 17355.4355 - val_mae: 84.8080
Epoch 4/10
469/469 [==============================] - 2s 4ms/step - loss: 18452.8251 - mae: 85.7017 - val_loss: 17585.8125 - val_mae: 86.8053
Epoch 5/10
469/469 [==============================] - 2s 4ms/step - loss: 17927.5303 - mae: 85.4813 - val_loss: 17324.6055 - val_mae: 83.7834
Epoch 6/10
469/469 [==============================] - 2s 4ms/step - loss: 18186.2085 - mae: 85.4019 - val_loss: 17604.5566 - val_mae: 87.2761
Epoch 7/10
469/469 [==============================] - 2s 4ms/step - loss: 18287.3563 - mae: 85.5081 - val_loss: 17387.5039 - val_mae: 85.7324
Epoch 8/10
469/469 [==============================] - 2s 4ms/step - loss: 18571.5448 - mae: 86.5604 - val_loss: 17302.2324 - val_mae: 84.8773
Epoch 9/10
469/469 [==============================] - 2s 4ms/step - loss: 18059.3360 - mae: 84.8311 - val_loss: 17361.1367 - val_mae: 85.5691
Epoch 10/10
469/469 [==============================] - 2s 4ms/step - loss: 18366.9713 - mae: 85.8968 - val_loss: 17023.9160 - val_mae: 83.9831
86.65617735763784
```

```
In [ ]: #2005
        y_year_pred_2005 = reg.predict(X_year_2005)

        #2010
        y_year_pred_2010 = reg.predict(X_year_2010)

        #2015
        y_year_pred_2015 = reg.predict(X_year_2015)

        print("MEAN 2005")
        print(np.mean(y_year_2005),np.mean(y_year_pred_2005))
        print("Standard deviation 2005")
        print(np.sqrt(np.var(y_year_2005)),np.sqrt(np.var(y_year_pred_2005)))


        print("MEAN 2010")
        print(np.mean(y_year_2010),np.mean(y_year_pred_2010))
        print("Standard deviation 2010")
        print(np.sqrt(np.var(y_year_2010)),np.sqrt(np.var(y_year_pred_2010)))


        print("MEAN 2015")
        print(np.mean(y_year_2015),np.mean(y_year_pred_2015))
        print("Standard deviation 2015")
        print(np.sqrt(np.var(y_year_2015)),np.sqrt(np.var(y_year_pred_2015)))

        plot_graphs(y_year_2005,y_year_pred_2005,"Year-2005")
        plot_graphs(y_year_2010,y_year_pred_2010,"Year-2010")
        plot_graphs(y_year_2015,y_year_pred_2015,"Year-2015")
```

```
MEAN 2005
47.477777777777774 96.60640201623632
Standard deviation 2005
60.27486628913282 44.75556760630732
MEAN 2010
48.0 96.23320118559711
Standard deviation 2010
52.08919700326013 38.94352508698204
MEAN 2015
43.7 98.30013917047046
Standard deviation 2015
42.14443419163832 31.951731018336222
```


Year-2005
Ground truth
Prediction
Amount of rainfall
0
25
50
75
100
125
150
175
200
8
84
5
67
34
72
176
98
95
202
106
125
0
114
0
49
0
54
APR
MAY
JUN
JUL
AUG
SEP
OCT
NOV
DEC


Year-2010
Ground truth
Prediction
Amount of rainfall
0
25
50
75
100
125
150
175
0
74
6
72
41
78
143
105
96
177
112
127
4
124
1
51
24
54
APR
MAY
JUN
JUL
AUG
SEP
OCT
NOV
DEC


Year-2015
Ground truth
Prediction
Amount of rainfall
0
20
40
60
80
100
120
140
160
29
119
16
85
48
72
130
103
88
164
69
121
9
95
0
59
0
61
APR
MAY
JUN
JUL
AUG
SEP
OCT
NOV
DEC

TESTING AND TRAINING DATA FOR SPECIFIC STATE- PUNJAB

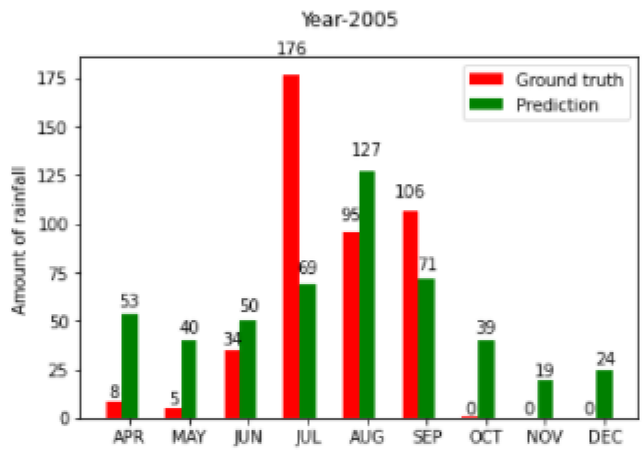
In [ ]:
```
# spliting training and testing data only for Punjab
punjab = np.asarray(data[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
       'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].loc[data['SUBDIVISION'] == 'PUNJAB'])

