School of Engineering and Applied Science (SEAS), Ahmedabad University

B.Tech (ICT) Semester VI / M.tech / PhD: Machine Learning (CSE 523)

- **Group No:** S_ECC4
  1. Jainam Chhatbar (1741002)
  2. Dhairya Dudhatra (1741058)
  3. Charmil Gandhi (1741059)
  4. Jinesh Patel (1741076)

- **Project Title:** Landslide Susceptibility Assessment by Novel Hybrid Machine Learning Algorithms
- **Project Area:** Climate and Environment
- **URL links:** Click here to go to our project drive

1 Introduction

1.1 Background

Landslides are one of the most dangerous natural occurrence which happen very often where there is loose terrain, high slopes, rainfall and high anthropogenic activities like construction and mining [1]. Landslides have vast effects on environmental condition and economics of impacted area [3]. It is not possible to control the landslides, but the damage caused by it can reduced by establishing a system which can predict the landslides or area highly prone to them for better management

Many researches have been done and machine learning has been used greatly to predict the occurrences of landslides based on the type of terrain and location [2]. Researchers have attempted to improve the accuracy of the prediction by applying various decision-trees machine learning algorithms such as random-forests [4], ID-3 decision trees [5], naive Bayes tree [6, 7], kernel logistic regression [8], support vector machine [9] etc. Application of hybrid or ensemble models have shown promising results in correctly predicting the landslide susceptibility on smaller and complex datasets.

Machine Learning is effective in representing connection between different natural indicators and different responses such floods [13, 14], wild fire [15], landslides [16–20] and so forth. In our base article, different Landslides Conditioning Factors(LCF) were utilized to set up a susceptible landslide prediction, for example, distance from roads, distance from rivers, inclination, curvature, vegetation, measure of precipitation, type of soil and so forth [21]. Hybrid models, for example, Bagging, Random Space, and Random Forest [22] as base classifiers were utilized for spatial prediction of landslides. The significant variables were determined by chi-square attribute evaluation technique.

1.2 Motivation

A significant amount of research has been done for forecasting of potential landslide occurrences on the basis of soil and slope of terrain, distance from hydrological water bodies etc. But prediction of landslide triggers have yet to be researched more. Our main goal was to determine the primary triggers for landslides in various locations around the globe. By considering the initial cause of a landslide, we can extend this research to predict the likelihood of a landslide occurrence.

2 Data Acquisition / Explanation of Data set

The dataset used for this project is the Global Landslide Catolog(GLC) export published by NASA [10].This dataset provides many incidences of landslides each with a corresponding trigger. There are in total 22 features in this dataset, out of which we chose 11 features to use for this project (see Table 1). After preprocessing, this dataset includes 9133 landslide triggers with 7966 being rain, 562 tropical cyclone, 133 snow, 129 monsoon, 93 mining, 89 earthquake, 86 construction and 74 flooding.

GLC data features representation	
Feature	**Description**
event_month	The calendar month of the landslide incident
event_time	Time at which landslide event took place
landslide_category	The type of landslide movement-slide(rock slide, debris slide, earth slide), creep, debris_flow, earth_flow, snow_avalanche, lahar, roack_fall, earth_fall, compex(combination of two or mote of the categories.)
landslide_size	The general size of the landslide(small, medium, large and very large)
fatality_count	The number of fatalities due to the landslide
injury_count	The number of injuries due to the landslide
country_name	Name of the country where landslide occurred
population	Population count of the location at which the landslide took place
longitude	Exact longitude of the landslide location
latitude	Exact latitude of the landslide location
landslide_trigger	The trigger that caused the landslide - rain, construction, earthquake, flooding, freeze-thaw, mining, monsoon, snow, and tropical cyclone

3 Machine Learning Concept Used

3.1 Classification

We followed the flow as seen in figure 1. Our problem involved classifying the landslide triggers using supervised learning. Since we had a multiple class problem with 8 landslide triggers, we chose to use classification methods such as Support Vector Machine(SVM) and Random Forest Classifier(RFC). The main reason to use these classifier methods was because of they can be used to train multiple classes while handling the problem of class imbalance. The first step was to pre-process our dataset.

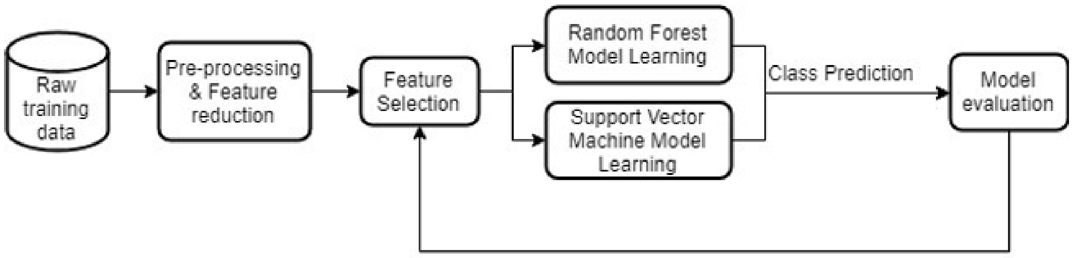

Raw training data
Pre-processing & Feature reduction
Feature Selection
Random Forest Model Learning
Support Vector Machine Model Learning
Class Prediction
Model evaluation

Figure 1: Workflow

3.1.1 Data Pre-Processing

As we have mentioned above, we have used the Global Landslide Catalog(GLC) dataset in order to train our model. Before pre-processing, the dataset consisted of over 11,000 landslide occurences from 1988 to 2017. Large portion of our dataset was related to rainfall triggers, thus causing data imbalance. The original dataset consisted of 17 landslide trigger labels.

As mentioned above, we used the Global Landslide Catalog (GLC) dataset in order to train our model. Prior to preprocessing, the dataset consisted of just over 11,000 landslide occurrences from 1988 to 2017. Because of this a large part of our data was related to rainfall based triggers, which caused imbalance in data. The original dataset consisted of 17 landslide trigger labels. Some of these labels had less than 5 occurrences. After data pre-processing, we narrowed down our class labels to 8 unique landslide triggers which are as follows:

- Rain
- Earthquake
- Construction
- Flooding
- Monsoon
- Snow
- Mining
- Tropical Cyclone

Our dataset consisted of multiple class labels which were associated with rainfall. So, we combined them all into on major class namely Rain because having more than one rainfall related class would create noise within our dataset. After completion of pre-processing, we found that majority of data was imbalanced with *'Rain'* as a majority class and all the other class as minorities. Data distribution for each class after pre-processing can be seen in figure 2.

