Nat. Hazards Earth Syst. Sci., 24, 1913–1928, 2024
https://doi.org/10.5194/nhess-24-1913-2024

Natural Hazards
and Earth System
Sciences
Open Access
EGU

Addressing class imbalance in soil movement predictions

Praveen Kumar[1], **Priyanka Priyanka**[1], **Kala Venkata Uday**[2], **and Varun Dutt**[1]

[1]Applied Cognitive Science Lab, Indian Institute of Technology Mandi, Himachal Pradesh, 175075, India
[2]Geotechnical Lab, Indian Institute of Technology Mandi, Himachal Pradesh, 175075, India

Correspondence: Praveen Kumar (dr.praveenkumar.ml@gmail.com)

Received: 27 June 2023 – Discussion started: 10 August 2023
Revised: 14 April 2024 – Accepted: 25 April 2024 – Published: 6 June 2024

Abstract. Landslides threaten human life and infrastructure, resulting in fatalities and economic losses. Monitoring stations provide valuable data for predicting soil movement, which is crucial in mitigating this threat. Accurately predicting soil movement from monitoring data is challenging due to its complexity and inherent class imbalance. This study proposes developing machine learning (ML) models with oversampling techniques to address the class imbalance issue and develop a robust soil movement prediction system. The dataset, comprising 2 years (2019–2021) of monitoring data from a landslide in Uttarakhand, has a 70 : 30 ratio of training and testing data. To tackle the class imbalance problem, various oversampling techniques, including the synthetic minority oversampling technique (SMOTE), K-means SMOTE, borderline-SMOTE, and adaptive SMOTE (ADASYN), were applied to the training dataset. Several ML models, namely random forest (RF), extreme gradient boosting (XGBoost), light gradient boosting machine (LightGBM), adaptive boosting (AdaBoost), category boosting (CatBoost), long short-term memory (LSTM), multilayer perceptron (MLP), and a dynamic ensemble, were trained and compared for soil movement prediction. A 5-fold cross-validation method was applied to optimize the ML models on the training data, and the models were tested on the testing set. Among these ML models, the dynamic ensemble model with K-means SMOTE performed the best in testing, with an accuracy, precision, and recall rate of 0.995, 0.995, and 0.995, respectively, and an F1 score of 0.995. Additionally, models without oversampling exhibited poor performance in training and testing, highlighting the importance of incorporating oversampling techniques to enhance predictive capabilities.

1 Introduction

Landslides pose a significant threat to infrastructure, resulting in numerous fatalities and substantial economic losses each year (Parkash, 2011). These destructive events occur globally, particularly in hilly and mountainous regions, driven by gravity and characterized by the movement of large rocks, debris, and soil (Crosta, 1998). Factors such as heavy rainfall, earthquakes, and the impacts of climate change contribute to the occurrence and severity of landslides (Crosta, 1998).

Monitoring, predicting, and warning people about slope movements in landslide-prone areas are crucial for mitigating landslide risks. Advanced technologies like the Global Positioning System (GPS), light detection and ranging (lidar), the geographic information system (GIS), and remote sensing have proven effective for assessing and analyzing slope failure hazards (Ray et al., 2020). However, their high cost and the need for specialized expertise limit their accessibility, especially in developing countries where cost-effective internet of things (IoT) technologies are necessary (Pathania et al., 2020).

Machine learning (ML) models have been extensively studied for predicting soil movement in landslide-prone areas (Kumar et al., 2021a, b, 2023). This prediction problem could be divided into classification and regression tasks. The classification task aims to predict the degree of soil movement using various ML models. On the other hand, the regression task involves estimating the acceleration or displacement of soil under observation.

One common challenge in landslide prediction is a class imbalance, where certain classes have significantly more data samples than others. This imbalance can adversely affect the performance of ML models. To address class imbalance

Published by Copernicus Publications on behalf of the European Geosciences Union.

issues, techniques such as the synthetic minority oversampling technique (SMOTE), K-means SMOTE, borderline-SMOTE, and the adaptive synthetic minority oversampling technique (ADASYN) are employed to balance the dataset (Chawla et al., 2002; Douzas et al., 2018; Han et al., 2005; He et al., 2008).

Several researchers have dedicated their efforts to addressing class imbalance problems in ML. Notably, Chawla et al. (2002) introduced SMOTE, Douzas et al. (2018) devised K-means SMOTE, Han et al. (2005) proposed borderline-SMOTE, and He et al. (2008) introduced the adaptive synthetic minority oversampling technique (ADASYN). These techniques were developed to generate synthetic data and balance imbalanced datasets.

The field of soil movement prediction requires further investigation, particularly considering the complexities associated with a class imbalance in the datasets. Despite extensive research on ML models' predictive abilities for soil movement in landslides, there still needs to be more understanding regarding how class imbalance affects the models' performance and accuracy. This study aims to bridge this knowledge gap by examining different approaches to tackle class imbalance and exploring diverse ML models to improve the prediction of soil movement. Various multivariate classification models, including random forest (RF); adaptive boosting (AdaBoost); extreme gradient boosting (XGBoost); light gradient boosting machine (LightGBM); category boosting (CatBoost); long short-term memory (LSTM); multilayer perceptron (MLP); and an ensemble of RF, AdaBoost, XGBoost, LightGBM, and CatBoost, are developed to predict soil movement when coupled with class imbalance techniques (Kumar et al., 2019; Semwal et al., 2022; Wu et al., 2020; Pathania et al., 2021; Zhang et al., 2022; Sahin, 2022; Kumar et al., 2020; Kumar et al., 2023).

This study delves into the field of soil movement prediction, making significant advancements by developing specialized ML models and techniques tailored to this domain. A notable aspect that has received limited attention in the existing literature is the challenge of class imbalance in landslide datasets. While previous research has primarily focused on ML models for soil movement prediction, this work addresses the issue of imbalanced data head-on. Multiple variants of SMOTE and other balancing strategies are introduced and implemented to enhance the efficacy and accuracy of the ML models.

Additionally, this research explores using cost-effective internet of things (IoT) technologies in developing regions to improve the investigation and assessment of landslide hazards. The dataset used in this study spans 2 years, from June 2019 to June 2021, and was collected by an inexpensive IoT monitoring station in Uttarakhand, India. This real-world dataset captures the distinctive characteristics and patterns of soil movements prevalent in the landslide-prone area. By employing a comprehensive methodology, this work advances soil movement prediction and effectively addresses the challenge of class imbalance. It commences with a thorough overview of the collected data, emphasizing the measured weather and soil-related factors. Various SMOTE variants and other balancing techniques are employed to rectify the class imbalance, resulting in the generation of synthetic samples and ensuring a balanced representation of soil movement classes. The intricate correlations and patterns in the soil movement data are captured using a variety of ML models, including RF; AdaBoost; XGBoost; LightGBM; CatBoost; MLP; LSTM; and a dynamic ensemble of RF, AdaBoost, XGBoost, and CatBoost. Overall, this study's findings show potential for accurately reducing landslide risks, increasing the accuracy of landslide prediction, and encouraging the use of cost-effective IoT technologies in landslide-prone locations.

2 Background

Several techniques have been proposed to address the challenge of learning from imbalanced datasets, where the classification categories are not evenly represented. For example, Chawla et al. (2002) proposed SMOTE, which involves generating synthetic minority-class examples to balance the dataset. SMOTE has been shown to improve model performance compared to only undersampling the majority class. Douzas et al. (2018) introduced K-means SMOTE, a method that combines SMOTE with K-means clustering to effectively overcome imbalances between and within classes without generating unnecessary noise. Additionally, Han et al. (2005) developed a borderline SMOTE method that focuses on oversampling only the minority examples near the class boundary. Experimental results indicate that borderline-SMOTE1 and borderline-SMOTE2 outperform SMOTE and random oversampling methods in terms of the true positive rate and F value. Lastly, He et al. (2008) developed ADASYN, which addresses class imbalance by generating more synthetic data for minority-class examples that are harder to learn. ADASYN reduces bias and adaptively shifts the classification decision boundary toward challenging examples. Simulation analyses have demonstrated the effectiveness of ADASYN across various evaluation metrics. These techniques offer valuable approaches to mitigate the impact of imbalanced data in classification tasks. These class imbalance techniques have limited the exploration and application of landslide datasets. Existing studies primarily focus on the general imbalanced dataset scenario but need to consider the unique characteristics and challenges associated with landslide datasets. Therefore, research is required for systematic studies that compare the performance and effectiveness of techniques such as SMOTE, K-means SMOTE, borderline-SMOTE, and ADASYN in the specific context of soil movement prediction across various evaluation metrics. By bridging this literature gap, we can enhance the accuracy and reliability of models for predicting soil movement in landslide-prone areas and contribute to improved landslide risk mitigation strategies.