X = None; y = None
for i in range(punjab.shape[1]-3):
    if X is None:
        X = punjab[:, i:i+3]
        y = punjab[:, i+3]
    else:
        X = np.concatenate((X, punjab[:, i:i+3]), axis=0)
        y = np.concatenate((y, punjab[:, i+3]), axis=0)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.01, random_state=42)
```

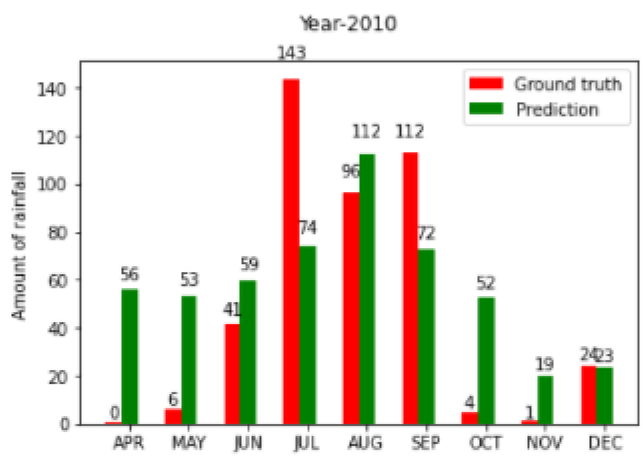
In [ ]:
```
from sklearn import linear_model

# linear model
reg = linear_model.ElasticNet(alpha=0.5)
reg.fit(X_train, y_train)
y_pred = reg.predict(X_test)
print(mean_absolute_error(y_test, y_pred))
```

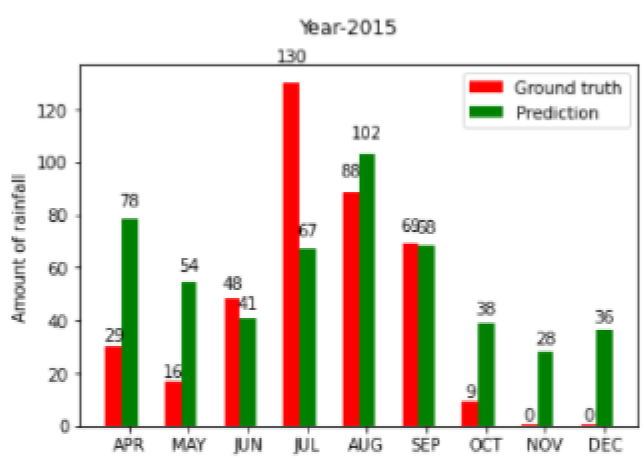
```
46.288437492334076
```

In [ ]:
```
#2005
y_year_pred_2005 = reg.predict(X_year_2005)

#2010
y_year_pred_2010 = reg.predict(X_year_2010)

