

Design of Deep Learning Model for Predicting Rainfall

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering

by

RAVI SAI SARAN (38110455)

RELANGI GOWTHAM KRISHNA SAI (38110460)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

**Accredited with Grade “A” by NAAC | 12B Status by UGC | Approved by
AICTE**

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600 119

MAY – 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC I 12B Status by
UGC I Approved by AICTE

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **RAVI SAI SARAN (Reg. No. 38110455)** and **RELANGI GOWTHAM KRISHNA SAI (Reg. No.**

38110460) who carried out the project entitled "**Design of Deep Learning Model for Predicting Rainfall**" under my supervision from November 2021 to March 2022.

Internal Guide

Dr. R. Jeberson Retna Raj M.E., Ph.D.,

Head of the Department

Dr. L. Lakshmanan M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **RAVI SAI SARAN** hereby declare that the Project Report entitled “**Design of Deep Learning Model for Predicting Rainfall**” is done by under the guidance of **Dr. R. Jeberson Retna Raj M.E., Ph.D.**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D.**, Dean, School of Computing and **Dr. L. Lakshmanan M.E., Ph.D.**, and **Dr. S. Vigneshwari M.E., Ph.D.**, Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. R. Jeberson Retna Raj M.E., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Rainfall is one of the major source of freshwater for all the organism around the world. Rainfall prediction model provides the information regarding various climatological variables on the amount of rainfall.

In recent days, Deep Learning enabled the self-learning data labels which allows to create a data-driven model for a time series dataset. It allows to make the anomaly/change detection from the time series data and also predicts the future event's data with respect to the events occurred in the past.

This project deals with obtaining models of the rainfall precipitation by using Deep Learning Architectures (LSTM and ANN) and determining the better architecture with RMSE of LSTM and RMSE of ANN claiming that for any time series dataset, Deep Learning models will be effective and efficient for the modellers.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
1	INTRODUCTION	1
	1.1 INTRODUCTION	1
	1.2 NEURAL NETWORK	1
	1.3 ENSEMBLE LEARNING	2
	1.4 MACHINE LEARNING	3
2	LITERATURE SURVEY	4
	2.1 LITERATURE SURVEY 1	4
	2.2 LITERATURE SURVEY 2	5
	2.3 LITERATURE SURVEY 3	6
	2.4 LITERATURE SURVEY 4	6
	2.5 LITERATURE SURVEY 5	7
	2.6 LITERATURE SURVEY 6	7
	2.7 LITERATURE SURVEY 7	7
	2.8 LITERATURE SURVEY 8	8
3	AIM AND SCOPE OF THE INVESTIGATION	9
	3.1 AIM OF PROJECT	9
	3.2 SCOPE	9
4	EXPERIMENTAL OR MATERIALS AND METHODS USED	10
	4.1 SYSTEM REQUIREMENTS	10

4.1.1 HARDWARE SPECIFICATION	10
4.1.2 SOFTWARE SPECIFICATION	10
4.2 EXISTING SYSTEM	10
4.2.1 DRAWBACKS OF EXISTING SYSTEM	11
4.3 PROPOSED SYSTEM	11
4.3.1 ADVANTAGES	11
4.4 LSTM	11
4.5 ARTIFICIAL NEURAL NETWORK(ANN)	17
4.6 LANGUAGE SPECIFICATION	18
4.7 HISTORY OF PYTHON	18
4.8 APPLICATION OF PYTHON	18
4.9 FEATURES OF PYTHON	19
4.10 DATA FLOW DIAGRAM	19
4.11 ARCHITECTURAL DESIGN	21
4.12 MODULES	22
4.12.1 DATA COLLECTION	22
4.12.2 DATA CLEANING	22
4.12.3 DATA SELECTION	22
4.12.4 DATA PRE-PROCESSING	23
4.12.5 DATA PREDICTION	23
4.13 UML DIAGRAMS	23
4.13.1 GOALS	24
4.13.2 USE CASE DIAGRAM	24
4.13.3 SEQUENCE DIAGRAM	25

5	RESULTS AND DISCUSSIONS	27
	5.1 EXECUTING THE PROJECT	27
	5.2 RESULTS	27
6	SUMMARY AND CONCLUSIONS	31
	6.1 CONCLUSION	31
	6.2 FUTURE ENHANCEMENT	31
	REFERENCES	32
	APPENDICES	35
	A. SAMPLE CODE	35
	B. SCREENSHOTS	41

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
ANN	ARTIFICIAL NEURAL NETWORK
LSTM	LONG SHORT TERM MEMORY
AVG	AVERAGE

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
4.1	Squashing function	13
4.2	Structure of an LSTM network	13
4.3	Step 1	14
4.4	Step 2	15
4.5	Step 3	16
4.6	Step 4	17
4.7	Data Flow Diagram	20
4.8	Architecture Diagram	21
4.9	Use case Diagram	25
4.10	Sequence Diagram	26
5.1	The attributes of Dataset	27
5.2	Frequency of Avg rainfall	28
5.3	Comparison of avg rainfall with month	28
5.4	The Prediction of rainfall	29

LIST OF TABLES

TABLE No.	TABLE NAME	PAGE No.
5.1	PERFORMANCE OF LSTM	30
5.2	PERFORMANCE OF ANN	30

CHAPTER 1

INTRODUCTIO

N

1.1 INTRODUCTION

Rainfall Prediction is one of the most challenging tasks. Though already many algorithms have been proposed but still accurate prediction of rainfall is very difficult. In an agricultural country like India, the success or failure of the crops and water scarcity in any year is always viewed with greatest concern. A small fluctuation in the seasonal rainfall can have devastating impacts on agriculture sector. Accurate rainfall prediction has a potential benefit of preventing casualties and damages caused by natural disasters. Under certain circumstances such as flood and drought, highly accurate rainfall prediction is useful for agriculture management and disaster prevention. In this paper, various algorithms have been analyzed. Data mining techniques are efficiently used in rainfall prediction.

1.2 NEURAL NETWORK

Generally, a Neural Network (NN) contains a compilation of artificial neurons which are unified to execute some computation on input data and produces output forms. In deep learning, these layered portrayals are learned through models called neural network (NN), organized in exacting layers stacked on top of one another. The term neural organization is a reference to neurobiology, yet albeit a portion of the focal ideas in deep learning were created to some degree by drawing motivation from our comprehension of the mind, deep-learning models are not modelling of the cerebrum. There's no proof that the mind executes anything like the learning instruments utilized in current deep-learning models. You may run over pop-science articles broadcasting that deep learning works like the cerebrum or was designed according to the mind, yet that isn't the situation. It would be mistaking and counterproductive for newcomers to the field to consider deep learning being in any capacity identified with neurobiology; you needn't bother with that cover of "much

the same as our personalities" persona and secret, and you should fail to remember anything you may have found out about speculative connections between deep learning and science. For our motivations, deep learning is a numerical system for learning portrayals from information.

Deep learning has arrived at a degree of public consideration and industry speculation at no other time found throughout the entire existence of Artificial Intelligence, yet it isn't the primary effective type of machine learning. It's protected to say that a large portion of the machine-learning calculations utilized in the business today aren't deep learning calculations. Deep learning isn't generally the correct apparatus for the work—at times there isn't sufficient information for deep learning to be pertinent, and in some cases the issue is better tackled by an alternate calculation. In the event that deep learning is your first contact with machine learning, at that point you may end up in a circumstance where the sum total of what you have is the deep-learning hammer, and each machine-learning issue begins to resemble a nail.

1.3 ENSEMBLE LEARNING

In statistics and machine learning, ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. Unlike a statistical ensemble in statistical mechanics, which is usually infinite, a machine learning ensemble consists of only a concrete finite set of alternative models, but typically allows for much more flexible structure to exist among those alternatives.

An ensemble is itself a supervised learning algorithm, because it can be trained and then used to make predictions. The trained ensemble, therefore, represents a single hypothesis. This hypothesis, however, is not necessarily contained within the hypothesis space of the models from which it is built. Thus, ensembles can be shown to have more flexibility in the functions they can represent. This flexibility can, in theory, enable them to over-fit the training data more than a single model would, but in practice, some ensemble techniques (especially bagging) tend to reduce problems related to over-fitting of the training data. Empirically, ensembles tend to yield better results when there is a significant diversity among the models.

Many ensemble methods, therefore, seek to promote diversity among the models they

combine. Although perhaps non-intuitive, more random algorithms (like random decision trees) can be used to produce a stronger ensemble than very deliberate algorithms (like entropy-reducing decision trees). Using a variety of strong learning algorithms.

1.4 MACHINE LEARNING

Machine Learning combines computer science, mathematics, and statistics. Statistics is essential for drawing inferences from the data. Mathematics is useful for developing machine learning models and finally, computer science is used for implementing algorithms. However, simply building models is not enough. You must also optimize and tune the model appropriately so that it provides you with accurate results. Optimization techniques involve tuning the hyperparameters to reach an optimum result. The world today is evolving and so are the needs and requirements of people. Furthermore, we are witnessing a fourth industrial revolution of data. In order to derive meaningful insights from this data and learn from the way in which people and the system interface with the data, we need computational algorithms that can churn the data and provide us with results that would benefit us in various ways. Machine Learning has revolutionized industries like medicine, healthcare, banking, and several other industries. Therefore, Machine Learning has become an essential part of modern industry. Data is expanding exponentially and in order to harness the power of this data, added by the massive increase in computation power, Machine Learning has added another dimension to the way we perceive information. Machine Learning is being utilized everywhere. The electronic devices you use, the applications that are part of your everyday life are powered by powerful machine learning algorithms. With an exponential increase in data, there is a need for having a system that can handle this massive load of data. Machine Learning models like Deep Learning allow the vast majority of data to be handled with an accurate generation of predictions. Machine Learning has revolutionized the way we perceive information and the various insights we can gain out of it. These machine learning algorithms use the patterns contained in the training data to perform classification and future predictions. Whenever any new input is introduced to the ML model, it applies its learned patterns over the new data to make future predictions.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY 1

“Indian Summer Monsoon Rainfall (ISMR) Forecasting using Time Series Data: A Fuzzy-Entropy-Neuro based Expert System” Pritpal Singh 30 July 2018.

The authors presented a model to forecast Indian Summer Monsoon Rainfall on the basis of monthly and seasonal timescales. To forecast, time series dataset was used, spanning from 1871 till 2014. The dataset was classified in two parts (1) 1871-1960 used as training data, and (2) 1961- 2014 used as test data. Statistical analysis reported the dynamic nature of rainfall in monsoon, which could not be predicted effectively with mathematical and statistical models. So, the authors in this research recommended to use three techniques for this type of prediction: Fuzzy Set, Entropy and Artificial Neural Network. By using these three techniques, a forecasting model is developed to deal with the dynamic nature of the ISMR. In proposed model, fuzzy set theory is used to handle uncertainties which are inherited in dataset. The entropy computational concept was modified in this model and used to provide the input as a degree of membership in the entropy function. That entropy function was referred as Fuzzy Information-Gain (FIG). Then, each fuzzified rule was defuzzified using the ANN. The value of FIG of each fuzzy- set was then used as input into ANN. The proposed model was named as “Fuzzy-Entropy-Neuro Based Expert System for ISMR Forecasting” because it is the integration of fuzzy set, entropy and ANN. To evaluate the performance of proposed model following accuracy measures were used: Standard Deviations (SDs), Correlation Coefficient (CC), Root Mean Square Error (RMSE) and Performance Parameter (PP). According to results the proposed model is effective and efficient in comparison with other existing models.

2.2 LITERATURE SURVEY 2

“An Extensive Evaluation of Seven Machine Learning Methods for Rainfall Prediction in Weather Derivatives” Sam Cramer, Michael Kampouridis, Alex A. Freitas , Antonis K. Alexandridis 2017

The researchers in compared the predictive performance of latest and state of the art method named “Markov chain extended with rainfall prediction” with the other widely used machine learning techniques: Support Vector Regression, Genetic Programming, M5 Rules, M5 Model trees, Radial Basis Neural Networks, and k-Nearest Neighbors. Daily rainfall datasets were collected from 42 cities of two continents, with very diverse climatic features. 20 cities were selected from around the Europe and 22 from around the USA. There were two reasons of choosing two continents for data extraction, first is to perform the experiment on different climates having diverse weather and second was the geographical locations as the selected cities were very far apart from each other. The ultimate goal was to not bias the experiment to particular climate type or for particular geographic location. According to results the accumulating rainfall amounts can bring good results as compared to prediction using daily rainy data. While using the accumulated data, Support Vector Regression, Radial Basis Functions, and Genetic Programming overall performed well however Radial Basis Functions performed better then modern technique of “Markov chain”. For all selected datasets, each technique used the same parameters so it was not guaranteed that the best possible set of parameters was used for all the techniques. During the experiment, the researchers have noted a relationship between predictive accuracy and climatic attributes such as: volatile nature of rainfall, amount of maximum rainfall and the interquartile range of rainfall. Moreover no significant difference was noted in algorithms’ prediction error among the cities of both the continents (USA and Europe). Issue regarding the discontinuity in rainfall data was solved with the help of accumulated rainfall amounts.

2.3 LITERATURE SURVEY 3

“A Hybrid Model for Statistical Downscaling of Daily Rainfall” Sahar Hadi

Poura , Shamsuddin Shahida , Eun-Sung Chungb 2019

Authors in proposed a hybrid technique to downscale the daily rainfall by integrating two methods: 1) Random Forest, and 2) Support Vector Machine. RF was selected due to its robustness in classification and it was used to predict whether it will be rain or not whereas SVM were selected due to its feature to fit in non-linear data and it was used to predict the amount of rainfall, if it will occur. The proposed model was evaluated by downscaling daily rainfall at three stations, Dungun, Besut, and Kemaman on the east cost of peninsular, Malaysia. Daily rainfall time series data from 1961 till 2000 was collected from Department of Irrigation and Drainage Malaysia. Total of 26 climatic features were collected from National Centre for Environmental Prediction re analysis dataset, which were used as predictors for downscaling the models. To assess homogeneity in rainfall time series, various quality control activities were performed. Histograms for the dataset were created to reflect the problems moreover Student's t test was also used to identify any variance in the means between two segments of dataset which finally found homogeneous at all three locations. According to results the hybrid technique is capable to downscale the rainfall with Nash-Sutcliffe efficiency within range of 0.90-0.93, which is much higher than RF and SVM models.