Several researchers developed various ML models to predict soil movement and prediction problems in other fields (Kumar et al., 2019; Semwal et al., 2022; Wu et al., 2020; Pathania et al., 2021; Zhang et al., 2022; Sahin, 2022; Kumar et al., 2020). For example, Kumar et al. (2019) developed an ensemble of ML models (RF, bagging, stacking, and voting) for predicting soil movement at the Tangni landslide in Uttarakhand, India. These models were compared with sequential minimal optimization (SMO) and autoregression (AR). The results indicate that the ensemble models outperformed the SMO and AR models in predicting soil movement. Furthermore, Semwal et al. (2022) developed sequential minimal optimization regression (SMOreg), instance-based learning (IBk), RF, linear regression (LR), MLP, and ensemble ML models to predict root tensile strength for different vegetation species. The results show that the MLP performed better than the other models, providing more accurate predictions of root tensile strength. Next, Wu et al. (2020) developed a decision tree (DT) with AdaBoost and bagging ensembles for mapping the susceptibility of landslides in Long County, Shaanxi Province, China. Researchers developed the technique with an ensemble of an alternating decision tree (ADTree) with bagging and AdaBoost to map landslide susceptibility. The results revealed that the ensemble of ADTree and AdaBoost performed better than the individual ADTree model and ensemble of ADTree and bagging. Similarly, Pathania et al. (2021) developed a novel ensemble gradient boosting model, called SVM–XGBoost (support vector machine), for soil movement warning at the Gharpa landslide, Mandi, India. They compared the performance of SVM–XGBoost with other models such as individual SVMs, DTs, RFs, XGBoost instances, naïve Bayes (NB) classifiers, and different variants of XGBoost. The results showed that the SVM–XGBoost model performed better than other models in soil movement prediction. In their research, Kumar et al. (2021b) directed their attention toward predicting soil movement, specifically at the Tangni landslide site in India. To enhance the accuracy of their predictions, they explored various variants of long short-term memory (LSTM) models. They introduced a novel ensemble approach called BS-LSTM, which combined bidirectional and stacked LSTM models. The findings of their study indicated that the BS-LSTM model outperformed the other LSTM variants in accurately predicting soil movement. Similarly, Zhang et al. (2022) conducted a study to assess the susceptibility of landslides using gradient boosting ML techniques coupled with class-balancing methods. Their investigation specifically focused on the aftermath of the 2018 Hokkaido earthquake and employed diverse datasets and methodologies to predict the susceptibility of specific areas prone to landslides. Compared to well-established models such as XGBoost and LightGBM, the proposed model showcased superior performance in accurately assessing landslide susceptibility. Furthermore, Sahin (2022) developed multiple ML models, including XGBoost, CatBoost, gradient boosting machine (GBM), and LightGBM, to model the susceptibility of landslides. By leveraging a comprehensive landslide inventory map and relevant conditioning factors stored in a geodatabase, the study employed feature selection techniques and compared the predictive capabilities of ensemble methods with the widely used RF model. The results highlighted that CatBoost exhibited the highest predictive capability, followed by XGBoost, LightGBM, and GBM, while RF demonstrated comparatively lower predictive capability. The study used a geodatabase with a landslide inventory map and conditioning factors. Feature selection techniques were applied, and the performance of XGBoost, CatBoost, GBM, and LightGBM was compared to RF. The results revealed that CatBoost had the highest prediction capability, followed by XGBoost, LightGBM, and GBM. The literature gap in the context of soil movement prediction is the limited exploration and evaluation of ML models in combination with synthetic data generated by SMOTE techniques. While various ML models, such as ensemble models (e.g., RF), neural network models (MLP and LSTM), and gradient boosting ML models (e.g., AdaBoost, XGBoost, LightGBM, CatBoost), have been developed and applied for soil movement prediction, their utilization in conjunction with synthetic data generated by SMOTE techniques has received less attention in the literature. Incorporating SMOTE-generated synthetic data into the training process of these models can address the issue of class imbalance in landslide datasets and improve their performance in predicting soil movement. Therefore, further research is needed to investigate the effectiveness of these ML models when combined with SMOTE techniques in the context of soil movement prediction, thereby filling the existing literature gap.

The models of RF; AdaBoost; XGBoost; LightGBM; CatBoost; MLP; LSTM; and an ensemble of RF, AdaBoost, XGBoost, LightGBM, and CatBoost were chosen to predict soil movement based on their proven effectiveness in previous research. RF is excellent at capturing complex relationships and has outperformed non-ensemble models in predicting debris flow and landslide susceptibility. AdaBoost has successfully predicted soil movement alerts ahead of time. At the same time, XGBoost and LightGBM have demonstrated their ability to achieve balanced and precise predictions, especially in earthquake-induced landslide susceptibility assessments. Among gradient boosting models, CatBoost stands out for its superior prediction capability, making it a well-suited option for modeling landslide susceptibility. On the other hand, when it comes to predicting root tensile strength, MLP has demonstrated higher accuracy in its predictions. Additionally, LSTM, a robust recurrent neural network architecture, is particularly effective in capturing temporal dependencies and long-term patterns in sequential data. Collectively, these models offer a diverse set of capabilities that prove valuable in the prediction of soil movement.

3 Data collection and description

The dataset for predicting soil movement was collected from an actual landslide site in Uttarakhand, India. The monitored landslides are characterized as shallow landslides with debris flow, occurring at elevations ranging from 1450 to 1920 m. The slopes in the landslide zones in the upper parts are made up of weathered limestone and dolomitic limestone, whereas the lower slopes exhibit black carbonaceous slate. The slates are highly weathered and leached, adorned with white and yellow encrustation. These are covered with a thin veneer of debris, mainly consisting of pebble- and cobble-sized limestone, sandstone, and slate embedded in a sand–silt–clay matrix. Additional context includes an annual rainfall of 4190 mm in the area, as reported by Gupta et al. (2016). Spanning a duration of 2 years, from June 2019 to June 2021, this dataset holds valuable insights into the behavior of soil in response to various environmental factors. To gather these data, a cost-effective landslide monitoring station (LMS) was carefully deployed at the landslide. Equipped with a range of sensors, the LMS diligently recorded critical weather and soil-related parameters. Weather-wise, it diligently captured temperature readings in degrees Celsius, humidity levels as a percentage, rainfall measurements in inches per hour (in. h^{-1}; equivalent to 25.4 $\mathrm{mm\,h}^{-1}$), atmospheric pressure in millibars (mb), and even sunlight intensity in lux (lx). These meticulous recordings shed light on the prevailing weather conditions experienced at the precise location of the landslide. The LMS relied on an accelerometer sensor to monitor the soil conditions with utmost precision. An advanced sensor was utilized to measure the acceleration of the soil in three directions: Ax, Ay, and Az (in $\mathrm{m\,s}^{-2}$). This provided valuable insights into the soil's movement and stability. Additionally, a gyroscope sensor was employed to capture the angular rotation of the soil along the Wx, Wy, and Wz axes (in degrees per second). This sensor enhanced the understanding of the soil's behavior by accurately detecting its angular movements. Furthermore, the LMS was equipped with a capacitive soil moisture sensor, enabling it to measure the volumetric moisture content of the soil in percentage. The LMS transmitted all these 12 attributes, including weather parameters, soil g force, angular rotation, and soil moisture content, to the cloud every 10 min. The dataset obtained from the LMS consisted of approximately 39 000 data points, covering a wide range of environmental and soil-related attributes. Table 1 showcases the statistics for the recorded soil movement prediction parameters. For each attribute, the table provides the mean value, representing the average measurement, along with the standard deviation (SD), indicating the variability in the data. The minimum and maximum values highlight the range of measurements observed, offering insights into the extreme values and overall data distribution.

4 Methodology

4.1 Data pre-processing

The sensors installed at the landslide locations experienced malfunctions, resulting in multiple missing values within the collected data. To address this issue, we employed a method to fill these gaps by replacing the missing values with the average values recorded at the corresponding timestamps during the previous week. By calculating the average values for parameters such as light intensity, humidity, temperature, and pressure from the same time periods in the preceding week, we obtained estimates to replace the skewed or missing data points.

4.2 Class labeling

The dataset contained values for acceleration and angular rotation in three directions: x, y, and z. The changes in acceleration and angular rotation were calculated by subtracting the current values from the past values, allowing for the assessment of movement. Four categories were defined to classify the movement data: no movement, low movement, moderate movement, and high movement. These categories were determined based on standard deviation thresholds derived from the acceleration and angular rotation values. Specifically, values from the mean were categorized as follows: within ±1 standard deviation as no movement, within ±2 standard deviations as low movement, within ±3 standard deviations as moderate movement, and exceeding ±3 standard deviations as high movement. This classification approach considered the variability in acceleration and angular rotation changes to determine the intensity of movement.

During the analysis, each timestamp was assigned to a movement class based on the class associated with the highest standard deviation observed in any acceleration or angular rotation element. If an individual element had the highest standard deviation at a specific timestamp, that timestamp was assigned to the corresponding movement class with the maximum standard deviation.

Table 2 presents the distribution of movement intensity within the dataset, which consisted of 38 900 data points. The table shows the percentage distribution of movement categories: high, moderate, low, and no movement. The majority of the dataset (97.8 %) falls under the no-movement category, indicating a lack of significant movement. On the other hand, the high-movement category represents only a small fraction (1.1 %) of the dataset. Additionally, the moderate-movement category comprises 0.7 % of the samples, while the low-movement category accounts for 0.4 % of the dataset. This distribution highlights the class imbalance issue present in the dataset, which needs to be taken into account when developing a classification model for predicting soil movement.