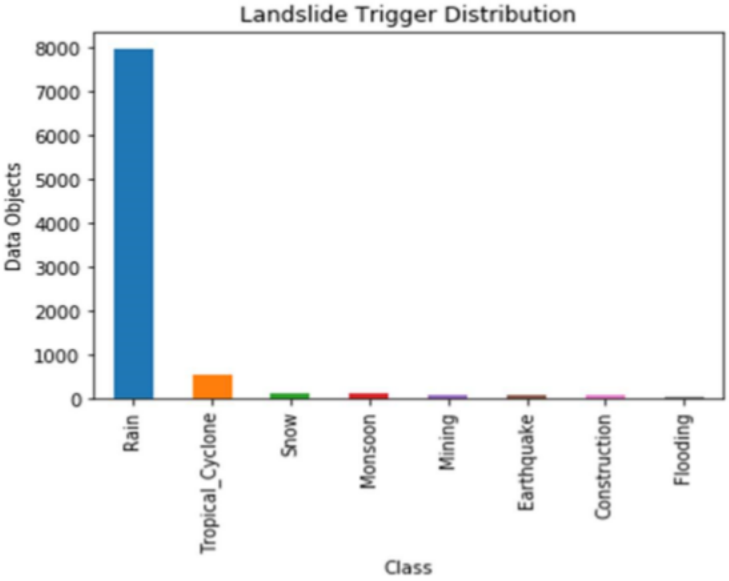


Figure 2: Distribution for each class after data pre-processing

As our dataset consisted of large class imbalance with label Rain making up about 87% of total data, we used oversampling methods of ADASYN and SMOTE to balance again the dataset. First method Adaptive Synthetic Sampling Approach(ADASYN) uses a weighted distribution for different minority classes. This lead to generation of more synthetic data for minority class which were harder to learn [11]. Then, Synthetic Minority Over Sampling Technique(SMOTE) over-samples the minorities by taking every minority class and introducing synthetic examples along the line segments which joined the k-nearest neighbours [12].

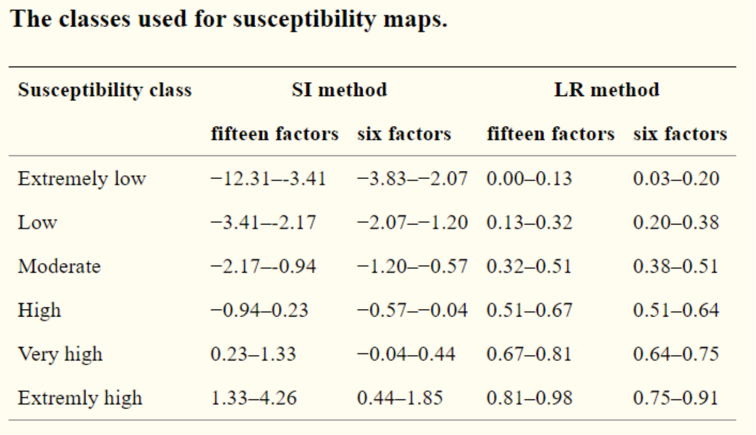
3.2 Principal Component Analysis

Case 1: Considering that dimensions of data set is less than number of samples

For PCA too, we have considered Global Landslide Catalog (GLC). In GLC , we first mapped all the text fields to integer values.

3.3 Linear Regression

Usually 15 factors are used frequently to determine the susceptibility of landslides. Out of these 15 factors, we have just considered the 6 most significant factors to be used in our linear regression model to show relation between number of landslides and susceptibility index (LSI). Landslide susceptibility mapping is essential to describe the propensity of a landslide in a susceptible area. Such results can provide helpful information for the disaster managers, for urban planners, and decision makers in the landslide-prone area. 15 factors considered :



The classes used for susceptibility maps.

Susceptibility class	SI method		LR method	
	fifteen factors	six factors	fifteen factors	six factors
Extremely low	−12.31–-3.41	−3.83–−2.07	0.00–0.13	0.03–0.20
Low	−3.41–-2.17	−2.07–−1.20	0.13–0.32	0.20–0.38
Moderate	−2.17–-0.94	−1.20–−0.57	0.32–0.51	0.38–0.51
High	−0.94–0.23	−0.57–−0.04	0.51–0.67	0.51–0.64
Very high	0.23–1.33	−0.04–0.44	0.67–0.81	0.64–0.75
Extremly high	1.33–4.26	0.44–1.85	0.81–0.98	0.75–0.91

Figure 3: Susceptibility Class Table

- Elevation
- Slope angle
- Slope Aspect
- Total Curvature
- Profile Curvature
- Plan Curvature
- CTI (Hydrological Factor)
- SPI (Hydrological Factor)
- Drainage Density
- Distance to drainage network
- Lithology
- Density of geological boundaries
- Distance to geological boundaries
- Distance to faults
- NDVI (Normalized difference vegetation index)

The above mentioned 15 features have some sub-classes from which the subclass which had text inputs were mapped to integers accordingly. We have selected the features using the Certainty Factor(CF) for that feature Out of these 15 factors, the 6 factors that were selected:

- Slope angle
- Slope Aspect
- Drainage Density
- Lithology
- Distance to geological boundaries
- Distance to faults

NOTE: This dataset was generated by us as the original dataset was not available anywhere. So results are not accurate.

4 Pseudo Code/ Algorithm

4.1 Classification

After pre-processing and oversampling for minority classes was accomplished, we selected Support Vector Machines(SVM) and Random Forest Classifier(RFC) to predict the landslide trigger. Both of these methods are used frequently with supervised learning classification problems which have class imbalance. We chose both SVM and RFC and compared the performance of the both the methods.

Random Forest Classifier(RFC) is ensemble algorithm which creates a set of decision trees from our training dataset. Using majority of votes from each decision tree, the random forest decides the class label of the test data. We trained our model with the help of RandomForestClassifier library provided by Scikits-learn and oversampled training data from ADASYN. Pseudo-code for the mentioned method can be seen below:

```
#Training and test data assumed to be pre-processed and over-sampled
training_features #oversampled
training_labels   #oversampled
test_features
test_labels

#Call our model with desired hyper-parameters
rf = RandomForestClassifier(n_estimators, random_State, class_weight, min_samples_leaf)
rf.fit(train_features, train_labels); #fit our model to training data
rf_predictions = rf.predict(test_features) determine our predicted labels

#Function to analyze our model performance
def model_report(model_predictions):
  triggers = [0,1,2,3,4,5,6,7]
  print(classification_report(test_labels, model_predictions, triggers))
  Confusion_matrix = ConfusionMatrix(test_labels, model_predictions)
  print(Confusion_matrix)
model_report(rf_prediction)
```

Support Vector Machines(SVM) separates classes through linear separators. SVM was originally designed for binary classification, but it can also be used to solve multiple class classification problems. We implemented one-vs-one which trains a seperate classifier for each different pair of labels. We get :

$$Classifiers = \frac{N(N-1)}{2}$$

Here N is the number of different class labels. We chose this method because of its insensitivity towards problems of imbalanced datasets. But, it has a tradeoff of being computationally expensive.

