Smart Method For Predicting Flooding In Cites Current Advances And Challenges

Nandini Boddhul[1], Prajakta khune[2], Prajakta thorat[3], Aarti Karvande[4], Prof. Amar chipde[5]
[1, 2, 3, 4, 5] Dept of civil Engineering
[1, 2, 3, 4, 5] DIT Pimpri, Pune

Abstract- *A quick urbanization in India and a worldwide environmental change has expanded metropolitan flood openness in our country and expanded a flood hazard has decreased a property estimation in a flood an inclined region. Metropolitan flooding instigated by a deluge has caused a broad interruption and harm around the world. The primary issues are related with water logging, water assets and the amphibian climate for an example a substantial downpour in a city definitely prompts water logging, which prompts an incredible danger to the vocation, a property and transportation offices disasters because of a metropolitan flood uncovered a key issue that restricts the improvement of metropolitan nature in the country. This is attributed to a rise in impermeable surfaces, which results in a decrease in a rainfall penetration and an increase in the surface runoff in Indian cities. It was also intended to have more in- depth thinking to address the nation's water logging scarcity. However, recognizing and forecasting rainfall-induced urban flooding also presents difficulties and information gaps.*

Keywords- Linear regression algorithm, Flood prediction.

I. INTRODUCTION

In urban areas all over the world, frequent flooding has become a serious concern. This is because the impervious surface covers the majority of the population, and rainfall appears to increases in the city as a result of the heat an island effect and a global warming.

Dredging rivers, upgrading drainage networks, and erecting flood walls are all necessary, but they significantly alter the urban Riverside environment. Besides, these public works are deficient as the metropolitan flood catastrophe unavoidably disintegrates without diminishing a spillover. It isimportant to improves a water maintenance and an invasion inside the whole metropolitan watershed where there is normally countless private properties and ventures have arranged. The office has a water maintenance work as well as environmental and instructive jobs. The reason for this examination is to report how to foresee the metropolitan flooding Utilizing python language by a linear regression algorithm .

1.1 LITERATURE SURVEY

1.In this paper, we have a tendency to investigate the flooding an attack detection and a bar for sensible meter networks.

The default unexpected On-Demand Distance Vector (AODV) the protocol is at a risk of flooding attacks as a result of intermediate meters forward packets blindly. To detect and prevent flooding attacks, we have a tendency to propose a brand the new AODV-based the routing protocol referred to as the flooding awareness AODV (FLOW-AODV) considering each with and while not web Protocol (IP) spoofing. If no informatics spoofing exists, we have a tendency to modify the route request forwarding procedure byadding new options.

2. Urban rainy flooding could be a threatening natural hazard in urban areas everywhere the globe, particularly in recent Years given its increasing a frequency of prevalence. So as to the forestall flood prevalence and mitigate the next aftermath, urban water managers aim to predict precipitation characteristics, as well as the peak intensity, arrival time and the period, in an order that they'll any warn
inhabitants in risky areas and take emergency actions once forecasting the inclement flood.

3. The present study aims to utilizes GIS abstraction analysis functions, information from measurement and Rain-Gauge stations,satellite pictures, and thematic information layers within the style of Artificial Neural Network algorithmic rule for prediction of discharge values and spatial modeling of floods in Kan River Basin set in capital of Iran province. AN optimized artificial neural network of 7 inputs, as well as slope, slope curvature, flow accumulation, NDVI, geologic units, soil type, and precipitation data beside eight, sixteen and one neurons for the primary, second and output hidden layers, severally, was designed and developed.

4. Paper explores opportunities of a machine learning ways for a prediction of the flooding phenomena in Pattani watercourse victimization open data. The study factors embrace information assortment amount and site and configuration of prediction models. The analysis of quality characteristics for many machines learning-based algorithms.

5. High accuracy models an area unit has needed for wise to the higher cognitive process in the urban flood management. This paper a replacement the holistic framework for exploitation data collected from multiple sources for setting parameters of the 2D flood model. This illustrates the importance of characteristic key urban options from the parcel information for capturing the high resolution flood processes.

6. Identifying floods and manufacturing a flood status map the area unit crucial steps for the decision-makers to forestall and manage disasters. Lots of studies have used machine learning models to provide reliable status maps. Nevertheless, The most analysis ignores the importance of developing applicable feature engineering strategies.

II. METHODOLOGY

Linear regression algorithmic rule shows a linear relationship between dependent (y) and one or additional freelance (y) variables, therefore referred to as as regression toward the mean. regression toward the mean model provides a sloping line representing the connection between thevariables.

INPUTE DATA REQUIRED:

District	Year	June	July	August
Kolhapur	2018	191.954	228	256.432
Kolhapur	2019	183.003	319.53	171.593
Kolhapur	2020	117.784	151.87	118.046

Formula for linear regression ,

$Y = a + bX + E$

Y = dependent variable (target variable)

X = independent variable (predictor variable)

a = intercept of the line (gives an additional degree of freedom)

b = linear regression coefficient (scale factor to each input value)

E = random error

The values for X and Y variables are training datasets for linear regression model representation .