Table 1. Summary statistics of recorded parameters for soil movement prediction dataset.

Parameter	Mean	SD	Min	Max
Temperature (°C)	16.18	10.48	0.00	39.00
Humidity (%)	66.69	35.46	0.00	99.00
Rain (in. h^{-1})	0.00	5.60	0.00	15.00
Pressure (mb)	1040.96	27.96	921.61	1065.41
Light (lx)	5025.35	10 154.75	0.00	54 612.00
Ax ($\mathrm{m\,s}^{-2}$)	0.02	1.23	−28.02	40.25
Ay ($\mathrm{m\,s}^{-2}$)	0.00	1.37	−100.08	100.08
Az ($\mathrm{m\,s}^{-2}$)	0.00	2.28	−149.61	315.61
Wx ($^\circ\,\mathrm{s}^{-1}$)	0.00	15.86	−1994.51	1997.24
Wy ($^\circ\,\mathrm{s}^{-1}$)	0.00	15.85	−1998.05	1998.73
Wz ($^\circ\,\mathrm{s}^{-1}$)	0.00	6.95	−932.00	932.00
Moisture (%)	80.00	20.30	40.00	100.00

Table 2. Class distribution of soil movement data points.

Movement class	Number of data points	Percentage
High movement	423	1.1 %
Moderate movement	146	0.7 %
Low movement	268	0.4 %
No movement	38 063	97.8 %

4.3 Sliding-window packets

The technique of sliding-window packets involves dividing a given dataset into fixed-length subsequences or packets and their corresponding labels. To achieve this, a sequence length parameter is used to determine the length of each subsequence. The sliding-window approach is then employed, where a window starts at the beginning of the dataset and moves through the data with a step size of 1. A subsequence of the specified length is extracted from the dataset at each window position. The label for prediction is taken from the next position after the window.

The technique of sliding-window packets aims to predict future values or events based on preceding subsequences. For instance, if the sequence length is set to 5, the sliding window will select five consecutive values from the dataset as a subsequence at each step. The label for prediction will be the value at the sixth position. This process continues until the end of the dataset is reached, resulting in multiple subsequences and their respective labels. Once the packets are created, they are flattened to form a single feature vector. For instance, if the sequence length is 5 and the dataset has 12 features, each packet will contain 60 elements (5 × 12). This transformation allows for the packets to be treated as individual samples with multiple features suitable for ML models. The primary purpose of creating these packets is to address prediction tasks involving sequences where the input data's order and dependencies are crucial. The model can effectively capture and learn patterns and relationships within the sequential data by utilizing the sliding-window packets. The flattened packets generated using the sliding-window technique are inputs in oversampling techniques.

4.4 Oversampling

In our analysis, we encountered a significant class imbalance issue in the labeled data. The no-movement class, which represents the majority of the data, had a large number of data points. All other classes, including high movement, moderate movement, and low movement, represent minority classes, each constituting only 1 %, 0.7 %, and 0.4 % of the total data, respectively. This class imbalance posed a challenge for building an effective classification model, as the skewed data distribution made it difficult to classify the minority class accurately.

To overcome the class imbalance challenge, we implemented several oversampling techniques, with a particular focus on SMOTE and its extensions (Chawla et al., 2002; Douzas et al., 2018; Han et al., 2005; He et al., 2008). SMOTE, which stands for the synthetic minority oversampling technique, addresses the imbalance by generating synthetic data points for the minority class (Chawla et al., 2002). By utilizing the characteristics of existing samples from the minority classes, we created new data points, thereby increasing the representation of the high-movement, moderate-movement, and low-movement classes. In addition to the standard SMOTE, we also explored other variations such as K-means SMOTE (Douzas et al., 2018) and borderline-SMOTE (Han et al., 2005) to further enhance the balance of class distribution.

Furthermore, we utilized ADASYN, an extension of SMOTE that explicitly addresses the classification boundary of the minority class (He et al., 2008). ADASYN assigns higher weights to the minority examples that are more challenging to classify, leading to the generation of additional artificial data points for these instances. By incorporating ADASYN into our oversampling strategy, we enhanced the balance of the class distribution further and improved the classification accuracy for all classes.

Figure 1 illustrates the application of the K-means SMOTE technique for addressing the class imbalance. Figure 1 depicts a scatterplot where the red crosses represent the minority-class samples, while the black dots represent the majority-class samples. The green crosses indicate the newly generated synthetic samples by the K-means SMOTE algorithm. The dashed line represents the decision boundary separating the two classes. K-means SMOTE operates by following two simple steps iteratively (Douzas et al., 2018). Firstly, it assigns each observation to the nearest cluster centroid among the K available. Secondly, it updates the position of the centroids so that they are positioned at the center between the assigned observations. The imbalance ratio (IR) shown in Fig. 1 helps K-means SMOTE determine the appropriate amount of oversampling for the minority class, ensuring a balanced representation of the classes in synthetic samples. The parameter K in all SMOTE techniques was varied from 2 to 5 in this experiment to observe how different numbers for nearest neighbors impact the diversity and quality of synthetic samples created, thereby affecting the performance of the model on imbalanced data.

4.5 Machine learning models

Various models were employed to classify the soil movement. The specific models will be discussed in the following subsection. To evaluate the accuracy of these models, the dataset was divided into two groups: training data (70 %) and testing data (30 %). Random sampling was used to select 70 % of the data points for training the classification models mentioned below, while the remaining 30 % of the dataset was reserved for model evaluation.

4.5.1 AdaBoost

AdaBoost enhances ML model performance by combining results from multiple weak learners, techniques slightly better than random guessing (Wu et al., 2020). In the AdaBoost model, the number of trees sets the maximum weak models, impacting performance and overfitting. The learning rate influences each model's contribution, with a higher rate giving more weight. The maximum depth parameter prevents weak models from becoming too complex. Table 3 details the AdaBoost model's parameter range.

4.5.2 XGBoost

XGBoost, a gradient boosting ensemble ML model with decision trees (Chen and Guestrin, 2016), excels in structured data handling. The number of trees in XGBoost determines boosting rounds, impacting performance with a computational complexity trade-off. The learning rate influences convergence speed and generalization ability, and the maximum depth parameter prevents overfitting for enhanced interpretability. See Table 3 for the XGBoost model's parameter range.

4.5.3 LightGBM

LightGBM, a gradient boosting framework for tasks like ranking and classification (Ke et al., 2017), stands out with its leaf-wise approach, reducing loss, improving accuracy, and ensuring efficient learning. The number of trees in the model influences boosting rounds for potential performance enhancement. The learning rate parameter balances convergence speed and accuracy, while the maximum depth parameter controls complexity and prevents overfitting. See Table 3 for the LightGBM model's parameter range.

4.5.4 CatBoost

CatBoost, short for category boosting, is an ML model developed by Yandex and released as an open-source tool (Prokhorenkova et al., 2018). In the CatBoost model, the choice of the loss function significantly impacts performance. Loss functions like log, entropy, or hinge are tailored for specific classification problems, influencing results. Table 3 outlines the range of parameters for the CatBoost model for fine tuning and optimizing CatBoost's performance on a given dataset.

4.5.5 Random forest

RF, an ensemble learning method combining predictions from multiple decision trees (Breiman, 2001), constructs regression or classification models. Known for handling relationships and non-linearities without requiring variable independence assumptions, RF excels in various industries, including landslide prediction and site recognition. Optimizing RF performance involves adjusting parameters like the number of trees (DTs), splitting criteria (Gini or entropy), and maximum tree depth, controlling robustness, accuracy, and complexity. Table 3 details parameter ranges for the RF model.

4.5.6 Multilayer perceptron

MLP, a neural network architecture introduced by Rosenblatt et al. (1961), features interconnected layers: input, hidden, and output. Neurons calculate weighted sums, passing through activation functions to capture intricate relationships. Dropout layers prevent overfitting by deactivating neurons randomly during training, enhancing generalization. Versatile for classification, the MLP's look-back period influences temporal dependency capture, while the number of layers and nodes per layer governs complexity. Table 3 outlines parameter ranges for the MLP model.

