AI-Based Landslide Risk Monitoring System in Northeast India Using Machine Learning and Explainable AI

Om Chetan Mohan[1], Anand P Nair[2], Shraddha Patil[3], Bhumika Rohidas Shelar[4]
[1,2,3,4]*Pillai College of Arts Commerce and Science Empowered Autonomous*

***Abstract*—Hill districts in Northeast India lose roads, fields and homes to landslides almost every monsoon. We wanted to see how far a machine-learning model can go in estimating that risk using only terrain, climate and land-cover data. The dataset was cleaned and explored first. We then trained Logistic Regression, Random Forest, XGBoost and LightGBM in Python on Google Colab and compared them on accuracy, precision, recall, F1-score and ROC-AUC. To open up the XGBoost model, we applied SHAP (SHapley Additive exPlanations). Relief, mapping-unit area, slope, the vegetation index (NDVI) and rainfall sum had the largest influence on the predictions. The predicted probabilities were also drawn as a probability map and a five-class susceptibility map, which show where the risk is highest. The workflow is small and easy to repeat, and it can serve as a base for later work with bigger datasets and spatially independent validation.**

***Index Terms*—Explainable AI, Landslide susceptibility, LightGBM, Machine learning, Northeast India, Random Forest, SHAP, XGBoost.**

I. INTRODUCTION

Every monsoon, slopes across Northeast India give way. A landslide is simply soil, rock or debris coming loose and moving down a slope, and long or intense rain is one of the usual causes. The region has steep hills and heavy rain together, so roads, farms, houses and far-off villages are at risk year after year.

The older way of studying landslides used past records, field trips, geological surveys and expert opinion. That work is valuable. It is also slow, and it is hard to repeat across a wide area. Today satellite images, elevation models and rainfall data are easy to download, which lets us ask a computer to find the link between these factors and landslides that have already occurred [6], [9].

Here we set up a machine-learning workflow for classifying landslide locations in Northeast India and tested four models' side by side. Logistic Regression acts as the plain baseline. Random Forest [3] builds many decision trees and combines their votes. XGBoost [1] and LightGBM [2] also use trees, but they add them in sequence so that each one corrects the errors left by the earlier ones. The question we cared about was a simple one: do these stronger models actually beat the baseline?

We cover data analysis, model building, model comparison and interpretation. A hardware sensor network and a web application are outside the scope. Everything was run in Python on Google Colab, so anyone can repeat the experiments.

II. PROBLEM STATEMENT AND OBJECTIVES

Northeast India combines steep slopes, fragile rock, intense seasonal rain and land use that is changing fast. Landslides damage villages, roads and fields, and the rough terrain means that rescue and repair can take a long time to arrive. There is rarely a single cause. Rain, slope, elevation, soil, vegetation and land use work together, and every place has its own mix. What is needed is one reliable system that brings these factors together for this region and gives a risk level for each location.

Our objectives were:

1) to find the environmental and geographical factors that go with landslides in the region;
2) to gather, combine and clean the data the models need;
3) to see which factors, matter most for landslide risk;
4) to build machine-learning models that sort locations by risk, and to compare them;
5) to score the models with accuracy, precision, recall, F1-score and ROC-AUC;
6) to show the results as risk maps that are easy to read,

7) to state what the work cannot do yet and how it could grow into real-time monitoring.

III. LITERATURE REVIEW

The first landslide studies leaned on field surveys, basic statistics and GIS susceptibility maps [7]. They were useful, but they cope poorly when many factors affect one another in complicated ways. Reviews written later describe a gradual shift to statistical and data-driven models as digital data became easier to get [6]. Since then, Decision Tree, Random Forest, Support Vector Machine and Logistic Regression have all been used to relate terrain and environment to past landslides. A comparison of these methods found that the choice of algorithm can shift the results by a noticeable amount, so it makes sense to test several of them on the same data [9]. Rain is an important trigger, yet it does not explain a landslide on its own. How a slope responds also depends on terrain, soil, vegetation and what people do to the land.

A. Research Gaps

Reading this work, we noticed six gaps. Northeast India differs greatly from one area to the next. Many studies work with very few predictors. Landslide records are uneven. A static map is not the same thing as a system that changes when conditions change. Accuracy alone does not tell users why a prediction was made, and they want to know. Finally, algorithms can disagree on identical data. We try to deal with these gaps in this paper.