#2015
y_year_pred_2015 = reg.predict(X_year_2015)

print("MEAN 2005")
print(np.mean(y_year_2005),np.mean(y_year_pred_2005))
print("Standard deviation 2005")
print(np.sqrt(np.var(y_year_2005)),np.sqrt(np.var(y_year_pred_2005)))


print("MEAN 2010")
print(np.mean(y_year_2010),np.mean(y_year_pred_2010))
print("Standard deviation 2010")
print(np.sqrt(np.var(y_year_2010)),np.sqrt(np.var(y_year_pred_2010)))


print("MEAN 2015")
print(np.mean(y_year_2015),np.mean(y_year_pred_2015))
print("Standard deviation 2015")
print(np.sqrt(np.var(y_year_2015)),np.sqrt(np.var(y_year_pred_2015)))

plot_graphs(y_year_2005,y_year_pred_2005,"Year-2005")
plot_graphs(y_year_2010,y_year_pred_2010,"Year-2010")
plot_graphs(y_year_2015,y_year_pred_2015,"Year-2015")
```

```
MEAN 2005
47.477777777777774 55.03656527267601
Standard deviation 2005
60.27486628913282 30.41770901028018
MEAN 2010
48.0 58.238901382104736
Standard deviation 2010
52.08919700326013 26.165211080203
MEAN 2015
43.7 57.267900910533314
Standard deviation 2015
42.14443419163832 22.727067908124504
```





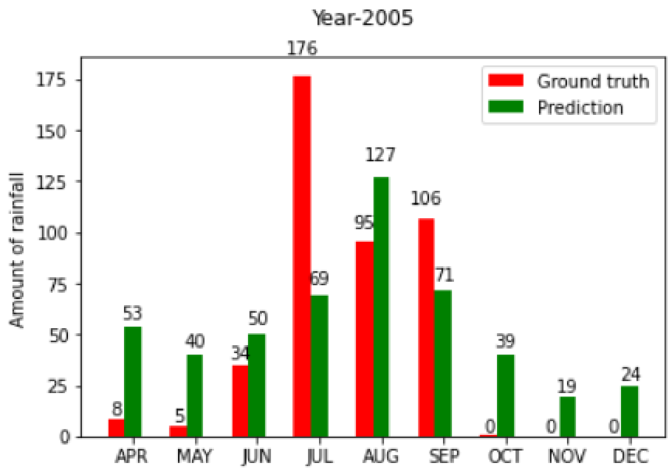


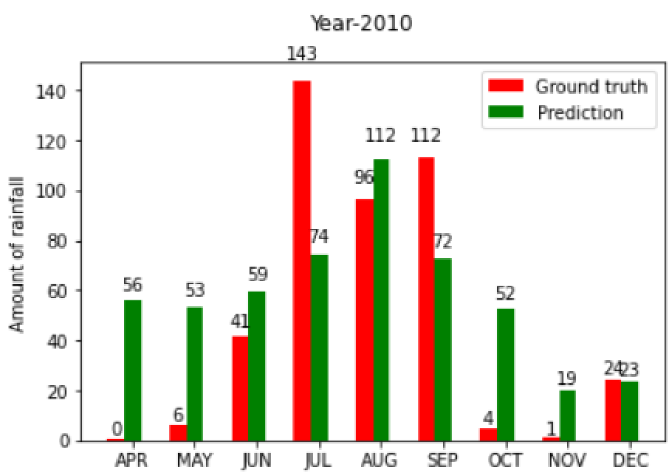
In [ ]:
```python
from sklearn.svm import SVR

# SVM model
clf = SVR(kernel='rbf', gamma='auto', C=0.5, epsilon=0.2)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
print(mean_absolute_error(y_test, y_pred))
```

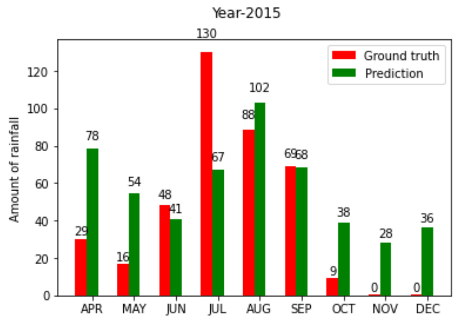
```
52.8911225532616
```

In [ ]:
```python
#2005
y_year_pred_2005 = reg.predict(X_year_2005)

#2010
y_year_pred_2010 = reg.predict(X_year_2010)