2.4 LITERATURE SURVEY 4

Manjunath N, Muralidhar B R, Sachin Kumar S, Vamshi K, Mrs. Savitha P, "RAINFALL PREDICTION USING MACHINE LEARNING AND DEEP LEARNING TECHNIQUES : A REVIEW" 2021

Manjunath et al suggested a Rainfall forecasting model using ML and DL algorithms. Agriculture is one of the main business for economic activity in India. Annual rainfall is very important for agriculture activities. Also, the prediction of rainfall is difficult and it is important to know in advance about the rainfall in order to take precautions by protecting the crop from heavy downpour. For rainfall prediction ML algorithms are mostly used. However, the forecasts are never 100% accurate

2.5 LITERATURE SURVEY 5

Tanvi Patil, Dr. Kamal Shah, "Weather Forecasting Analysis using Linear and Logistic Regression Algorithm", International Research Journal of Engineering and Technology (IRJET) 2021

The objective of the system is to predict the weather over a given period. The weather condition at any instance is described by using different kinds of variables. Out of these attributes, only significant attributes are used in the process of weather prediction. The existing weather condition attributes are used to fit a model and by using the machine learning techniques and extrapolating the information, the future variations in the attributes are analyzed.

2.6 LITERATURE SURVEY 6

N. Divya Prabha, P. Radha, "Prediction Of Weather And Rainfall Forecasting Using Classification Techniques", International Research Journal of Engineering and Technology (IRJET) 2019

Contrasted with different spots where rainfall information isn't accessible, it consumes a large chunk of the day to build up a solid water overview for a long time. Improving complex neural organizations is intended to be a brilliant instrument for anticipating the stormy season. This downpour succession was affirmed utilizing a complex perceptron neural organization. Estimations like MSE (Early Modeling), NMSE (Usually Early Error), and the arrangement of informational collections for transient arranging are clear in the examination of different organizations, like Adanaive. AdaSVM.

2.7 LITERATURE SURVEY 7

Senthamil Selvi, Seetha N, "Rainfall Prediction by using Time-series data in analysis of ANN Mmodels", International Research Journal of Engineering and Technology (IRJET) 2019

In this paper, Artificial Neural Network (ANN) technique has been used to develop weather forecasting models for rainfall prediction using rainfall data of India. In this model, Feed Forward Neural Network (FFNN) using BackPropagation Algorithm has been used. The performance of both the models has - been assessed based on Regression Analysis, Mean Square Error (MSE) and Magnitude of Relative Error (MRE). This paper also gives some future directions for rainfall prediction research.

2.8 LITERATURE SURVEY 8

YashasAthreya, VaishaliBV, SagarK, SrinidhiHR, "Flood Prediction and Rainfall Analysis using Machine Learning", International Research Journal of Engineering and Technology (IRJET) 2021

Yashas Athreya et.al proposed a model based on Flood Prediction and Rainfall Analysis using Machine Learning algorithms. The main goal of employing this application is to prevent immediate impacts of the flood. This application can be easily used by the common people or government to predict the occurrence of flood beforehand. The Prediction of flood is done by making use of previous data, map the flood concentration as well , and then provide them necessary help through helpline for further evacuation or any other necessary precautions.

CHAPTER 3

AIM AND SCOPE OF PRESENT INVESTIGATION

3.1 AIM OF PROJECT

The purpose of this document is deep learning models for the prediction of rainfall using machine learning algorithms. In detail, this document will provide a general description of our project, including user requirements, product perspective, and overview of requirements, general constraints. In addition, it will also provide the specific requirements and functionality needed for this project – such as interface, functional requirements and performance requirements.

3.2 SCOPE

The scope of this SRS document persists for the entire life cycle of the project. This document defines the final state of the software requirements agreed upon by the customers and designers. Finally at the end of the project execution all the functionalities may be traceable from the SRS to the product. The document describes the functionality, performance, constraints, interface and reliability for the entire life cycle of the project.

CHAPTER 4

EXPERIMENTAL OR MATERIALS AND METHODS USED

4.1 SYSTEM REQUIREMENTS

4.1.1 Hardware Specification

- System - Windows 7/10
- Speed - 2.4GHZ
- Hard disk - 40GB
- Monitor - 15VGA color
- RAM - 4GB

4.1.2 Software Specification

- 4.1.2.1 Python
- 4.1.2.2 Jupyter Notebook
- 4.1.2.3 NumPy, Keras

4.2 EXISTING SYSTEM

Agriculture is the strength of our Indian economy. Farmer only depends upon monsoon to be their cultivation. The good crop productivity needs good soil, fertilizer and also good climate. Weather forecasting is the very important requirement of the each farmer. Due to the sudden changes in climate/weather, The people are suffered economically and physically. Weather prediction is one of the challenging problems in current state. The main motivation of this paper to predict the weather using various data mining techniques. Such as classification, clustering, decision tree and also neural networks. Weather related information is also called the meteorological data. In this paper the most commonly used weather parameters are rainfall, wind speed, temperature and cold.

4.2.1 DRAWBACKS OF EXISTING SYSTEM

Classification

Clustering

Decision tree

4.3 PROPOSED SYSTEM

Rainfall is significant for food conveyance, water assets, and many other ecological actions. Long haul dry spells or substantial rainfall during extreme development and advancement can altogether diminish yields. India is a rural nation and its economy is to a great extent dependent on farming creation. In this way, rainfall determining is turning out to be an ever-increasing number of significant in farming nations like India. Rainfall gauging has become one of the world's most squeezing science and innovation issues in ongoing hundreds of years. The model uses high-level architecture where it is trained and also fit using the deep learning algorithms and then the prediction is done to know how the model is performing. This model type helps in predicting the rainfall, hence it helps to reduce the damage on agricultural production and floods.

Here, we will be using Deep Learning algorithms for obtaining the best from that we use. Firstly we take dataset and then it is thoroughly goes under pre-processing and cleaning will be done. Once the data visualization is done to the dataset, then we split the dataset into test data set and we train the test data set with training dataset. Finally on comparison with both the algorithms (ANN and LSTM), subsequently we take the algorithm which gives the best accuracy. After that, we implement the model by that best model.

4.3.1 ADVANTAGES

Numerical Weather Prediction

Statistical Weather Prediction

Synoptic Weather Prediction

4.4 LSTM

Long Short-Term Memory networks, which are commonly known as "LSTMs," are a special kind of Recurrent Neural Networks that are capable enough of learning

long-term dependencies.

Numerous choices have been created to kill the issue of lost inclinations and rushes in profound reusing pipes. The most renowned of these is the Longevity of Remembrance. Typically, the LSTM audit segment attempts to "recall" all that the organization has seen up until this point, and to "overlook" it and the article data. This finishes with the presentation of a progression of exercises called "doors" for certain exercises. Each piece of the LSTM makes the forward cell push ahead, which is characterized as the chose and changed data over to be put away through the past LSTM areas. Long and short memory pipelines have 4 kinds of doors for a similar capacity depicted beneath: -

Failed to remember the Door (f): This decides the degree to which past data was disregarded.

Input entryway (I): This decides the quantity of data that can be written before the cell.

Entrance Door (g): This is regularly viewed as a piece of the entry, and many books on LSTM currently quit showing it and think it is a long way from the entry. It was utilized to change data over to compose inside the front cell of the information family and added a little line to the information and zeros. This is done to diminish the catch time, as Zeru- implies fast info. In spite of the fact that house to house travel is exceptionally restricted and is constantly controlled in a reasonable manner, it is a decent practice to do this house-to-house LSTM approach.

Leave Gate (o): This decides the result (the following mystery) made from the back to front.

Long Short-Term Memory networks, which are commonly known as "LSTMs," are a special kind of Recurrent Neural Networks that are capable enough of learning long-term dependencies.

What are long-term dependencies?

It has happened many times that we only require recent data in order to perform questions in a model. But at the same time, we may also need data that has been previously obtained.

Consider the following example to have a better understanding of it.

Let's suppose there is a language model, which is trying to predict the next word on the basis of the previous ones. Assume that we are trying to predict the last

word in the sentence say, "The car runs on the road".

Here the context is quite simple because the last word always ends up being a road. By incorporating Recurrent Neural Networks, the gap present between the former information and the existing necessities can be easily associated.

That is the reason why Vanishing and Exploding Gradient problems do not exist, followed by making this LSTM networks to easily handle long-term dependencies. LSTM encompasses a layer of neural network in the form of a chain. In a standard recurrent neural network, the repeating module consists of one single function as shown in the image given below:

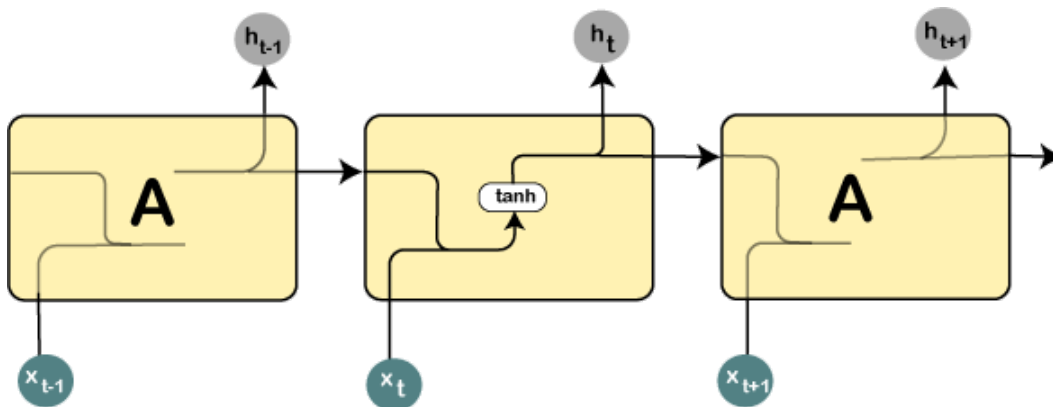


Fig 4.1 Squashing function

From the image given above, it can be seen that there is a tanh function in the layer, which is called as squashing function. So, what is a squashing function?

The squashing function is mainly used in between the range of -1 to +1 so that the values can be manipulated on the basis of inputs.

Now, let us consider the structure of an LSTM network:

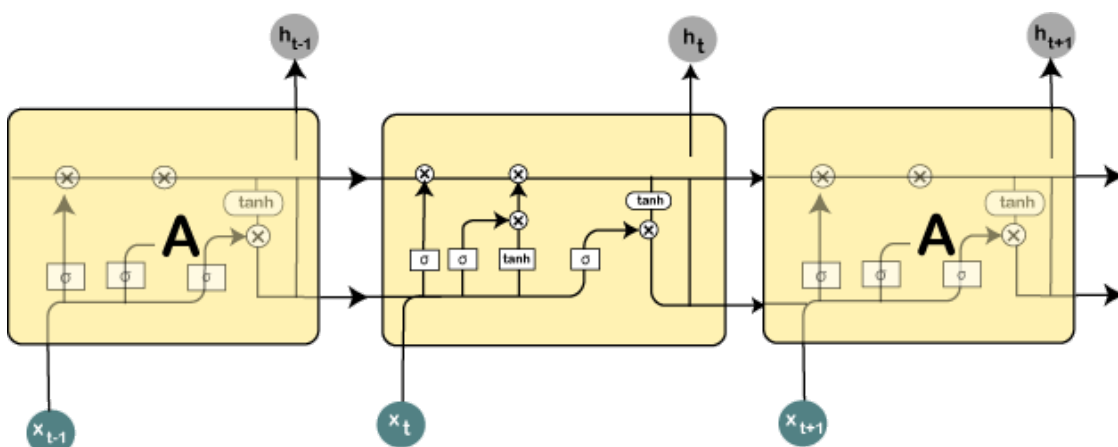


Fig 4.2 Structure of an LSTM network

Here all those functions that are present in the layers have their own structures as and when it comes to LSTM networks. The cell state is represented by the horizontal line, acts as a conveyor belt that carries the data linearly crossways the data channel.

Let us consider a step-by-step approach to understand LSTM networks better.

Step 1:

The first step in the LSTM is to identify that information which is not required and will be thrown away from the cell state. This decision is made by a sigmoid layer, which called the forget gate layer.

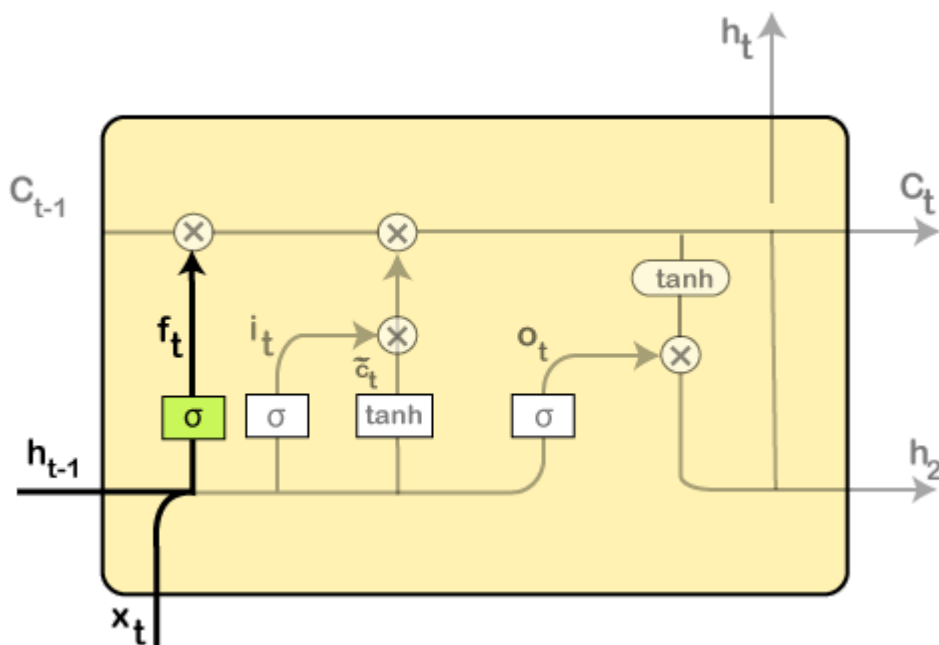


Fig 4.3 Step 1

The highlighted layer in the above is the sigmoid layer which is previously mentioned.