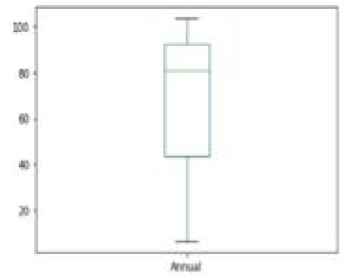
```
In [10]: import pandas as pd #for dataframe
         import numpy as np #for calculation
         import math
         import sklearn #for ML
         from sklearn.model_selection import train_test_split
         from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score
         import matplotlib.pyplot as plt #for plotting graph
         from sklearn import preprocessing
```

```
In [11]: data=pd.read_csv("rainfall1.csv") #to import data as a input
```

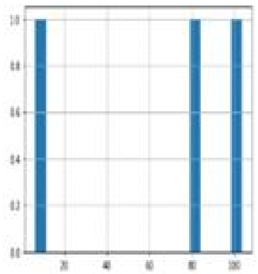
```
In [12]: print("data head")
         print(data.head()) #for showing top 5 rows

data head
        State District  Year    Jan     Feb     Mar     Apr     May  \
0  Maharashra Kolhapur  2018  21.725  25.488  38.288   2.811   8.093
1  Maharashra Kolhapur  2019  22.725  26.488  39.288  22.906  45.447
2  Maharashra Kolhapur  2020  23.725  27.488  40.288   4.528  94.681

       Jun    July      Aug      Sep      Oct     Nov     Dec   Annual  \
0  191.954  228.00  256.432  243.942  152.958  36.694  34.780  103.311
1  183.003  319.53  171.593  369.323  139.869   3.440  45.780    6.459
2  117.787  151.87  118.046   71.556    0.820   9.123  43.868   80.648

   Jan-Feb  Mar-May  Jun-Sep  Oct-Dec
0   68.508   57.086  158.553   22.083
1   73.985   59.661   38.950  179.969
2   76.403   47.832  163.176  114.788
```

```
In [78]: print("info") #checking the data type of attributes
         print(data.info())

         info
         <class 'pandas.core.frame.DataFrame'>
         RangeIndex: 18 entries, 0 to 17
         Data columns (total 20 columns):
          #   Column    Non-Null Count  Dtype
         ---  ------    --------------  -----
          0   State     18 non-null     object
          1   District  18 non-null     object
          2   Year      18 non-null     int64
          3   Jan       18 non-null     float64
          4   Feb       18 non-null     float64
          5   Mar       18 non-null     float64
          6   Apr       18 non-null     float64
          7   May       18 non-null     float64
          8   Jun       18 non-null     float64
          9   July      18 non-null     float64
          10  Aug       18 non-null     float64
          11  Sep       18 non-null     float64
          12  Oct       18 non-null     float64
          13  Nov       18 non-null     float64
          14  Dec       18 non-null     float64
          15  Annual    18 non-null     float64
          16  Jan-Feb   18 non-null     float64
          17  Mar-May   18 non-null     float64
          18  Jun-Sep   18 non-null     float64
          19  Oct-Dec   18 non-null     float64
```