IV. METHODOLOGY

A. Dataset and Input Factors

Each row in the dataset is one mapping unit of the study area, and there are about 34 input variables per row. The values are averages of terrain, climate and land factors. A label y marks whether a landslide was recorded in that unit (1) or not (0). Table I lists the main groups of inputs.

TABLE I MAIN INPUT FACTORS USED IN THE MODELS

Feature	Meaning
Rlf_mean	Average terrain relief
Slope_mean	Average slope steepness
VCv_mean,	Vertical and horizontal curvature
HCv_mean	
NDVI_mean	Vegetation greenness index
RnSum_mean	Average rainfall sum
pga_mean	Peak ground acceleration (earthquake shaking)
area	Area of the mapping unit
litho_0 – litho_8	Rock type (lithology) classes
lcover_*	Land-cover classes (forest, cropland, etc.)

B. Data Checking and Preprocessing

Before any modelling we went through the data one check at a time: size, duplicate rows, missing values, data types, class distribution, unique values, constant columns and identifier columns. Missing values were dealt with during preprocessing. Columns that repeated information or had no predictive value, for example fid and ID, were dropped from the inputs. Summary statistics and a correlation analysis gave us a first feel for the numeric variables and how they relate to the target.

Landslide units are far fewer than non-landslide units, so we split the data with stratification, which keeps the same class ratio in every part. There were three parts. The training set fitted the models. The validation set was used to compare models and to pick the probability threshold. The test set stayed untouched until the final evaluation and the mapping.

C. Models

Four classifiers were trained on exactly the same data partitions: Logistic Regression as the baseline, Random Forest, XGBoost and LightGBM. We coded them in Python with scikit-learn [5], XGBoost and LightGBM. Identical stratified splits for every model keep the comparison fair. XGBoost got extra tuning, because it handles nonlinear links between terrain and environmental variables well.

D. Evaluation Metrics

Landslide units are rare, so accuracy by itself can look good even when a model misses most real landslides. We therefore read several measures together. Let TP, TN, FP and FN be true positives, true negatives, false positives and false negatives, and let P and R be precision and recall. Then:

$$\text{Accuracy} = (TP+TN)/(TP+TN+FP+FN) \quad (1)$$

$$\text{Precision} = TP/(TP+FP) \quad (2)$$

$$\text{Recall} = TP/(TP+FN) \quad (3)$$

F1 = 2PR/(P+R) (4)

ROC-AUC tells us how well a model separates landslide units from the rest over all thresholds [8]. We also looked at confusion matrices and ROC curves. For the final XGBoost model we did not keep the usual 0.5 cut-off. The threshold was picked on the validation set instead, which gave a better balance of precision and recall for this imbalanced data.

E. Explainability and Mapping

To see why the final XGBoost model predicts what it does, we computed SHAP (SHapley Additive exPlanations) values [4]. They give each feature's contribution to a prediction and show which factors push the landslide probability up or down. After evaluation, the model was run over all the geospatial mapping units to get a probability for each one. We plotted these on a map and sorted them into five susceptibility classes: Very Low, Low, Moderate, High and Very High. The map shows how the predicted risk is spread over the study area.

V. RESULTS AND DISCUSSION

A. Relationships Between Variables

The correlation heatmap is in Fig. 1. Relief (Rlf_mean) and slope (Slope_mean) move together very strongly, which makes sense, since steep ground usually has bigger height differences.

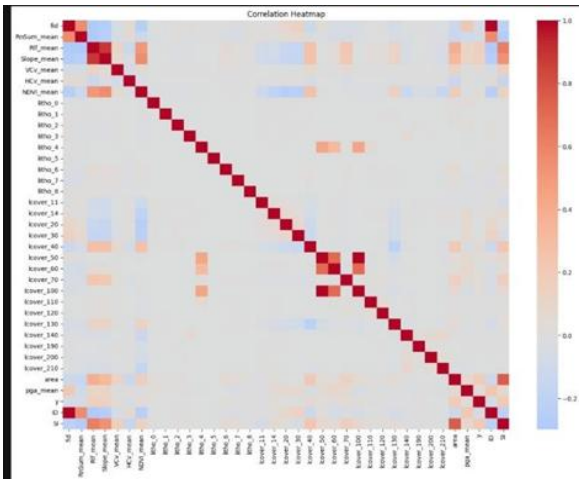

Fig. 1. Correlation heatmap of the input variables.

