Addressing Class Imbalance in Soil Movement Predictions

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

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

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

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 two years (2019-2021) of monitoring data from a landslide in Uttarakhand, was split into a 70:30 ratio for training and testing. To tackle the class imbalance problem, various oversampling techniques, including Synthetic Minority Oversampling Technique (SMOTE), K-Means SMOTE, Borderline SMOTE, ~~Support Vector Machine 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 (Light GBM), Adaptive Boosting (AdaBoost), Category Boosting (CatBoost), ~~Multilayer Perceptron (MLP),~~ Long Short-Term Memory (LSTM), Multilayer Perceptron (MLP), and dynamic ensemble models, 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 ~~99.68% each and an F1-score of 0.9968. The RF model with K-Means SMOTE stood out as the second-best performer, achieving an impressive accuracy, precision, and recall rate of 99.64% each and an F1-score of 0.9964. These results show that ML models with class imbalance techniques have the potential to significantly improve soil movement predictions in landslide-prone areas.~~ 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.

Keywords: Soil Movement Prediction; Class Imbalance; Oversampling; Machine Learning; Landslide Prone Areas.

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 Global Positioning System (GPS~~, LIDAR,~~), Light Detection and Ranging (LiDAR), 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 ~~low~~ cost-effective 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; Kumar et al., 2021b~~).~~, Kumar et al., 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

~~* Corresponding author: Praveen Kumar~~
~~*E-mail address:* dr.praveenkumar.ml@gmail.com~~

class imbalance issues, techniques such as Synthetic Minority Oversampling Technique (SMOTE), K-Means SMOTE, ~~Support Vector Machine SMOTE (SVM SMOTE),~~ Borderline SMOTE, and Adaptive Synthetic Minority Oversampling Technique (ADASYN) are employed to balance the dataset (Chawla et al., 2002; Douzas et al., 2018; ~~Tang et al., 2008;~~ Han et al., 2005; He et al., 2008).

~~One crucial literature gap~~ Several researchers have dedicated their efforts to addressing class imbalance problems in ~~landslide~~ ML. Notably, Chawla et al. (2002) introduced the SMOTE, Douzas et al. (2018) devised the K-Means SMOTE, Han et al. (2005) proposed the 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 ~~is the need for more focus on addressing the challenges posed by~~ requires further investigation, particularly considering the complexities associated with a class imbalance in ~~landslide~~the datasets. ~~Although machine learning models have been extensively studied~~Despite extensive research on ML models' predictive abilities for ~~predicting~~ soil movement in landslides, ~~the impact of~~there still needs to be more understanding regarding how class imbalance ~~on their~~affects the models' performance and accuracy ~~has received less attention. In this paper, we explore the development~~. 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 ~~different ML models and class imbalance techniques to predict landslides. By developing SMOTE variants and other balancing techniques, we aim to overcome the challenge of imbalanced data~~soil movement. Various multivariate classification models, including Random Forest (RF), Adaptive Boosting (AdaBoost), Extreme Gradient Boosting (XGBoost), Light Gradient Boosted Machine (Light GBM), Category Boosting (CatBoost), ~~Multilayer Perceptron (MLP),~~ Long Short-Term Memory (LSTM), Multilayer Perceptron (MLP), and an ensemble of RF, AdaBoost, XGBoost, Light GBM, 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).

~~The novelty of this paper lies in its exploration and development of different ML models and class imbalance techniques designed explicitly for landslide prediction. While previous research has studied ML models for predicting~~ This study delves into the field of soil movement ~~in landslides, this paper focuses on addressing~~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~~, which has received limited attention in the literature. The paper introduces and applies various SMOTE variants and other balancing techniques to overcome~~. While previous research has primarily focused on ML models for soil movement prediction, this work addresses the issue of imbalanced data~~, improving the performance~~ head-on. Multiple variants of the SMOTE and other balancing strategies are introduced and implemented to enhance the efficacy and accuracy of the ML models.

Additionally, ~~the paper presents the development of a range of multivariate classification models, including RF, AdaBoost, XGBoost, Light GBM, CatBoost, MLP, LSTM, and an ensemble of RF, AdaBoost, XGBoost, Light GBM, and CatBoost, showcasing the comprehensive and innovative approach taken to predict soil movement in landslide-prone areas. The findings of~~ this ~~study have significant implications for mitigating landslide risks and improving the prediction of soil movements in landslide-prone areas. By effectively predicting landslides, it becomes possible to reduce economic losses and prevent fatalities caused by these destructive events. Furthermore, developing and deploying low-~~research explores using cost~~-~~-effective Internet of Things (IoT) technologies in developing ~~countries can enhance the~~ regions to improve the investigation and assessment ~~and analysis~~ of landslide hazards~~, making them more accessible and efficient. In this study, we utilized data from a low-cost IoT monitoring station located in Uttarakhand, spanning~~. The dataset used in this study spans two years, from June 2019 to June 2021~~. This dataset~~, and was ~~used to train the ML models. By leveraging this real-world data, we aimed to capture the specific~~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, Light GBM, CatBoost, MLP, LSTM, and a dynamic ensembling 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.

~~In what follows, we discuss the methodology employed in this study to address the class imbalance challenge and improve landslide prediction. First, we provide an overview of the dataset collected from the monitoring station in Uttarakhand, India, highlighting the weather and soil-related parameters measured over two years. Next, we describe the application of class imbalance techniques, including SMOTE variants and other balancing methods, to overcome the imbalanced nature of the dataset. We explain how these techniques help generate synthetic samples and balance the representation of different landslide classes. Then, we present the development and implementation of various ML models, including RF, AdaBoost, XGBoost, Light GBM, CatBoost, MLP, LSTM, and an ensemble of RF, AdaBoost, XGBoost, Light GBM, and CatBoost. We highlight each model's unique features and advantages in capturing the complex relationships and patterns in the landslide data. Finally, we discuss the implications of our findings for mitigating landslide risks, improving landslide prediction accuracy, and the potential for deploying low-cost IoT technologies in landslide-prone areas.~~

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 the 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. ~~Tang et al. (2008) modified SVMs by incorporating different rebalancing heuristics, such as cost-sensitive learning, over-sampling, and undersampling. Among the variations of SVM, the granular SVMs-repetitive undersampling model (GSVM-RU) has been found to be the most effective and efficient.~~ 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 true positive rate and F-value. Lastly, He et al. (2008) developed the 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 exploration and application for 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 ~~have worked on developing different~~developed various ML models to predict soil movement and ~~solve~~ 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) ~~predicted the landslide debris-flow using~~developed an ensemble ~~and non-ensemble~~of ML models ~~focuses~~

