Flood Prediction And Rainfall Analysis Using LightGradient Boosted Machine

Dr. Shikha Chadha[1]*, Ms. Rosey Chauhan[2], Ms. Neha Gupta[3]

***Corresponding author**: - Dr. Shikha Chadha
*1 AP, IT Deptt., JSS Academy of Technical Education, Noida, shikhaverma@jssaten.ac.in
2 AP, IT Deptt., JSS Academy of Technical Education, Noida, roseychauhan@jssaten.ac.in
3 AP, IT Deptt., JSS Academy of Technical Education, Noida, neha.gupta@jssaten.ac.in

Abstract

Floods are among the most destructive, most devastating natural disasters complex to model. Research into the development of flood prediction models contribute to risk reduction, policy proposals, minimize loss of human life, and to reduce flood damage. Imitation of complexity statistical discourses on flood processes, over the past two decades, through this research it is aimed to analyze rainfall and historic flood data of Kerala. We will use different ML models to predict floods and also compare the accuracy of these models. The different classifiers which are used in this project are: - GaussianNB, Random Forest, Logistic Regression, Decision Tree, LGBM, XG Boost. A comparison of these models is studied and found that LGBM is the ideal model for the prediction in terms of classification. The rainfall prediction accuracy is 70.98%, and the false predictionof droughts and floods is the lowest, 8% and 4%. Classification model Light Gradient Boost Machine is applied on the flood dataset of the state Kerala and obtained an accuracy of 88% with high recall of 90.5% indicating that there are less false negative cases.

Keywords: Risk Reduction, Policy Proposals, GaussianNB, Random Forest, Logistic Regression, Decision Tree, XGBoost

DOI Number: 10.14704/nq.2022.20.9.NQ44194 **Neuro Quantology 2022; 20(9):1690-1696**

1. INTRODUCTION

Flooding is one of the most common natural disasters on the planet and causes catastrophic consequences. The formation of a flood is closely related to the rainfall intensity, climatic conditions, topographical characteristics, human activities, and drainage conditions of the area. Floods often occur in small, remote, and mountainous basins. They strike quickly with little orno warning, which causes or threatens serious disruption of life and property including death and injury to an oversized number of individuals, and requires therefore, mobilization of efforts in way over that which are normally provided by statutory emergency services. Flooding has immediate consequences such as loss of life, property damage, crop devastation, livestock loss, and worsening of health due to waterborne infections. Floods can also traumatize victims and their families for long periods of time. States like Bihar, Uttarakhand observe floods very frequently. According to statistics, every time a flood occurs, there is a significant loss of life and property. According to our observation the stats for loss have decreased subsequently in the 21st Century due to various Machine learning models and flood prediction but as much as we have seen we have observed that all models use rainfall analysis for flood prediction which are not very accurate we are willing to use some of the other parameters to predict the upcoming flood according to availability of dataset of major rivers. This will help in more accurate flood prediction June 2013 North Indian floods Heavy rain due to a burst of a cloud caused severe floods and landslides in the North Indian states, mainly Uttarakhand and nearby states. More than 5,700 people were presumed dead [12]. Floods like these are rare but very dangerous; these cannot be predicted; they are basically flash floods. But using flood runoff data on time series analysis may help in predicting the danger zone during the expected time of the year. There are several ways to predict or reduce the risks using artificial intelligence, machinelearning, and deep learning. By utilizing this method at the moment, it facilitates tasks in visualizing, analyzing, and predicting natural floods.

The objectives of this research are: -

- To be able to see and visualize the historical trends of annual rainfall.
- To analyze the flood trends in Kerala
- To know any possibility of flood due to heavy rainfall in the near future.