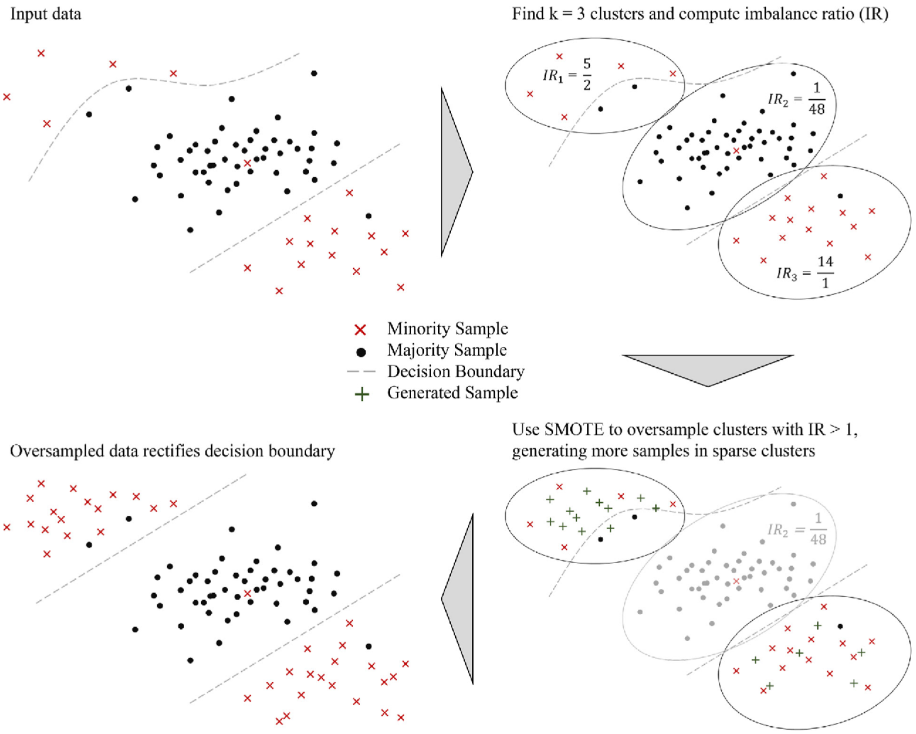
Input data
Find k = 3 clusters and compute imbalance ratio (IR)
$IR_1 = \frac{5}{2}$
$IR_2 = \frac{1}{48}$
$IR_3 = \frac{14}{1}$
× Minority Sample
● Majority Sample
– – Decision Boundary
+ Generated Sample
Use SMOTE to oversample clusters with IR > 1, generating more samples in sparse clusters
$IR_2 = \frac{1}{48}$
Oversampled data rectifies decision boundary

Figure 1. K-means SMOTE effectively addresses within-class imbalance by oversampling safe areas (Douzas et al., 2018).

4.5.7 LSTM

LSTM is a recurrent neural network that captures long-term dependencies in sequential data (Hochreiter and Schmidhuber, 1997). It excels in various applications, including natural-language processing and time series forecasting. In our LSTM model, experiments explored different parameters: LSTM unit sizes (32, 64, 128, 256), activation functions (sigmoid; tanh; ReLU, rectified linear unit), and a look-back period ranging from 3 to 10. We chose the categorical cross-entropy loss function for multi-class classification. Table 3 details the parameter range for the LSTM model.

4.5.8 Dynamic ensembling

Dynamic ensembling is a highly effective technique in ML that takes advantage of the adaptability and ongoing improvement of predictive models (Ko et al., 2008). It involves creating a versatile and continuously evolving ensemble by harnessing the strengths of multiple models, including RF, CatBoost, XGBoost, LightGBM, and AdaBoost. Traditionally, ensembling methods like bagging and boosting have focused on fixed ensembles. However, dynamic ensembling goes beyond this by introducing the ability to add or remove models based on their performance dynamically. In the case of dynamic ensembling with the models, as mentioned earlier, the monitoring criterion used is accuracy. Accuracy as the monitoring criterion ensures that the dynamic ensemble maintains a high level of accuracy in its predictions. If a model falls below a predefined accuracy threshold, it is considered underperforming and may be replaced to enhance the ensemble's overall performance.

Dynamic ensembling offers numerous advantages, including handling concept drift, where the underlying data distribution changes over time. By incorporating new models that capture updated patterns and relationships in the data, the dynamic ensemble can effectively adapt to conceptualize drift and maintain accurate predictions.

The dynamic ensemble model utilized base models such as RF, CatBoost, XGBoost, LightGBM, and AdaBoost. Each base model was trained individually with the same default parameter settings as their standalone counterparts. The range of parameters for the dynamic ensemble model is mentioned in Table 3.

5 Model execution, minimization, and handling class imbalance

A rigorous process was followed to develop an effective model for predicting the intensity of soil movement. The dataset has a 70 : 30 ratio, with 70 % allocated for training and 30 % for testing. To tackle the class imbalance issue in the training data, oversampling techniques were applied exclusively to the training set, ensuring a balanced representation of all three classes. The oversampling methods were not extended to the testing data, preserving its original distribution. In this study, we developed two methods, referred to as

Table 3. The range of parameters varied in the models.

Model	Parameter	Range of parameter
AdaBoost	Number of trees	[10, 100] in steps of 5
	Learning rate	[0.1, 2] in steps of 0.1
XGBoost	Number of trees	[10, 100] in steps of 5
	Learning rate	[0.05, 0.55] in steps of 0.05
	Maximum depth	[5, 50] in steps of 5
LightGBM	Number of trees	[10, 100] in steps of 5
	Learning rate	[0.05, 0.55] in steps of 0.05
	Maximum depth	[5, 50] in steps of 5
CatBoost	Loss function	Log, entropy, hinge
	Learning rate	[0.1, 2] in steps of 0.1
	Maximum depth	[3, 33] in steps of 3
RF	Number of trees	[10, 100] in steps of 5
	Criteria	Gini, entropy
	Maximum depth	[5, 50] in steps of 5
MLP	Look-back period	3 to 10
	Layers	[1, 3]
	Nodes per layer	[50, 250] in steps of 50
	Learning rate	[0.1, 0.9] in steps of 0.1
LSTM	Look-back period	3 to 10
	LSTM units	32, 64, 128, 256
	Activation function	Sigmoid, tanh, ReLU
	Learning rate	[0.1, 0.9] in steps of 0.1

the method with five training datasets (5-TD) and the method with 5-fold cross-validation (5-CV). The 5-TD method was employed for parameter variation analysis across different datasets. On the other hand, the 5-CV method was utilized for conducting 5-fold cross-validation to analyze the performance of the ML models.

5.1 5-TD method

For the 5-TD method, the training dataset was split into five training datasets, each utilized for parameter variation analysis. This involved training and optimizing the ML model for each dataset independently using the grid search method. Since each dataset possessed different optimal parameters, we calculated the mean and SD of the ML-optimized parameter values across all datasets to assess parameter variability. This enabled us to observe parameter variations across the ML models, providing insights into the sensitivity of the models to different dataset characteristics and parameter configurations. A lower SD implied that the model maintained consistency across each dataset and demonstrated robust generalization capabilities. Conversely, a higher SD suggested that the model encountered difficulties maintaining consistency across datasets, potentially hindering its ability to learn general patterns effectively. The evaluation primarily focused on F1 score metrics to determine how effectively the models predicted the intensity of soil movements in each of the five datasets.

5.2 5-CV method

For the 5-CV method, a suite of ML models underwent training using a 5-fold cross-validation approach (Kumar et al., 2023). In the 5-CV method, the training data were split into five datasets, where each dataset was alternately used for validation, while the others were used for training. The models were optimized by employing a grid search methodology and optimized based on performance on the five validation sets, and a single set of best-performing parameters was selected for each model. Subsequently, the models with the best parameters found during training were tested on the independent testing data, and their performance metrics were reported as indicative of their predictive capabilities. The evaluation primarily focused on F1 score metrics to determine how effectively the models predicted the intensity of soil movement across the five validation sets and the test set.

6 Results

6.1 Parameter analysis result

Upon scrutinizing the parameter analysis presented in Table 4 from the 5-TD method, a discernible trend emerged: models trained with oversampling techniques exhibit a notably smaller SD than their counterparts trained without oversampling. For instance, when examining the AdaBoost model, we observe that the SD of the parameter of number of trees was 0 for the oversampling case. In contrast, it stood at 16.43 for the dataset without oversampling. This phenomenon underscores the stabilizing effect of oversampling on parameter estimates, mitigating the variability that may arise from imbalanced datasets.

Similarly, in the case of the RF model, the SD of the parameter of number of trees was 0 with oversampling, indicating consistent parameter values across folds. Conversely, for the dataset without oversampling, the SD increased to 21.21, suggesting greater variability in parameter estimates. This trend persisted across various models and parameters, highlighting the robustness imparted by oversampling techniques in stabilizing model performance.

Overall, these examples underscore the importance of oversampling in reducing parameter variability and ensuring consistent model behavior, particularly in scenarios involving imbalanced datasets.

6.2 Optimized parameters

In the 5-CV method, we optimized the parameters separately for the ML models using a 5-fold cross-validation process on the full training dataset. In analyzing various SMOTE techniques, the parameter K, representing the count of nearest neighbors for synthesizing new samples, was consistently optimized at a value of 4. Table 5 presents each model's optimized parameter values obtained through the grid search in

Table 4. The result of parameter variation analysis across five datasets from the 5-TD method.