4.2 Principal Component Analysis

Consider an i.i.d. dataset X = $x_1, \ldots, x_N$, $x_n \in R^D$, with mean 0 and covariance matrix,

$$\text{S} = \frac{1}{N}\sum_{n=1}^{N} x_n x_n^T$$

Furthermore, we assume there exists a low-dimensional compressed representation of x_n,

$$z_n = B^T x_n \in R^M$$

where, B:=$[b_1, \cdots_n] \in R^{D\times M}$ Projection matrix

And we assume that columns of B are orthonormal, i.e,

$$b_i^T b_j = 0 \text{ if and only if i} \neq j \text{ and } b_i^T b_i = 1$$

We seek an M-dimensional subspace U $\subseteq R^D$, dim(U) = M < D.

Now, we want to maximize the variance of first coordinate z_1 of z.

$$V_1 := V_1[z_1] = \frac{1}{N}\sum_{n=1}^{N} z_{1n}^2 \text{ , } z_{1n} = b_1^T x_n$$

$$V_1 = V_1[z_1] = \frac{1}{N}\sum_{n=1}^{N}(b_1^T x_n)^2$$

$$V_1 = V_1[z_1] = \frac{1}{N}\sum_{n=1}^{N} b_1^T x_n x_n^T b_1$$

$$V_1 = V_1[z_1] = b_1^T(\frac{1}{N}\sum_{n=1}^{N} x_n x_n^T)b_1$$

$$V_1 = b_1^T S b_1$$

The Lagragian is obtained as :

$$\mathcal{L}(b_1, \lambda_1) = b_1^T S b_1 + \lambda_1(1 - b_1^T b_1)$$

Taking partial derivatives of $\mathcal{L}$ w.r.t b_1 and $_1$ and equating to zero, we get

$$Sb_1 = \lambda_1 b_1 \text{ and } b_1^T b_1 = 1$$

Now, variance is,

$$V_1 = b_1^T S b_1 = \lambda_1 b_1^T b_1$$

$$V_1 = \lambda_1$$

So, we can reconsturct the data points by,

$$\widetilde{x}_n = b_1 z_{1n}$$

$$\widetilde{x}_n = b_1 b_1^T x_n \in R^D$$

$$Z = \begin{cases} CF1 + CF2\text{-}CF1CF2 & CF1,\, CF2 \geq 0 \\ CF1 + CF2 + CF1CF2 & CF1,\, CF2 < 0 \\ \frac{CF1+CF2}{1-\min(|CF1|,\,|CF2|)} & CF1,\ CF2,\ \text{opposite signs} \end{cases}$$

4.3 Linear Regression

As mentioned above, that we have used Certainty Factor to choose the parameters whose value varies between -1 and 1. Using this certainty factor(CF), Z value for every feature was found out. Once the CF values for classes of the causative factors are obtained, these factors are then incorporated pairwise using the combination rule: The negative value of Z signifies: that particular feature does not contribute significantly in landslides' occuring and were considered for regression model. As multiple features are considered to determine the susceptibility of landslides, Multiple Linear Regression Model (LMR) was used. The value of each class corresponding to a feature (6 significant feature) were used to find the coefficient θ using Maximum Likelihood Estimation (MLE). The values of θ were further used in predictor function which would predict the number of landslides for a given Landslide Susceptibility Index (LSI). The regression model can be represented as :

$$\text{P} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots \theta_6 x_6 + \epsilon$$

In above equation, θ_n represents coefficients of independent parameters and ϵ is error.

4.3.1 Analysis

Let data be D:$(x_1, y_1), (x_2, y_2), \ldots (x_N, y_N)$. For Likelihood function:

$$\text{P(y}|x) = N(y|f(x),\sigma^2)$$

where $X_n \in R^D$ are inputs and $Y_n \in$ R are noisy function values.

$$\text{y} = \text{f(x)} + \epsilon \text{ where } \epsilon\ \ \text{N}(0,\sigma^2)\text{: IID Gaussian Noise}$$

Considering the parameters are linear in the model

$$\text{P(y}|x,\theta) = \text{N(y|x}^T\theta,\ \sigma^2)$$

$$\text{y} = x^T\theta + \epsilon \text{ where } \theta \in R^D$$

Using Probabilistic Graphical Model:

$$\text{P(Y}|X,\theta) = \textstyle\prod_{n=1}^{N}\text{P}(y_n|\text{x}_n,\ \theta)$$

$$\text{P(Y}|X,\theta) = \textstyle\prod_{n=1}^{N}\text{N}(y_n|\text{x}_n^T\theta,\sigma^2)$$

For maximum likelihood of parameters:

$$\theta_{ML} = \arg\ max_\theta \text{p(Y}|X,\theta)$$

Taking negative log-likelihood:

$$L(\theta)\text{= -log[p(Y}|X\text{,}\theta\text{)]}$$

$$L(\theta)\text{= -log}\left[\prod_{n=1}^{N}\text{p}(y_n|\text{x}_n,\ \theta)\right]$$

$$L(\theta)\text{= -}\sum_{n=1}^{N}\text{log[p}(y_n|\text{x}_n,\ \theta)\text{]}$$

$$L(\theta)\text{= -}\sum_{n=1}^{N}\text{log}\ \left[\frac{1}{\sqrt{2\pi\sigma^2}}\exp(\frac{(y_n-x^T\theta)^2}{2\sigma^2})\right]$$

$$L(\theta)\text{= }\frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n-x_n^T\theta)^2$$

$$L(\theta)\text{= }\frac{1}{2\sigma^2}\|y-x\theta\|^2$$

On differentiating above equation with respect to θ and equating it to 0, we get:

$$\theta_{ML}=(X^TX)^{-1}X^TY$$

Now, if inputs are in non-linear transformation,

$$P(y|x,\theta)=\text{N}(y|\phi^T(x)\theta,\sigma^2)$$

$$P(y|x,\theta)=\sum_{k=0}^{k-1}\theta_k\phi_k+\epsilon$$

where ϕ: $R^D\rightarrow\text{R}^K$ Non-linear transformation of input X. Now taking negative log-likelihood:

$$\text{-log[P(y}|X\text{,}\theta\text{)]}=\frac{1}{2\sigma^2}(y-\phi\theta)^T(y-\phi\theta)+c$$