2. LITERATURE REVIEW

A real-time flood disaster prediction system is proposed [1] with the help of machine learning techniques. The basic framework of the flood-based real-time forecast model proposed for this project is based on the relationship between the accumulating rainfall and the accumulated volume represented by the ambiguous model. But after using LHS and PNN he realized that it is difficult to provide real-time information on flood-prone urban areas. Floods are the foremost calamitous natural disasters as informed in [2], which are strenuous to model, where Machine Learning models were toughstone through the chemical analysis of sapness, accuracy, effective-ness, and speed are particularly the most investigated to produce an in-depth overview on assorted ML algorithms utilized in the domain. The performance comparison of Machine-learning models presents an in-depth understanding of various techniques within the framework of global evaluation by using Deep-Learning, Artificial Neural -Networks (ANNs), Multilayer-Perceptron (MLP), Adaptive Neuro-Fuzzy Inference System (ANFIS). We concluded that it compares all the DL and ML algorithms to conclude that which of the following is the best algorithm technique, best of them are Artificial Neural Networks with an accuracy of 90 to 95.

Development of a method for flood detection based on Sentinel images and classifier is proposed [3]. The basis of the proposed route is based on the threshold. Thus,gamma PDF is used to select two thresholds in a SAR image. An object-based classification (OBC) has been implemented. A fast ICA algorithm was implemented which is not only efficient in the speckle reduction and keeping the details, but also very fast.

Flood susceptibility modelling is introduced [4] using advanced ensemble machine learning models. The five machine learning models, such as ANN, SVM, RF, subspace random, and Dagging, were employed to compute the maps of floodsusceptibility for each pixel in the basin.

Efficient rainfall prediction and analysis was worked on [5] using Machine Learning Techniques. The proposed system forecasts rainfall with two machine learning techniques: logistic regression and random forest, for a more precise solution. The results reveal that the characterization procedures used worked well for the no-downpourclass but not so well for the downpour class.

Idea of urban flood prediction was worked on [6] using machine Learning and GIS. The GIS model was able to adjust the sensitivity of the predictions obtained fromthe ML model based on the hot spots observed in the history. The project methodology was structured in accordance with the Cross Industry Standard Process for Data Mining (CRISP-DM).

A model for evaluation of a global ensemble flood predictionsystem was prepared [7] in Peru. This study presents a test framework, in which the Global Flood Awareness System (GloFAS) capabilities are tested in Peru in 2009–2015. The simulated GloFAS release compared to the visibility of 10 river scales. The results of post-processing forecasts show a decrease in variability. Similar to immature predictions, skill scores indicate that performancedecreases with LT growth.

A flood susceptibility modelling is proposed [8] using advanced ensemble machine learning models based on concerns mentioned above, the main aim of the present study was to investigate and compare the ANN, SVM, RF, RS, and Dagging model's ability to predict flood susceptibility maps. In the case of Dagging model, the smallest area percentage (14.03%) comes under the very high class, followed by low (14.89%), high (15.61%), moderate (23.78%), and very low 31%, 70%, 42% classes.

A hybrid model is proposed [9] for fast and probabilistic urban pluvial flood prediction this technique is based on the characteristics of these sub catchments, lumped conceptual models were employed to simulate the hydrological responses of the sub catchments for given rainfall events. Flood warning issued for probabilities above 0.5 achieved a balanced accuracy of approximately 86%.

A rainfall forecasting model is proposed [10] using machine learning methods. His forecasting model uses four different ML algorithms, which are Bayesian Linear Regression (BLR), Boosted Decision Tree Regression (BDTR), Decision Forest Regression (DFR) and Neural Network Regression (NNR). He used two methods forecasting with the help of auto correlation, Forecasting using projected errors.

A new approach for the prediction is proposed [11] of water level in association with flood severity using the ensemble model uses the latest breakthroughs in the Internet of Things (IoT) and machine learning to automate flood data processing, which could be valuable in preventing natural disasters .His results indicated that early warning of flood severity can be obtained using appropriate ensemble machine learning based data science techniques thus he tried to propose IoT-enabled Flood Severity Prediction via Ensemble Machine Learning .