~~on~~(RF, Bagging, Stacking, and Voting) for predicting ~~debris-flow~~soil movement at the Tangni landslide in Uttarakhand, India. ~~The study compares ensemble ML models (RF, Bagging, Stacking, and Voting)~~ These models were compared with ~~non-ensemble model (~~Sequential Minimal Optimization (SMO) and Autoregression (AR). The results indicate that the ensemble models~~, specifically Bagging, Stacking, and RF, performed better than the non-ensemble~~ outperformed the SMO and AR models in predicting ~~weekly debris-flow.~~soil movement. Furthermore, Semwal et al. (2022) developed the ~~Sequential Minimal Optimization Regression (~~SMOreg~~)~~,, Instance-based Learning (IBk), RF, Linear Regression (LR), ~~Multi-layer Perceptron (~~MLP~~)~~,, as well as ensemble ML models to predict root tensile strength for different vegetation species. ~~The researchers investigate the relationship between root tensile strength and root and shoot characteristics of vegetation.~~ The results show that the ~~ensemble via~~ MLP ~~outperformed all~~ performed better than the other ~~individual~~ models, providing more accurate predictions of root tensile strength. Next, Wu et al. (2020) developed the decision tree (DT) with AdaBoost and bagging ensembles for ~~landslide~~mapping the susceptibility ~~mapping~~ of landslides in Longxian County, Shaanxi Province, China. ~~They used~~Researcher developed the technique with ensemble of Alternating Decision Tree (ADTree) ~~along with ensemble techniques such as~~with Bagging and AdaBoost to map landslide susceptibility. ~~The performance of the models was evaluated using receiver operating characteristic (ROC) curve, area under the ROC curve (AUC), and statistical measures.~~ The results ~~showed~~revealed that ~~the~~ensemble of ADTree~~-~~ and AdaBoost model ~~had the highest success rate and prediction rate, outperforming the~~performed better than the individual ADTree model and ensemble of ADTree~~-~~ and Bagging model. Similarly, Pathania et al. (2021) developed a novel ensemble gradient boosting model, called SVM-XGBoost, for ~~generating specific warnings about impending~~ soil movements warning at ~~a~~Gharpa landslide~~ site in Gharpa Hill~~, Mandi, India. They compared the performance of SVM-XGBoost with other models such as individual SVMs, DTs, RF, XGBoost, Naïve Bayes (NB), ~~decision trees (DTs), RF, SVMs, XGBoost,~~ and different variants of XGBoost. The results showed that the SVM-XGBoost model ~~outperformed the~~performed better than other models in soil movement prediction. In their research, Kumar et al. (2021b) directed their attention toward predicting soil movement~~ alerts 10 minutes ahead of time. Similarly, Kumar et al. (2021b) developed different variants of LSTM and an ensemble of LSTM models named BS-LSTM in their research to predict soil movement~~, specifically at the Tangni landslide site in India. ~~The results demonstrated~~ 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 ~~for~~in accurately predicting soil movement~~ prediction. Furthermore,~~. Similarly, Zhang et al. (2022) conducted a study ~~on earthquake-induced landslide~~to assess the susceptibility ~~assessment~~of landslides using ~~a novel model based on~~ gradient~~-~~boosting ML ~~and~~ techniques coupled with class~~-~~balancing methods. ~~The research~~Their investigation specifically focused on the aftermath of the 2018 Hokkaido earthquake and ~~used different~~employed diverse datasets and ~~methods~~ methodologies to predict the susceptibility of specific ~~parts of~~areas prone to landslides. ~~The results demonstrated that~~Compared to well-established models such as XGBoost and Light GBM, the proposed model ~~with the dice cross entropy (DCE) loss function and either XGBoost or Light GBM achieved more balanced and precise~~showcased superior performance in accurately assessing landslide susceptibility~~ assessments compared to existing models. Next,~~. Furthermore, Sahin (2022) ~~compared four recent gradient boosting~~developed multiple ML models, including XGBoost, CatBoost, Gradient Boosting Machine (GBM), ~~Categorical Boosting (CatBoost), Extreme Gradient Boosting (XGBoost), and~~ and Light GBM, ~~for modeling landslide~~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, Light GBM, 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 ~~prediction~~ performance of ~~ensemble methods~~XGBoost, CatBoost, GBM, and Light GBM was compared to RF. The results ~~showed~~revealed that CatBoost had the highest prediction capability, followed by XGBoost, Light GBM, and GBM.~~ RF had the lowest prediction capability.~~ 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 networks models (MLP and LSTM), and gradient boosting ML models (e.g., AdaBoost, XGBoost, Light GBM, 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 RF, AdaBoost, XGBoost, Light GBM, CatBoost, MLP, LSTM, and an ensemble of RF, AdaBoost, XGBoost, Light GBM, and CatBoost models 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 Light GBM have demonstrated their ability to achieve balanced and precise predictions, especially in earthquake-induced landslide susceptibility assessments. Among gradient-boosting models, CatBoost ~~has the highest~~stands out for its superior prediction capability~~ among gradient-boosting models, ~~, making it a ~~suitable choice~~well-suited option for ~~modeling~~modelling landslide susceptibility. ~~MLP as the base model, has provided more accurate predictions of~~On the other hand, when it comes to predicting root tensile strength~~. Finally~~, MLP has demonstrated higher accuracy in its predictions. Additionally, LSTM~~ is,~~ a robust recurrent neural network architecture~~ that captures~~, is particularly effective in capturing temporal dependencies and long-term patterns in sequential data. ~~These~~Collectively, these models ~~collectively~~ offer a ~~range~~diverse set of capabilities~~, making them~~ that prove valuable ~~tools for~~ in the prediction of soil movement~~ prediction~~.