The calculation is done by considering the new input, and the previous timestamp is, which eventually leads to the output of a number between 0 and 1 for each number in that cell state.

As a typical binary, 1 represents to keep the cell state while 0 represents to trash it.

$$f_t = \sigma(w_f [h_{t-1}, x_t] + b_f)$$

where, w_f = Weight

h_{t-1} = Output from previous

timestamp x_t = New input

bf = Bias

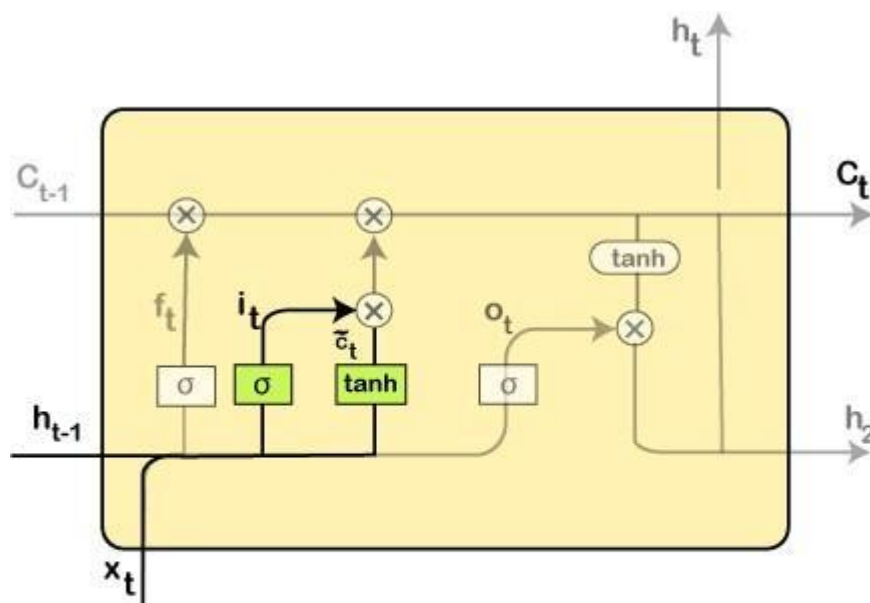
Considering a gender classification problem, it necessitates observing the correct gender when we are using the network.

Step 2:

Next, we will decide which information we will store in the cell state. It further consists of the following steps:

The sigmoid layer, which is also known as the "input gate layer," will make decisions about those values that are needed to be updated.

A vector of new candidate values is created so that they can be added to the state by the tanh layer.



$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$$

$$\tilde{c}_t = \tanh(w_c[h_{t-1}, x_t] + b_c)$$

Fig 4.4 Step 2

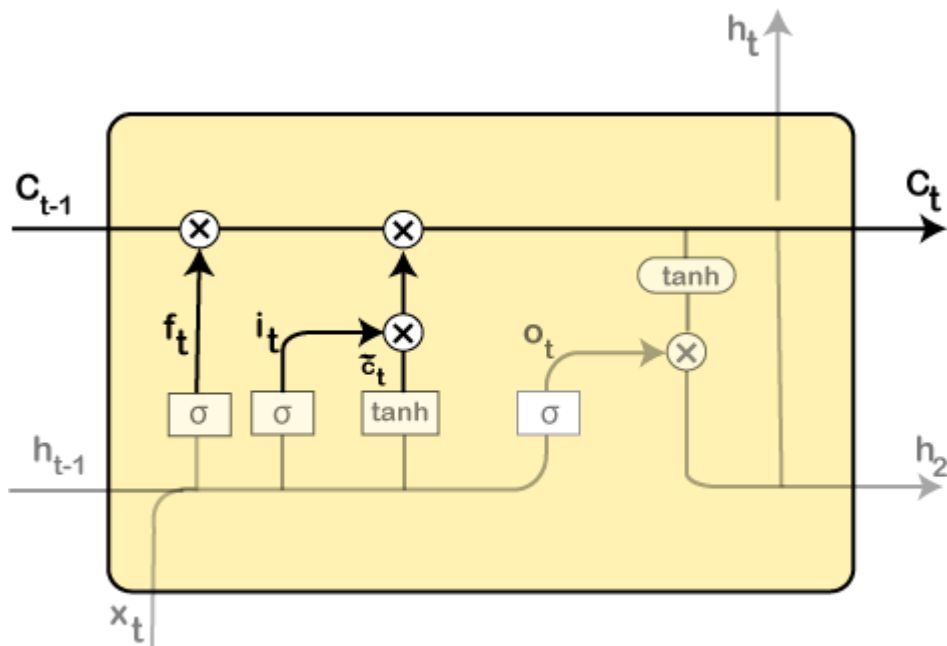
Then the new input, as well as the preceding timestamp's input, will get passed through a sigmoid function that will result in the value $i(t)$, which will then be multiplied by $c(t)$ followed by adding it to the cell state.

In the next step, we will combine both of them so as to update the state. Step 3:

In the 3rd step, the previous cell state C_{t-1} will get updated into the new cell

state Ct.

And for that, we will need to multiply the old state (C_{t-1}) by $f(t)$, keeping the things aside that we thought that we earlier decided to leave.



$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$$

Fig 4.5 Step 3

Next, we will add $i_t * \tilde{c}_t$, which is the new candidate values. It has been actually scaled by how much we wanted to update each state value.

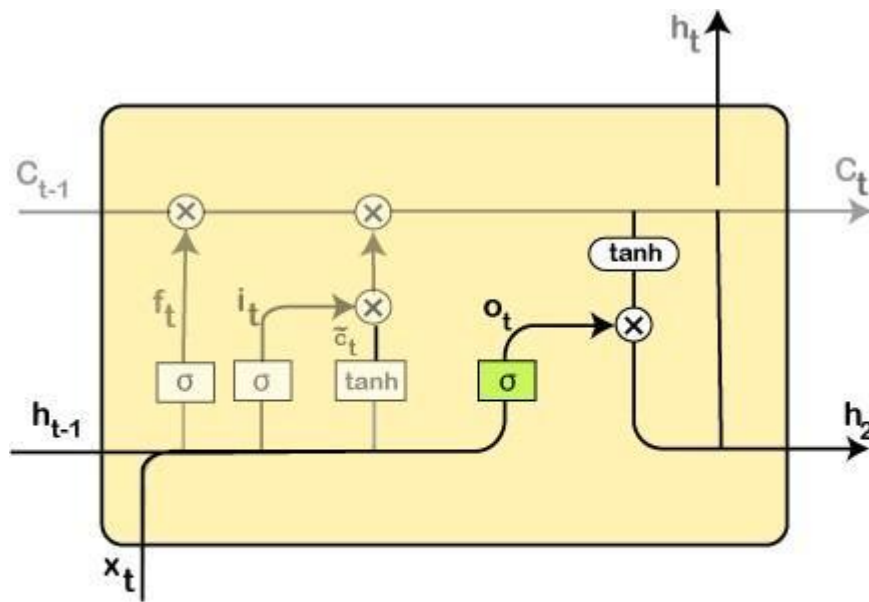
In the second step, we decided to do make use of the data, which is only required at that stage. However, in the third step, we have executed it.

Step 4:

In the 4th step, we will run the sigmoid layer that will decide for those parts of the cell state that will result in the output.

Next, we will put the cell state through tanh, which means we will be pushing the values in between the range of -1 and 1.

And then, further, we will multiply it with the sigmoid gate's output so that only the decided parts results in the output.



$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(c_t)$$

Fig 4.6 Step 4

4.5 ARTIFICIAL NEURAL NETWORK(ANN)

An artificial neural network typically contains three or more layers, all of which are interconnected. The first layer comprises input neurons, which will send data to deeper layers before sending final output data to the last output layer in a sequential order. The inner layers are now hidden, and they are formed by units that are responsible for the variation in information received from layer to layer. In this case, each layer serves as an input and output layer, allowing the ANN to comprehend complex concepts. The inner layers that are present here are referred to as the Neural layer. The units in the neural layer are interested in learning more about the information and using the ANN's internal mechanism to do so. Only when these principles are fulfilled do the units provide a changed result, which is then authorized as an output to the following layer.

There are some more set of learning rules that make use of backpropagation. It sticks to a process like ANN can be able to adjust it's output results taking the consideration of errors that are in the account. The information is sent backward every time whenever the output is labelled as an error. In this way the weight gets

updated proportionally for responsibility of the error.

Thereupon errors are used to revise the weight of Artificial neural networks unit connections to know about the dissimilarities among actual one and the desired outcome. Eventually ANN will learn to reduce the chances of getting errors or unwanted results.

4.6 LANGUAGE SPECIFICATION

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language.

4.7 HISTORY OF PYTHON

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

4.8 APPLICATION OF PYTHON

- Easy-to-learn – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- Easy-to-read – Python code is more clearly defined and visible to the eyes.
- Easy-to-maintain – Python's source code is fairly easy-to-maintain.
- A broad standard library – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- Interactive Mode – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- Portable – Python can run on a wide variety of hardware platforms and has

the same interface on all platforms.

- Extendable – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- Databases – Python provides interfaces to all major commercial databases.
- GUI Programming – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- Scalable – Python provides a better structure and support for large programs than shell scripting.

4.9 FEATURES OF PYTHON

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java

4.10 Data Flow Diagram

- The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
- The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
- DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
- DFD is also known as bubble chart. A DFD may be used to represent a

system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

- The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
- The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
- DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
- DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

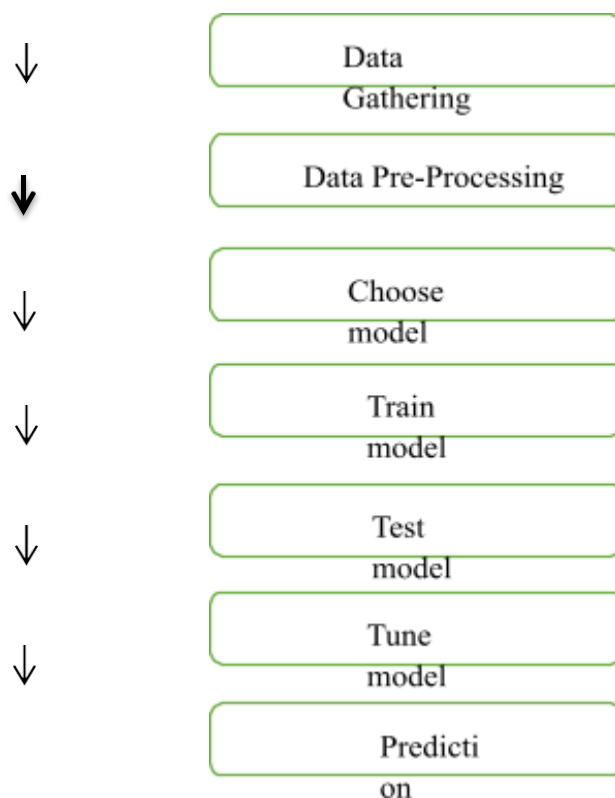


Fig 4.7 Data Flow Diagram

4.11 ARCHITECTURAL DESIGN

The system architecture of the proposed system is shown in figure 3.1. The first and most important thing to work with deep learning models, we need to have huge amount of data because without data, one cannot train DL/AI models. One of the most crucial parts while creating a Deep learning model is Collecting and preparing the dataset. If the dataset is not well prepared or not pre-processed the technology that is applied behind the DL/AI project cannot work. We got this dataset from Kaggle and it has the data that point out the features on climatological normal of district wise rainfall (in mm) which is estimated with the data of the period 1951- 2015.

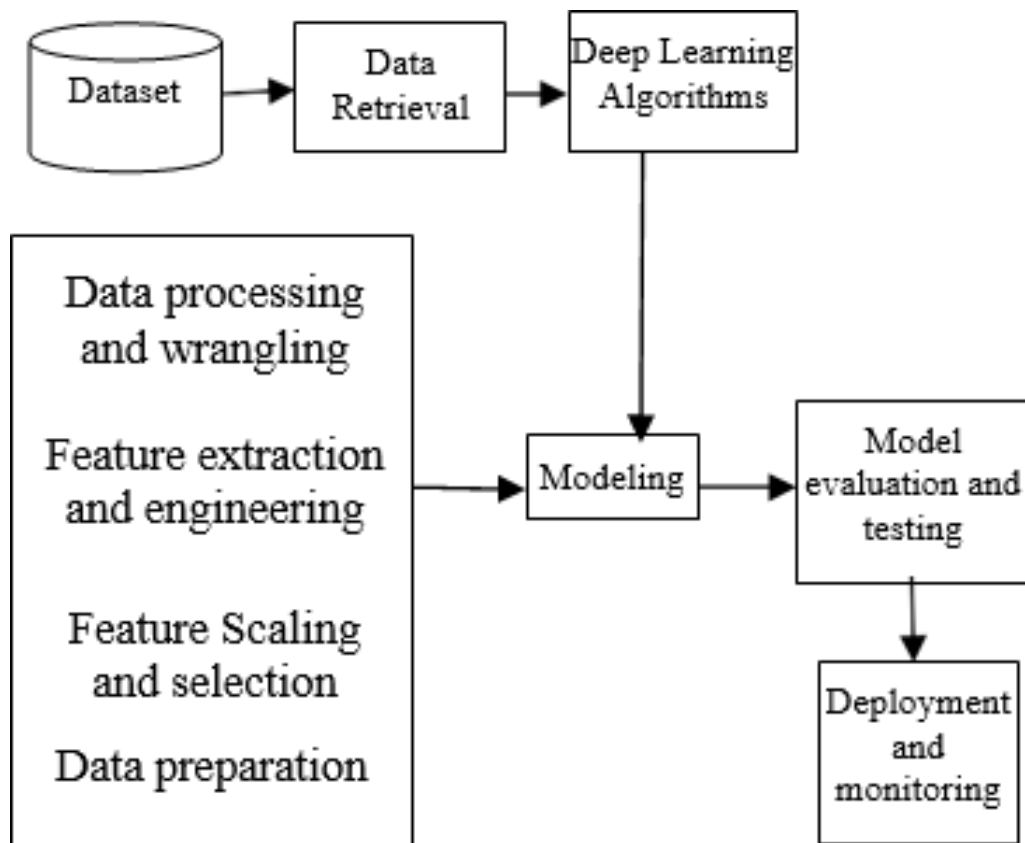


Fig: 4.8 architecture diagram

While coming into preparation, the next thing we do is Data Pre-processing. It is the procedure of preparing the raw data and making it suitable for a Deep Learning model, it's a crucial step while creating a model. This process is important for cleaning the data and making sure whether it is suitable for making a DL model, it also enhances the accuracy and efficiency of the model.