3. SYSTEM ARCHITECTURE AND METHODOLOGY

System architecture contains the steps involved in the workflow of research. The first step is to collect the data, data can be collected via secondary and primary sources. Data is collected from secondary data source which is the pre trained dataset. Then data analysis is applied to find the missing values if found then fill them. After that data exploratory analysis is done to analyze dataset. Here we plot different graphs such as boxplot, bar plot, gg-plot, re-plot, heatmap are plotted to analyze the trend our data. On the basis of the analysis, we conclude the further working steps. First seasonality is checked and then analysis of deseasonalized monthly and seasonal data is done which indicated that the seasonality is there, but it is random. In the end comparison of different classification models for flood analysis is done to find the best model. Different regression models are also compared for rainfall analysis. Then the best classification model is applied on flood dataset and check the accuracy of the result.

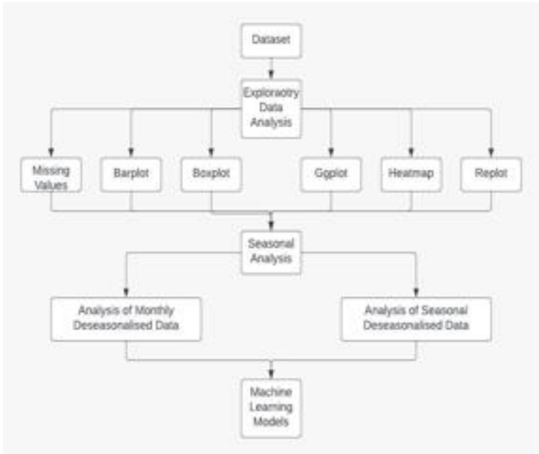


Fig 3.1 workflow of research

3.1 Dataset

```
# the data set contains 117 rows and 19 columns
df = pd.read_csv('C:/Users/ayush/Documents/Kerala-Rainfall-Historical.csv')
print(df.shape)
df.head()
```

(117, 19)

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	JF	MAM	JJAS	OND
0	Kerala	1901	28.7	44.7	51.6	160.0	174.7	824.6	743.0	357.5	197.7	266.9	350.8	48.4	3248.6	73.4	386.2	2122.8	666.1
1	Kerala	1902	6.7	2.6	57.3	83.9	134.5	390.9	1205.0	315.8	491.6	358.4	158.3	121.5	3326.6	9.3	275.7	2403.4	638.2
2	Kerala	1903	3.2	18.6	3.1	83.6	249.7	558.6	1022.5	420.2	341.8	354.1	157.0	59.0	3271.2	21.7	336.3	2343.0	570.1
3	Kerala	1904	23.7	3.0	32.2	71.5	235.7	1098.2	725.5	351.8	222.7	328.1	33.9	3.3	3129.7	26.7	339.4	2398.2	365.3
4	Kerala	1905	1.2	22.3	9.4	105.9	263.3	850.2	520.5	293.6	217.2	383.5	74.4	0.2	2741.6	23.4	378.5	1881.5	458.1

Fig 3.2 Dataset

Monthly rainfall for the Indian state of Kerala from 1901 to 2017 [14]. For each year,monthly rainfall is shown in each column. There is also an annual rainfall column.

3.2 Removing null values

Fig 3.3: Removing null values

Neuro Quantology | August 2022 | Volume 20 | Issue 9 | Page 1690-1696 | doi: 10.14704/nq.2022.20.9.NQ44194

3.2.1 Distribution of Rainfall and Regression

```
sns.set_context("notebook")
plt.style.use('ggplot')
plt.subplots(figsize=(16, 8))
g1=sns.regplot(data=df_season, x='YEAR', y='ANNUAL', order=1, ci=95, label='Annual',
               marker='o', scatter_kws={'s':10}, color='black')
g2=sns.regplot(data=df_season, x='YEAR', y='JF', order=1, ci=95, label='Jan-Feb',
               marker='o', scatter_kws={'s':10}, color='orange')
g3=sns.regplot(data=df_season, x='YEAR', y='MAM', order=1, ci=95, label='Mar-May',
               marker='o', scatter_kws={'s':10}, color='green')
g4=sns.regplot(data=df_season, x='YEAR', y='JJAS', order=1, ci=95, label='Jun-Sep',
               marker='o', scatter_kws={'s':10}, color='red')
g5=sns.regplot(data=df_season, x='YEAR', y='OND', order=1, ci=95, label='Oct-Dec',
               marker='o', scatter_kws={'s':10}, color='blue')
g1.set_title("Distribution of Kerala's Rainfall and the Regression", y=1)
g1.set(xlabel='Year', ylabel='Precipitation (mms)')
plt.xticks(rotation=0)
plt.legend()
sns.set_context("paper")
plt.show()
```