3. Data Collection and Description

The ~~data used~~dataset for predicting soil movement ~~prediction~~ was collected from ~~a real-world~~an actual landslide ~~location~~site in Uttarakhand, India~~, over~~. The monitored landslides are characterized as shallow landslides with debris flow, occurring at elevations ranging from 1450 m 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 ~~period~~sand–silt–clay matrix. Additional context includes an annual rainfall of 4190 mm in the area, as reported by Gupta et al. (2015). Spanning a duration of two years, from June 2019 to June 2021~~. The monitoring was performed using a low-~~, this dataset holds valuable insights into the behaviour of soil in response to various environmental factors. To gather this data, a cost~~ IoT based~~-effective landslide monitoring station (LMS) ~~specifically installed~~was carefully deployed at the ~~site. The LMS was equipped~~landslide. Equipped with ~~various~~a range of sensors~~ to measure different~~, the LMS diligently recorded critical weather and soil-related parameters. ~~In terms of weather parameters, the LMS recorded the~~ Weather-wise, it diligently captured temperature readings in degrees Celsius, humidity ~~in~~levels as a percentage, rainfall ~~(Rain)~~measurements in inches per hour (in/hr), atmospheric pressure in millibars (mb), and even sunlight ~~(Light)~~intensity in lux. ~~These measurements provided valuable information about~~These meticulous recordings shed light on the prevailing weather conditions experienced at the precise location of the landslide~~ location. To monitor the soil conditions, the~~. The LMS ~~employed~~relied on an accelerometer sensor to ~~measure the soil~~ monitor the soil conditions with utmost precision. An advanced sensor was utilized to measure the acceleration ~~(in m/s^2)~~of the soil in three directions: Ax, Ay, and Az~~.~~ (in m/s^2). This provided valuable insights into the soil's movement and stability. Additionally, a gyroscope sensor was ~~utilized~~employed to capture the angular rotation of the soil along the Wx, Wy, and Wz axes (in degrees per second~~, represented by Wx, Wy, and Wz. These measurements helped in understanding the movement and stability of the soil. The LMS's~~). This sensor enhanced the understanding of the soil's behaviour by accurately detecting its angular movements. Furthermore, the LMS was equipped with a capacitive soil moisture sensor~~ measures~~, enabling it to measure the ~~soil's~~ volumetric moisture content ~~(Mos)~~of the soil in ~~percent~~percentage. The LMS transmitted all these twelve attributes, including weather parameters, soil g-force, angular rotation, and soil moisture content, to the cloud every ten minutes. The dataset obtained from the LMS consisted of approximately thirty-nine thousand data points, covering a wide range of environmental and soil-related attributes. Table 1 ~~is showcasing~~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 (~~Std. Dev.),~~stdev), indicating the variability of the data. The minimum and maximum values highlight the range of measurements observed, offering insights into the extreme values and overall data distribution.

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

Parameter	Mean	~~Std. Dev.~~stdev	Min	Max
Temperature ($^\circ C$)	16.18	10.48	0.00	39.00
Humidity (%)	66.69	35.46	0.00	99.00
Rain (in/hr)	0.00	5.60	0.00	15.00
Pressure (mb)	1040.96	27.96	921.61	1065.41
Light (lux)	5025.35	10154.75	0.00	54612.00
Ax (m/s^2)	0.02	1.23	-28.02	40.25
Ay (m/s^2)	0.00	1.37	-100.08	100.08
Az (m/s^2)	0.00	2.28	-149.61	315.61
Wx ($^\circ/s$)	0.00	15.86	-1994.51	1997.24
Wy ($^\circ/s$)	0.00	15.85	-1998.05	1998.73
Wz ($^\circ/s$)	0.00	6.95	-932.00	932.00
Moisture (%)	80.00	20.30	40.00	100.00

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 ~~included three ~~contained values for acceleration ~~due to gravity in the x, y, and z directions and three values for ~~and angular rotation in ~~degree per second in the~~three directions: x, y, and z~~ directions~~. The ~~change~~changes in acceleration and angular rotation ~~was derived by calculating the difference between~~were calculated by subtracting the current ~~and ~~values from the past values~~. The ~~, allowing for the assessment of movement. Four categories were defined to classify the movement data~~ were categorized into four classes (~~: no movement, ~~small~~low movement, moderate movement, and ~~large~~high movement~~) using~~. These categories were determined based on standard deviation thresholds ~~calculated for ~~derived from the acceleration and angular rotation values. Specifically, values within ± 1 standard deviation from the mean ~~of acceleration and angular rotation were classified ~~were categorized as no movement, ± 2 standard deviations as ~~small~~low movement, ± 3 standard deviations as moderate movement, and values exceeding ± 3 standard deviations as ~~large~~high movement. This classification approach ~~enabled the determination of movement intensity based on~~ considered the variability ~~of~~in acceleration and angular rotation changes to determine the ~~change in acceleration and angular rotation. In our~~ intensity of movement.

During the analysis, ~~we categorized~~ each timestamp was assigned to a movement class based on the class ~~that corresponds to~~associated with the ~~maximum~~highest standard deviation ~~of~~observed in any acceleration or ~~velocity~~angular rotation element. ~~This means that if any~~If an individual element ~~among the acceleration and velocity measurements~~ had the highest standard deviation at a ~~particular~~specific timestamp, ~~we assigned~~ that timestamp was assigned to the corresponding movement class ~~associated~~ with ~~that~~the maximum standard deviation.

Table 2 ~~provides~~presents the distribution of movement intensity within the dataset~~. The dataset consists of a total~~, which consisted of 38,900 data points. ~~It~~The table shows the percentage distribution of ~~different~~ movement categories~~, including~~: high~~ intensity, medium intensity~~, moderate, low~~ intensity~~, and no movement. ~~Most~~The majority of the dataset (97.8%) falls under the "No Movement" category, ~~while the~~ indicating a lack of significant movement. On the other hand, the high movement category represents only a small fraction (1.1~~%). The~~ %) 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 ~~of movement~~ highlights the class imbalance issue present in the dataset, which needs to be ~~addressed~~taken into account when ~~building~~developing a classification model for predicting soil movement.

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	38063	97.8%

4.3. Sliding Window Packets

The sliding window packets technique involves ~~creating~~dividing a given dataset into fixed-length subsequences or packets ~~from a given dataset~~ and their corresponding labels. ~~The~~ To achieve this, a sequence length parameter ~~determines~~is used to determine the length of each subsequence. ~~To generate these subsequences, the~~ The sliding window approach is ~~applied. The sliding~~then employed, where a window starts at the beginning of the dataset and moves ~~across~~through the data with a step size of 1. ~~At each position of the window, a~~A subsequence of the specified length is extracted from the dataset~~, and the~~ at each window position. The label for prediction is taken from the next position after the window.~~ This means that the sliding window packets technique is typically used for predicting future values or events based on the preceding subsequence.~~

The sliding window packets technique aims to predict future values or events based on preceding subsequences. For instance, if the sequence length is set to five, 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 ~~6th~~sixth position. This process continues until the end of the dataset is reached, resulting in multiple subsequences and their ~~corresponding~~respective labels.