In data pre-processing it has some steps to be followed are acquiring the dataset, importing libraries and datasets, find the missing data, splitting into training set and testing set, and finally feature scaling. The libraries that we imported are NumPy, pandas, matplotlib, time and OS. We now divide the dataset into two parts: training and testing. The training set is a subset of the dataset used to train the deep learning model. While the test set is a subset of the dataset used to test the deep learning model, the model predicts the output using the test set. Final step in Pre-processing is Feature Scaling. This technique is used to standardize the independent variables of the dataset in a specific range. Here we put all our variables in same range e.g all the variables are scaled between -1 to 1. And then we use Deep learning algorithms LSTM and ANN for the prediction of rainfall.

4.12 MODULES

4.12.1 Data Collection

It is known as the procedure of collecting information from distinct sources. In an effort to use the data we accumulate to develop practical and deep learning solutions, it must be gathered and reserved such a way that makes sense for the problem additionally make correct use of it. The data utilized in this action was gathered by the meteorological association. The conversation covers the period from 1901 to 2015. This segment of the review talks about the accompanying systems: Cleaning the data, picking the data, changing the data, cleaning the data.

4.12.2 Data Cleaning

Data cleaning is to recognize and eliminate errors and duplicate data, in an effort to create a good dataset. This enhances the quality of the training data for analytics and enables precise decision-making. At last the information cleaning framework was changed into an effective information mining framework.

4.12.3 Data Selection

Data selection is the process of defining the appropriate source and data type, it also picks the suitable instruments to collect the data. At this stage, the data identified with the examination tree is the choice tree and is extricated from the

informational index. The meteorological informational index had ten attributes that

pre-owned two elements later on. Because of the idea of cloud information, all qualities are the equivalent, and the vast majority of the qualities that are absent in sunlight-based information are not utilized in the investigation.

4.12.4 Data Pre-Processing

Data Preprocessing is here Information handling is the method involved with getting ready essential data and involving it in AI. This is the first and most significant stage in the advancement of AI. When constructing a vehicle preparing machine, we don't consistently get spotless and helpful data, procedure of developing the raw data and make it suitable for accomplishing any instructive work. In this manner, we use information handling data.

4.12.5 Data Prediction

Data prediction - The proposed model uses an inside and out text data transformation calculation that contains key components of computerized information in the information section step and gives mechanical progression at the expectation stage.

4.13 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to

express the design of software projects.

4.13.1 GOALS:

The Primary goals in the design of the UML are as follows:

Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.

Provide extendibility and specialization mechanisms to extend the core concepts.

Be independent of particular programming languages and development process.

Provide a formal basis for understanding the modeling language.

Encourage the growth of OO tools market.

Support higher level development concepts such as collaborations, frameworks, patterns and components.

Integrate best practices.

4.13.2 USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

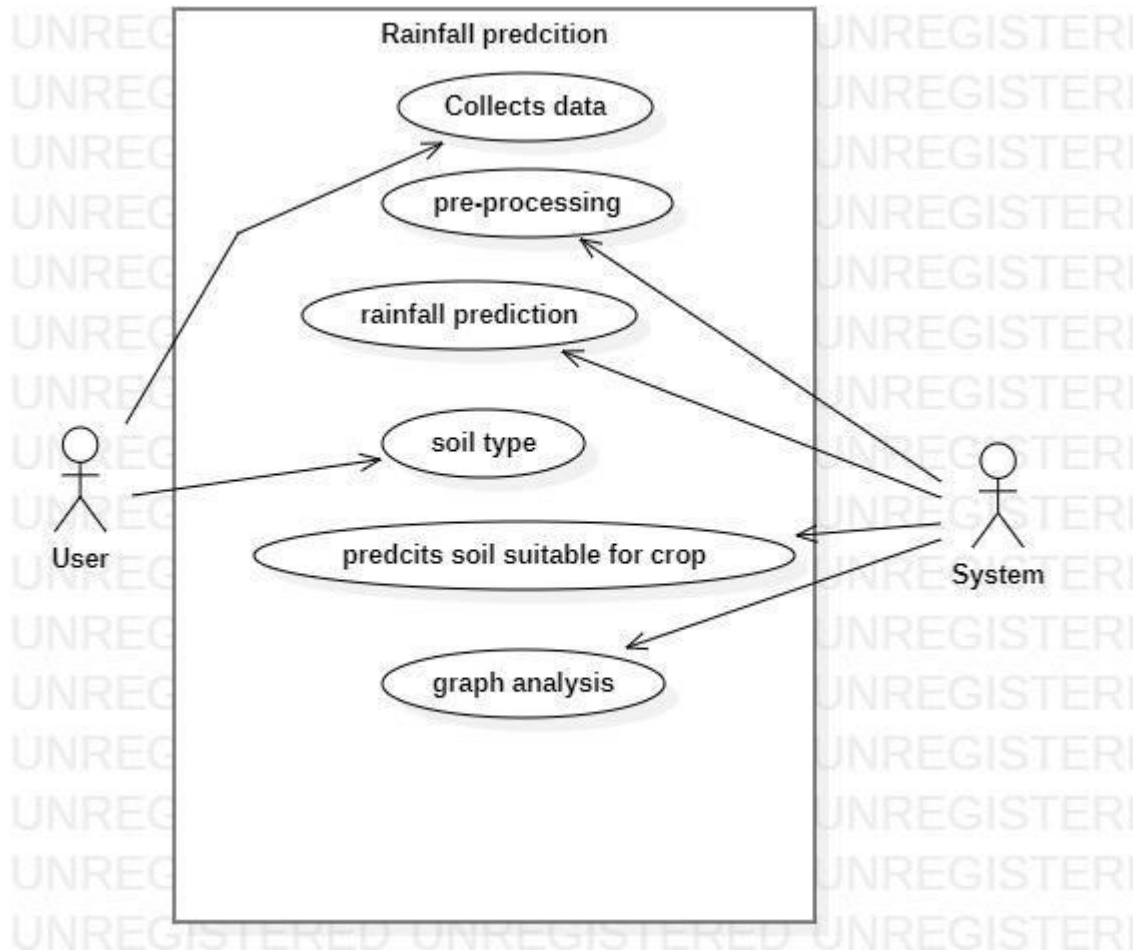


Fig 4.9 Use case diagram

4.13.3 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

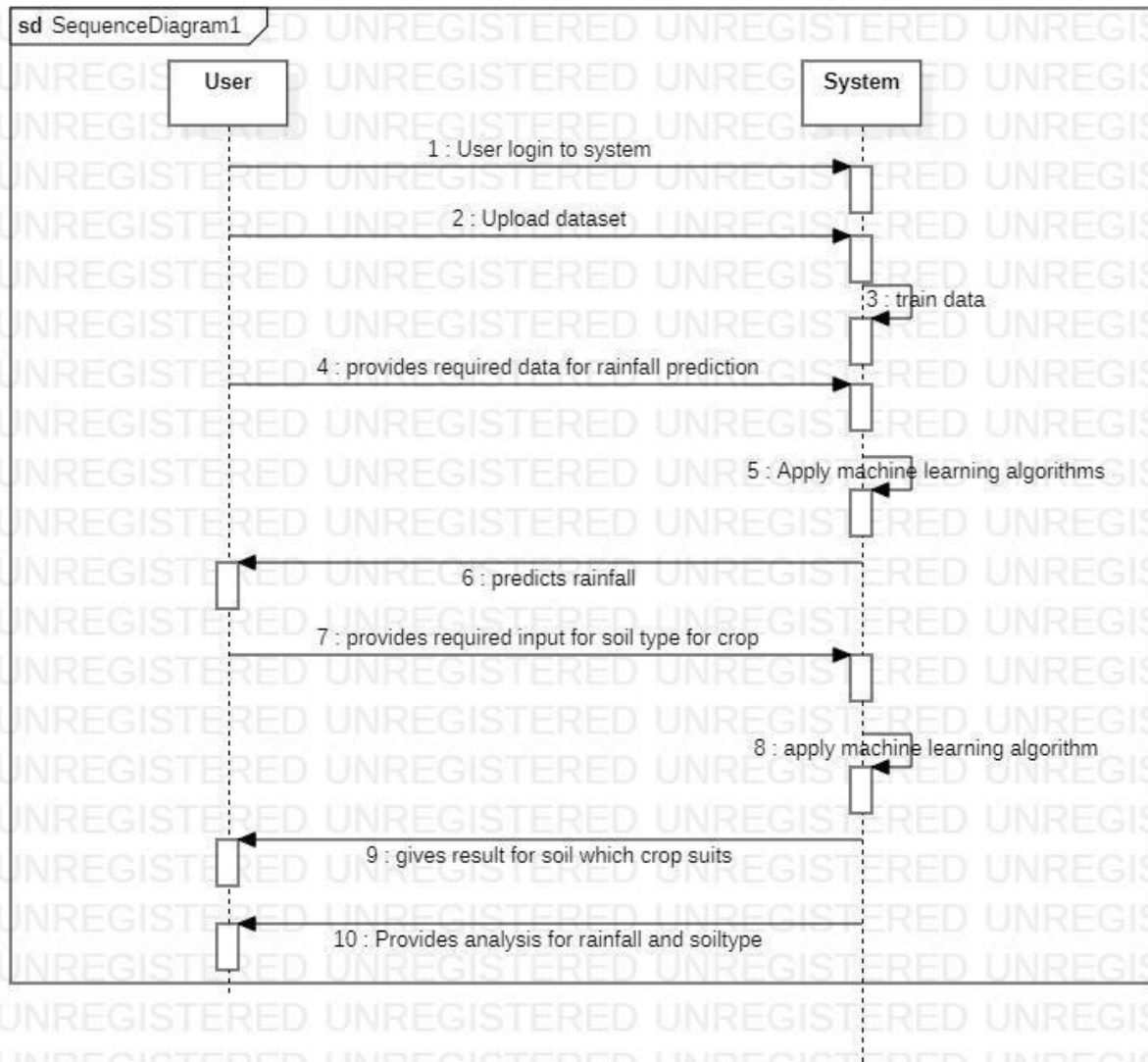


Fig 4.10 Sequence Diagram

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 EXECUTING THE PROJECT

The system is implemented with the dataset called rainfall in India 1901-2015 from Kaggle, where we further preprocess the data and selected the attributes called year, month, avg_rainfal and date for prediction of the rainfall.

5.2 RESULTS

	INDEX	YEAR	Month	avg_rainfall	Date
0	0	1901	1	51.8	1901-01-01
115	115	1901	2	19.6	1901-02-01
230	230	1901	3	11.9	1901-03-01
345	345	1901	4	1.1	1901-04-01
460	460	1901	5	65.6	1901-05-01
575	575	1901	6	66.3	1901-06-01
690	690	1901	7	245.9	1901-07-01
805	805	1901	8	319.4	1901-08-01
920	920	1901	9	155.1	1901-09-01
1035	1035	1901	11	7.3	1901-11-01
1150	1150	1901	12	0.1	1901-12-01
1	1	1902	1	4.6	1902-01-01

Fig 5.1 The attributes of dataset

We have taken the data set called rainfall in india 1901- 2015 from Kaggle where we further preprocessed the data and selected the attributes called year, month, avg_rainfal and date for prediction of the rainfall.

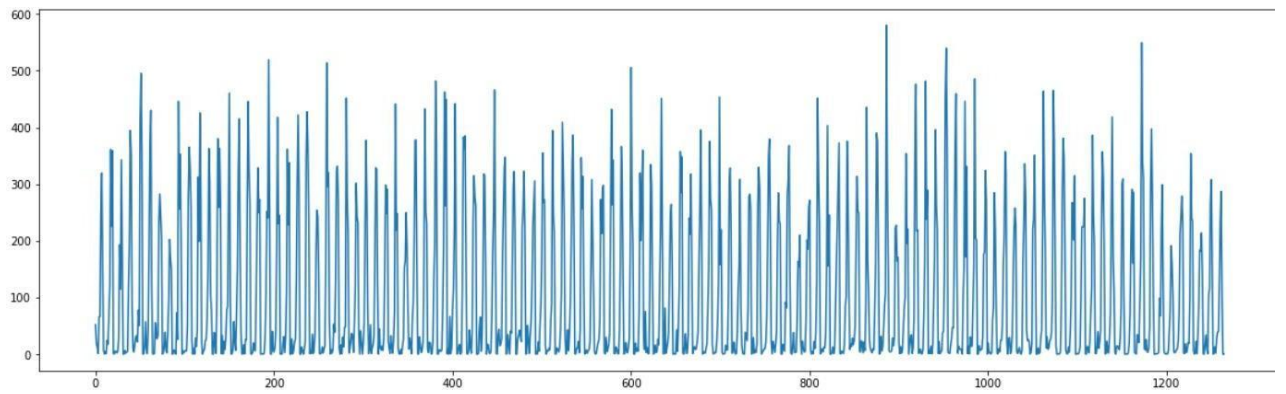


Fig 5.2 Frequency of average rainfall

Here we plotted this graph on basis of avg_rainfall attribute. This graph describes the frequency of avg_rainfall from the dataset.