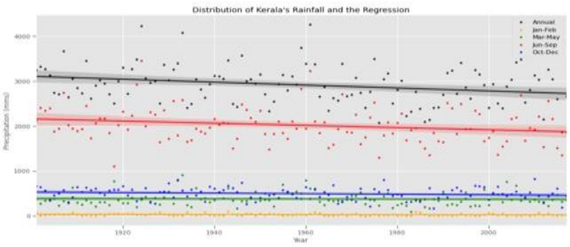
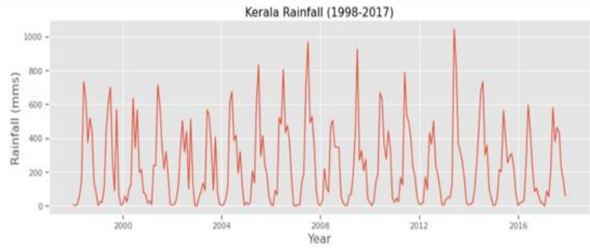


Fig 3.4: Distribution of Rainfall and Regression

Dispersion of the data points from the regression line varies therefore there is more variance in the dataset hence there is a seasonality with a possibility of many outliners and we can see a trend of decreasing rainfall annually.

3.2.2 Checking for Seasonality

```
sns.relplot(data=df_sorted[-240:], x='date', y='mms', kind='line', height=4, aspect=3)
plt.title("Kerala Rainfall (1998-2017)", fontsize=16)
plt.xlabel('Year', fontsize=16)
plt.ylabel('Rainfall (mms)', fontsize=16)
plt.xticks(rotation=0)
plt.yticks(rotation=0)
plt.show()
```



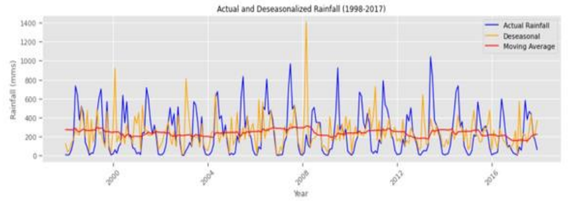
This line plot shows the rainfall in Kerala from 1998 to 2017. It seems the precipitation increases up and down periodically every year.

Fig 3.5 Checking for Seasonality

Above graph shows us that our dataset is periodic with crest and drought in certain months

3.2.3 Seasonal Analysis of Monthly Data

```
sns.set_context("notebook")
plt.subplots(figsize=(16, 4))
g1=sns.lineplot(data=df_sorted[-240:], x='date', y='mms', label='Actual Rainfall', color='blue')
g2=sns.lineplot(data=df_sorted[-240:], x='date', y='deseasonal', label='Deseasonal', color='orange')
g3=sns.lineplot(data=df_sorted[-240:], x='date', y='mov_avg', label='Moving Average', color='red')
g1.set_title('Actual and Deseasonalized Rainfall (1998-2017)', y=1)
g1.set(xlabel='Year', ylabel='Rainfall (mms)')
plt.xticks(rotation=44)
plt.legend()
plt.show()
```