~~After creating~~ Once the packets are created, they are flattened to form a single feature vector. For ~~example~~instance, if the sequence length is five and the dataset has twelve features, each packet will contain sixty ~~features~~elements (5x12). This transformation allows the packets to be treated as individual samples with multiple features suitable for ML models.

The primary ~~objective~~purpose of creating these packets is to address ~~sequence-based~~ prediction tasks~~,~~ involving sequences where the input data's order and dependencies ~~of the input data play a~~are crucial~~ role~~. The model can effectively capture and learn patterns and relationships within the sequential data by utilizing the sliding window packets. The flattened packets generated ~~through~~using the sliding window technique ~~were used as~~are inputs in oversampling techniques.

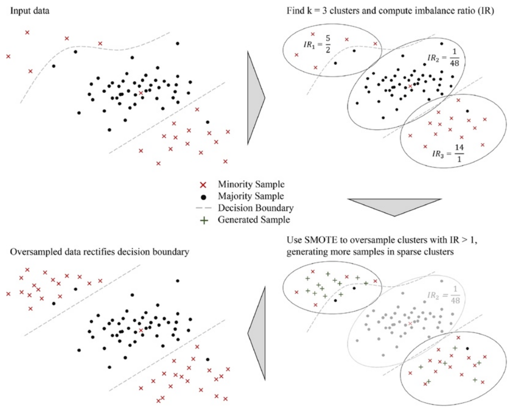
4.4. Oversampling

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. ~~On the~~All other ~~hand, the~~classes, including "High Movement~~" class, which represents the~~," "Moderate Movement," and "Low Movement," represent minority ~~class, had~~classes, each constituting only ~~a minimal representation with just~~ 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.

~~We employed various~~To overcome the class imbalance challenge, we implemented several oversampling techniques~~ to address this issue, focusing on the~~, with a particular focus on SMOTE and its extensions (Chawla et al., 2002; Douzas et al., 2018; ~~Tang et al., 2008;~~ Han et al., 2005; He et al., 2008). SMOTE~~ creates~~, which stands for Synthetic Minority Oversampling Technique, addresses the imbalance by generating synthetic data points for the minority class ~~to balance its representation~~ (Chawla et al., 2002). By ~~generating new data points using~~utilizing the characteristics of existing ~~minority class~~ samples~~, we increased the number of instances in~~from the ~~"High Movement" class. Additionally~~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 ~~of SMOTE, including KMeans~~such as K-Means SMOTE (Douzas et al., 2018), ~~SVM SMOTE (Tang et al., 2008),~~ and Borderline SMOTE (Han et al., 2005) to further ~~improve~~enhance the balance of ~~the~~ class distribution.

Furthermore, we utilized the 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.

~~Fig.~~Figure 1~~.~~ illustrates the application of the K-Mean SMOTE technique for addressing the class imbalance. The ~~figure~~Fig. 1 depicts a scatter plot 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-Mean SMOTE algorithm. The dashed line represents the decision boundary separating the two classes. K-Mean SMOTE operates by following two simple steps iteratively ~~[8].~~(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~~centre between the assigned observations. The ~~information~~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 value of k~~The parameter 'k' in all SMOTE techniques was ~~selected 4~~varied from 2 to 5 in this experiment to observe how different numbers of nearest neighbors impact the diversity and quality of synthetic samples created, thereby affecting the performance of the model on imbalanced data.



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

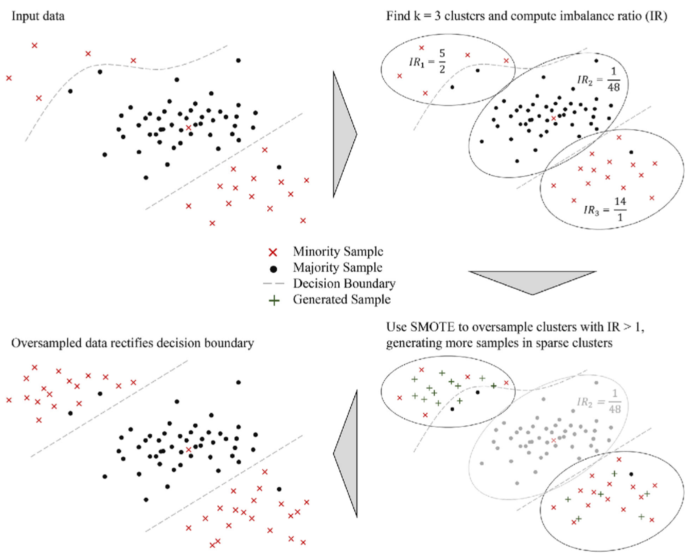


Figure 1: K-Means SMOTE effectively addresses within-class imbalance by oversampling safe areas (Douzas et al., 2018).

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~~, also known as Adaptive Boosting, is a probabilistic classification meta-~~ enhances ML model ~~designed to enhance the~~ performance ~~of ML models (Wu et al., 2020). It achieves this~~ by combining ~~the~~ results ~~of~~from multiple weak learners, ~~which are learning~~ techniques ~~with~~ slightly better than random guessing (Wu et al., 2020). ~~capabilities. Through an adaptive process, AdaBoost adjusts subsequent weak models to focus on the cases that were misclassified by earlier models. This adaptive nature helps improve the overall accuracy of the model and reduces the risk of overfitting in certain situations. Although individual weak learners may not perform well on their own, their collective strength allows the final model to converge as a powerful learner capable of making more accurate predictions.~~
~~For~~In the AdaBoost model, the number of ~~estimators determines~~trees sets the maximum ~~number of weak models to be combined. Increasing the number of estimators can improve the model's~~ weak models, impacting

performance ~~but may also increase the risk of~~and overfitting. The learning rate ~~controls the~~ influences each model's contribution~~ of each weak model~~, with a higher ~~learning~~ rate giving more weight~~ to each model.~~. The maximum depth parameter ~~limits the depth of individual~~prevents weak models~~, preventing them~~ from becoming too complex~~ and overfitting the data.~~. Table 3 ~~shows the~~details the AdaBoost model's parameter range~~ of hyperparameters for the AdaBoost model~~.