Model	Parameter	With oversampling		Without oversampling	
		Mean	SD	Mean	SD
AdaBoost	Number of trees	80	0	62	16.43
	Learning rate	0.66	0.22	0.9	0
XGBoost	Number of trees	50	0	50	0
	Maximum depth	20	0	10	0
	Learning rate	0.5	0	0.68	0.16
LightGBM	Number of trees	50	0	50	0
	Maximum depth	20	0	20	0
	Learning rate	0.5	0	0.6	0.12
CatBoost	Number of trees	50	0	50	0
	Maximum depth	20	0	20	0
	Learning rate	0.8	0	0.66	0.13
RF	Number of trees	80	0	50	21.21
	Maximum depth	20	0	20	0
MLP	Look-back period	2.8	0.44	3.6	1.34
	Layers	2	0	2	0
	Nodes in first layer	130	67.08	130	67.08
	Nodes in second layer	200	0	60	54.77
	Learning rate	0.78	0.16	0.64	0.28
LSTM	Look-back period	4.6	0.89	4	1.41
	Layers	2	0	2	0
	Nodes in first layer	90	22.36	70	27.39
	Nodes in second layer	160	54.77	100	61.24
	Learning rate	0.84	0.08	0.86	0.05

5-CV on the training dataset. These parameters were carefully fine-tuned to ensure the best fit for the given data. In the case of AdaBoost, the optimized values included 80 trees and a learning rate of 0.6. The optimized values for the XGBoost model consisted of 50 trees, a learning rate of 0.3, and a maximum depth of 10. These settings were determined to enhance the model's performance in terms of both speed and accuracy.

Similarly, the LightGBM model underwent parameter optimization, selecting 50 trees, a learning rate of 0.5, and a maximum depth of 20. Next, the CatBoost model was also optimized, leading to entropy selection as the loss function, 50 trees, a learning rate of 0.8, and a maximum depth of 20. In the RF model, the optimized values were 80 for the number of trees and 20 for the maximum depth, and the evaluation criteria were set to Gini. Likewise, the MLP model optimized its parameters with a look-back period of 3, 2 layers, and 200 nodes per layer. Similarly, the LSTM model consists of 2 layers with 100 and 200 nodes in the first and second layers and utilizes a ReLU activation function. Lastly, the dynamic ensemble model in this study incorporated the optimized RF, CatBoost, XGBoost, LightGBM, and AdaBoost models to improve the accuracy of landslide analysis predictions. By leveraging the strengths of these individually optimized models, as mentioned above, the dynamic ensemble model aimed to improve the accuracy and reliability of landslide analysis predictions.

6.2.1 Train–test results

Table 6 presents the training results of different classification models evaluated using 5-fold cross-validation on the training dataset and various oversampling techniques for landslide prediction, utilizing the 5-CV method. In Table 6, C0, C1, C2, and C3 represent no-movement, low-movement, moderate-movement, and high-movement classes' accuracies, respectively. These results provide valuable insights into the performance of each model when trained on the training dataset with and without oversampling. The XGBoost model with K-means SMOTE emerged as the best model in training, achieving outstanding accuracy, precision, recall, and F1 scores of 0.999, 0.999, 0.999, and 0.999, respectively. The dynamic ensemble model with the K-means SMOTE and borderline-SMOTE techniques also performed similarly with F1 scores of 0.998. It demonstrates remarkable predictive capability by achieving perfect accuracy in oversampling scenarios. When the XGBoost model was trained without over-

Table 5. The best value of the parameters was calibrated from the training data using the 5-CV method.

Model	Parameter	Best value of parameter
AdaBoost	Number of trees	80
	Learning rate	0.6
XGBoost	Number of trees	50
	Learning rate	0.3
	Maximum depth	10
LightGBM	Number of trees	50
	Learning rate	0.5
	Maximum depth	20
CatBoost	Loss function	Entropy
	Learning rate	0.8
	Number of trees	50
	Maximum depth	20
RF	Number of trees	80
	Criteria	Gini
	Maximum depth	20
MLP	Look-back period	3
	Layers	2
	Nodes per layer	200 in both layers
	Learning rate	0.6
LSTM	Look-back period	5
	LSTM units	100 in first and 200 in second layer
	Activation function	ReLU
	Learning rate	0.9

sampling, its accuracy, precision, recall, and F1 score were notably lower, with values of 0.999, 0.999, 0.971, and 0.985, respectively.

Table 7 presents the test results of various classification models combined with different oversampling techniques for landslide prediction (here models were trained using the 5-CV method). In Table 7, C0, C1, C2, and C3 represent no-movement, low-movement, moderate-movement, and high-movement classes' accuracies, respectively. Among them, the dynamic ensemble model utilizing the K-means SMOTE technique demonstrated exceptional performance in accurately predicting landslides on unseen data. It achieves impressive accuracy, precision, and recall rates of 0.995, 0.995, and 0.995, respectively, along with an F1 score of 0.995. These outstanding results confirm the effectiveness of the dynamic ensemble approach when combined with K-means SMOTE for accurate soil movement prediction. Similarly, the borderline-SMOTE technique also showed similar performance with accuracy, precision, recall, and an F1 score of 0.995 for all. When the model is tested without oversampling, its accuracy, precision, recall, and F1 score are notably lower, with values of 0.981, 0.646, 0.397, and 0.462, respectively. The best-performing model is highlighted in bold in Tables 6 and 7.

Moreover, it is noteworthy that K-means SMOTE consistently outperformed other oversampling techniques across all models during the test performance evaluations, establishing itself as the optimal technique. Notably, it is crucial to highlight the impact of oversampling on the performance of the dynamic ensemble model. This underscores the discernible effectiveness of K-means SMOTE in generating oversampling for the soil movement dataset. The success of K-means SMOTE can be attributed to its ability to identify clusters within the minority class and select similar features for oversampling. The IR employed by K-means SMOTE aids in determining the appropriate degree of oversampling for the minority class, ensuring a balanced representation of classes in synthetic samples.

Moreover, the absence of oversampling techniques negatively impacted the models' performance in both training and testing. Without oversampling, the models exhibited lower accuracy, precision, recall, and F1 scores during training and testing, emphasizing the challenges posed by class imbalance. In the absence of a balanced representation through oversampling, the models struggled to effectively learn and generalize from the imbalanced dataset. Consequently, this underscores the pivotal role of oversampling in mitigating class imbalance issues, leading to substantial enhancements

Table 6. Results of ML models obtained from the training dataset using 5-fold cross-validation in the 5-CV method. The best model with oversampling techniques and its results are shown in bold.

Model	Oversampling technique	Accuracy					Precision	Recall	F1 score
		C0	C1	C2	C3	Overall			
AdaBoost	SMOTE	0.942	0.562	0.640	0.817	0.747	0.748	0.747	0.747
	K-means SMOTE	0.948	0.760	0.675	0.855	0.807	0.809	0.807	0.806
	Borderline-SMOTE	0.919	0.565	0.667	0.815	0.740	0.741	0.740	0.740
	ADASYN	0.934	0.552	0.649	0.798	0.740	0.741	0.740	0.740
	Without oversampling	0.995	0.250	0.243	0.341	0.980	0.575	0.465	0.506
XGBoost	SMOTE	0.995	0.999	0.999	0.997	0.998	0.998	0.998	0.998
	K-means SMOTE	**0.997**	**0.999**	**0.999**	**0.998**	**0.999**	**0.999**	**0.999**	**0.999**
	Borderline-SMOTE	0.996	0.999	0.999	0.998	0.998	0.998	0.998	0.998
	ADASYN	0.994	0.999	0.999	0.997	0.998	0.998	0.998	0.998
	Without oversampling	1.000	0.995	0.953	0.906	0.999	0.999	0.971	0.985
LightGBM	SMOTE	0.984	0.994	0.999	0.988	0.991	0.991	0.991	0.991
	K-means SMOTE	0.991	0.998	0.998	0.996	0.996	0.996	0.996	0.996
	Borderline-SMOTE	0.985	0.999	0.999	0.995	0.995	0.995	0.995	0.995
	ADASYN	0.983	0.994	0.998	0.987	0.991	0.991	0.991	0.991
	Without oversampling	1.000	1.000	1.000	0.976	0.994	0.999	0.999	0.996
CatBoost	SMOTE	0.990	0.999	0.999	0.997	0.997	0.997	0.997	0.997
	K-means SMOTE	0.991	0.999	0.999	0.997	0.997	0.997	0.997	0.997
	Borderline-SMOTE	0.992	0.999	0.999	0.997	0.997	0.997	0.997	0.997
	ADASYN	0.991	0.999	0.999	0.997	0.996	0.996	0.996	0.996
	Without oversampling	0.999	0.924	0.916	0.735	0.997	0.997	0.903	0.946
RF	SMOTE	0.920	0.892	0.951	0.905	0.921	0.923	0.921	0.922
	K-means SMOTE	0.920	0.921	0.959	0.902	0.925	0.928	0.925	0.926
	Borderline-SMOTE	0.948	0.969	0.988	0.959	0.967	0.967	0.967	0.967
	ADASYN	0.921	0.898	0.945	0.899	0.915	0.917	0.915	0.915
	Without oversampling	1.000	0.701	0.682	0.537	0.992	0.995	0.742	0.841
MLP	SMOTE	0.959	0.976	0.997	0.952	0.961	0.961	0.961	0.961
	K-means SMOTE	0.940	0.996	0.984	0.957	0.974	0.974	0.974	0.974
	Borderline-SMOTE	0.968	0.974	0.989	0.913	0.964	0.964	0.964	0.964
	ADASYN	0.929	0.975	0.981	0.984	0.961	0.961	0.961	0.961
	Without oversampling	0.997	0.016	0.000	0.056	0.980	0.693	0.336	0.381
LSTM	SMOTE	0.882	0.841	0.881	0.896	0.875	0.884	0.875	0.877
	K-means SMOTE	0.980	0.996	0.992	0.968	0.984	0.984	0.984	0.984
	Borderline-SMOTE	0.946	0.954	0.997	0.965	0.966	0.966	0.966	0.966
	ADASYN	0.955	0.979	0.997	0.955	0.971	0.971	0.971	0.971
	Without oversampling	0.999	0.859	0.925	0.700	0.995	0.979	0.871	0.919
Dynamic ensemble	SMOTE	0.992	0.999	0.999	0.999	0.997	0.997	0.997	0.997
	K-means SMOTE	0.994	0.999	0.999	0.999	0.998	0.998	0.998	0.998
	Borderline-SMOTE	0.997	0.999	0.999	0.998	0.998	0.998	0.998	0.998
	ADASYN	0.992	0.999	0.999	0.998	0.997	0.997	0.997	0.997
	Without oversampling	1.000	0.951	0.944	0.770	0.997	0.999	0.916	0.954

in predictive accuracy and overall model robustness during training and testing evaluations.