The line plot demonstrates the deseasonalized rainfall has many outliners. The moving average numbers go smoothly.

```
df_sorted['period'] = df_sorted.index + 1

X_train = df_sorted['period']
y_train = df_sorted['mms']
X_train = sm.add_constant(X_train)
olsreg = sm.OLS(y_train, X_train)
olsreg = olsreg.fit()
print('Original Rainfall')
print('Intercept: {}, Coefficient: {}'.format(olsreg.params[0], olsreg.params[1]))
print('R-Squared:', olsreg.rsquared)

Original Rainfall
Intercept: 257.92735438711924, Coefficient: -0.021455734565719836
R-Squared: 0.0011508387916198126
```

```
X_train = df_sorted['period']
y_train = df_sorted['deseasonal']
X_train = sm.add_constant(X_train)
olsreg = sm.OLS(y_train, X_train)
olsreg = olsreg.fit()
print('Deseasonalized Rainfall')
print('Intercept: {}, Coefficient: {}'.format(olsreg.params[0], olsreg.params[1]))
print('R-Squared:', olsreg.rsquared)
print(olsreg.summary())

Deseasonalized Rainfall
Intercept: 264.4360132473951, Coefficient: -0.0307920960935075
R-Squared: 0.005512372162586376
```

Fig 3.5 Seasonal Analysis of Monthly Data

The moving average numbers go smoothly and the deseasonalized rainfall has many outliners. The R-Squared number only increases by 0.6% and the deseasonalized r-squared number is 0.006 which tells our prediction will be imprecise to be useful.

```
sns.set_context("notebook")
plt.subplots(figsize=(16, 4))
g1=sns.lineplot(data=season_sorted[-40:], x='year_s', y='q_mms', ci=None, label='Actual Rainfal
l', color='blue')
g2=sns.lineplot(data=season_sorted[-40:], x='year_s', y='deseasonal', ci=95, label='Deseasonal',
color='orange')
g3=sns.lineplot(data=season_sorted[-40:], x='year_s', y='mov_avg', label='Moving Average', color
='red')
g1.set_title('Actual and Deseasonalized Rainfall Optimized by 4 Seasons(1998-2017)', y=1)
g1.set(xlabel='Year', ylabel='Rainfall (mms)')
plt.xticks(rotation=44)
plt.legend()
sns.set_context("paper")
plt.show()
```

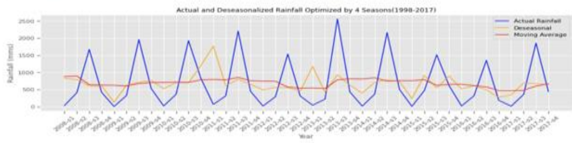


Fig 3.6: Deseasonalized rainfall

The moving average numbers go smoothly and the deseasonalized rainfall has many outliners. The R-Squared number only increases by 0.15% and the deseasonalized r-squared number is 0.015 which tells our prediction will be imprecise to be useful. Hence our data is seasonal but cannot be calculated using a factor.

4. Comparison of Classification Machine Learning Models

```
bar_x = ['GaussianNB', 'Random Forest', 'Logistic Regression',
     'Decision Tree', 'LGBM', 'XGBoost']
bar_y = [accuracy_gnb, accuracy_rf, accuracy_lr, accuracy_dt, accuracy_lgb, accuracy_xgb]
rounded_y = [round(num, 4) for num in bar_y]

fig = px.bar(x=bar_x, y=bar_y, text=rounded_y, title='Accuracy Comparison between Models',
             color=rounded_y, color_continuous_scale='sunsetdark',
             labels={'x':'Machine Learning Models', 'y':'Accuracy Score'}, height=500)
fig.update_layout(title_x=0.5)
fig.show()
```