Maximum likelihood parameter estimation for parameter :

$$\theta_{ML}=(\phi^T\phi)^{-1}\phi^Ty$$

5 Coding and Simulation

5.1 Classification

5.1.1 Simulation Framework

To determine the best hyper-parameters for both the models the most challenging task. We took an approach of trial and error and using Scikit-learn's GridSearchCV tool, received the performance scores and adjusted the hyper-parameters as needed and trained our models again. With GridSearchCV, we obtained a starting point from where we can declare the hyper-parameters for SVM and RFC. The result of running GridSearchCV to calculate the C parameter is as follows:

From C parameter,we can know how much we want to avoid mis-classifying each training example. In our case, the best score came from a value 1 of C parameter which was 0.878997. We also used GridSearchCV to identify the parameters for RFC which gave the following best scores:

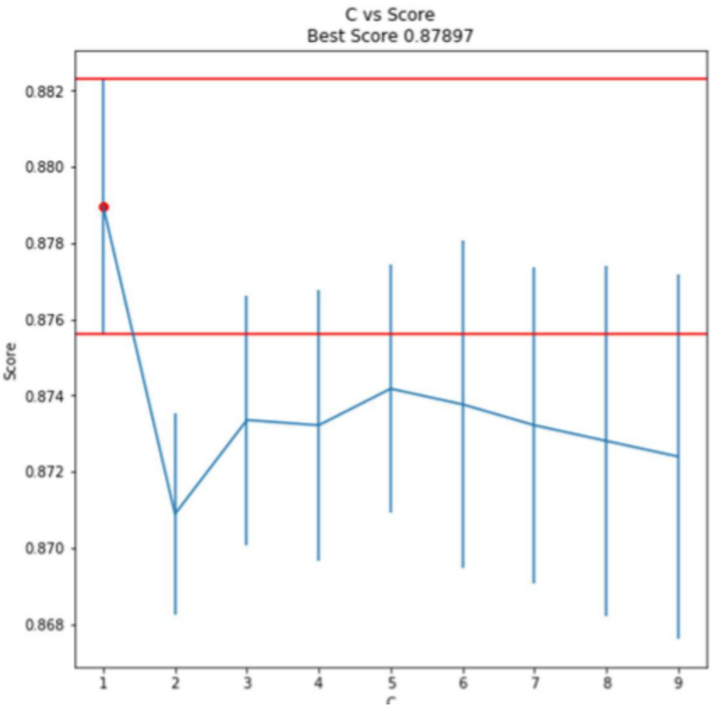


Figure 4: Grid search results for SVM hyperparameter C

- n_estimators = 1500
- random_state = 50
- min_samples = 75
- class_weight = 'balanced'

By using these hyper-parameters in our models, we found that overall performance increased

5.1.2 Evaluation Metrics

We used precision, recall and f1-score for evaluating the performance of our models. These performance metrics allowed us to view how each class in our model reacted to the hyper-parameters that we found above.

- **Precision:** The part of positive class predictions that were correct. This helped determine when the costs of false positive values were high.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

- **Recall:** The part of actual true positive values that were correct. This value is most useful when analyzing our model performance.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

- **f1-score:** A measure which used when we need to search for a balance between precision and recall while having an unbalanced class distribution.

$$f1 - score = 2 \times \frac{precision \times recall}{precision + recall} \tag{3}$$

5.2 Principal Component Analysis

Case 1: We have considered 2000 samples and 11 features. So dimension of our matrix X (which is data set) is [2000 × 11]. As all our features are not in the same range, normalization is required to change the values of numeric columns in data set to common scale and then Covariance Matrix (S) was formed :

$$S = \tfrac{1}{N} \textstyle\sum (X^T \times X)$$

The dimensions of Covariance Matrix (S) is [11 × 11]. The next step was to find Eigen Values and Eigen Vectors related to number of features i.e 11. and then we formed the Projection Matrix given by $(B^T.B)$. Matrix B is formed by the eigen vectors of all components. Dimensions of B is [11 × 11].

Case 2: Considering that dimensions of data set is larger than number of samples
In this case, we have assumed that dimensions of data set i.e features are more than the number of samples. As mentioned in Case 1, we have 11 features. So, for Case 2, we have just considered 7 samples of data. So for this case, dimensions of **X** is [7 × 11], though, the performance will not be great. The Covariance Matrix (S) is given by:

$$S = \frac{1}{N} \sum (X \times X^T)$$

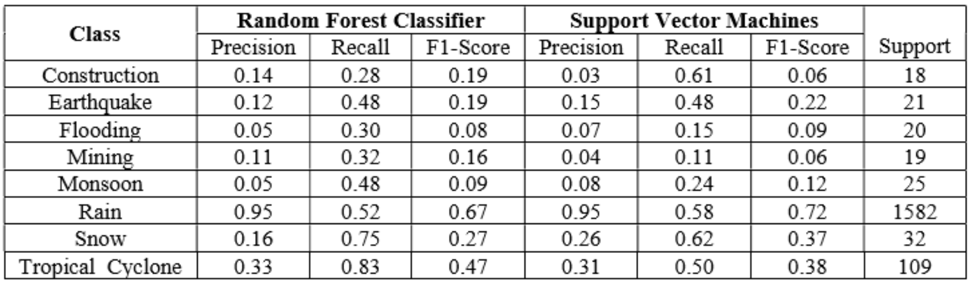
Thus, dimensions of Covariance Matrix (S) is [7 × 7]. The next step is to Eigen Values and Eigen Vectors related to number of features i.e 11. Next we formed the Projection Matrix given by $(B^T.B)$. Matrix B is formed by the eigen vectors of all components. Dimensions of B is [11 × 11].

5.3 Results

5.3.1 Classification

During experiment, we immediately saw that after over-sampling our minority classes with ADASYN and SMOTE, there was a bias towards minority class. Because of imbalanced class labels, accuracy score was misleading to the true performance of our model. Because of this reason, we evaluated model's performance based on precision, recall and f1-score. Over sampling provided an even distribution of training data. But we had relatively low performance scores.