Models trained with oversampling techniques consistently demonstrate comparable performance across both training and testing datasets, indicating a lack of overfitting. Conversely, models trained without oversampling, notably RF, MLP, LSTM, and a dynamic ensemble, exhibit signs of overfitting, as evidenced by significantly higher performance metrics in the training dataset relative to the testing dataset. This observation underscores the effectiveness of oversampling techniques in mitigating overfitting by enhancing the model's ability to generalize to unseen data.

Table 7. Results of ML models obtained from the testing dataset in the 5-CV method.

Model	Oversampling technique	Accuracy					Precision	Recall	F1 score
		C0	C1	C2	C3	Overall			
AdaBoost	SMOTE	0.939	0.548	0.436	0.763	0.932	0.383	0.671	0.442
	K-means SMOTE	0.946	0.583	0.436	0.681	0.939	0.382	0.662	0.445
	Borderline-SMOTE	0.917	0.595	0.462	0.756	0.911	0.374	0.682	0.423
	ADASYN	0.995	0.226	0.205	0.230	0.978	0.514	0.414	0.447
	Without oversampling	0.931	0.524	0.436	0.681	0.924	0.360	0.643	0.412
XGBoost	SMOTE	0.991	0.976	0.974	0.837	0.989	0.774	0.945	0.846
	K-means SMOTE	0.993	0.952	0.949	0.785	0.990	0.787	0.920	0.842
	Borderline-SMOTE	0.994	0.905	0.769	0.733	0.990	0.803	0.850	0.823
	ADASYN	0.990	0.988	0.974	0.830	0.988	0.761	0.946	0.837
	Without oversampling	0.996	0.250	0.026	0.333	0.980	0.553	0.401	0.447
LightGBM	SMOTE	0.983	0.905	0.974	0.748	0.980	0.656	0.903	0.750
	K-means SMOTE	0.984	0.917	0.872	0.704	0.980	0.654	0.869	0.737
	Borderline-SMOTE	0.990	0.738	0.667	0.637	0.983	0.695	0.758	0.720
	ADASYN	0.981	0.917	0.974	0.741	0.978	0.638	0.903	0.735
	Without oversampling	0.996	0.214	0.205	0.326	0.980	0.547	0.435	0.472
CatBoost	SMOTE	0.986	0.964	0.974	0.852	0.984	0.705	0.944	0.799
	K-means SMOTE	0.988	0.952	0.974	0.815	0.986	0.726	0.932	0.810
	Borderline-SMOTE	0.990	0.798	0.641	0.689	0.984	0.720	0.779	0.743
	ADASYN	0.987	0.988	0.974	0.859	0.985	0.722	0.952	0.814
	Without oversampling	0.997	0.226	0.179	0.311	0.981	0.611	0.428	0.487
RF	SMOTE	0.988	0.988	0.974	0.970	0.988	0.763	0.980	0.851
	K-means SMOTE	0.995	0.917	0.821	0.867	0.993	0.885	0.900	0.889
	Borderline-SMOTE	0.991	0.976	0.974	0.956	0.991	0.801	0.974	0.875
	ADASYN	0.989	0.988	0.974	0.978	0.988	0.757	0.982	0.848
	Without oversampling	0.998	0.190	0.051	0.289	0.980	0.676	0.382	0.440
MLP	SMOTE	0.958	1.000	1.000	0.948	0.958	0.554	0.977	0.671
	K-means SMOTE	0.965	0.988	0.974	0.830	0.964	0.578	0.939	0.689
	Borderline-SMOTE	0.937	0.750	0.641	0.659	0.932	0.444	0.747	0.518
	ADASYN	0.927	1.000	0.974	0.963	0.928	0.554	0.966	0.652
	Without oversampling	0.995	0.012	0.026	0.015	0.974	0.380	0.262	0.270
LSTM	SMOTE	0.878	0.774	0.897	0.815	0.877	0.451	0.841	0.522
	K-means SMOTE	0.981	0.869	0.923	0.763	0.977	0.693	0.884	0.766
	Borderline-SMOTE	0.948	0.917	1.000	0.919	0.948	0.527	0.946	0.636
	ADASYN	0.953	0.952	1.000	0.911	0.953	0.552	0.954	0.661
	Without oversampling	0.996	0.488	0.667	0.415	0.985	0.804	0.642	0.704
Dynamic ensemble	SMOTE	0.978	0.999	0.999	0.997	0.994	0.994	0.994	0.994
	K-means SMOTE	**0.999**	**1.000**	**0.979**	**1.000**	**0.995**	**0.995**	**0.995**	**0.995**
	Borderline-SMOTE	0.982	0.999	0.999	0.997	0.995	0.995	0.995	0.995
	ADASYN	0.979	0.999	0.999	0.997	0.994	0.994	0.994	0.994
	Without oversampling	0.998	0.167	0.128	0.296	0.981	0.646	0.397	0.462

Comparing the dynamic ensemble model with other classification models, it becomes evident that the dynamic ensemble model with K-means SMOTE consistently outperformed the rest, highlighting the effectiveness in accurately predicting landslides.

These findings underscore the importance of carefully selecting appropriate ML models and employing suitable oversampling techniques to address the class imbalance challenge in soil movement prediction. They provide valuable insights into the performance and suitability of these models and techniques for enhancing landslide prediction accuracy, ultimately enabling proactive measures to mitigate landslide risks.

In Fig. 2, we juxtaposed the performance metrics obtained using K-means SMOTE against those obtained without oversampling across various machine learning models. In Fig. 2, the blue bars represent the F1 score achieved with K-means SMOTE (oversampling), while the orange bars represent the F1 score without oversampling. Notably, when comparing the performance in the test dataset using the F1 score metric, the oversampling dataset generated with K-means SMOTE consistently yielded superior results compared to the without oversampling approach. For instance, in the case of the AdaBoost model, K-means SMOTE resulted in an F1 score of 0.412 for the technique without oversampling, whereas it achieved an F1 score of 0.445 for K-means SMOTE. Similarly, in the XGBoost model, the F1 score improved from 0.447 without oversampling to 0.842 with K-means SMOTE. This trend persisted across various other models such as LightGBM, CatBoost, RF, MLP, LSTM, and a dynamic ensemble, where K-means SMOTE consistently demonstrated superior performance in terms of the F1 score compared to the technique without oversampling. These results underscore the effectiveness of K-means SMOTE in enhancing the predictive performance of ML models for soil movement prediction tasks.

Figure 3 illustrates the confusion matrix, depicting the performance of the dynamic ensemble model in both the training and testing datasets, utilizing the K-means SMOTE oversampling technique. The confusion matrix provides a comprehensive overview of the model's classification accuracy by presenting the true and predicted labels across different classes. The dynamic ensemble model demonstrates robust performance in the training dataset, as evidenced by the high counts along the diagonal, indicating a substantial number of correct predictions across all classes. Similarly, in the testing dataset, the model maintains its efficacy, with the majority of samples correctly classified across various classes.

7 Discussion and conclusions

In summary, the threat posed by landslides requires the development of effective prediction frameworks, although modeling the chaotic nature of natural data remains challenging. The analyzed dataset exhibited a significant class imbalance, with the majority class dominating the samples. This distribution imbalance necessitated careful consideration and appropriate techniques to address the issue.

Various oversampling techniques were employed to tackle the class imbalance, including SMOTE and its extensions (K-means SMOTE, borderline-SMOTE, and ADASYN). ADASYN, which focuses on the minority-class boundary, effectively generated synthetic data points and improved the class distribution balance.