4.5.2. XGBoost

XGBoost~~ is an~~ , a gradient-boosting ensemble ML model ~~based on gradient boosting and uses~~ with decision trees (Chen and Guestrin, 2016~~). While deep neural networks excel~~), excels in ~~predicting unstructured data such as images and text, decision tree-based methods are considered superior for dealing with~~ structured data~~. Decision trees are particularly effective when the data has a well-defined structure or specific features, making them a preferred choice for certain prediction problems involving structured information.~~

~~In the XGBoost model, the~~ handling. The number of ~~estimators~~ trees in XGBoost determines ~~the number of~~ boosting rounds~~ or iterations. Increasing the number of estimators can improve the model's~~ , impacting performance ~~but also increases~~ with a computational complexity trade-off. The learning rate ~~controls the step size during the boosting process and affects the model's~~ influences convergence speed and generalization ability~~. The~~, and the maximum depth parameter ~~restricts the depth of the decision trees in the ensemble, preventing~~prevents overfitting ~~and promoting~~for enhanced interpretability. See Table 3 ~~shows the range of hyperparameters~~ for the XGBoost ~~model~~model's parameter range.

4.5.3. Light GBM

Light GBM~~ is~~, a gradient-boosting framework ~~that utilizes a decision-branching technique for various ML~~ for tasks ~~such as~~like ranking and classification (Ke et al., 2017~~). Unlike other boosting methods that divide the tree lengthwise or layerwise, Light GBM employs a~~), stands out with its leaf-wise approach, ~~where the tree is divided leaf by leaf, selecting the best split at each step. This leaf-wise strategy reduces~~reducing loss~~ and improves accuracy compared to traditional boosting techniques. Additionally, Light GBM is known for its speed and efficiency, earning its name "Light" due to its fast execution.~~

~~In the Light GBM model, the number of estimators determines the number of~~, improving accuracy, and ensuring efficient learning. The number of trees in the model influences boosting rounds~~, with a higher number potentially improving the model's~~ for potential performance enhancement. The learning rate ~~controls the step size during boosting and affects the trade-off between~~parameter balances convergence speed and ~~model~~ accuracy~~. The~~, while the maximum depth parameter ~~limits the depth of the decision trees, controlling the model's~~ controls complexity and ~~the risk of~~prevents overfitting. See Table 3 ~~shows the~~for the Light GBM model's parameter range~~ of hyperparameters for the Light GBM model~~.

4.5.4. CatBoost

CatBoost, short for Category Boosting, is ~~a~~an ML model developed by Yandex and ~~recently made available~~released as an open-source tool (Prokhorenkova et al., 2018). ~~It is designed to be easily integrated with other popular frameworks like TensorFlow and Core ML, expanding its compatibility across different platforms. CatBoost is known for its versatility and ability to handle diverse datasets, making it suitable for solving various industry problems. In addition, it boasts industry-leading performance, delivering high-quality results and predictive accuracy.~~

In the CatBoost model, the choice of the loss function ~~can~~ significantly ~~impact the model's~~impacts performance. ~~Different loss~~Loss functions~~, such as~~ like log, entropy, or hinge~~, handle various~~ are tailored for specific classification problems~~ and may yield different~~, influencing results. Table 3 ~~shows~~outlines the range of

~~hyperparameters~~parameters for the CatBoost model for fine-tuning and optimizing CatBoost's performance on a given dataset.

4.5.5. Random Forest

~~The regression or classification model was constructed using the~~ RF ~~technique~~, an ensemble learning method combining predictions from multiple decision trees (Breiman, 2001~~). This approach offers several advantages. It is commonly employed in scenarios involving various regression or classification tasks, as it does not require the assumption of variable independence in the data distribution. Using a combination of categorical and numerical variables, RF can account~~), constructs regression or classification models. Known for handling relationships and non-linearities without ~~the need for dummy variables. The RF model has been widely used across~~ requiring variable independence assumptions, RF excels in various industries ~~and has demonstrated exceptional performance. It has also shown promise in~~, including landslide prediction and site recognition ~~applications.~~
~~The RF technique utilizes several~~. Optimizing RF performance involves adjusting parameters ~~to optimize its performance. The first parameter is~~like the number of ~~estimators, which determines the number of decision~~ trees ~~in the ensemble. The second parameter is the criteria for~~(DTs), splitting ~~nodes~~criteria (Gini ~~and~~or Entropy~~) in the decision trees. Gini measures node impurity based on the probability of misclassifying a randomly chosen element, while Entropy measures information gain at each split. The third parameter is the~~), and maximum tree depth ~~of the decision trees. It limits the number of levels a tree can grow.~~, controlling robustness, accuracy, and complexity. Table 3 ~~shows the range of hyperparameters~~details parameter ranges for the RF model.

4.5.6. Multilayer Perceptron

The MLP ~~is~~, a ~~popular~~ neural network architecture ~~for classification (~~introduced by Rosenblatt~~,~~ in 1961~~). It consists of~~ , features interconnected layers~~, including an~~: input ~~layer~~, hidden ~~layers~~, and ~~an~~ output ~~layer.~~(Rosenblatt et al., 1961). Neurons ~~in the MLP compute a~~calculate weighted ~~sum of inputs and pass it~~sums, passing through ~~an~~ activation ~~function~~functions to ~~learn complex~~capture intricate relationships. ~~Hidden layers extract informative features from the input data.~~ Dropout layers ~~are used to~~ prevent overfitting by ~~randomly~~ deactivating neurons randomly during training~~. This regularization technique improves the MLP's~~ , enhancing generalization ~~ability and reduces reliance on specific patterns. The MLP is a versatile and effective approach~~. Versatile for ~~solving~~ classification ~~problems.~~
~~In~~, the ~~MLP model, the~~MLP's look-back period ~~determines the number of previous time steps considered for prediction, affecting the model's ability to capture temporal dependencies. The number of~~ influences temporal dependency capture, while the number of layers and nodes per layer ~~control the network's~~governs complexity ~~and capacity to learn complex representations from the data.~~. Table 3 ~~shows the range of hyperparameters~~outlines parameter ranges for the MLP model.

4.5.7. LSTM

The LSTM is a recurrent neural network ~~architecture designed to analyze~~ that captures long-term dependencies in sequential data (Hochreiter and Schmidhuber, 1997). ~~It overcomes the vanishing gradient problem and~~It excels ~~at capturing long-term dependencies in sequences. LSTM networks use memory cells and gating mechanisms to selectively retain or forget information over time, allowing them to process and predict sequences of varying lengths. They have been successfully applied~~ in various ~~domains~~applications, including natural language processing~~, speech recognition,~~ and time series forecasting.
In our LSTM model, ~~we conducted~~ experiments ~~to analyze the effects of various parameters on its performance. We investigated the impact of different~~explored different parameters: LSTM unit sizes (32, 64, 128, ~~and~~ 256~~) to understand how the model's complexity influences its ability to capture patterns in the data. Additionally, we explored different~~), activation functions (sigmoid, tanh, ~~and~~ ReLU~~) to assess their impact on the model's~~

~~capability to learn complex relationships within the dataset. The~~), and a look-back period ranging from 3 to 10. We chose the categorical cross-entropy loss function ~~was chosen for its suitability in~~ for multi-class classification~~ tasks. Furthermore, we varied the look-back period from 3 to 10 to evaluate how it affects the model's ability to capture temporal dependencies.~~. Table 3 ~~shows~~details the parameter range ~~of hyperparameters~~ for the LSTM model.