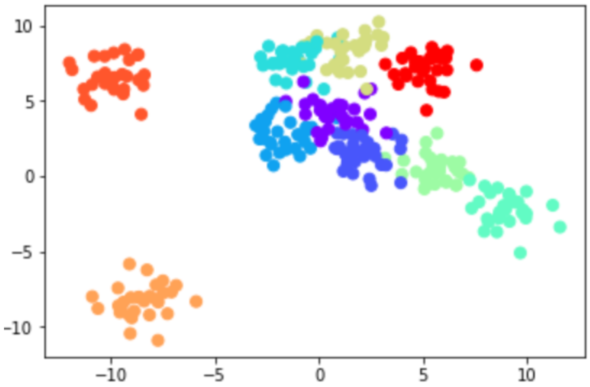
To improve the performance, we removed the noisy data from dataset by filtering each attribute by unique values, we removed data that occured infrequently and filled the missing values with median of that attribute. Feature creation was used to split the event_date attribute into event_month column. After such procedures, we saw improvement in models.
It could be seen that because support levels of test data, precision and f1-score of minority classes were so



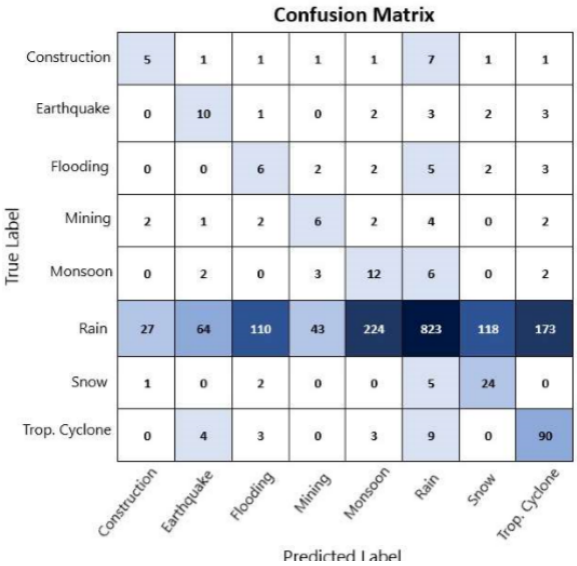
Class	Random Forest Classifier			Support Vector Machines			
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Support
Construction	0.14	0.28	0.19	0.03	0.61	0.06	18
Earthquake	0.12	0.48	0.19	0.15	0.48	0.22	21
Flooding	0.05	0.30	0.08	0.07	0.15	0.09	20
Mining	0.11	0.32	0.16	0.04	0.11	0.06	19
Monsoon	0.05	0.48	0.09	0.08	0.24	0.12	25
Rain	0.95	0.52	0.67	0.95	0.58	0.72	1582
Snow	0.16	0.75	0.27	0.26	0.62	0.37	32
Tropical Cyclone	0.33	0.83	0.47	0.31	0.50	0.38	109

Figure 5: Performance of RFC with ADASYN and SVM with SMOTE

low. Almost all minority classes have support values less than 100, and there is very little possibility of mis-classification.



The precision and recall scores can also be represented with the help of Confusion Matrix with actual values and predicted labels.

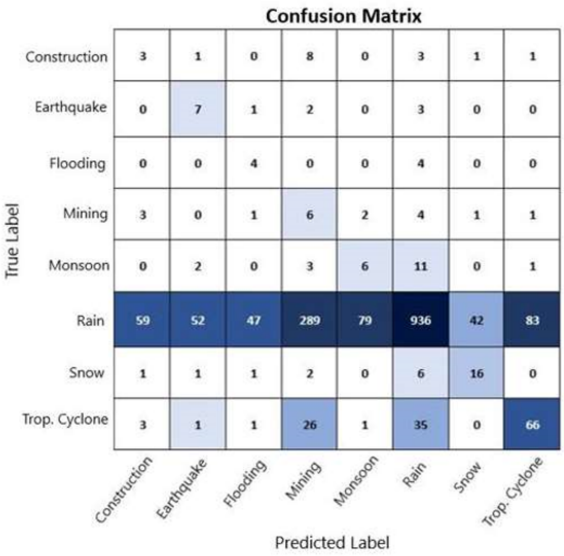


Confusion Matrix

True Label

	Construction	Earthquake	Flooding	Mining	Monsoon	Rain	Snow	Trop. Cyclone
Construction	5	1	1	1	1	7	1	1
Earthquake	0	10	1	0	2	3	2	3
Flooding	0	0	6	2	2	5	2	3
Mining	2	1	2	6	2	4	0	2
Monsoon	0	2	0	3	12	6	0	2
Rain	27	64	110	43	224	823	118	173
Snow	1	0	2	0	0	5	24	0
Trop. Cyclone	0	4	3	0	3	9	0	90

Predicted Label



Confusion Matrix

True Label

	Construction	Earthquake	Flooding	Mining	Monsoon	Rain	Snow	Trop. Cyclone
Construction	3	1	0	8	0	3	1	1
Earthquake	0	7	1	2	0	3	0	0
Flooding	0	0	4	0	0	4	0	0
Mining	3	0	1	6	2	4	1	1
Monsoon	0	2	0	3	6	11	0	1
Rain	59	52	47	289	79	936	42	83
Snow	1	1	1	2	0	6	16	0
Trop. Cyclone	3	1	1	26	1	35	0	66

Predicted Label

Figure 6: Confusion Matrix for Random Forest Classifier (RFC) with ADASYN

Figure 7: Confusion Matrix for Support Vector Machines (SVM) with SMOTE

5.3.2 Principal Component Analysis

Case 1:

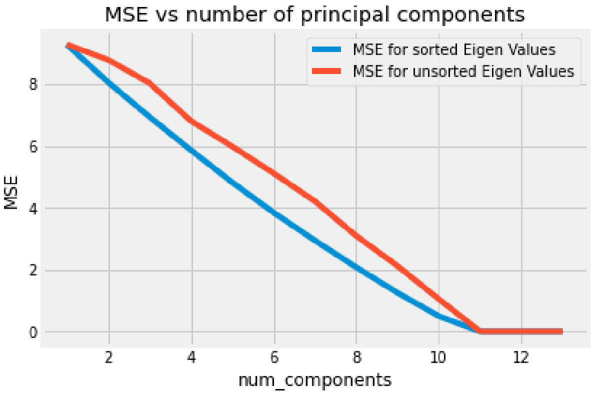
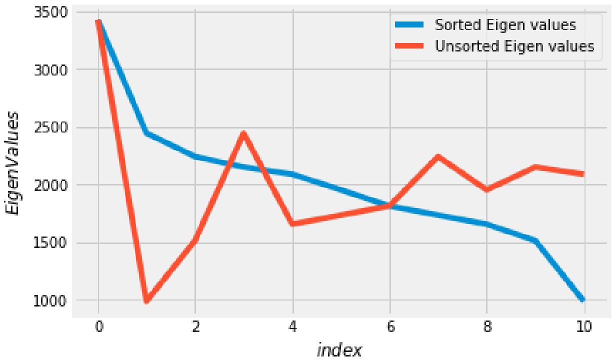


Figure 8: MSE vs Number Of Components

As seen in *Figure 8*, MSE decreases as the number of components increases. Thus, as the dimensions increases, we are covering maximum spread and the error i.e loss in information decreases. Also, significant change is observed in both the case. Though, the trend remains the same : MSE decreases with increase in number of components. Also we can observe from *Figure 11*, that variance increases as number of principal components increases. As shown in figure, we have considered 11 components, so variance increases till 11 and then becomes constant. Also as seen in *Figure 9*, As the both: sorted and unsorted eigen values are displayed. We are observing such graphs because in unsorted eigen values, for same index, it's value may differ from the value in unsorted eigen value.