Three land-cover columns (lcover_50, lcover_60 and lcover_100) also move together. Most other pairs show weak correlation, so the features mostly carry different information. The identifier columns fid and ID are perfectly correlated, which supports dropping them.

B. Important Factors

Fig. 2 ranks the features by mean absolute SHAP value. Relief (Rlf_mean) is far ahead of the rest. Area, slope, NDVI and rainfall sum (RnSum_mean) follow. Peak ground acceleration and the two curvature values matter less but are still visible, and single land-cover classes matter least.

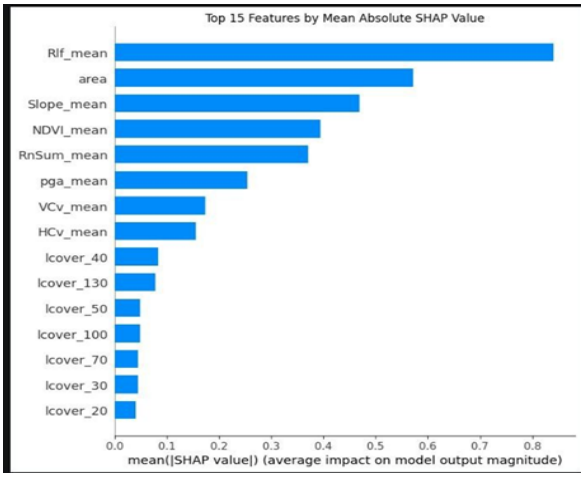

Fig. 2. Top 15 features by mean absolute SHAP value.

Fig. 3 adds the direction of each effect. High relief, high slope and large area push the predictions towards a landslide. High NDVI, that is dense vegetation, pushes the other way,

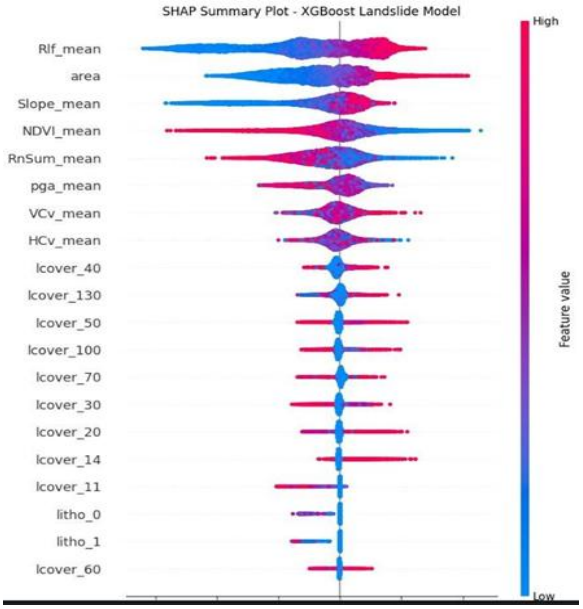

Fig. 3. SHAP summary plot for the XGBoost landslide model.

which fits the idea that plants help hold the soil together. We did not expect the next result: high rainfall sum and high peak ground acceleration both sit on the negative side. This may come from how the data is spread across the region rather than from the physics. Both factors should be rechecked with better rainfall data before we draw any firm conclusion.

C. Maps of Landslide Risk

Fig. 4 shows where landslides actually happened in the test set, and Fig. 5 shows the probabilities the model gave for the same area. The two maps look alike. The biggest clusters of real landslides sit in the eastern and central hill belt, and so do the highest predicted probabilities, which go above 0.8. The rest of the area mostly has low probabilities.

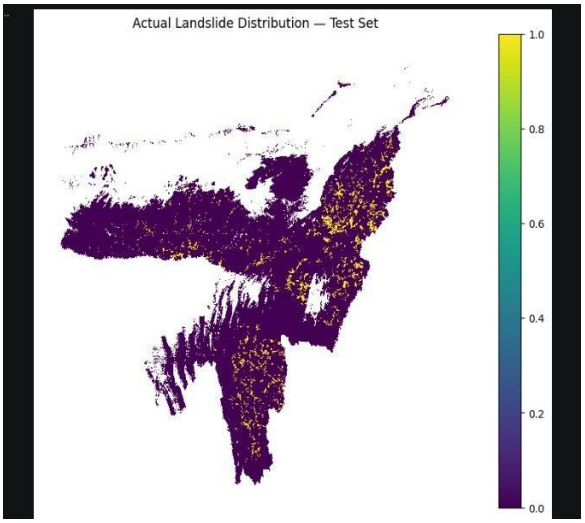

Fig. 4. Actual landslide distribution in the test set.