#2015
y_year_pred_2015 = reg.predict(X_year_2015)

print("MEAN 2005")
print(np.mean(y_year_2005),np.mean(y_year_pred_2005))
print("Standard deviation 2005")
print(np.sqrt(np.var(y_year_2005)),np.sqrt(np.var(y_year_pred_2005)))


print("MEAN 2010")
print(np.mean(y_year_2010),np.mean(y_year_pred_2010))
print("Standard deviation 2010")
print(np.sqrt(np.var(y_year_2010)),np.sqrt(np.var(y_year_pred_2010)))


print("MEAN 2015")
print(np.mean(y_year_2015),np.mean(y_year_pred_2015))
print("Standard deviation 2015")
print(np.sqrt(np.var(y_year_2015)),np.sqrt(np.var(y_year_pred_2015)))

plot_graphs(y_year_2005,y_year_pred_2005,"Year-2005")
plot_graphs(y_year_2010,y_year_pred_2010,"Year-2010")
plot_graphs(y_year_2015,y_year_pred_2015,"Year-2015")
```

```
MEAN 2005
47.477777777777774 55.03656527267601
Standard deviation 2005
60.27486628913282 30.41770901028018
MEAN 2010
48.0 58.238901382104736
Standard deviation 2010
52.08919700326013 26.165211080203
MEAN 2015
43.7 57.267900910533314
Standard deviation 2015
42.14443419163832 22.727067908124504
```







In [ ]:

```
model.fit(x=np.expand_dims(X_train, axis=2), y=y_train, batch_size=64, epochs=10, verbose=1, validation_split=0.1, shuffle=True)
y_pred = model.predict(np.expand_dims(X_test, axis=2))
print(mean_absolute_error(y_test, y_pred))
```