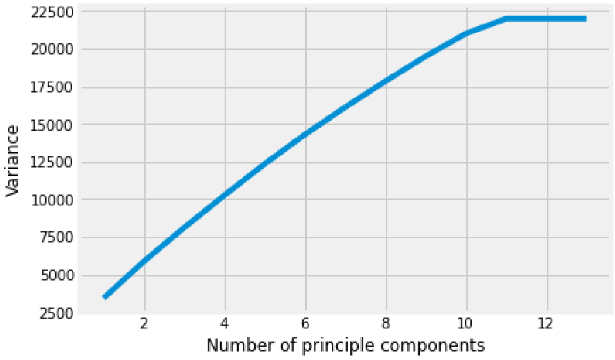


Figure 9: Eigen Values vs Index

Figure 10: Variance vs Number Of Principal Components

Case 2: The trend followed here is very similar to the one in Case 1. MSE decreases as the number of component increases which signifies that the loss in information also decreases. .

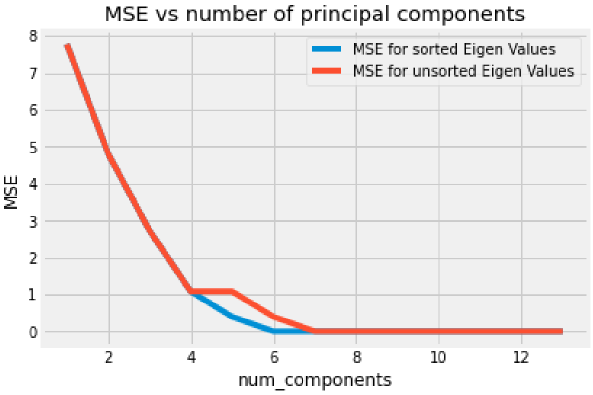
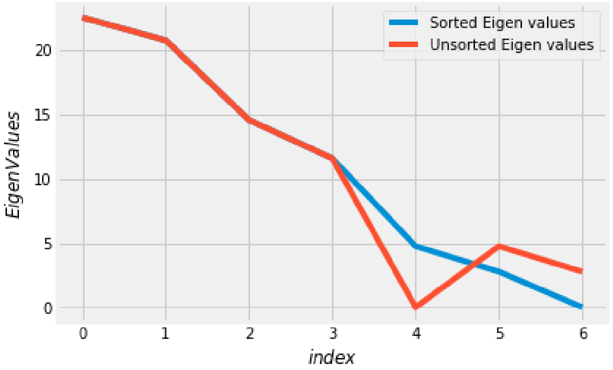


Figure 11: MSE vs Number Of Components

From *Figure 14*, it can be noted that variance increases with the increase in number of principal components which is the similar trend followed in Case 1.



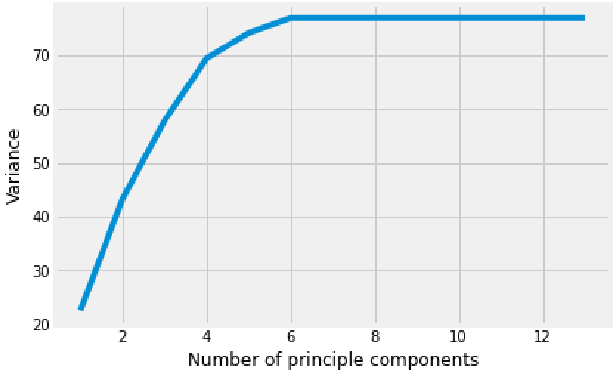


Figure 12: Eigen Values vs Index

Figure 13: Variance vs Number Of Principal Components

5.3.3 Linear Regression

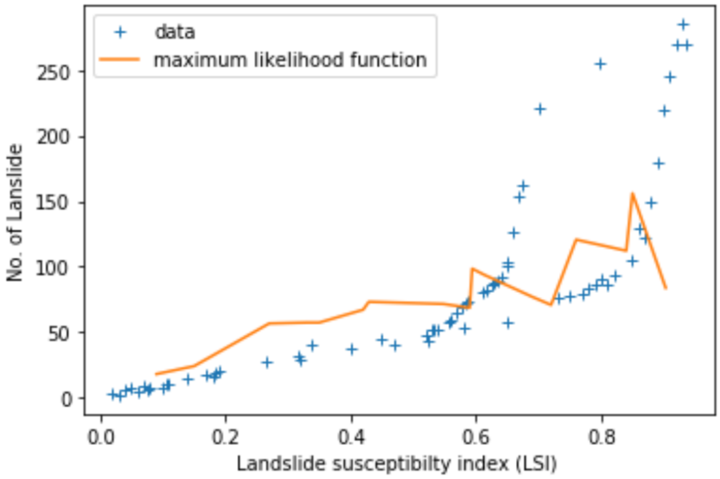


Figure 14: MLE With Intercept

The *figure 14* represents the MLE calculation taking intercept into consideration. This signifies that the output y before we started the data will not always be 0. For this, it is always a good practice to consider the intercept.

Figure 15 represents the MLE calculation without taking intercept into consideration. This signifies that the output y before we started the data will be 0. Thus the predicted data will always pass through origin.