Multiple classification models, such as AdaBoost, XGBoost, LightGBM, CatBoost, RF, MLP, LSTM, and a dynamic ensemble, were evaluated to predict soil movement. The grid search approach and 5-CV were employed to optimize the parameters of each model. Within the 5-CV framework, the parameter analysis was conducted on each fold treated as an independent dataset, allowing for a comprehensive assessment of parameter variability across different dataset splits. This approach facilitated the identification of optimal parameter configurations that yielded consistent performance across diverse dataset distributions. By treating each fold as an independent dataset, the parameter analysis provided insights into the variability in parameter values, thereby enhancing our understanding of how the models generalize to unseen data.

The ML models' training results highlight oversampling's significant impact on model performance. The dynamic ensemble model, particularly when coupled with K-means SMOTE, emerges as the standout performer in the training phase. This model demonstrates superior predictive capabilities by achieving remarkable accuracy, precision, recall, and F1 scores of 0.998, 0.998, 0.998, and 0.998, respectively.

Furthermore, these models were tested to assess their ability to generalize well to unseen data. The testing results showcased the dynamic ensemble model with K-means SMOTE as the top performer, achieving an outstanding accuracy of 0.995, precision of 0.995, recall of 0.995, and F1 score of 0.995. This confirms that the exceptional performance observed in training extends to the testing phase, emphasizing the robustness and reliability of the dynamic ensemble approach with K-means SMOTE. Moreover, the dynamic ensemble model incorporating borderline-SMOTE emerges as the second-best model in the test phase, showcasing high accuracy, precision, and recall rates of 0.995, 0.995, and 0.995, respectively, along with an F1 score of 0.995. This result reinforces the reliability and robustness of the model in tackling landslide prediction tasks.

The superior performance of the K-means SMOTE technique can be attributed to its ability to identify clusters within the minority class and generate synthetic samples that maintain the underlying structure of the data. By considering the IR, K-means SMOTE ensures a balanced representation of classes in the synthetic samples, contributing to improved model generalization and predictive accuracy. Furthermore, the lack of oversampling adversely affects both training and testing performances. The models face challenges in learning and generalizing from the imbalanced dataset without a balanced representation.

On the other hand, the success of the dynamic ensemble model, comprising AdaBoost, XGBoost, LightGBM, CatBoost, and RF, can be attributed to the complementary strengths of these diverse algorithms. Ensemble methods leverage the collective decision-making power of multiple models, each capturing different aspects of the underlying data patterns. The combination of boosting algorithms like AdaBoost, gradient boosting methods like XGBoost, tree-based models like LightGBM and CatBoost, and the robustness of RF creates a robust and versatile ensemble that excels

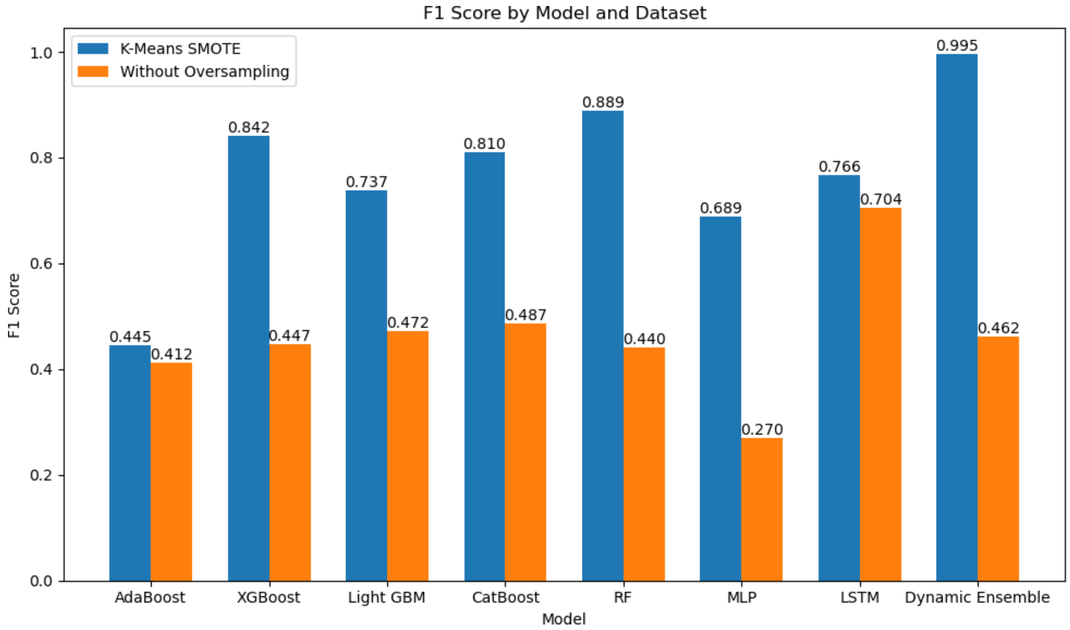

Figure 2. Comparison of F1 score performance between K-means SMOTE and techniques without oversampling across various ML models for soil movement prediction in testing. Blue bars represent F1 scores achieved with K-means SMOTE, while orange bars represent F1 scores obtained without oversampling.

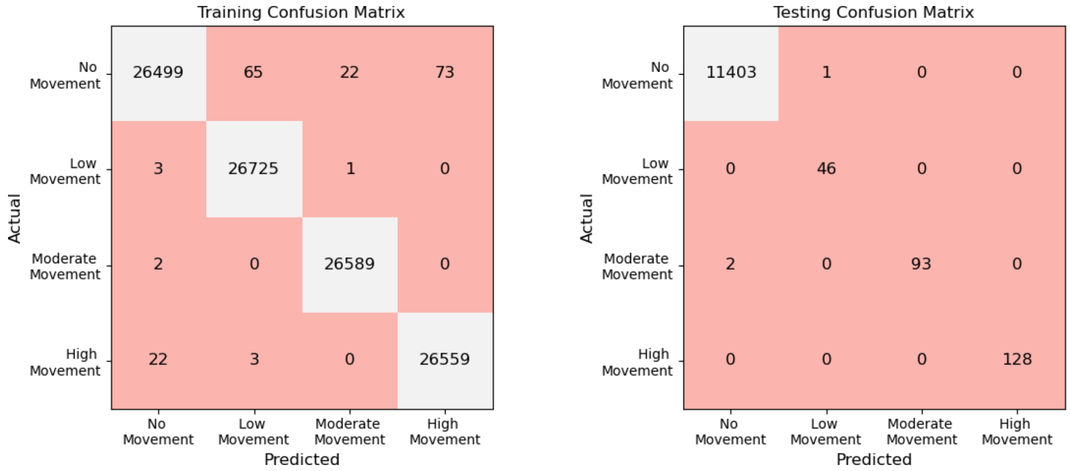

Figure 3. Confusion matrix depicting the performance of the dynamic ensemble model on the training and testing datasets using the K-means SMOTE oversampling technique.

in handling various aspects of the dataset, contributing to its overall superior performance.

In summary, the findings underscore the critical role of oversampling techniques, especially K-means SMOTE, in enhancing the predictive performance of landslide prediction models. The success of the dynamic ensemble model further highlights the importance of ensemble techniques in aggregating diverse model predictions for improved accuracy.

The superior performance demonstrated by oversampling techniques compared to those without oversampling can be attributed to several factors. Firstly, oversampling techniques address class imbalance by generating synthetic samples for minority classes, thus providing the model with more representative training data. This allows for the ML model to learn the underlying patterns of the minority class more effectively, leading to improved classification performance. Additionally, oversampling techniques help reduce the risk of overfitting by providing a more balanced representation of the dataset, enhancing the model's ability to generalize to unseen data. Moreover, by increasing the diversity of the training data, oversampling techniques enable the model to capture a wider range of variation within the dataset, resulting in better generalization performance. Overall, using oversampling techniques ensures that the ML model is better equipped to handle imbalanced datasets, leading to enhanced predictive performance in soil movement prediction tasks.

Furthermore, the parameter analysis reveals that oversampling techniques add generalized information to the dataset, making it more consistent across different datasets. This reduced variability in the dataset allows for ML models to learn these generalized patterns more effectively. As evident in the parameter analysis results, oversampling techniques lead to a smaller SD in parameter values across different models, indicating improved consistency and generalization. This further supports the notion that oversampling techniques help mitigate overfitting and enhance the overall performance of ML models in soil movement prediction tasks.

Despite these achievements, it is crucial to acknowledge the study's limitations. The generalizability of the findings to different geological conditions or regions may be restricted due to the specificity of the dataset. While effective, the synthetic data points generated through oversampling may only capture part of the complexity inherent in real-world landslide occurrences. The choice of classification models and parameter settings introduces a level of bias, with alternative configurations potentially yielding different results. Additionally, relying on historical data may limit the model's ability to account for future changes or unforeseen events, such as changes in rainfall intensity, seismic activity, or human influences.

In future work, the exploration of encoder–decoder or transformer models in the class-imbalanced movement dataset is planned. These models, known for their success in sequence-to-sequence tasks, may improve classification accuracy and address class imbalance challenges. This avenue of experimentation aims to provide valuable insights into the suitability of advanced models for analyzing and modeling imbalanced movement data.