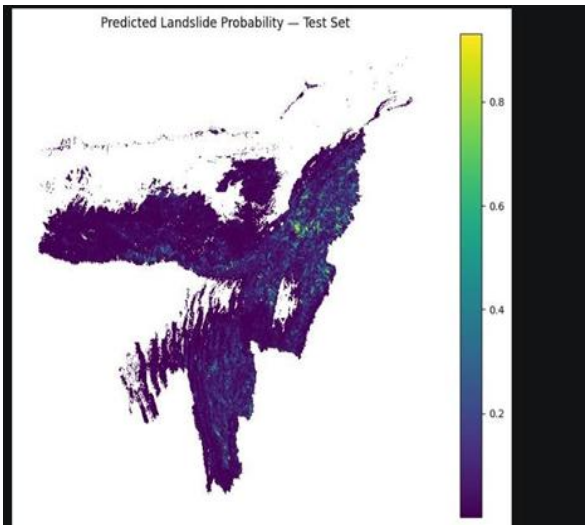

Fig. 5. Predicted landslide probability in the test set.

In Fig. 6 the probabilities are grouped into five susceptibility classes. Most of the region is Very Low. High and Very High show up only as small, scattered patches in the eastern hill belt. That is helpful for planning, since attention can go to those small risky zones first.

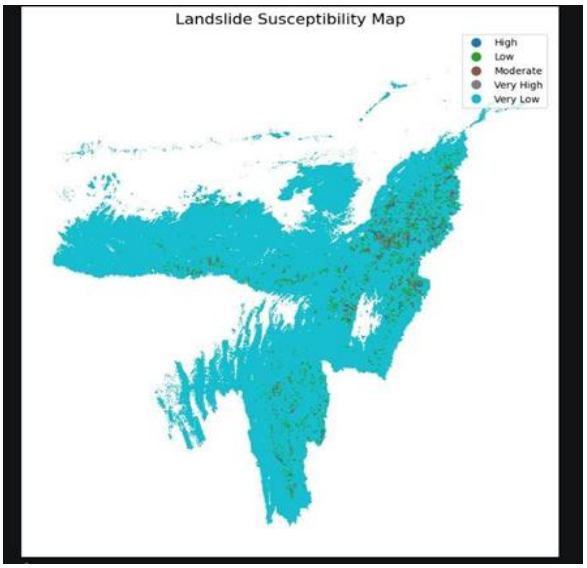

Fig. 6. Landslide susceptibility map with five risk classes.

D. Choice of Probability Threshold

Fig. 7 plots precision, recall and F1-score on the validation set for cut-offs from 0.01 up to about 0.5. At very low cut-offs recall is high, and it drops steadily as the cut-off grows. Precision creeps up. F1 stays low and is nearly flat, around 0.2 from roughly 0.08 to 0.2, with its peak near 0.11. We used 0.11. At that point precision is about 0.16 and recall about 0.30.

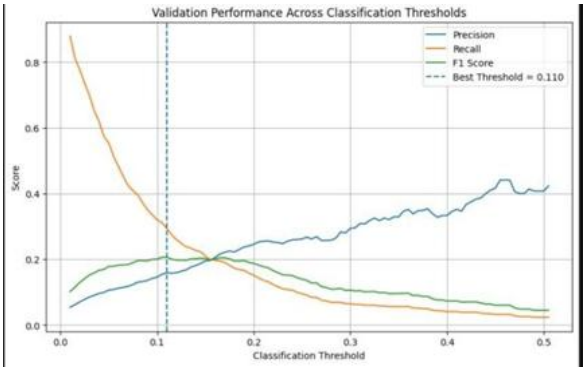

Fig. 7. Validation precision, recall and F1-score across classification thresholds.