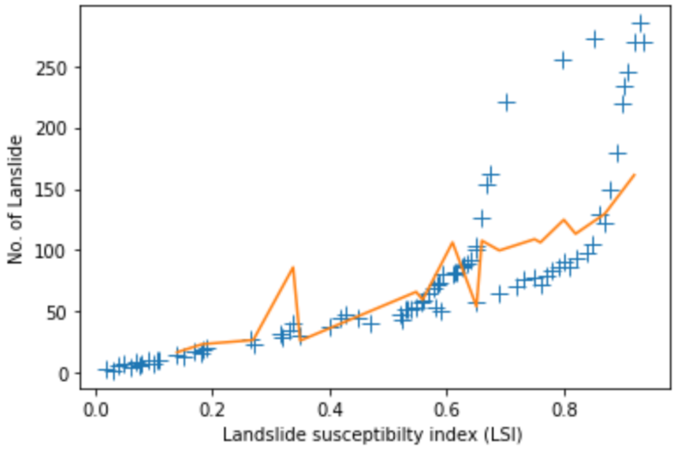
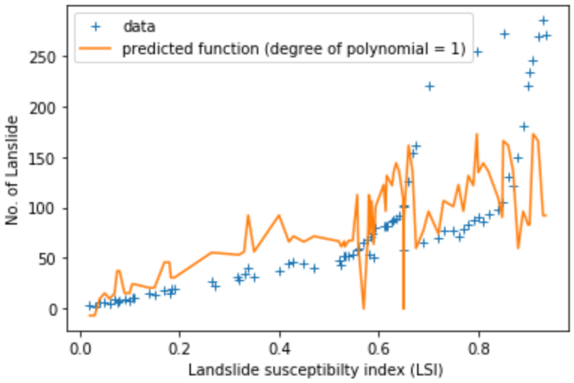


Figure 15: MLE Without Intercept

The below 2 figures are for polynomial with degree 1 and degree 2. As it can be seen from figure, for polynomial with degree 1, the predicted function results into much more haphazard data, where as polynomial with degree 2 performs better than polynomial with degree 2 but still fails to fit with original data.



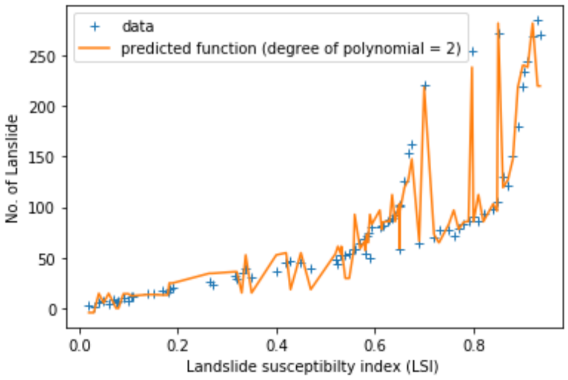
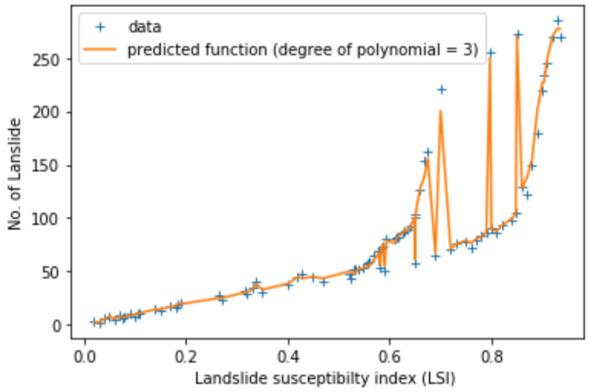


Figure 16: Polynomial Degree 1

Figure 17: Polynomial Degree 2

The *figure 19* and *figure 20* are for polynomials with degree 3 and degree 4. As seen from figures above, the predicted function performs better with polynomial of degree 3 and 4 as the predicted values fit with the original data.



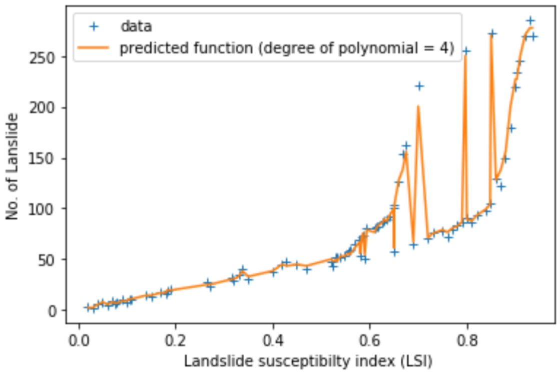


Figure 18: Polynomial Degree 3

Figure 19: Polynomial Degree 4

6 Conclusions

The methods that we used showed the relationship between various triggers of landslides and their features. We were able to balance the dataset by over-sampling using ADASYN and SMOTE. Random Forest Classifier performed slightly better than SVM. SVM had mean absolute error of 1.03, where as RFC had a mean absolute error of 0.86. reduction exploits structure and correlation and allows us to work with a more compact representation of the data without losing much information.

Principal Component Analysis (PCA) is one of the many methods for Dimensionality Reduction. In both the cases, MSE decreases with increase in number of components which signifies decrease in loss of information as number of components increase. The eigen values also decrease with increase in number of components. Variance increases with the increase in number of principal components.

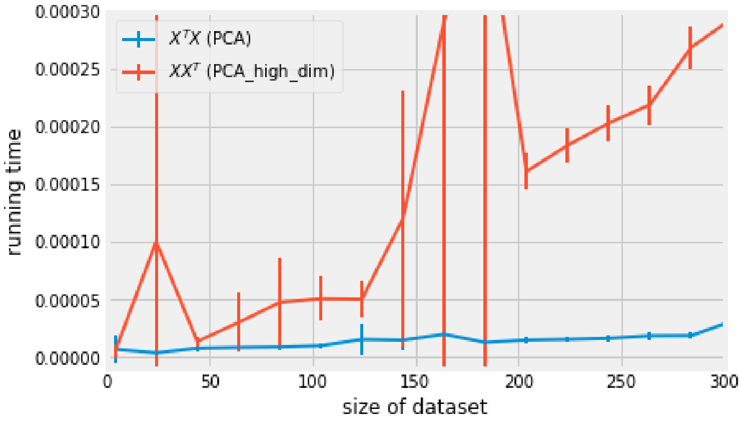


Figure 20: Run Time vs Size of Dataset

We have also compared the running time of both cases. As observed in above figure, run time for PCA of higher dimension data is comparatively larger than run time for PCA with more samples.

A prior knowledge of appropriate causative factors is required to successfully determine the landslide susceptibility. Positive value of Certainty Factor(CF) represents that those features have strong influence on landslide occurrence which increases the Landslide susceptibility Index of that place.