To sum up, the study contributes to understanding landslide risks and supports the development of effective preventive measures. The combination of robust oversampling techniques, ensemble modeling, and a systematic approach to parameter tuning yields a promising framework for accurate landslide prediction. The work presented lays the groundwork for future research to refine models and address the inherent challenges in landslide prediction tasks.

Code availability. The code is available on Zenodo at https://doi.org/10.5281/zenodo.11446683 (Kumar, 2024).

Data availability. The dataset referred to in this manuscript is not publicly accessible due to privacy concerns, third-party licensing restrictions, and ethical considerations. Sharing raw data publicly could compromise sensitive geographical information and lead to misuse. Therefore, the dataset can be made available upon request to the corresponding author, Praveen Kumar, or Varun Dutt at varun@iitmandi.ac.in, ensuring adherence to ethical guidelines.

Author contributions. The paper benefited from the collaborative efforts of each author. PK played a central role in conceptualizing the research, drafting the original manuscript, and conducting experiments. PP contributed significantly to the project by curating data, developing methodologies, and ensuring methodological accuracy. KVU provided valuable insights and contributed to the project by validating the data. VD supervised the experiment, ensuring adherence to best practices and providing guidance throughout the research process. Together, the collective contributions of all authors have enriched the paper, resulting in a comprehensive and robust study.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. While Copernicus Publications makes every effort to include appropriate place names, the final responsibility lies with the authors.

Special issue statement. This article is part of the special issue "Reducing the impacts of natural hazards through forecast-based action: from early warning to early action". It is a result of the EGU General Assembly 2022, Vienna, Austria, 23–27 May 2022.

Acknowledgements. We want to acknowledge and express our sincere gratitude to the Department of Science and Technology (DST), India, and the District Disaster Management Authority (DDMA) of Kangra, Kinnaur, and Mandi for their invaluable financial support towards this research project. We are also immensely grateful to the Indian Institute of Technology (IIT) Mandi for generously providing us with the necessary infrastructure, including research facilities and computational resources, that have been instrumental in the successful execution of this study.

Financial support. This research has been supported by the District Disaster Management Authority (grant nos. IITM/DDMA-Kan/KVU/357, IITM/DDMA-Kinn/VD/345, IITM/DDMA-M/VD/325, and IITM/DDMA-M/VD/358).

Review statement. This paper was edited by Gabriela Guimarães Nobre and reviewed by two anonymous referees.

References

Breiman, L.: Random forests, Mach. Learn., 45, 5–32, 2001.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P.: SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res., 16, 321–357, 2002.

Chen, T. and Guestrin, C.: Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, California, USA, 13–17 August 2016, 785–794, https://doi.org/10.1145/2939672.2939785, 2016.

Crosta, G.: Regionalization of rainfall thresholds: an aid to landslide hazard evaluation, Environ. Geol., 35, 131–145, 1998.

Douzas, G., Bacao, F., and Last, F.: Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE, Inform. Sciences, 465, 1–20, 2018.

Gupta, V., Bhasin, R. K., Kaynia, A. M., Tandon, R. S., and Venkateshwarlu, B.: Landslide hazard in the Nainital township, Kumaun Himalaya, India: the case of September 2014 Balia Nala landslide, Nat. Hazards, 80, 863–877, 2016.

Han, H., Wang, W. Y., and Mao, B. H.: Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning, in: Advances in Intelligent Computing: International Conference on Intelligent Computing, ICIC 2005, Hefei, China, 23–26 August 2005, Proceedings, Part II, 878–887, https://doi.org/10.1007/11538059_91, 2005.

He, H., Bai, Y., Garcia, E. A., and Li, S.: ADASYN: Adaptive synthetic sampling approach for imbalanced learning, in: 2008 IEEE International Joint Conference on Neural Networks, IEEE World Congress on Computational Intelligence, Hong Kong, China, 1–8 June 2008, 1322–1328, https://doi.org/10.1109/IJCNN.2008.4633969, 2008.

Hochreiter, S. and Schmidhuber, J.: Long short-term memory, Neural Comput., 9, 1735–1780, 1997.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T. Y.: Lightgbm: A highly efficient gradient boosting decision tree, Adv. Neur. In., 30, 3149–3157, ISBN 9781510860964, 2017.

Ko, A. H., Sabourin, R., and Britto Jr., A. S.: From dynamic classifier selection to dynamic ensemble selection, Pattern Recogn., 41, 1718–1731, 2008.

Kumar, P.: bluecodeindia/SMOTE-and-Classification-Models: v1.0.0-initial-release, Zenodo [code], https://doi.org/10.5281/zenodo.11446683, 2024.

Kumar, P., Sihag, P., Pathania, A., Agarwal, S., Mali, N., Chaturvedi, P., Singh, R., Uday, K. V., and Dutt, V.: Landslide debris-flow prediction using ensemble and non-ensemble machine-learning methods, in: International Conference on Time Series and Forecasting, Granda, Spain, 25–27 September 2019, Vol. 1, 614–625, ISBN 978-84-17970-78-9, 2019.

Kumar, P., Sihag P., Pathania A., Agarwal S., Mali N., Singh R., Chaturvedi P., Uday K. V., and Dutt V.: Predictions of weekly slope movements using moving-average and neural network methods: a case study in Chamoli, India, in: Soft Computing for Problem Solving 2019: Proceedings of SocProS 2019, Liverpool, UK, 2–4 September 2019, Vol. 2, 67–81, https://doi.org/10.1007/978-981-15-3287-0_6, 2020.

Kumar, P., Sihag, P., Sharma, A., Pathania, A., Singh, R., Chaturvedi, P., and Dutt, V.: Prediction of Real-World Slope Movements via Recurrent and Non-recurrent Neural Network Algorithms: A Case Study of the Tangni Landslide, Indian Geotechnical Journal, 51, 788–810, 2021a.

Kumar, P., Sihag, P., Chaturvedi, P., Uday, K. V., and Dutt, V.: BS-LSTM: an ensemble recurrent approach to forecasting soil movements in the real world, Front. Earth Sci., 9, 696–792, 2021b.

Kumar, P., Priyanka, P., Dhanya, J., Uday, K. V., and Dutt, V.: Analyzing the Performance of Univariate and Multivariate Machine Learning Models in Soil Movement Prediction: A Comparative Study, IEEE Access, 11, 62368–62381, 2023.

Parkash, S.: Historical records of socio-economically significant landslides in India, Journal of South Asia Disaster Studies, 4, 177–204, 2011.

Pathania, A., Kumar, P., Sihag, P., Chaturvedi, P., Singh, R., Uday, K. V., and Dutt, V.: A low-cost, sub-surface IoT framework for landslide monitoring, warning, and prediction, in: Proceedings of 2020 International Conference on Advances in Computing, Communication, Embedded and Secure Systems, Cochin, India, 28–30 May 2020, https://www.researchgate.net/publication/339850328_A_Low_Cost_Sub-Surface_IoT_Framework_for_Landslide_Monitoring_Warning_and_Prediction (last access: 30 May 2024) 2020.

Pathania, A., Kumar, P., Priyanka, P., Maurya, A., Uday, K. V., and Dutt, V.: Development of an Ensemble Gradient Boosting Algorithm for Generating Alerts About Impending Soil Movements, in: Machine Learning, Deep Learning and Computational Intelligence for Wireless Communication: Proceedings of MDCWC 2020, Trichy, India, 22–24 October 2020, 365–379, https://doi.org/10.1007/978-981-16-0289-4_28, 2021.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A.: CatBoost: unbiased boosting with categorical features, Adv. Neur. In., 31, 6639–6649, https://doi.org/10.48550/arXiv.1706.09516, 2018.

Ray, R. L., Lazzari, M., and Olutimehin, T.: Remote sensing approaches and related techniques to map and study landslides, Landslides Investig. Monit., https://doi.org/10.5772/intechopen.93681, 2020.

Rosenblatt, F.: Principles of neurodynamics. perceptrons and the theory of brain mechanisms, Cornell Aeronautical Lab Inc Buffalo NY, https://doi.org/10.21236/AD0256582, 1961.

Sahin, E. K.: Comparative analysis of gradient boosting algorithms for landslide susceptibility mapping, Geocarto Int., 37, 2441–2465, 2022.

Semwal, T., Priyanka, P., Kumar, P., Dutt, V., and Uday, K. V.: Predictions of Root Tensile Strength for Different Vegetation Species Using Individual and Ensemble Machine Learning Models, in: Trends on Construction in the Digital Era: Proceedings of ISIC 2022, Guimaraes, Portugal, 7–9 September 2022, 87–100, https://doi.org/10.1007/978-3-031-20241-4_7, 2022.

Wu, Y., Ke, Y., Chen, Z., Liang, S., Zhao, H., and Hong, H.: Application of alternating decision tree with AdaBoost and bagging ensembles for landslide susceptibility mapping, Catena, 187, 104396, https://doi.org/10.1016/j.catena.2019.104396, 2020.

Zhang, S., Wang, Y., and Wu, G.: Earthquake-Induced Landslide Susceptibility Assessment Using a Novel Model Based on Gradient Boosting Machine Learning and Class Balancing Methods, Remote Sens., 14, 5945, https://doi.org/10.3390/rs14235945, 2022.