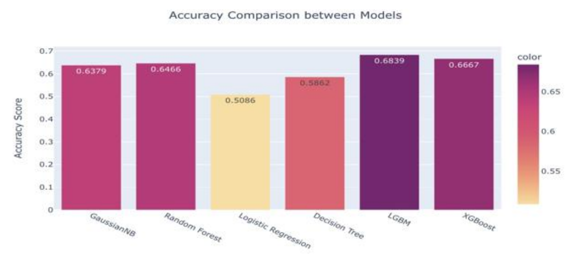


Fig: 3.7

Despite LGBM and XGBoost, the accuracy of GaussianNB and Random Forest appears to be acceptable on this bar chart. Despite this, LGBM has the greatest accuracy rating [15], with a score of 70.98 percent. It's possible that it's the basis for this Kerala rainfall forecast. The confusion matrix will be reviewed in the following section of this analysis, and the best answer will bechosen.

```
fig = go.Figure()
fig.add_trace(go.Bar(x=f_drought_list, y=bar_x, name='False Prediction of Droughts',
                     marker_color='lightgrey', text=f_drought_list,
                     textposition='auto', orientation='h'))
fig.add_trace(go.Bar(x=f_flood_list, y=bar_x, name='False Prediction of Floods',
                     marker_color='lightsalmon', text=f_flood_list,
                     textposition='auto', orientation='h'))
fig.update_layout(barmode='group', xaxis_tickangle=0)
fig.update_layout({'showlegend':True,
                   'legend':{'x':0.7, 'y':0.94,
                             'bgcolor':'rgb(246, 228, 129)'}})
fig.update_xaxes(
        title_text = "Machine Learning Models",
        title_font = {"size": 16},
        title_standoff = 12)
fig.update_yaxes(
        title_text = "False Prediction Rate",
        title_font = {"size": 16},
        title_standoff = 12)
fig.update_layout(title_text="False Prediction of Droughts and Floods (The Lower, The Better)",
title_x=0.5, title_y=0.88)
fig.show()
```

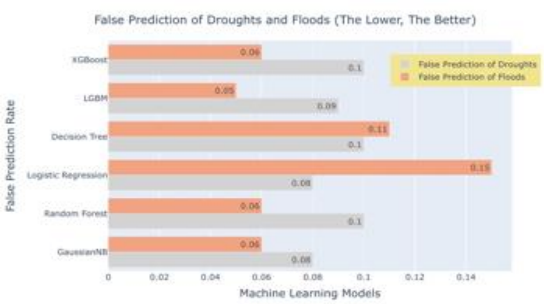


Fig: 3.8

The bar chart shows that LGBM still has the lowest chance of being wrong about natural disasters [16]. LGBM should be the solution for this classification analysis, which is an encouraging result.

5.Comparison of regression machine learning models


```
linear_x = ['LinearRegression', 'Ridge', 'Lasso', 'XGBRegressor', 'ElasticNet']
linear_mae = [mae_lrs, mae_rgs, mae_lss, mae_xgbs, mae_ens]
linear_rmse = [rmse_lrs, rmse_rgs, rmse_lss, rmse_xgbs, rmse_ens]

linear_mae = [round(num, 2) for num in linear_mae]
linear_rmse = [round(num, 2) for num in linear_rmse]

fig = go.Figure()
fig.add_trace(go.Bar(x=linear_x, y=linear_mae, name='MAE',
                     marker_color='lightblue', text=linear_mae,
                     textposition='auto'))
fig.add_trace(go.Bar(x=linear_x, y=linear_rmse, name='RMSE',
                     marker_color='lightgreen', text=linear_rmse,
                     textposition='auto'))
fig.update_layout(barmode='group', xaxis_tickangle=0)
fig.update_layout({'showlegend':True})
fig.update_xaxes(
        title_text = "Machine Learning Models",
        title_font = {"size": 16},
        title_standoff = 12)
fig.update_yaxes(
        title_text = "Errors",
        title_font = {"size": 16},
        title_standoff = 12)
fig.update_layout(title_text="MAE and RMSE to Evaluate the Prediction (The Lower, The Better)",
                  title_x=0.5, title_y=0.88)
fig.show()
```

Fig: 3.9

Despite XGBregressor's superiority, the Ridge may be the second-best option due to its lower MAE and RMSE [17]. The coefficients in the Ridge model also show a steadily decreasing slope in the previous section. As a result, Ridge is regarded as an appropriate regression model for predicting rainfall in Kerala in this study.