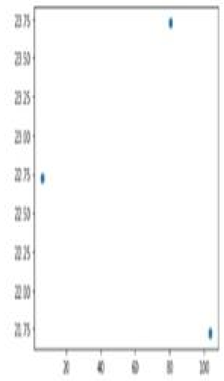
```
In [14]: print("box plot data of annual rainfall in year") #box plot
         data['Annual'].plot(kind='box',sharex=False,sharey=False)

         box plot data of annual rainfall in year

Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x2505934250b0>
```




```
In [15]: print("annual rainfall of all state") #this shows annual rainfall of all state
         data['Annual'].hist(bins=20)

         annual rainfall of all state

Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x250593b0e50>
```



```
In [13]: print("scatter plot of annual and janurary attribute") #visulize the data using different plotting techniques like scatter plo
         plt.scatter(data.Annual,data.Jan)

         scatter plot of annual and janurary attribute

Out[13]: <matplotlib.collections.PathCollection at 0x2505829d640>
```



```
In [16]: print("histogram of jan to dec rainfall of all state")
         data['Jan'].hist(bins=20)
         data['Feb'].hist(bins=20)
         data['Mar'].hist(bins=20)
         data['Apr'].hist(bins=20)
         data['May'].hist(bins=20)
         data['Jun'].hist(bins=20)
         data['July'].hist(bins=20)
         data['Aug'].hist(bins=20)
         data['Sep'].hist(bins=20)
         data['Oct'].hist(bins=20)
         data['Nov'].hist(bins=20)
         data['Dec'].hist(bins=20)

         histogram of jan to dec rainfall of all state
```

```
In [23]: #train  the model using train_test_split and test it using various performance measure( like mean squared error,root mean square
         # and also polt the scatter plot of expected and predicted values.

         from sklearn import linear_model
         print("_multiple linear regression model between annual flood and periodic flood_")
         y=data['Annual']
         x=data[['Jun-Sep']]
         train_x,test_x,train_y,test_y=train_test_split(x,y,test_size=0.1,shuffle=False)
         print("Train x shape",train_x.shape,"; Test_x",test_x.shape)
         print("Train y shape",train_y.shape,"; Test_y",test_y.shape)
         ln=linear_model.LinearRegression()
         ln.fit(train_x,train_y)
         pred=ln.predict(test_x)
         print("mean squared error",mean_squared_error(test_y,pred))
         print("root mean squared error",np.sqrt(mean_squared_error(test_y,pred)))
         print("r2_score",r2_score(test_y,pred))
         print("mean absolute error",mean_absolute_error(test_y,pred))
         plt.scatter(pred,test_y)
         plt.xlabel("Train_X")
         plt.ylabel("Train_Y")
         plt.show()
```

```
        expected.append("flood is coming with low ratio")
predicted=[]
for i in pred:
    if i>2000:
        predicted.append("flood is coming with  high ratio")
    else:
        predicted.append("flood is coming with low ratio")
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
acc=accuracy_score(predicted,expected)
clas=classification_report(predicted,expected)
print("accuracy")
print(acc)
print("classification")
print(clas)
```

```
accuracy
1.0
classification
                                precision    recall  f1-score   support

flood is coming with low ratio       1.00      1.00      1.00         1

                      accuracy                           1.00         1
                     macro avg       1.00      1.00      1.00         1
                  weighted avg       1.00      1.00      1.00         1
```

III. RESULT


```
        expected.append("flood is coming with low ratio")
predicted=[]
for i in pred:
    if i>2000:
        predicted.append("flood is coming with  high ratio")
    else:
        predicted.append("flood is coming with low ratio")
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
acc=accuracy_score(predicted,expected)
clas=classification_report(predicted,expected)
print("accuracy")
print(acc)
print("classification")
print(clas)
```

```
accuracy
1.0
classification
                                precision    recall  f1-score   support

flood is coming with low ratio       1.00      1.00      1.00         1

                      accuracy                           1.00         1
                     macro avg       1.00      1.00      1.00         1
                  weighted avg       1.00      1.00      1.00         1
```


REFERENCES

[1] Qianke , xintian "Urban pluvial flooding prediction by machine learning approaches" Advances in water resources. Vol145 page. No. 103719.2020

[2] Young Jiana "Urban pluvial flooding and storm water management. A contemporary review of chains challenges and sponge cities strategy" Environmental science and policy vol 80 pages no. 32-143 Nov. 2017.

[3] Mohammad Hossein Jahangir "spatial prediction of flood zonation mapping in kan river Basin, Iron using artificial neural network algorithm" Weather and climate extremes vol. 25 page. No. 100215 . 2019

[4] Md Ranquibull "An effective ADOV based flooding detection and prevention for smart meter network" Science direct. Vol 129.page no. 454-460 . 2018.

[5] Yuntao wang "An integrated framework for high resolution urban flood modelling considering multiple information sources and urban features" Environmental modelling software vol 107.page no.85-95 2018.

[6] Zhice fang,Yiwang "predicting flood susceptibility using LSTM neural networks" Journal of hydrology vol. 594 page no.125734.2021 .

[7] Jessica almond " sustainable blue-green infrastructure: a social practice approach to understanding community preferences and stewardship" science direct vol. 191.page no.103639. Aug.2019.

[8] J.B. Williams " Residence perception of sustainable drainage system as highly functional blue-green infrastructure" science direct vol.190.page no. 103610.july2019.

[9] Daniel coles “Beyond flood hotspot modelling emergency services accessibility during flooding in York, UK “ Journal of hydrology vol. 546 page no. 419-436 .2017 .
[10] X.J.pan “ coupling of urban green space system and sponge city construction” earth and environmental science vol.346.page no. 012014,2019.
[11] Liping Dai “ governance of the sponge city program in chain Wuhan as a case study” vol. 34. Page no. 578-596. Sep.2017.
[12] E. MC. Bean “ Evolutionv of a double pipe technology performance for sponge city” advancement in civil engineering and technology vol.1.page no.5 Aug 2018.
[13] R.K. Kamble “ Ecosystem approach to flood desarter risk reduction” international journal of environment.vol.-2.page no. 2854. Nov.2013.
[14] Faith Kashunchan “ Sponge city in China- A break through of planning and flood risk management in urban context” vol.-
10.page no.-1016March 2018.
[15] Zongmin Li “ A literature mining of sponge city : trends, FOCI and challenges standing ahead “ vol -10 page no-1182 April 2018.125734.2021.