```
Epoch 1/10
15/15 [==============================] - 0s 11ms/step - loss: 6404.7183 - mae: 57.2172 - val_loss: 3536.7673 - val_mae: 42.8480
Epoch 2/10
15/15 [==============================] - 0s 7ms/step - loss: 4373.5698 - mae: 42.5351 - val_loss: 2386.7100 - val_mae: 33.2118
Epoch 3/10
15/15 [==============================] - 0s 6ms/step - loss: 4277.8765 - mae: 40.7706 - val_loss: 2232.6953 - val_mae: 33.0882
Epoch 4/10
15/15 [==============================] - 0s 7ms/step - loss: 4104.6807 - mae: 42.0731 - val_loss: 2314.5825 - val_mae: 34.6880
Epoch 5/10
15/15 [==============================] - 0s 7ms/step - loss: 4099.7026 - mae: 43.4880 - val_loss: 2323.6750 - val_mae: 34.8766
Epoch 6/10
15/15 [==============================] - 0s 7ms/step - loss: 4036.4912 - mae: 42.2419 - val_loss: 2167.4119 - val_mae: 33.0055
Epoch 7/10
15/15 [==============================] - 0s 7ms/step - loss: 4026.0129 - mae: 41.5690 - val_loss: 2195.8015 - val_mae: 33.3416
Epoch 8/10
15/15 [==============================] - 0s 7ms/step - loss: 4002.3433 - mae: 41.7327 - val_loss: 2218.3132 - val_mae: 33.4872
Epoch 9/10
15/15 [==============================] - 0s 7ms/step - loss: 4006.6270 - mae: 41.6921 - val_loss: 2213.7324 - val_mae: 33.4936
Epoch 10/10
15/15 [==============================] - 0s 7ms/step - loss: 3976.2031 - mae: 41.6140 - val_loss: 2197.7429 - val_mae: 33.1164
48.226309221441085
```

V. EXPERIMENTS RESULTS

Prediction Observations

Training on complete dataset

ALGORITHM	MEAN ABSOLUTE ERROR
Linear Regression	96.32435229744083
SVR	127.1600615632603
Artificial neural network	86.65617735763784

Training on Punjab dataset

ALGORITHM	MEAN ABSOLUTE ERROR
Linear Regression	48.288437492334076
SVR	52.8911225532616
Artificial neural network	46.226309221441085

The results of PSNR from the attacked image and original image drives the fact that our scheme is robust to distortion on applications of digital image processing attacks and geometrical attacks.

VI. COMPARATIVE STUDY

The results obtained in some other studies are as follows:

1. The performance analysis of the two models is done using mean square error, root mean square error; mean absolute percentage error, and prediction accuracy. The prediction accuracy of the neural network is 77.17% and that of the fuzzy logic is 68.92%. The results show that the neural network model is better than the fuzzy logic model. (https://www.ijsce.org/wp-content/uploads/papers/v5i6/D2689095415.pdf) [7]

2. For a study, The error values of the estimated rainfall are: ·

Root Mean Square Errors (RMSE) of the whole dataset: 78.65

Area 1 – 45.96 Area 2 – 102.36 Area 3 – 121.96 Area 4 – 77.23

· Relative Error – 0.46 and· Absolute Error – 55.9

(https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.13.4364&rep=rep1&type=pdf)[1]

3. Study using Genetic ANN

Accuracy of the models used.

Method	FF_BPNN	**BPNN-GA**
Training	79.54	80.12
Validation	90.63	97.17
Testing	92.78	**98.78**

(https://acadpubl.eu/jsi/2018-119-10/articles/10a/60.pdf) [29]

From the models observed, and comparisons made, we see that Neural Nets provide the best results among the models seen so far, and additionally we can use more metrics for evaluation other than MAE, such as Root mean squared error to get a clearer picture of deviating values.

This will give us scope to improve on our model and reduce errors.

.

VII. CONCLUSION AND FUTURE WORK

- Artificial Neural Network performs better than SVR & Linear Regression
- Various visualizations of data are observed which helps in implementing the approaches for prediction
- Observations also indicate machine learning models won't work well for prediction of rainfall due to fluctuations in rainfall also because it is a natural phenomenon.
- But even if we want to show Rainfall prediction using ML algorithms, Artificial neural network is the best prediction algorithm because the Mean absolute error in this case is the least.

VIII. REFERENCES

Article link:
1) Rainfall Prediction Using Artificial Neural Networks
2) Rainfall Prediction using Machine Learning and Neural Network
3)Rainfall Prediction Using Neural Network
4) Rainfall prediction for the Kerala state of India using artificial intelligence approaches - ScienceDirect
5) A Survey On Rainfall Prediction Techniques
6) Evolving RBF neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm - ScienceDirect
7) https://www.ijsce.org/wp-content/uploads/papers/v5i6/D2689095415.pdf
8) A wavelet artificial neural network method for medium‐term rainfall prediction in Queensland (Australia) and the comparisons with conventional methods - Ghamariadyan - 2021 - International Journal of Climatology - Wiley Online Library
9) Development of Water Level Prediction Models Using Machine Learning in Wetlands: A Case Study of Upo Wetland in South Korea
10) Artificial neural network technique for rainfall forecasting applied to Alexandria, Egypt
11) A Survey on Rainfall Prediction using Artificial Neural Network
12) Development and Analysis of Artificial Neural Network Models for Rainfall Prediction by Using Time-Series Data
13) Monthly Rainfall Forecasting Using One-Dimensional Deep Convolutional Neural Network
14) A Survey on Rainfall Prediction using Artificial Neural Network
15) Annual and Non-Monsoon Rainfall Prediction Modelling Using SVR-MLP: An Empirical Study From Odisha
16) A Rainfall Prediction Model using Artificial Neural Network
17) Rainfall forecasting using neural network: A survey
18) Artificial Neural Network Models for Rainfall Prediction in Pondicherry
19) Prediction of Rainfall Using Intensified LSTM Based Recurrent Neural Network with Weighted Linear Units
20) Improving Monsoon Precipitation Prediction Using Combined Convolutional and Long Short Term Memory Neural Network
21) Analysis of rainfall prediction using machine learning data mining and satellite techniques | B.Shardoor | International Journal of Engineering & Technology
22) Soft Computing Models for Weather Forecasting
23) An Approach for Rainfall Prediction using Soft Computing
24) Performance comparison of artificial neural network models for daily rainfall prediction | SpringerLink
25) Rainfall prediction using hybrid neural network approach - IEEE Conference Publication
26) A rainfall prediction model using artificial neural network - IEEE Conference Publication
27) Performance Enhancement Model for Rainfall Forecasting Utilizing Integrated Wavelet-Convolutional Neural Network | SpringerLink
28) Surface and high-altitude combined rainfall forecasting using convolutional neural network | SpringerLink
29) Rainfall Prediction Using an Optimised Genetic-Artificial Neural Network Model
30) International Journal of Recent Technology and Engineering