4.5.8. Dynamic Ensembling

Dynamic ensembling is a ~~powerful~~highly effective technique in ML that ~~leverages~~ takes advantage of the adaptability and ~~continuous~~ongoing improvement of predictive models (Ko et al., 2008). ~~Dynamic ensembling creates~~It involves creating a ~~flexible~~versatile and continuously evolving ensemble by ~~combining~~harnessing the strengths of multiple models, ~~such as~~including RF, CatBoost, XGBoost, Light GBM, 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 concept drift and maintain accurate predictions.

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

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

Model	~~Hyperparameter~~Parameter	Range of ~~Hyperparameter~~Parameter
AdaBoost	Number of ~~Estimators~~Trees	[~~15, 200~~10, 100] in steps of ~~10~~5
	Learning Rate	[0.1, 2] in steps of 0.1
	~~Maximum Depth~~	~~[3, 33] in steps of 3~~
XGBoost	Number of ~~Estimators~~Trees	[~~50, 1100~~10, 100] in steps of ~~50~~5
	Learning Rate	[0.05, 0.55] in steps of 0.05
	Maximum Depth	[~~3, 33~~5, 50] in steps of ~~3~~5
Light GBM	Number of ~~Estimators~~Trees	[~~20, 400~~10, 100] in steps of ~~20~~5
	Learning Rate	[0.05, 0.55] in steps of 0.05
	~~Max~~Maximum Depth	[~~3, 33~~5, 50] in steps of ~~3~~5
CatBoost	Loss Function	Log, Entropy, Hinge
	Learning Rate	[0.1, 2] in steps of 0.1
	~~Max~~Maximum Depth	[3, 33] in steps of 3
RF	Number of ~~Estimators~~Trees	[~~1~~10, 100] in steps of 5
	Criteria	Gini, Entropy
	Maximum Depth	[~~1, 100~~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 step 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 step of 0.1

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 model was trained using grid search techniques, which systematically explored different combinations of hyperparameters to optimize its performance. The training phase utilized the labelled training data, split into~~ The dataset was partitioned into a 70:30 ratio, with 70% allocated for training and 30% for testing ~~purposes. One challenge encountered during the training process was~~ . To tackle the class imbalance issue~~. The number of samples available for the minority class was insufficient compared to the majority class. To address this,~~ 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 method 5 Training Datasets (5-TD) and method 5-fold cross-validation (5-CV). Method 5-TD was employed~~. By generating synthetic data points~~ for ~~the minority class, we were able to balance the dataset and mitigate the bias toward the majority class. Once the~~ parameter variation analysis across different datasets. On the other hand, method 5-CV was utilized for conducting 5-fold cross-validation to analyze the performance of the ML models.

5.1. Method 5-TD

For method 5-TD, the training dataset was ~~balanced, multiple ML models were trained using the training data. The primary objective was to optimize the models' parameters for improved performance. Each model was subjected to the grid search technique, systematically exploring~~split into five training datasets, each utilized for parameter variation analysis. This involved training and optimizing the ML model on each dataset independently using the grid search method. Since each dataset possessed different ~~parameter combinations to identify the~~ optimal ~~settings. After training~~parameters, we calculated the ~~models with~~mean and stdev of the ML-optimized ~~parameters, they were tested on separate test data~~parameter values across all datasets to assess ~~their predictive capability. The evaluation process~~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 stdev implied that the model maintained consistency across each dataset and demonstrated robust generalization capabilities. Conversely, a higher stdev suggested that the model encountered difficulties maintaining consistency across datasets, potentially hindering its ability to learn general patterns effectively. The evaluation primarily focused on ~~measuring accuracy~~F1 score metrics to determine ~~the models' effectiveness in predicting~~how effectively the models predicted the intensity of soil movements in each of the 5 datasets.

5.2. Method 5-CV

For method 5-CV, 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 was split into 5 datasets, where each dataset was alternately used for validation while the others were used for training. The models were optimized by employing grid search methodology and optimized based on performance on the 5 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 5 validation sets and the test set.

6. Results

6.1. Parameter Analysis Result

Upon scrutinizing the parameter analysis presented in Table ~~3 showcases the~~ 4 from method 5-TD, a discernible trend emerged: models trained with oversampling techniques exhibit notably smaller stdevs than their counterparts trained without oversampling. For instance, when examining the AdaBoost model, we observe that the stdev of the number of trees parameter 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 stdev of the number of trees parameter was 0 with oversampling, indicating consistent parameter values across folds. Conversely, for the dataset without oversampling, the stdev 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 behaviour, particularly in scenarios involving imbalanced datasets.

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

Model	Parameter	With Oversampling		Without Oversampling	
		Mean	stdev	Mean	stdev
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
Light GBM	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

6.2. Optimized Parameters

In method 5-CV, we optimized ~~hyperparameter values determined through~~ 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 four. Table 5 presents each model's optimized parameter values obtained through the grid search ~~method.~~in 5-CV on the training dataset. These ~~hyperparameters~~parameters were carefully fine-tuned to ~~find~~ensure the best fit for the given data. In the ~~XGBoost~~case of AdaBoost, the optimized values included ~~800 for the number of estimators, 0.3 for the~~ 80 trees and a learning rate~~, and 9 for the~~ 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 ~~parameter~~ settings were ~~selected after careful evaluation~~ determined to enhance the model's ~~ability to generalize and capture complex patterns in the data. In the Light GBM model, the optimized values consisted of 220 for the number of estimators, 0.25 for the number of hidden layers, and 12 for the maximum depth. These settings were determined to improve the model's performance in terms of~~ performance in terms of both speed and accuracy.