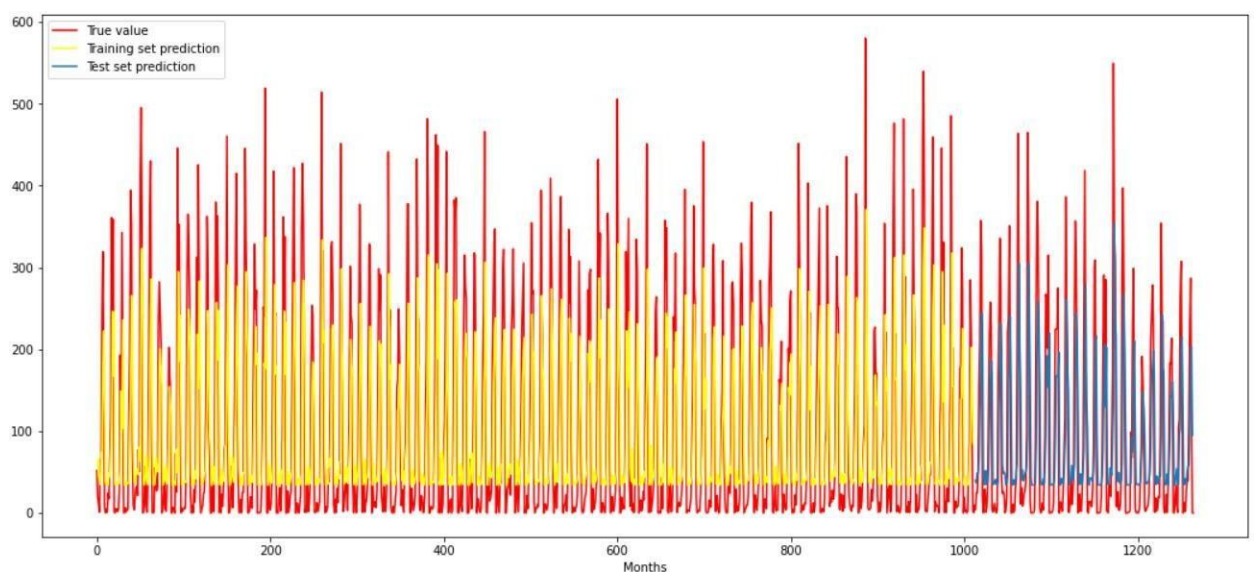


Fig 5.3 Comparision of average rainfall value with month

Figure 5.3 graph shows that the comparison of Avg rainfall value with months. This graph also compares the true values with training set predictions and test set predictions.

```
↳ ['No rain',  
    'No rain',  
    'No rain',  
    'No rain',  
    'No rain',  
    'Rain',  
    'Rain',  
    'Rain',  
    'Rain',  
    'No rain',  
    'No rain',  
    'No rain',  
    'No rain',  
    'No rain',  
    'No rain',  
    'No rain',  
    'Rain',  
    'Rain',  
    'Rain',  
    'Rain',  
    'No rain',  
    'No rain',  
    'No rain',  
    'No rain',  
    'No rain',  
    'No rain',  
    'No rain',  
    'Rain',  
    'Rain',  
    'Rain',  
    'Rain',  
    'No rain',
```

Fig 5.4 The prediction of rainfall

After comparing the accuracies of both LSTM and ANN we have selected LSTM algorithm for prediction of rainfall and rainfall is predicted as per the figure above. If the avg_rainfal is above 100 it is predicted as “Rain” or else if the value of avg_rainfall is below 100 it is predicted as “No rain”.

LSTM	accuracy	RMSE
Train Data	96.57%	107.35
Test Data	94.45%	94.53

Table 5.1 Performance of LSTM algorithm

The above table shows the accuracy and rmse values when we are using lstm for prediction. As mentioned above the accuracy of train data and test data is 96.57% and 94.45% respectively. The RMSE values of train data and test data are 107.35 and 94.53 respectively. Table II shows the accuracy of ANN algorithm in prediction process.

ANN	Accuracy	RMSE
Train data	58.71%	109.28
Test data	94.45%	97.31

Table 5.2 Performance of ANN algorithm

The above table shows the accuracy and rmse values when we are using ANN for prediction. As mentioned above the accuracy of train data and test data is 58.71.57% and 94.45% respectively. the Root Means Square Error (RMSE) values of train data and test data are 109.28 and 97.31 respectively. Both ANN and LSTM achieve same in accuracy in the given dataset and the RMSE value is lower in LSTM comparing with ANN. Regarding RMSE value, lower the value is better in performance. In such case, LSTM returns better performance.

CHAPTER 6

SUMMARY AND CONCLUSIONS

6.1 CONCLUSION

Forecasting the weather is challenging and to achieve better accuracy is complex due to the inherent nature of environment and climatic conditions. It is subjective to change because of dynamic nature of environmental properties. The deep learning based prediction of rainfall is successfully implemented using the available data sets. With the applications of LSTM and ANN algorithms, the system predict the rainfall for a given time periods. The dataset includes daily records of parameters like avg_rainfall, month, year, date. The experimental analysis clearly demonstrates that the proposed model bring about a promising outcome. These algorithms LSTM and ANN based predictive model is suitable for numerous applications requiring rainfall prediction, Agriculture as such.

6.2 FUTURE ENHANCEMENT

In future, the severity of damages, loss of crop, vegetation, agriculture can be estimated with the help of other deep learning algorithms. Later on, we intend to build the work on harvest and yield estimating and downpour gauging.

REFERENCES

- [1] Singh, Pritpal. (2017). Indian summer monsoon rainfall (ISMR) forecasting using time series data: A fuzzy-entropy-neuro based expert system. *Geoscience Frontiers*. 9. 10.1016/j.gsf.2017.07.011.
- [2] Cramer, Sam & Kampouridis, Michael & Freitas, Alex & Alexandridis, Antonios. (2017). An extensive evaluation of seven machine learning methods for rainfall prediction in weather derivatives. *Expert Systems with Applications*. 85. 10.1016/j.eswa.2017.05.029.
- [3] Hadipour, Sahar & Shahid, Shamsuddin & Chung, Eun-Sung. (2016). A Hybrid Model for Statistical Downscaling of Daily Rainfall. *Procedia Engineering*. 154. 1424-1430.10.1016/j.proeng.2016.07.514.
- [4] Manjunath N, Muralidhar B R, Sachin Kumar S, Vamshi K, Mrs. Savitha P, "RAINFALL PREDICTION USING MACHINE LEARNING AND DEEP LEARNING TECHNIQUES : A REVIEW", *International Research Journal Of Engineering And Technology (IRJET)* e-ISSN: 2395-0056 Volume: 08 Issue: 08 | Aug 2021 www.irjet.net p-ISSN: 2395-0072.
- [5] Tanvi Patil, Dr. Kamal Shah, "Weather Forecasting Analysis using Linear and Logistic Regression Algorithm", *International Research Journal of Engineering and Technology (IRJET)* e-ISSN: 2395-0056 Volume: 08 Issue: 06 | June 2019 www.irjet.net p-ISSN: 2395-0072.
- [6] N. Divya Prabha, P. Radha, "Prediction Of Weather And Rainfall Forecasting Using Classification Techniques", *International Research Journal of Engineering and Technology (IRJET)* e-ISSN: 2395-0056 Volume: 06 Issue: 02 | Feb 2019 www.irjet.net p-ISSN: 2395-0072.
- [7] Senthamil Selvi, Seetha N, "Rainfall Prediction by using Time-series data in analysis of ANN Models", *International Research Journal of Engineering*

- [8] YashasAthreya, VaishaliBV, Sagark, SrinidhiHR, "Flood Prediction and Rainfall Analysis using Machine Learning", International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 08 Issue: 07 | July 2021 www.irjet.net p-ISSN: 2395-0072.
- [9] O. M. S and A. J P, "MapReduce and Optimized Deep Network for Rainfall Prediction in Agriculture," in The Computer Journal, vol. 63, no. 1, pp. 900-912, Jan. 2020, doi: 10.1093/comjnl/bxz164.
- [10] X. Zhang, S. N. Mohanty, A. K. Parida, S. K. Pani, B. Dong and X. Cheng, "Annual and Non-Monsoon Rainfall Prediction Modelling Using SVR-MLP: An Empirical Study From Odisha," in IEEE Access, vol. 8, pp. 30223-30233, 2020, doi: 10.1109/ACCESS.2020.2972435.
- [11] N. K. A. Appiah-Badu, Y. M. Missah, L. K. Amekudzi, N. Ussiph, T. Frimpong and E. Ahene, "Rainfall Prediction Using Machine Learning Algorithms for the Various Ecological Zones of Ghana," in IEEE Access, vol. 10, pp. 5069-5082, 2022, doi: 10.1109/ACCESS.2021.3139312.
- [12] Y. Kim and S. Hong, "Very Short-Term Rainfall Prediction Using Ground Radar Observations and Conditional Generative Adversarial Networks," in IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1-8, 2022, Art no. 4104308, doi: 10.1109/TGRS.2021.3108812.
- [13] M. I. Khan and R. Maity, "Hybrid Deep Learning Approach for Multi-Step-Ahead Daily Rainfall Prediction Using GCM Simulations," in IEEE Access, vol. 8, pp. 52774-52784, 2020, doi: 10.1109/ACCESS.2020.2980977.
- [14] S. Manandhar, S. Dev, Y. H. Lee, Y. S. Meng and S. Winkler, "A Data-Driven Approach for Accurate Rainfall Prediction," in IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 11, pp. 9323-9331, Nov. 2019,

doi: 10.1109/TGRS.2019.2926110.

- [15] A. Naha Biswas, Y. H. Lee and S. Manandhar, "Rainfall Forecasting Using GPS-Derived Atmospheric Gradient and Residual for Tropical Region," in IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1-10, 2022, Art no. 5802110, doi: 10.1109/TGRS.2021.3131217.

APPENDICES

A. SAMPLE CODE

```
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

from matplotlib import

rcParams

rcParams['figure.figsize']=10,6

from keras.layers.core import Dense, Activation,

Dropout from keras.layers.recurrent import LSTM

from keras.models import Sequential

import time

from sklearn.metrics import

mean_squared_error from matplotlib import

pyplot

from sklearn.preprocessing import MinMaxScaler

from sklearn.preprocessing import StandardScaler

# Input data files are available in the "../input/" directory.

# For example, running this (by clicking run or pressing

Shift+Enter) will list the files in the input directory

import os

# Any results you write to the current directory are saved as output.
```

```
dataset=pd.read_csv("rainfall in india 1901-2015.csv",encoding =  
"ISO- 8859-1")
```

```
dataset.dtypes
```

```

groups =
dataset.groupby('SUBDIVISION')[['YEAR','JAN','FEB','MAR','APR','MAY',
','JUN','JUL','AUG','SEP','NOV','DEC']]

data=groups.get_group(('BIHAR'))

data.head()

data=data.melt(['YEAR']).reset_index()

data.head()

df=
data[['YEAR','variable','value']].reset_index().sort_values(by=['YEAR','in
dex'])

df.head()

df.columns=['INDEX','YEAR','Month','avg_rainfall'

]

d={'JAN':1,'FEB':2,'MAR'
:3,'APR':4,'MAY':5,'JUN':6,'JUL':7,'AUG':8,'SEP':9,

'OCT':10,'NOV':11,'DEC':12}

df['Month']=df['Month'].map(d

) df.head(12)

df['Date']=pd.to_datetime(df.assign(Day=1).loc[:,['YEAR','Month','Day']]

)

df.head(12)

cols=['avg_rainfall']

dataset=df[cols]

```

```
dataset.head()
```

```
series=dataset
```

```
series.head()
```

```
series.shape
```

```
pyplot.figure(figsize=(20,6))
```

```
) pyplot.plot(series.values)
```

```
pyplot.show()
```

```
# Get the raw data values from the pandas data
```

```
frame. data_raw = series.values.astype("float32")
```

```
# We apply the MinMax scaler from sklearn
```

```
# to normalize data in the (0, 1) interval.
```

```
scaler = MinMaxScaler(feature_range = (0,
```

```
1)) dataset = scaler.fit_transform(data_raw)
```

```
# Print a few values.
```

```
dataset[0:5]
```

```
# Using 60% of data for training, 40% for validation.
```

```
TRAIN_SIZE = 0.80
```

```
train_size = int(len(dataset) * TRAIN_SIZE)
```

```
test_size = len(dataset) - train_size
```

```
train, test = dataset[0:train_size, :], dataset[train_size:len(dataset), :]
```

```
print("Number of entries (training set, test set): " + str((len(train), len(test))))
```

```
# FIXME: This helper function should be rewritten using numpy's shift function. See  
below.
```

```
def create_dataset(dataset, window_size =
```

```
1): data_X, data_Y = [], []
```

```
for i in range(len(dataset) - window_size - 1):
```

```
    a = dataset[i:(i + window_size), 0]
```

```
data_X.append(a)
```

```

        data_Y.append(dataset[i + window_size, 0])
    return(np.array(data_X), np.array(data_Y))

# Create test and training sets for one-step-ahead regression.
window_size = 1
train_X, train_Y = create_dataset(train,
window_size) test_X, test_Y = create_dataset(test,
window_size) print("Original training data shape:")
print(train_X.shape)

# Reshape the input data into appropriate form for Keras.
train_X = np.reshape(train_X, (train_X.shape[0], 1, train_X.shape[1]))
test_X = np.reshape(test_X, (test_X.shape[0], 1, test_X.shape[1]))
print("New training data shape:")
print(train_X.shape)

def fit_model(train_X, train_Y, window_size = 1):
    model = Sequential()
    model.add(LSTM(2000,activation = 'tanh', input_shape = (1, window_size)))
    model.add(Dropout(0.2))
    model.add(Dense(500))
    model.add(Dropout(0.4))
    model.add(Dense(500))
    model.add(Dropout(0.4))
    model.add(Dense(400))
    model.add(Dropout(0.4))
    model.add(Dense(1, activation = 'linear'))
    model.compile(loss =
"mean_squared_error",
                    optimizer = "adam",metrics=["accuracy"])
    model.fit(train_X,
        train_Y,
        epochs =

```

```
10,  
batch_size = 64,  
)
```

```

    return(model)

# Fit the first model.
model1 = fit_model(train_X, train_Y, window_size)
scores = model1.evaluate(train_X, train_Y)
print("Traning Accuracy:",100-scores[0]*100," %")
print("Testing Accuracy:",100-scores[1]*100," %")

import math

def predict_and_score(model, X, Y):
    # Make predictions on the original scale of the data.
    pred = scaler.inverse_transform(model.predict(X))
    # Prepare Y data to also be on the original scale for interpretability.
    orig_data = scaler.inverse_transform([Y])
    # Calculate RMSE.
    score = math.sqrt(mean_squared_error(orig_data[0], pred[:, 0]))
    return(score, pred)

rmse_train, train_predict = predict_and_score(model1, train_X, train_Y)
rmse_test, test_predict = predict_and_score(model1, test_X, test_Y)
print("Training data score: %.2f RMSE" % rmse_train)
print("Test data score: %.2f RMSE" % rmse_test)

# Start with training predictions.
train_predict_plot = np.empty_like(dataset)
train_predict_plot[:, :] = np.nan
train_predict_plot[window_size:len(train_predict) + window_size, :] = train_predict

# Add test predictions.
test_predict_plot = np.empty_like(dataset)
test_predict_plot[:, :] = np.nan
test_predict_plot[len(train_predict) + (window_size * 2) + 1:len(dataset) - 1, :] =
test_predict

# Create the plot.