6. Applying LGBM on flood data

```
import lightgbm as lgb

model = lgb.LGBMClassifier(learning_rate=0.35,max_depth=-5,random_state=42)
z=model.fit(x_train,y_train,eval_set=[(x_test,y_test),(x_train,y_train)],
        verbose=20,eval_metric='logloss')

[20]    training's binary_logloss: 0.104713    valid_0's binary_logloss: 0.380188
[40]    training's binary_logloss: 0.0236525   valid_0's binary_logloss: 0.351076
[60]    training's binary_logloss: 0.00656326  valid_0's binary_logloss: 0.379676
[80]    training's binary_logloss: 0.00186193  valid_0's binary_logloss: 0.402474
[100]   training's binary_logloss: 0.000505759 valid_0's binary_logloss: 0.484752

print('Training accuracy {:.4f}'.format(model.score(x_train,y_train)))
print('Testing accuracy {:.4f}'.format(model.score(x_test,y_test)))

Training accuracy 1.0000
Testing accuracy 0.8750
```

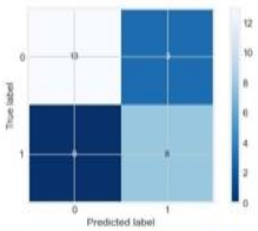
Fig: 3.10

Accuracy -88%

7.RESULTS

```
from sklearn import metrics
metrics.plot_confusion_matrix(model,x_test,y_test,cmap='Blues_r')
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x24ded7a6610>
```




```
print(metrics.classification_report(y_test,model.predict(x_test)))
```

```
              precision    recall  f1-score   support

           0       1.00      0.81      0.90        16
           1       0.73      1.00      0.84         8

    accuracy                           0.88        24
   macro avg       0.86      0.91      0.87        24
weighted avg       0.91      0.88      0.88        24
```

Fig: 3.11

Model accuracy is 88% and recall is 90.5%. This indicates that there are less false negativecases.

8. PERFORMANCE EVALUATION

The goal of this study is to choose a rainfall forecasting prediction model. Because Kerala sends a significant amount of agricultural products to India [18], this analysis focuses on maximizing production while minimizing the risks of natural disasters.

In general, predicting seasonal factors would yield good results if the issue had regular seasonal factors, such as a retailer's sales[19]. However, when using the seasonal factor technique, the results appear to be poor due to the presence of too many outliers. The R-squared scores are too low to effectively predict rainfall. As a result, the next stage is to switch to a machine learning method.

In terms of prediction, the worst-case scenario is that the model mispredicts natural catastrophes, which implies that when choosing a prediction model, the inaccurate prediction of droughts and floods should be taken into account. Following the investigation, LGBM is the best model for categorization prediction. The accuracy of the predictions is 70.98 percent, with the lowest incorrect predictions of droughts and floods at 8% and 4%, respectively.

After the evaluation, XGBRegressor contributes a higher correlation between the prediction and the actual rainfall. The XGBRegressor appears to be an excellent regression model for forecasting [20]. However, because XGBRegressor does not reveal the training element coefficients, it is unknown if it overfits or not. Thus, this analysis chooses Ridge to be the best choice for prediction of rainfall since it has low RMSE and also better coefficients.

The flood model which used the LGBM model had an accuracy of o.88 with a F-score of 0.9 recall 0.81and precision 0.73. This indicates that our model finds almost all false negative casesas our f1 score is high.

9. REFERENCES

H. J. Keum, K. Y. Han, and H. Il Kim, “Real-Time Flood Disaster Prediction System by Applying Machine Learning Technique,” KSCE J. Civ. Eng., vol. 24, no. 9, pp. 2835–2848, 2020.

A. Mosavi, P. Ozturk, and K. W. Chau, “Flood prediction using machine learning models: Literature review,” Water (Switzerland), vol. 10, no. 11, pp. 1–40, 2018, doi: 10.3390/w10111536.