Similarly, the ~~AdaBoost~~Light GBM model ~~had its hyperparameters~~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, ~~resulting in the~~ leading to entropy selection ~~of 25 for the number of estimators, 1.7 for the learning rate, and 20 for the maximum depth. These parameter values were chosen to enhance the model's adaptability and accuracy in classification tasks. The CatBoost model also underwent optimization, resulting in the selection of entropy~~ as the loss function, ~~0.9 as the~~a learning rate of 0.8, 50 trees, and ~~20 as the~~a maximum depth~~. These settings~~ of 20. In the RF model, the optimized values were ~~chosen to maximize~~80 for the ~~model's performance in terms of accuracy~~number of trees and ~~robustness. Similarly~~20 for the maximum depth, and the evaluation criteria were set to "Gini." Likewise, the MLP model ~~had its hyperparameters~~ optimized its parameters with a look-back period of 3, 2 layers, and 200 nodes per layer. ~~These settings were selected to enhance the model's ability to capture complex relationships and improve classification accuracy.~~ Similarly, ~~LSTM has 128 units~~the LSTM model consists of two layers with 100 and ~~tanh~~200 nodes in the first and second layers and utilizes a ReLU activation function. ~~In the RF model, the optimized values were 45 for the number of estimators, 25 for the maximum depth, and the evaluation criteria were set to "entropy." These values were chosen to maximize the model's accuracy and predictive power performance.~~ Lastly, the dynamic ~~ensembling~~ensemble model in this study incorporated the optimized RF, CatBoost, XGBoost, Light GBM, 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 ensembling model aimed to improve the accuracy and reliability of landslide analysis predictions.

Table ~~4~~5. The best value of the ~~hyperparameters~~parameters was calibrated from the training data using method 5-CV.

Model	~~Hyperparameter~~Parameter	Best Value of ~~Hyperparameter~~Parameter
AdaBoost	Number of ~~Estimators~~Trees	~~25~~80
	Learning Rate	~~1.7~~0.6
	~~Maximum Depth~~	~~20~~
XGBoost	Number of ~~Estimators~~Trees	~~800~~50
	Learning Rate	0.3

	Maximum Depth	~~9~~10
Light GBM	Number of ~~Estimators~~Trees	~~220~~50
	Learning Rate	0.~~25~~5
	~~Max~~Maximum Depth	~~12~~20
CatBoost	Loss Function	Entropy
	Learning Rate	0.~~9~~8
	Number of Trees	50
	~~Max~~Maximum Depth	20
RF	Number of ~~Estimators~~Trees	~~45~~ 80
	Criteria	~~Entropy~~Gini
	Maximum Depth	~~25~~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	~~128~~100 in first and 200 in second layer
	Activation Function	~~tanh~~ReLU
	Learning Rate	0.9

6.2.1. Train-Test Results

Table ~~5~~6 presents the training results of different classification models ~~combined with~~ evaluated using 5-fold cross-validation on the training dataset and various oversampling techniques for landslide prediction~~.~~, utilizing method 5-CV. 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 ~~RF~~XGBoost model with K-Mean SMOTE ~~emerges~~emerged as the best model in training, achieving outstanding accuracy, precision, recall, and F1~~ score of 100% and 1, respectively.~~ scores of 0.999, 0.999, 0.999, and 0.999, respectively. The dynamic ensemble model with K-Mean SMOTE and Borderline SMOTE techniques also performed similarly with 0.998 F1 scores. It demonstrates remarkable predictive capability by achieving perfect accuracy in ~~both oversampling and non-~~oversampling scenarios. When the XGBoost model was trained without oversampling, its accuracy, precision, recall, and F1 score were notably lower, with values of 0.999, 0.999, 0.971, and 0.985, respectively.

Table ~~6 showcases~~7 presents the test results of various classification models combined with different oversampling techniques for landslide prediction~~. The dynamic ensemble model with~~ (here models were trained using the method 5-CV). 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-Mean SMOTE technique ~~exhibits~~demonstrated exceptional performance in accurately predicting landslides on unseen data. It achieves impressive accuracy, precision, and recall ~~rate~~rates of ~~99.68% each~~0.995, 0.995, and 0.995, respectively, along with an F1~~-~~ score of 0.~~9968~~995. These outstanding results ~~reaffirm~~confirm the effectiveness of the dynamic ensemble approach ~~in combination~~when combined with K-Mean SMOTE for accurate ~~landslide~~soil movement prediction. ~~The best performed~~ 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 ~~shown~~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 Table 6 and Table 7.

~~Furthermore, the RF model with K-Mean SMOTE stands as the second-best model in the test phase, delivering a high accuracy, precision, and recall rate of 99.64% each and an F1-score of 0.9964. This highlights the reliability and robustness of the RF model in landslide prediction tasks.~~

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 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 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 Dynamic Ensemble, exhibit signs of overfitting, as evidenced by significantly higher performance metrics on 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.

Comparing the dynamic ensemble ~~and RF models~~model with~~ the~~ other classification models~~ and oversampling techniques~~, it becomes evident that the dynamic ensemble model with K-Mean SMOTE ~~and the RF model with K-Mean SMOTE~~ consistently ~~outperform~~outperformed the rest, ~~showcasing~~highlighting their effectiveness in ~~accurate landslide prediction~~accurately predicting landslides.

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

~~**Table 5.** The results of the ML models from training dataset.~~

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 without oversampling technique, 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 Light GBM, CatBoost, RF, MLP, LSTM, and Dynamic Ensemble, where K-Means SMOTE consistently demonstrated superior performance in terms of F1 score compared to 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 on both the training and testing datasets, utilizing the K-Mean 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.

Table 6. Results of ML models obtained from the training dataset using 5-fold cross-validation in method 5-CV.