```

```
plt.figure(figsize = (18, 8))
```

```
plt.plot(scaler.inverse_transform(dataset), label = "True value",color='red')
plt.plot(train_predict_plot, label = "Training set prediction",color='yellow')
plt.plot(test_predict_plot, label = "Test set prediction")
plt.xlabel("Months")
plt.legend()
plt.show()
test=list(test_predict
) lis=[]
for i in test:
    v=i
    if v[0]<100:
        x='No rain'
    else:
        x="Rain"
    lis.append(x)
lis
```

B. SCREENSHOTS

The first screenshot shows the initial setup of the Jupyter Notebook environment. The code imports necessary libraries: numpy, pandas, matplotlib, keras, and sklearn. It also sets up the rcParams for the figure size and imports the LSTM layer and Sequential model from keras. The code then imports time, mean_squared_error, pyplot, MinMaxScaler, and StandardScaler from sklearn. A comment indicates that input data files are available in the "../input/" directory. The code then imports the os module.

```
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

from matplotlib import rcParams
rcParams['figure.figsize']=10,6
from keras.layers.core import Dense, Activation, Dropout
from keras.layers.recurrent import LSTM
from keras.models import Sequential
import time
from sklearn.metrics import mean_squared_error
from matplotlib import pyplot
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler
# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list the files in the input directory

import os

# Any results you write to the current directory are saved as output.
```

The second screenshot shows the loading of the dataset and the resulting dtypes. The code uses pd.read_csv to load the "rainfall in india 1901-2015.csv" file, encoding it as "ISO-8859-1". The resulting dtypes are:

SUBDIVISION	object
YEAR	int64
JAN	float64
FEB	float64
MAR	float64
APR	float64
MAY	float64
JUN	float64
JUL	float64
AUG	float64
SEP	float64
OCT	float64
NOV	float64
DEC	float64
ANNUAL	float64
Jan-Feb	float64
Mar-May	float64
Jun-Sep	float64
Oct-Dec	float64
dtype:	object

The third screenshot shows the grouping of the data by SUBDIVISION and the resulting data. The code uses dataset.groupby to group the data by SUBDIVISION, and then uses get_group to get the data for the "BIHAR" group. The resulting data is shown as a table with columns for YEAR, JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, NOV, and DEC. The data is then melted into a long format using data.melt, and the index is reset.

```
groups = dataset.groupby('SUBDIVISION')['YEAR','JAN','FEB','MAR','APR','MAY','JUN','JUL','AUG','SEP','NOV','DEC']
data=groups.get_group(('BIHAR'))
data.head()
```

	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	NOV	DEC
897	1901	51.8	19.6	11.9	1.1	65.6	66.3	245.9	319.4	155.1	7.3	0.1
898	1902	4.6	0.7	24.3	17.3	66.3	118.2	361.0	225.5	358.7	1.1	0.0
899	1903	5.3	4.7	2.0	4.7	28.2	192.9	115.0	342.6	173.9	0.1	0.0
900	1904	6.3	1.7	3.5	5.3	118.7	191.6	394.4	351.3	84.4	10.6	3.8
901	1905	16.0	30.1	32.6	21.4	77.5	50.5	409.1	495.3	353.9	0.0	0.6

```
[ ] data=data.melt(['YEAR']).reset_index()
data.head()
```

index	YEAR	variable	value	
0	0	1901	JAN	51.8
1	1	1902	JAN	4.6
2	2	1903	JAN	5.3
3	3	1904	JAN	6.3
4	4	1905	JAN	16.0

```
[ ] data=data.melt(['YEAR']).reset_index()
data.head()
```

	index	YEAR	variable	value
0	0	1901	JAN	51.8
1	1	1902	JAN	4.6
2	2	1903	JAN	5.3
3	3	1904	JAN	6.3
4	4	1905	JAN	16.0

```
df= data[['YEAR','variable','value']].reset_index().sort_values(by=['YEAR','index'])
df.head()
```

	index	YEAR	variable	value
0	0	1901	JAN	51.8
115	115	1901	FEB	19.6
230	230	1901	MAR	11.9
345	345	1901	APR	1.1
460	460	1901	MAY	65.6

```
[ ] df.columns=['INDEX','YEAR','Month','avg_rainfall']
```

```
d={'JAN':1,'FEB':2,'MAR':3,'APR':4,'MAY':5,'JUN':6,'JUL':7,'AUG':8,'SEP':9,
  'OCT':10,'NOV':11,'DEC':12}
df['Month']=df['Month'].map(d)
df.head(12)
```

	INDEX	YEAR	Month	avg_rainfall
0	0	1901	1	51.8
115	115	1901	2	19.6
230	230	1901	3	11.9
345	345	1901	4	1.1
460	460	1901	5	65.6
575	575	1901	6	66.3
690	690	1901	7	245.9
805	805	1901	8	319.4
920	920	1901	9	155.1
1035	1035	1901	11	7.3
1150	1150	1901	12	0.1
1	1	1902	1	4.6

```
[ ]
```

	INDEX	YEAR	Month	avg_rainfall	Date
0	0	1901	1	51.8	1901-01-01
115	115	1901	2	19.6	1901-02-01
230	230	1901	3	11.9	1901-03-01
345	345	1901	4	1.1	1901-04-01
460	460	1901	5	65.6	1901-05-01
575	575	1901	6	66.3	1901-06-01
690	690	1901	7	245.9	1901-07-01
805	805	1901	8	319.4	1901-08-01
920	920	1901	9	155.1	1901-09-01
1035	1035	1901	11	7.3	1901-11-01
1150	1150	1901	12	0.1	1901-12-01
1	1	1902	1	4.6	1902-01-01

```
[ ] 1 1 1902 1 4.6 1902-01-01
```

```
[ ] cols=['avg_rainfall']
dataset=df[cols]
dataset.head()
```

avg_rainfall	
0	51.8
115	19.6
230	11.9
345	1.1
460	65.6

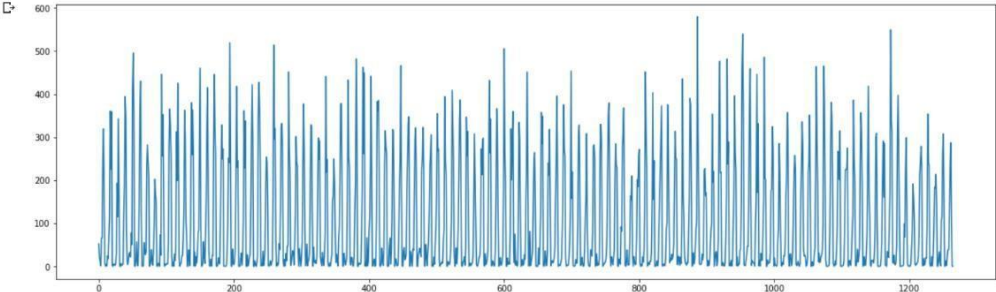
```
series=dataset
series.head()
series.shape
```

```
(1265, 1)
```

```
[ ] series.shape
```

```
(1265, 1)
```

```
pyplot.figure(figsize=(20,6))
pyplot.plot(series.values)
pyplot.show()
```



```
[ ] # Get the raw data values from the pandas data frame.
data_raw = series.values.astype("float32")

# We apply the MinMax scaler from sklearn
# to normalize data in the (0, 1) interval.
scaler = MinMaxScaler(feature_range = (0, 1))
dataset = scaler.fit_transform(data_raw)
```

```
[ ] # Print a few values.
dataset[0:5]
```

```
array([[0.08929496],
       [0.03378728],
       [0.02051371],
       [0.00189622],
       [0.11308396]], dtype=float32)
```

```
[ ] # Using 60% of data for training, 40% for validation.
TRAIN_SIZE = 0.60

train_size = int(len(dataset) * TRAIN_SIZE)
test_size = len(dataset) - train_size
train, test = dataset[0:train_size, :], dataset[train_size:len(dataset), :]
print("Number of entries (training set, test set): " + str((len(train), len(test))))
```

```
Number of entries (training set, test set): (1012, 253)
```

```
# FIXME: This helper function should be rewritten using numpy's shift function. See below.
def create_dataset(dataset, window_size = 1):
    data_X, data_Y = [], []
    for i in range(len(dataset) - window_size - 1):
        a = dataset[i:i + window_size, 0]
        data_X.append(a)
        data_Y.append(dataset[i + window_size, 0])
    return np.array(data_X), np.array(data_Y)
```

```
def fit_model(train_X, train_Y, window_size = 1):
    model = Sequential()

    model.add(LSTM(2000,activation = 'tanh', input_shape = (1, window_size)))
    model.add(Dropout(0.2))
    model.add(Dense(500))
    model.add(Dropout(0.4))
    model.add(Dense(500))
    model.add(Dropout(0.4))
    model.add(Dense(400))
    model.add(Dropout(0.4))
    model.add(Dense(1, activation = 'linear'))
    model.compile(loss = "mean_squared_error",
                  optimizer = "adam",metrics=["accuracy"])

    model.fit(train_X,
              train_Y,
              epochs = 10,
              batch_size = 64,
              )

    return(model)

# Fit the first model.
model1 = fit_model(train_X, train_Y, window_size)
```

```
Epoch 1/10
16/16 [=====] - 12s 538ms/step - loss: 0.0457 - accuracy: 0.0554
Epoch 2/10
16/16 [=====] - 9s 532ms/step - loss: 0.0374 - accuracy: 0.0535
Epoch 3/10
```

```
[ ] scores = model1.evaluate(train_X, train_Y)

32/32 [=====] - 2s 54ms/step - loss: 0.0365 - accuracy: 0.0554

[ ] print("Traning Accuracy:",100-scores[0]*100," %")
    print("Testing Accuracy:",100-scores[1]*100," %")

Traning Accuracy: 96.35279588401318 %
Testing Accuracy: 94.45544555783272 %

[ ]
```

```
import math
def predict_and_score(model, X, Y):
    # Make predictions on the original scale of the data.
    pred = scaler.inverse_transform(model.predict(X))
    # Prepare Y data to also be on the original scale for interpretability.
    orig_data = scaler.inverse_transform([Y])
    # Calculate RMSE.
    score = math.sqrt(mean_squared_error(orig_data[0], pred[:, 0]))
    return(score, pred)

rmse_train, train_predict = predict_and_score(model1, train_X, train_Y)
rmse_test, test_predict = predict_and_score(model1, test_X, test_Y)

print("Training data score: %.2f RMSE" % rmse_train)
```

```
[ ] v=list(train_X)

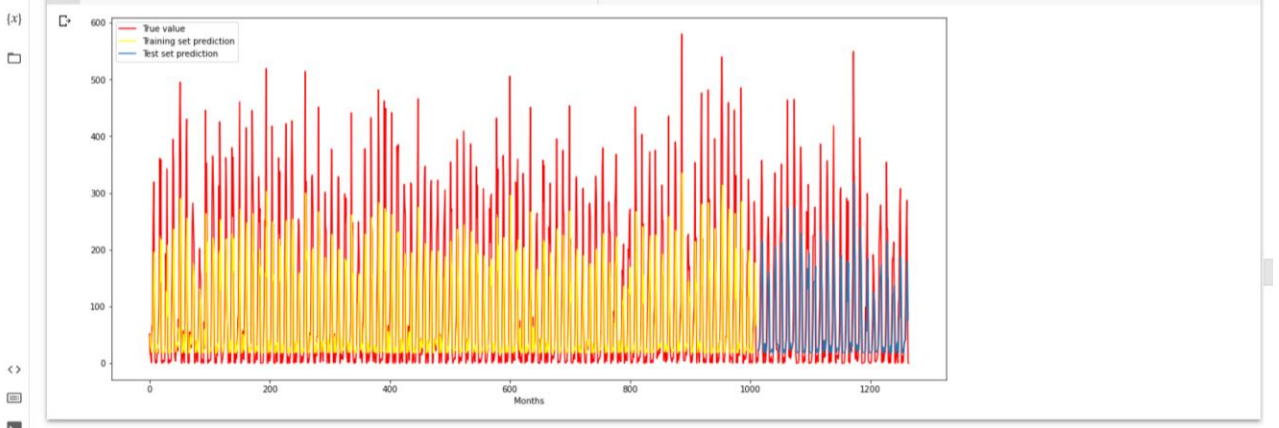
# Start with training predictions.
train_predict_plot = np.empty_like(dataset)
train_predict_plot[:, :] = np.nan
train_predict_plot[window_size:len(train_predict) + window_size, :] = train_predict

# Add test predictions.
test_predict_plot = np.empty_like(dataset)
test_predict_plot[:, :] = np.nan
test_predict_plot[len(train_predict) + (window_size * 2) + 1:len(dataset) - 1, :] = test_predict

# Create the plot.
plt.figure(figsize = (18, 8))
plt.plot(scaler.inverse_transform(dataset), label = "True value",color='red')
plt.plot(train_predict_plot, label = "Training set prediction",color='yellow')
plt.plot(test_predict_plot, label = "Test set prediction")
plt.xlabel("Months")

plt.legend()
plt.show()

plt.legend()
plt.show()
```



```
[x] ▶ test=list(test_predict)
lis=[]
for i in test:
    v=i
    if v[0]<100:
        x='No rain'
    else:
        x='Rain'

    lis.append(x)
lis
```

```
[- 'No rain',
  'No rain',
  'No rain',
  'No rain',
  'No rain',
  'Rain',
  'Rain',
  'Rain',
  'Rain',
  'Rain']
```

paper (1).docx

by Paper (1).docx Paper (1).docx

Submission date: 23-Mar-2022 02:16AM (UTC-0400)

Submission ID: 1790758264

File name: paper_1.docx (307.15K)

Word count: 3726

Character count: 20303

Design of Deep Learning Model for Predicting Rainfall

 Sai Saran Ravi
Department of Computer Science And
Engineering
Sathyabama Institute Of Science And
Technology
Chennai, India
saisaran131@gmail.com

 Relangi Gowtham Krishna Sai
Department of Computer Science And
Engineering
Sathyabama Institute Of Science And
Technology
Chennai, India
gowthamkrishnasair@gmail.com

 Jeberson Retna Raj R
Department of Information Technology
Sathyabama Institute Of Science And
Technology
Chennai, India
jebersonretnarajr@sathyabam.ac.in

Abstract— Precipitation of seasonal rainfall is ever important for all living creatures and human. Forecasting the rainfall is of important for better planning in agriculture and take precautionary measures for downpour and flash floods. Downpour is one of the main wellsprings of new water for all living things on the planet. Precipitation figures give data based on environmental change in conditions of precipitation. As of late, Deep Learning has been assisting with learning sign information that permits information displaying during informational collections. This permits us to get twists/changes from the news line and to foresee future data about the past. This article will clarify the improvement of precipitation designs through a top to bottom review. This paper examines downpour conjectures utilizing inside and out calculations, for example, LSTM and ANN, which will be valuable and great for the model.