Fig. 8 maps the test-set predictions at that threshold. Far more units are marked as landslides than in Fig. 4,

which matches the low precision in Fig. 7. The main hill belts are found, but the model gives many false alarms.

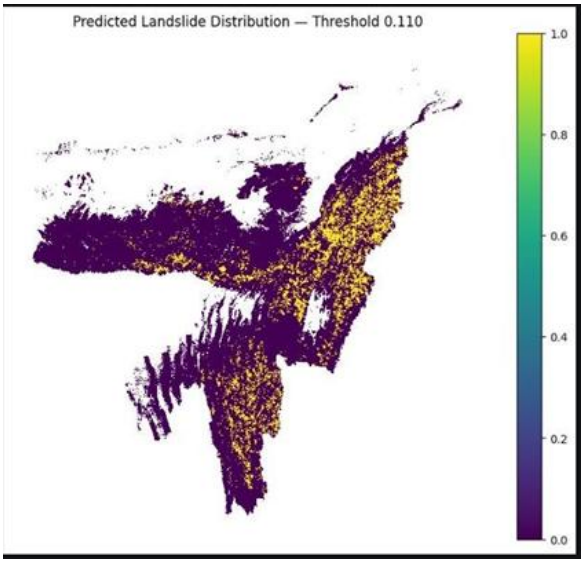

Fig. 8. Predicted landslide distribution in the test set at threshold 0.110.

E. Model Comparison

The models learn in different ways, and Table II sums this up. Logistic Regression fits a straight-line boundary. It is easy to read but cannot follow curved relationships. Random Forest, XGBoost and LightGBM are built from many decision trees and can pick up nonlinear patterns involving terrain, vegetation and rainfall. We picked XGBoost for the explanation and mapping work because it performs well on table-style data and works fully with SHAP.

TABLE II COMPARISON OF THE FOUR MODELS USED

Model	Type	Main strength
Logistic Regression	Linear	Simple baseline, easy to read
Random Forest	Bagged trees	Stable, handles nonlinear data
XGBoost	Boosted trees	High accuracy, works with SHAP
LightGBM	Boosted trees	Fast on large data

VI. LIMITATIONS

The work has limits. The dataset is small, and the quality of old landslide records affects the results. The test set comes from the same region as the training data, so we do not know how the model would do in a completely new area. The data is static, so the system cannot respond to daily rainfall, and the models were not tuned much. At the chosen threshold the precision is low: many areas are flagged although no landslide was recorded there. The odd direction of the rainfall and ground-shaking effects in the SHAP plot also needs a closer look.

VII. CONCLUSION AND FUTURE WORK

We built four machine-learning models for landslide classification in Northeast India, compared them, and used SHAP to explain the XGBoost model. Relief, mapping-unit area, slope, vegetation and rainfall sum came out as the main factors. The probability and susceptibility maps pick out small high-risk zones from large low-risk areas, which can help early planning and disaster preparedness.

Next steps include a larger dataset, more environmental variables, hyperparameter tuning and validation on areas the model has not seen. Feeding in live rainfall and satellite data would let the map refresh continuously and turn the static susceptibility map into a live risk monitor.

ACKNOWLEDGMENT

The authors are grateful to Pillai College of Arts, Commerce & Science (Empowered Autonomous), New Panvel, for the academic support during this work.

REFERENCES

[1] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 785–794.

[2] G. Ke *et al*., "LightGBM: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 3146–3154.

[3] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[4] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in*

Neural Information Processing Systems, vol. 30, 2017, pp. 4765–4774.

[5] F. Pedregosa *et al*., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, Oct. 2011.

[6] P. Reichenbach, M. Rossi, B. D. Malamud, M. Mihir, and F. Guzzetti, "A review of statistically-based landslide susceptibility models," *Earth-Science Reviews*, vol. 180, pp. 60–91, May 2018.

[7] F. Guzzetti, A. Carrara, M. Cardinali, and P. Reichenbach, "Landslide hazard evaluation: A review of current techniques and their application in a multi-scale study, Central Italy," *Geomorphology*, vol. 31, nos. 1–4, pp. 181–216, Dec. 1999.

[8] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, Jun. 2006.

[9] A. Merghadi *et al*., "Machine learning methods for landslide susceptibility studies: A comparative overview of algorithm performance," *Earth-Science Reviews*, vol. 207, Art. no. 103225, Aug. 2020.