A. Sharifi, “Development of a method for flood detection based on Sentinel-1 images and classifier algorithms,” Water Environ. J., vol. 35, no. 3, pp. 924–929, 2021.

M. Motta, M. de Castro Neto, and P. Sarmento, “A mixed approach for urban flood prediction using Machine Learning and GIS,” Int. J. Disaster Risk Reduct., vol. 56, no. July 2020, p. 102154, 2021.

K. Bischiniotis, B. van den Hurk, E. Zsoter, E. Coughlan de Perez, M. Grillakis, and J. C. J. H.Aerts, “Evaluation of a global ensemble flood prediction system in Peru,” Hydrol. Sci. J., vol. 64, no. 10, pp. 1171–1189, 2019, doi: 10.1080/02626667.2019.1617868.

W. M. Ridwan, M. Sapitang, A. Aziz, K. F. Kushiar, A. N. Ahmed, and A. El-Shafie, “Rainfall forecasting model using machine learning methods: Case study Terengganu, Malaysia,” Ain Shams Eng. J., vol. 12, no. 2, pp. 1651–1663, 2021, doi: 10.1016/j.asej.2020.09.011.

X. Li and P. Willems, A Hybrid Model for Fast and Probabilistic Urban Pluvial FloodPrediction, vol. 56, no. 6. 2020.

M. Khalaf, H. Alaskar, A. J. Hussain, T. Baker, Z. Maamar, R. Buyya, P. Liatsis, W. Khan,H. Tawfik, and D. Al-Jumeily, “IoT-Enabled flood severity prediction via ensemble machine learning models,” IEEE Access, vol. 8, pp. 70375–70386, 2020.

Y. Liu and L. Wu, “Geological Disaster Recognition on Optical Remote Sensing Images Using Deep Learning,” Procedia Comput.Sci., vol. 91, pp. 566–575, 2016.

T. K. Sahaet al., “How far spatial resolution affects the ensemble machine learning based flood susceptibility prediction in data sparse region,” J. Environ. Manage., vol. 297, no. December 2020, p. 113344, 2021, doi: 10.1016/j.jenvman.2021.113344.

X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. K. Wong and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting" in Advances in Neural Information Processing Systems, Red Hook, NY, USA:Curran, vol. 28, pp. 802-810, 2015.

“Floods in India.” Wikipedia, https://en.wikipedia.org/wiki/Floods_in_India. Accessed 26 May 2022.

Naive Bayes — scikit-learn 1.1.1 documentation.” Scikit-learn, https://scikit-learn.org/stable/modules/naive_bayes.html. Accessed 26 May 2022.

India-WRIS, https://indiawris.gov.in/wris/. Accessed 26 May 2022.

N. Gupta, R. Chauhan, and S. Chadha, “Unmanned Aerial Vehicle (UAV) for parcel delivery,” Int. J. Eng. Res. Technol., vol. 13, no. 10, pp. 2824–2830, 2020, doi: 10.37624/IJERT/13.10.2020.2824-2830.

U. Deep and L. Techniques, “A Survey: Melanoma Skin Cancer Detection Using Deep Learning Techniques,” pp. 3395–3401, 2021.

N. Gupta and K. K. Rana, “Disaster Prediction and Post Disaster Management Using Machine Learning And Bluetooth,” vol. 18, no. 5, pp. 274–292, 2021.

S. Indexed, N. Gupta, C. Science, and C. Science, “A SURVEY ON DISASTER MITIGATION AND MANAGEMENT: ISSUES, CHALLENGES ,” vol. 12, no. 3, pp. 787–798, 2021.

Chadha, S., S. Mittal, and V. Singhal. & quot; Ancient text character recognition using deep learning. & quot; International Journal of Engineering Research and Technology 3.9 (2020): 2177-2184.

Chadha, S., Mittal, S., & Singhal, V. (2019). An insight of script text extraction performance using machine learning techniques. International Journal of Innovative Technology and Exploring Engineering, 9(1),2581–2588. https://doi.org/10.35940/ijitee.A5224.119119.