Keywords—Rainfall Prediction, Weather Prediction, Deep Learning, ANN, LSTM;

I. INTRODUCTION

Rainfall arranging is perhaps the most troublesome task. Albeit numerous calculations have been set up, it is truly challenging to precisely anticipate rainfall. In a nation like India, the attention is on crop achievement, disappointment, and water deficiencies at whatever year. A slight change in the stormy season will contrarily affect horticulture. Appropriate rainfall arranging is fundamental to forestall debacle. Legitimate climate anticipating sometimes, like floods and dry seasons, can assist with overseeing farming and forestall fiascos. This article investigates various calculations. Information mining procedures are all around used to recognize rainfall.

The prediction of rainfall remained as a severe concern and it has become a big challenge for government and numerous industries too. Rainfall being a climatic factor it affects many human activities and this prediction is a critical task because many people depend on farming and generation of power. Rainfall prediction is necessary because rainfall is one the factor responsible for adverse natural events like flood, landslides. Many of these incidents have affected several parts of the country leading loss of lives and also creating several damage to the properties for years. Rainfall forecasting is also difficult task due to the distinct nature of weather conditions. For that reason it is better to have an proper approach for prediction of rainfall which makes it possible to take necessary precautions in time for a natural phenomenon. So here we introduced a deep learning algorithms using ANN (Artificial Neural Network) and LSTM (Long Short-Term Memory) being most helpful and attractive methods for forecasting tasks.

II. LITERATURE ANALYSIS

Pritpal Singh et al proposed a prediction model for Indian Summer Monsoon Rainfall (ISMR) using time-series data. The model employed with fuzzy entropy and neural expert system. In essence, the mathematical and statistical based models limited in accuracy. Therefore, fuzzy set and ANN based techniques is used to forecast the rainfall. Then this model is divided in training and test data sets. Numerous comparisons and statistical analysis demonstrated the efficacy of this proposed model. [1]. Sam Crame et.al introduced seven machine learning algorithms for predicting rainfall using the available weather derivatives. They applied and compared the performance of six prominent ML algorithms which includes Vector regression, neural networks, KNN, M5 rules and M5 model trees for predicting the rainfall. The system employs with the data set consists of 42 Cities with different climatic features based on this the tests are run [2]. In [3], a Statistical Downscaling based hybrid Model for predicting the rainfall is introduced. This model employs with SVM algorithm to predict the rainfall occurring days and whether rain will occur in a day or not. The hybrid model is implemented based on the features which include number of wet days, 95 percentile rainfall amounts in each month and distribution of observed rainfall reliably. The capability of the model is verified with three different rain-gauge locations in the east cost of peninsular Malaysia [3]. Manjunath et al suggested a Rainfall forecasting model using ML and DL algorithms. Agriculture is one of the main business for economic activity in India. Annual rainfall is very important for agriculture activities. Also, the prediction of rainfall is difficult and it is important to know in advance about the rainfall in order to take precautions by protecting the crop from heavy downpour. For rainfall prediction ML algorithms are mostly used. However, the forecasts are never 100% accurate [4]. Tanvi Patil et al presented a model for weather Forecasting Analysis using Linear and Logistic Regression Algorithms. The main objective of this model is to predict the weather over a given period. The weather condition at any instance is described by using different kinds of variables. Out of these attributes, only significant attributes are used in the process of weather prediction [5]. In [6], a classification method for weather prediction and forecasting rainfall is introduced. The model built with an intellectual tool employs with enhanced Multilayer Perception Neural Network applied to time Series data for prediction. The accuracy of prediction is also tested with the metrics MSE (Mean square error), and NMSE (Normalized mean square error) and compared with other models such as Adanaive, AdaSVM. Senthamil Selvi et al introduced a model for Predicting rainfall using Time-series data and analyzing the role of ANN models. Here, ANN

technique is used to develop weather forecasting for predicting the rainfall by using rainfall data of India. FFNN back propagation algorithm is used for forecasting the seasonal rainfall in India. The performance of both models is evaluated using regression analysis, Mean square error and Magnitude of Relative Error (MRE). However, it will not help in assessing the floods in advance due to the limited range of data [7]. In [8], Yashas Athreya et.al proposed a model based on Flood Prediction and Rainfall Analysis using Machine Learning algorithms. The main goal of employing this application is to prevent immediate impacts of the flood. This application can be easily used by the common people or government to predict the occurrence of flood beforehand. The Prediction of flood is done by making use of previous data, map the flood concentration as well, and then provide them necessary help through helpline for further evacuation or any other necessary precautions [8]. Oswalt manoj et al proposed a MapReduce framework based model for predicting rainfall. The model employed with convLSTM (convolutional long short-term memory) which is integrated with the gradient descent algorithm for achieving better accuracy [9]. In [10], rainfall is predicted in Odisha using multilayer perceptron and support vector regression. Appiah et al presented a rainfall prediction model for Ghana using machine learning classification algorithms and the accuracy of prediction is compared [11]. In [12], a short-range rainfall prediction model is introduced using Conditional Generative Adversary Network (CGAN) algorithm. Khan et al introduced a multistep ahead one-to-five days of rainfall prediction using multilayer perceptron and 1-D convolutional neural network approach [13]. Manandhar et al identified the weather features such as PVW, solar radiation, diurnal and seasonal features used for rainfall prediction [14]. Naha Biswas et al proposed a rainfall prediction model based on tropospheric gradient and atmospheric residual weather features. The algorithm incorporated with GPS based on Precipitable Water Vapor (PWV) for prediction [15].

III. PROPOSED RAINFALL PREDICTION MODEL

Rainfall is significant for food conveyance, water assets, and many other ecological actions. Long haul dry spells or substantial rainfall during extreme development and advancement can altogether diminish yields. India is a rural nation and its economy is to a great extent dependent on farming creation. In this way, rainfall determining is turning out to be an ever-increasing number of significant in farming nations like India. Rainfall gauging has become one of the world's most squeezing science and innovation issues in ongoing hundreds of years. The model uses high-level architecture where it is trained and also fit using the deep learning algorithms and then the prediction is done to know how the model is performing. This model type helps in predicting the rainfall. Hence it helps to reduce the damage on agricultural production and floods.

Here, we will be using Deep Learning algorithms for obtaining the best from that we use. Firstly we take dataset and then it is thoroughly goes under pre-processing and cleaning will be done. Once the data visualization is done to the dataset, then we split the dataset into test data set and we train the test data set with training dataset. Finally on comparison with both the algorithms (ANN and LSTM), subsequently we take the algorithm which gives the best accuracy. After that, we implement the model by that best model.

A. Long Short-Term Memory (LSTM)

Numerous choices have been created to kill the issue of lost inclinations and rushes in profound reusing pipes. The most renowned of these is the Longevity of Remembrance. Typically, the LSTM audit segment attempts to "recall" all that the organization has seen up until this point, and to "overlook" it and the article data. This finishes with the presentation of a progression of exercises called "doors" for certain exercises. Each piece of the LSTM makes the forward cell push ahead, which is characterized as the chose and changed data over to be put away through the past LSTM areas. Long and short memory pipelines have 4 kinds of doors for a similar capacity depicted beneath: -

Failed to remember the Door (f): This decides the degree to which past data was disregarded.

Input entryway (I): This decides the quantity of data that can be written before the cell.

Entrance Door (g): This is regularly viewed as a piece of the entry, and many books on LSTM currently quit showing it and think it is a long way from the entry. It was utilized to change data over to compose inside the front cell of the information family and added a little line to the information and zeros. This is done to diminish the catch time, as Zeru implies fast info. In spite of the fact that house to house travel is exceptionally restricted and is constantly controlled in a reasonable manner, it is a decent practice to do this house-to-house LSTM approach.

Leave Gate (o): This decides the result (the following mystery) made from the back to front.

B. Artificial Neural Network (ANN)

An artificial neural network typically contains three or more layers, all of which are interconnected. The first layer comprises input neurons, which will send data to deeper layers before sending final output data to the last output layer in a sequential order. The inner layers are now hidden, and they are formed by units that are responsible for the variation in information received from layer to layer. In this case, each layer serves as an input and output layer, allowing the ANN to comprehend complex concepts. The inner layers that are present here are referred to as the Neural layer. The units in the neural layer are interested in learning more about the information and using the ANN's internal mechanism to do so. Only when these principles are fulfilled do the units provide a changed result, which is then authorized as an output to the following layer.

There are some more set of learning rules that make use of backpropagation. It sticks to a process like ANN can be able to adjust its output results taking the consideration of errors that are in the account. The information is sent backward every time whenever the output is labelled as an error. In this way the weight gets updated proportionally for responsibility of the error.

Thereupon errors are used to revise the weight of Artificial neural networks unit connections to know about the dissimilarities among actual one and the desired outcome. Eventually ANN will learn to reduce the chances of getting errors or unwanted results.

IV. SYSTEM ARCHITECTURE

The system architecture of the proposed system is shown in figure 1. The first and most important thing to work with deep learning models, we need to have huge amount of data because without data, one cannot train DL/AI models. One of the most crucial parts while creating a Deep learning model is Collecting and preparing the dataset. If the dataset is not well prepared or not pre-processed the technology that is applied behind the DL/AI project cannot work. We got this dataset from Kaggle and it has the data that point out the features on climatological normal of district wise rainfall (in mm) which is estimated with the data of the period 1951-2015.

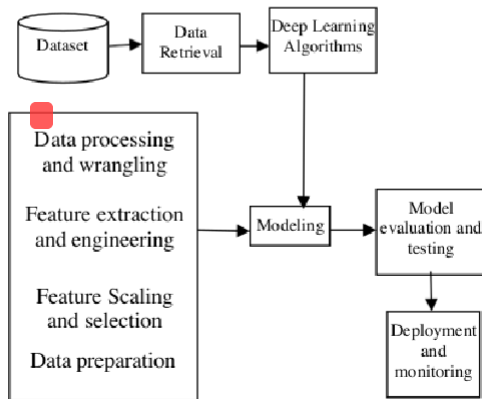


Fig 1. System Architecture

While coming into preparation, the next thing we do is Data Pre-processing. It is the procedure of preparing the raw data and making it suitable for a Deep Learning model, it's a crucial step while creating a model. This process is important for cleaning the data and making sure whether it is suitable for making a DL model, it also enhances the accuracy and efficiency of the model.

In data pre-processing it has some steps to be followed are acquiring the dataset, importing libraries and datasets, find the missing data, splitting into training set and testing set, and finally feature scaling. The libraries that we imported are NumPy, Pandas, matplotlib, time and OS. We now divide the dataset into two parts: training and testing. The training set is a subset of the dataset used to train the deep learning model. While the test set is a subset of the dataset used to test the deep learning model, the model predicts the output using the test set. Final step in Pre-processing is Feature Scaling. This technique is used to standardize the independent variables of the dataset in a specific range. Here we put all our variables in same range e.g all the variables are scaled between -1 to 1. And then we use Deep learning algorithms LSTM and ANN for the prediction of rainfall.

A. Data Collection

It is known as the procedure of collecting information from distinct sources. In an effort to use the data we accumulate to develop practical and deep learning solutions, it must be gathered and reserved such a way that makes sense for the problem additionally make correct use of it. The data utilized in this action was gathered by the meteorological association. The conversation covers the period from 1901 to 2015. This segment of the review talks about the accompanying systems: Cleaning the data, picking the data, changing the data, cleaning the data.

B. Data Cleaning

Data cleaning is to recognize and eliminate errors and duplicate data, in an effort to create a good dataset. This enhances the quality of the training data for analytics and enables precise decision-making. At last the information cleaning framework was changed into an effective information mining framework.

C. Data Selection

Data selection is the process of defining the appropriate source and data type, it also picks the suitable instruments to collect the data. At this stage, the data identified with the examination tree is the choice tree and is extricated from the informational index. The meteorological informational index had ten attributes that pre-owned two elements later on. Because of the idea of cloud information, all qualities are the equivalent, and the vast majority of the qualities that are absent in sunlight-based information are not utilized in the investigation.

D. Data Preprocessing

Data Preprocessing is here Information handling is the method involved with getting ready essential data and involving it in AI. This is the first and most significant stage in the advancement of AI. When constructing a vehicle preparing machine, we don't consistently get spotless and helpful data, procedure of developing the raw data and make it suitable for accomplishing any instructive work. In this manner, we use information handling data.

E. Data Prediction

Data prediction - The proposed model uses an inside and out text data transformation calculation that contains key components of computerized information in the information section step and gives mechanical progression at the expectation stage.