7 Contribution of team members

7.1 Technical contribution of all team members

Tasks	Jainam Chhatbar	Dhairya Dudhatra	Charmil Gandhi	Jinesh Patel
Understanding Base Article	Yes	Yes	Yes	Yes
Analysis	No	Yes	No	Yes
Code and Simulation	Yes	Yes	Yes	Yes

7.2 Non-Technical contribution of all team members

Tasks	Jainam Chhatbar	Dhairya Dudhatra	Charmil Gandhi	Jinesh Patel
Report Writing	Yes	No	Yes	No

References

[1] Binh, P., Biswajeet, P., Dieu, B. , Indra P., "A comparative study of different machine learning methods for landslide susceptibility assessment: A case study of Uttarakhand area (India)". *Environmental Modelling Software*, 2016, 84, https://doi.org/10.1016/j.envsoft.2016.07.005.

[2] Froude, M.J.; Petley, "D.N. Global fatal landslide occurence" form 2004 to 2016 *Nat. Hazards Earth Syst. Sci.*, 2018, 18, 2161-2181

[3] Pham,B.T.; Prakash,I; Singh,S.K.; Shirzadi,A.; Shahabi,H.; Bui,D.T. "Landslide susceptibilty model using Reduces error pruning tree and different ensemble techniques: Hybrid machine learning appraoches". *Catena*, 2019, 175, 203-218

[4] Chen, W.; Xie, X.; Peng, J.; Shahabi, H.; Hong, H.; Bui, D.T.; Duan, Z.; Li, S.; Zhu, A.-X. "Gis-based landslide susceptibility evaluation using a novel hybrid integration approach of bivariate statistical based random forest method". *Catena*, 2018, 164, 135–149.

[5] Tsangaratos,P.;Ilia,I.; Grof,G.; Ho,H.L.;"Landslide susceptibility mapping using a modified decision tree classifier in the Xanthi Perfection". *, Greece. Landslides*, 2016, 13, 305–320

[6] Thai Pham, B.; Prakash, I.; Dou, J.; Singh, S.K.; Trinh, P.T.; Trung Tran, H.; Minh Le, T.; Tran, V.P.; Kim Khoi, D.; Shirzadi, A.; "A novel hybrid approach of landslide susceptibility modeling using rotation forest ensemble and different base classifiers.". *Geocarto Int.*,2019, 1–25.

[7] Chen, W.; Hong, H.; Li, S.; Shahabi, H.; Wang, Y.; Wang, X.; Bin Ahmad, B."Flood susceptibility modelling using novel hybrid approach of reduced-error pruning trees with bagging and random subspace ensembles.". *J. Hydrol.*, 2019, 575, 864–873.

[8] Bui,D.T.;Panahi,M.;Shahabi,H.;Singh,V.P.;Shirzadi,A.;Chapi,K.;Khosravi,K.;Chen,W.;Panahi,S.;Li,S."Novel hybrid evolutionary algorithms for spatial prediction of floods.". *Sci. Rep.*, 2018, 8, 15364.

[9] TienBui,D.;Shahabi,H.;Shirzadi,A.;Chapi,K.;Hoang,N.D.;Pham,B.;Bui,Q.T.;Tran,C.T.;Panahi,M.;Bin Ahamd, B."A novel integrated approach of relevance vector machine optimized by imperialist competitive algorithm for spatial modeling of shallow landslides.". *Remote Sens.*,2018, 10, 1538.

[10] N. Global Landslide Catalog Export. NASA.,(2016, March 7). *https://data.nasa.gov/Earth-Science/Global-Landslide-Catalog-Export/dd9ewu2v*,

[11] Haibo H., Yang B., E. A. Garcia and Shutao L."ADASYN: Adaptive synthetic sampling approach for imbalanced learning,". *IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong*,2008, pp. 1322-1328.

[12] Chawla, N. Bowyer, K., Hall, L., Kegelmeyer, W. "SMOTE: Synthetic Minority Over-sampling Technique.". *JAIR.*,2002, *https://jair.org/index.php/jair/article/view/10302*

[13] Khosravi,K; Shahabi,H; Pham,B.T.; Adamawoski,J.; Shirzadi,A.; Pradhan,B.; Dou,J.; Ly,H.-B.; Grof,G.; Ho,H.L.;"A comparative assessment of flood susceptibility modeling using multi criteria decision-making analysis and machine learning methods". *J. Hydrol.*, 2019, 573, 311-323

[14] Bui,D.T.; Panahi,M.; Shahabi,H.; Singh,V.P.; Shirzadi,A.; Chapi,K.; Khosravi,K.; Chen,W.; Panahi,S.; Li,S."Novel hybrid evolutionary algorithms for spatial prediction of floods.". *Sci. Rep.*, 2018, 8, 15364

[15] Jaafari,A.; Zenner,E.K.; Panahi,M.; Shahabi,H."Hybrid artificial intelligence models based on a neuro-fuzzy system and metaheurestic optimization algorithms for spatial prediction of wildfire probability". *Agric. For. Meteorol.*, 2019, 266, 198-207

[16] Singh,S.K.; Taylor,R.W.; Rahman,M.M.; Pradhan,B."Developing robust arsenic awareness prediction models using machine learning algorithms". *J. Environ. Manag.*, 2018, 211, 125-137

[17] Pham,B.T.; Prakash,I.; Bui,D.T. "Spatial prediction of landslides using a hybrid machine learning approach based on random subspace and classification and regression trees". *Geomorphology*, 2018, 303, 256-270.

[18] Thai Pham,B.; Prakash,I.; Dou,J.; Singh,S.K.; Trinh,P.T.; Trung Tran, H.; Minh Le, T.; Tran,V.P.; Kim Khoi,D.; Shirazadi, A. "A novel approach of landslide susceptibility modeling using rotation forest ensemble and different base classifiers". *Geocarto Int.*, 2019, 1-25.

[19] Pradhan,B, "A comparative study on the predictive ability of the decision tree, support vector machine and neuro-fuzzy models in landslide susceptibility mapping using gis.". *Comput. Geosci.*, 2013, 51, 350-365.

[20] Bui,D.T.; Tuan,T.A.; Klempe,H.; Pradhan,B.; Revhaug,I. "Spatial prediction for shallow landslide hazards: A comparative assessment of the efficacy support vector machines, artificial neural networks, kernel logistic regression, and logistice model tree.". *Landslides*, 2016, 13, 361-378.

[21] Pourghasemi,H.R.;Yansari,Z.T.;Panagos,P.;Pradhan,B."Analysis and evaluation of landslide susceptibility: A review on articles published during 2005–2016 (periods of 2005–2012 and 2013–2016)". *Arab. J. Geosci.*, 2018, 11, 193.

[22] Freund, Y.; Mason, L. "The Alternating Decision Tree Learning Algorithm". *ICML: New Jersey, NY, USA*, 1999, pp. 124-133.