V. EXPERIMENTATION RESULTS

The system is implemented with the dataset called rainfall in India 1901-2015 from Kaggle, where we further preprocess the data and selected the attributes called year, month, avg_rainfal and date for prediction of the rainfall. Figure 2 shows the features of the dataset.

INDEX	YEAR	Month	avg_rainfall	Date
0	0	1901	1	51.8 1901-01-01
115	115	1901	2	19.6 1901-02-01
230	230	1901	3	11.9 1901-03-01
345	345	1901	4	1.1 1901-04-01
460	460	1901	5	65.6 1901-05-01
575	575	1901	6	66.3 1901-06-01
690	690	1901	7	245.9 1901-07-01
805	805	1901	8	319.4 1901-08-01
920	920	1901	9	155.1 1901-09-01
1035	1035	1901	11	7.3 1901-11-01
1150	1150	1901	12	0.1 1901-12-01
1	1	1902	1	4.6 1902-01-01

Fig 2. The attributes of the dataset

We have taken the data set called rainfall in india 1901-2015 from Kaggle where we further preprocessed the data and selected the attributes called year, month, avg_rainfal and date for prediction of the rainfall. Figure 3 shows the frequency of the rainfall.

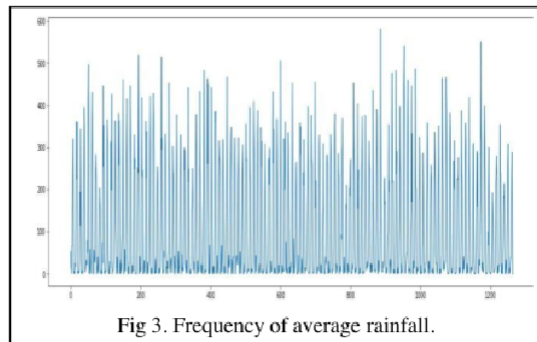


Fig 3. Frequency of average rainfall.

Here we plotted this graph on basis of avg_rainfall attribute. This graph describes the frequency of avg_rainfall from the dataset.

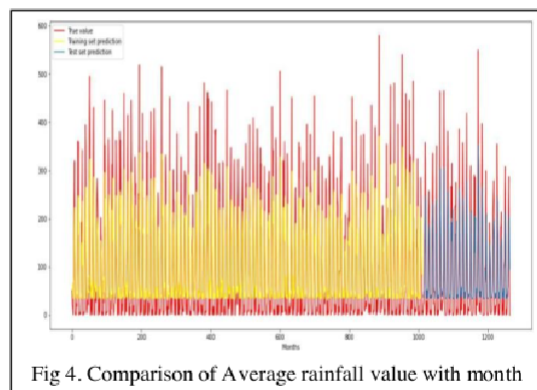


Fig 4. Comparison of Average rainfall value with month

Figure 4 graph shows that the comparison of Avg rainfall value with months. This graph also compares the true values with training set predictions and test set predictions.

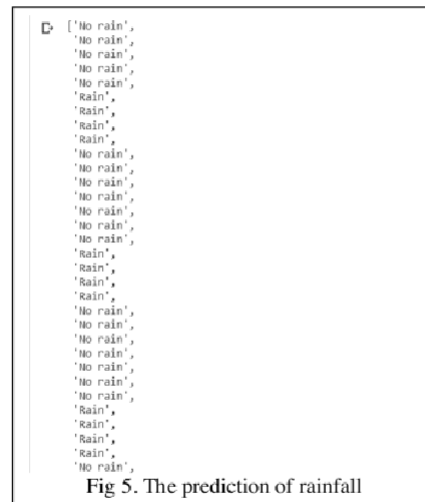


Fig 5. The prediction of rainfall

After comparing the accuracies of both LSTM and ANN we have selected LSTM algorithm for prediction of rainfall and rainfall is predicted as per the figure above. If the avg_rainfal is above 100 it is predicted as "Rain" or else if the value of avg_rainfall is below 100 it is predicted as "No rain". Table 1 shows the performance of LSTM algorithm.

TABLE I: PERFORMANCE OF LSTM ALGORITHM

LSTM	accuracy	RMSE
Train Data	96.57%	107.35
Test Data	94.45%	94.53

The above table shows the accuracy and rmse values when we are using lstm for prediction. As mentioned above the accuracy of train data and test data is 96.57% and 94.45% respectively. The RMSE values of train data and test data are 107.35 and 94.53 respectively. Table II shows the accuracy of ANN algorithm in prediction process.

TABLE II: PERFORMANCE OF ANN ALGORITHM

ANN	Accuracy	RMSE
Train data	58.71%	109.28
Test data	94.45%	97.31

The above table shows the accuracy and rmse values when we are using ANN for prediction. As mentioned above the accuracy of train data and test data is 58.71.57% and 94.45% respectively. the Root Means Square Error (RMSE) values of train data and test data are 109.28 and 97.31 respectively. Both ANN and LSTM achieve same in accuracy in the given dataset and the RMSE value is lower in

LSTM comparing with ANN. Regarding RMSE value, lower the value is better in performance. In such case, LSTM returns better performance.

VI. CONCLUSION

Forecasting the weather is challenging and to achieve better accuracy is complex due to the inherent nature of environment and climatic conditions. It is subjective to change because of dynamic nature of environmental properties. The deep learning based prediction of rainfall is successfully implemented using the available data sets. With the applications of LSTM and ANN algorithms, the system predict the rainfall for a given time periods. The dataset includes daily records of parameters like avg_rainfall, month, year, date. The experimental analysis clearly demonstrates that the proposed model bring about a promising outcome. These algorithms LSTM and ANN based predictive model is suitable for numerous applications requiring rainfall prediction, Agriculture as such. Later on, we intend to build the work on harvest and yield estimating and downpour gauging.

REFERENCES

- [1] Singh, Pritpal. (2017). Indian summer monsoon rainfall (ISMR) forecasting using time series data: A fuzzy-entropy-neuro based expert system. *Geoscience Frontiers*. 9. 10.1016/j.gsf.2017.07.011.
- [2] Cramer, Sam & Kampouridis, Michael & Freitas, Alex & Alexandridis, Antonios. (2017). An extensive evaluation of seven machine learning methods for rainfall prediction in weather derivatives. *Expert Systems with Applications*. 85. 10.1016/j.eswa.2017.05.029.
- [3] Hadipour, Sahar & Shahid, Shamsuddin & Chung, Eun-Sung. (2016). A Hybrid Model for Statistical Downscaling of Daily Rainfall. *Procedia Engineering*. 154. 1424-1430. 10.1016/j.proeng.2016.07.514.
- [4] Manjunath N, Muralidhar B R, Sachin Kumar S, Vamshi K, Mrs. Savitha P, "RAINFALL PREDICTION USING MACHINE LEARNING AND DEEP LEARNING TECHNIQUES : A REVIEW", *International Research Journal Of Engineering And Technology (IRJET)* e-ISSN: 2395-0056 Volume: 08 Issue: 08 | Aug 2021 www.irjet.net p-ISSN: 2395-0072.
- [5] Tanvi Patil, Dr. Kamal Shah, "Weather Forecasting Analysis using Linear and Logistic Regression Algorithm", *International Research Journal of Engineering and Technology (IRJET)* e-ISSN: 2395-0056 Volume: 08 Issue: 06 | June 2019 www.irjet.net p-ISSN: 2395-0072.
- [6] N. Divya Prabha, P. Radha, "Prediction Of Weather And Rainfall Forecasting Using Classification Techniques", *International Research Journal of Engineering and Technology (IRJET)* e-ISSN: 2395-0056 Volume: 06 Issue: 02 | Feb 2019 www.irjet.net p-ISSN: 2395-0072.
- [7] Senthamil Selvi, Seetha N, "Rainfall Prediction by using Time-series data in analysis of ANN Mmodels", *International Research Journal of Engineering and Technology (IRJET)* e-ISSN: 2395-0056 Volume: 06 Issue: 04 | Apr 2019 www.irjet.net p-ISSN: 2395-0072.
- [8] YashasAthreya, VaishaliBV, SagarK, SrinidhiHR, "Flood Prediction and Rainfall Analysis using Machine Learning", *International Research Journal of Engineering and Technology (IRJET)* e-ISSN: 2395-0056 Volume: 08 Issue: 07 | July 2021 www.irjet.net p-ISSN: 2395-0072.
- [9] O. M. S and A. J P, "MapReduce and Optimized Deep Network for Rainfall Prediction in Agriculture," in *The Computer Journal*, vol. 63, no. 1, pp. 900-912, Jan. 2020, doi: 10.1093/comjnl/bxz164.
- [10] X. Zhang, S. N. Mohanty, A. K. Parida, S. K. Pani, B. Dong and X. Cheng, "Annual and Non-Monsoon Rainfall Prediction Modelling Using SVR-MLP: An Empirical Study From Odisha," in *IEEE Access*, vol. 8, pp. 30223-30233, 2020, doi: 10.1109/ACCESS.2020.2972435.
- [11] N. K. A. Appiah-Badu, Y. M. Missah, L. K. Amekudzi, N. Ussiph, T. Frimpong and E. Ahene, "Rainfall Prediction Using Machine Learning Algorithms for the Various Ecological Zones of Ghana," in *IEEE Access*, vol. 10, pp. 5069-5082, 2022, doi: 10.1109/ACCESS.2021.3139312.
- [12] Y. Kim and S. Hong, "Very Short-Term Rainfall Prediction Using Ground Radar Observations and Conditional Generative Adversarial Networks," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-8, 2022, Art no. 4104308, doi: 10.1109/TGRS.2021.3108812.
- [13] M. I. Khan and R. Maity, "Hybrid Deep Learning Approach for Multi-Step-Ahead Daily Rainfall Prediction Using GCM Simulations," in *IEEE Access*, vol. 8, pp. 52774-52784, 2020, doi: 10.1109/ACCESS.2020.2980977.
- [14] S. Manandhar, S. Dev, Y. H. Lee, Y. S. Meng and S. Winkler, "A Data-Driven Approach for Accurate Rainfall Prediction," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 11, pp. 9323-9331, Nov. 2019, doi: 10.1109/TGRS.2019.2926110.
- [15] A. Naha Biswas, Y. H. Lee and S. Manandhar, "Rainfall Forecasting Using GPS-Derived Atmospheric Gradient and Residual for Tropical Region," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-10, 2022, Art no. 5802110, doi: 10.1109/TGRS.2021.3131217.

paper (1).docx

5%

8%

6%

ORIGINALITY REPORT

INTERNET SOURCES

PUBLICATIONS

STUDENT PAPERS

1

2 %

SIMILARITY INDEX

PRIMARY SOURCES

Submitted to Coventry University



Student Paper

%

1

1

%

2

Sahar Hadi Pour, Shamsuddin Shahid,

Eun- Sung Chung. "A Hybrid Model for

Statistical Downscaling of Daily Rainfall",

Procedia Engineering, 2016

Publication

hrcak.srce.hr



Internet Source

%

1

1

%

4

Submitted to Rutgers University,
New Brunswick

Student Paper

Submitted to University of Greenwich

1



Student Paper

%

1

Submitted to Universiti Teknologi Malaysia



Student Paper

%

1

%

7

Submitted to Indian Institute of Technology,
Madras

Student Paper

www.ru.ac.bd

8

"Intelligent Systems
Design and

9

Submitted to
University of
KwaZulu-Natal

Applications",
Springer Science
and Business
Media LLC, 2020

Student Paper

10

Publication

thesai.org

Internet Source

11

Aqsa Rahim, Zara
Hayat, Muhammad

www.ncbi.nlm.nih.g
ov

Internet Source

12

Abbas, Amna
Rahim, Muhammad
Abdul Rahim.

"Software Defect

Shabib Aftab,
Munir Ahmad,
Noureen Hameed,
Muhammad
Salman, Iftikhar Ali,
Zahid Nawaz.

Prediction with
Naïve Bayes
Classifier", 2021
International
Bhurban
Conference on
Applied Sciences
and Technologies
(IBCAST), 2021

13

"Rainfall Prediction
in Lahore City using
Data Mining
Techniques",
International

14

Journal of
Advanced
Computer Science
and Applications,
2018

Publication

1 %

<1 %

<1 %

<1 %

<1 %

<1 %

<1 %

15

Anik Naha
Biswas, Yee Hui
Lee, Shilpa
Manandhar.
"Rainfall
Forecasting Using
GPS- Derived
Atmospheric
Gradient and
Residual for
Tropical Region",
IEEE Transactions
on Geoscience
and Remote
Sensing, 2022

Publication

Systems (ICUAS),
2021

Publication

archive.org

Internet Source

mdpi-res.com

Internet Source

arxiv.org

Internet Source

16

17

18

19

20

Paloma Carrasco,
Francisco Cuesta,
Rafael Caballero,
Francisco J.
Perez-Grau, Antidio
Viguria.
"Monte-Carlo
Localization for
Aerial Robots using
3D LiDAR and
UWB sensing",
2021 International
Conference on
Unmanned Aircraft

Abhinav Sharma,
Akshay Khanna,
Muskaan
Bhargava, Rutwik
Pendse. "Rainfall
Prediction: Analysis
of Machine
Learning
Algorithms and
Ensemble
Techniques", 2021
7th International
Conference on
Signal Processing
and
Communication
(ICSC), 2021

<1 %

<1 %

<1 %

<1 %

<1 %

<1 %

21

Dipanjana Sarkar,
Raghav Bali,
Tushar Sharma.
"Practical Machine
Learning with
Python", Springer
Science and
Business Media
LLC, 2018

Publication

fall prediction

enabled by

optimized

convolutional

neural network-

22

Submitted to
SASTRA University

Student Paper

based feature

formation with

23

24

Shruti, Monika
Singh. "Intelligent
HealthCare
System using
Raspberry Pi",
2021 International
Conference on
Technological
Advancements
and Innovations
(ICTAI), 2021

Publication

adaptive long

short-term

memory

framework",

Concurrency and

Computation:

Practice and

Experience, 2022

Ananthajothi K,

Karthick T,

Amanullah M.

Publication

"Automated rain

<1 %

<1 %

<1 %

<1 %

Exclude quotes
bibliography

On Exclude
On

Exclude matches Off