Model	Oversampling Technique	Accuracy ~~(in %)~~					Precision ~~(in %)~~	Recall ~~(in %)~~	~~F1Score~~ F1 Score
		C0	C1	C2	C3	Overall			
AdaBoost	SMOTE	~~71.45~~0.942	~~71.41~~0.562	~~71.45~~0.640	0.~~7142~~817	0.747	0.748	0.747	0.747
	K-Means SMOTE	~~71.13~~0.948	~~71.07~~0.760	~~71.13~~0.675	0.~~7105~~855	0.807	0.809	0.807	0.806
	Borderline SMOTE	~~77.54~~0.919	~~77.48~~0.565	~~77.54~~0.667	0.~~7748~~815	0.740	0.741	0.740	0.740
	ADASYN	~~72.97~~0.934	~~72.91~~0.552	~~72.97~~0.649	0.~~7276~~798	0.740	0.741	0.740	0.740
	Without Oversampling	~~97.87~~0.995	~~97.73~~0.250	~~97.87~~0.243	0.~~9779~~341	0.980	0.575	0.465	0.506
~~CatBoost~~**XGBoost**	SMOTE	~~99.85~~0.995	~~99.85~~0.999	~~99.85~~0.999	0.~~9985~~997	0.998	0.998	0.998	0.998
	K-Means SMOTE	~~99.86~~**0.997**	~~99.86~~**0.999**	~~99.86~~**0.999**	**0.**~~9986~~**998**	**0.999**	**0.999**	**0.999**	**0.999**
	Borderline SMOTE	~~99.78~~0.996	~~99.78~~0.999	~~99.78~~0.999	0.~~9978~~998	0.998	0.998	0.998	0.998
	ADASYN	~~99.85~~0.994	~~99.85~~0.999	~~99.85~~0.999	0.~~9985~~997	0.998	0.998	0.998	0.998
	Without Oversampling	~~99.72~~1.000	~~99.72~~0.995	~~99.72~~0.953	0.~~9971~~906	0.999	0.999	0.971	0.985
LightGBM~~XGBoost~~	SMOTE	~~99.97~~0.984	~~99.97~~0.994	~~99.97~~0.999	0.~~9997~~988	0.991	0.991	0.991	0.991
	K-Means SMOTE	~~99.98~~0.991	~~99.98~~0.998	~~99.98~~0.998	0.~~9998~~996	0.996	0.996	0.996	0.996
	Borderline SMOTE	~~99.98~~0.985	~~99.98~~0.999	~~99.98~~0.999	0.~~9998~~995	0.995	0.995	0.995	0.995
	ADASYN	~~99.97~~0.983	~~99.97~~0.994	~~99.97~~0.998	0.~~9997~~987	0.991	0.991	0.991	0.991
	Without Oversampling	~~99.99~~1.000	~~99.99~~1.000	~~99.99~~1.000	0.~~9999~~976	0.994	0.999	0.999	0.996
CatBoost~~LightGBM~~	SMOTE	~~99.87~~0.990	~~99.87~~0.999	~~99.87~~0.999	0.~~9987~~997	0.997	0.997	0.997	0.997
	K-Means SMOTE	~~99.91~~0.991	~~99.91~~0.999	~~99.91~~0.999	0.~~9991~~997	0.997	0.997	0.997	0.997
	Borderline SMOTE	~~99.97~~0.992	~~99.97~~0.999	~~99.97~~0.999	0.~~9997~~997	0.997	0.997	0.997	0.997

Inserted Cells
Inserted Cells
Inserted Cells
Inserted Cells

	ADASYN	~~99.88~~0.991	~~99.88~~0.999	~~99.88~~0.999	0.~~9988~~997	0.996	0.996	0.996	0.996
	Without Oversampling	~~99.87~~0.999	~~99.87~~0.924	~~99.87~~0.916	0.~~9987~~735	0.997	0.997	0.903	0.946
RF	SMOTE	~~100~~0.920	~~100~~0.892	~~100~~0.951	~~1~~0.905	0.921	0.923	0.921	0.922
	K-Means SMOTE	~~100~~0.920	~~100~~0.921	~~100~~0.959	~~1~~0.902	0.925	0.928	0.925	0.926
	Borderline SMOTE	~~100~~0.948	~~100~~0.969	~~100~~0.988	~~1~~0.959	0.967	0.967	0.967	0.967
	ADASYN	~~100~~0.921	~~100~~0.898	~~100~~0.945	~~1~~0.899	0.915	0.917	0.915	0.915
	Without Oversampling	~~100~~1.000	~~100~~0.701	~~100~~0.682	~~1~~0.537	0.992	0.995	0.742	0.841
MLP	SMOTE	~~90.32~~0.959	~~90.49~~0.976	~~90.32~~0.997	0.~~9028~~952	0.961	0.961	0.961	0.961
	K-Means SMOTE	~~60.16~~0.940	~~73.24~~0.996	~~60.16~~0.984	0.~~5785~~957	0.974	0.974	0.974	0.974
	Borderline SMOTE	~~95.66~~0.968	~~95.72~~0.974	~~95.66~~0.989	0.~~9566~~913	0.964	0.964	0.964	0.964
	ADASYN	~~88.54~~0.929	~~88.81~~0.975	~~88.54~~0.981	0.~~8854~~984	0.961	0.961	0.961	0.961
	Without Oversampling	~~97.81~~0.997	~~96.43~~0.016	~~97.81~~0.000	0.~~9694~~056	0.980	0.693	0.336	0.381
LSTM	SMOTE	~~68.73~~0.882	~~68.81~~0.841	~~68.73~~0.881	0.~~6799~~896	0.875	0.884	0.875	0.877
	K-Means SMOTE	~~79.51~~0.980	~~79.89~~0.996	~~79.51~~0.992	0.~~7959~~968	0.984	0.984	0.984	0.984
	Borderline SMOTE	~~86.40~~0.946	~~86.66~~0.954	~~86.40~~0.997	0.~~8627~~965	0.966	0.966	0.966	0.966
	ADASYN	~~77.47~~0.955	~~77.72~~0.979	~~77.47~~0.997	0.~~7736~~955	0.971	0.971	0.971	0.971
	Without Oversampling	~~97.88~~0.999	~~95.80~~0.859	~~97.87~~0.925	0.~~9683~~700	0.995	0.979	0.871	0.919
Dynamic Ensemble	SMOTE	~~99.94~~0.992	~~99.94~~0.999	~~99.94~~0.999	0.~~9994~~999	0.997	0.997	0.997	0.997
	K-Means SMOTE	~~99.96~~0.994	~~99.96~~0.999	~~99.96~~0.999	0.~~9996~~999	0.998	0.998	0.998	0.998
	Borderline SMOTE	~~99.97~~0.997	~~99.97~~0.999	~~99.97~~0.999	0.~~9997~~998	0.998	0.998	0.998	0.998
	ADASYN	~~99.93~~0.992	~~99.93~~0.999	~~99.93~~0.999	0.~~9993~~998	0.997	0.997	0.997	0.997
	Without Oversampling	~~99.99~~1.000	~~99.99~~0.951	~~99.99~~0.944	0.~~9999~~770	0.997	0.999	0.916	0.954

Table ~~6. The results~~7. Results of ~~the~~ ML models obtained from ~~test~~the testing dataset in method 5-CV.

Model	Oversampling Technique	Accuracy ~~(in %)~~					Precision ~~(in %)~~	Recall ~~(in %)~~	F1-Score
		C0	C1	C2	C3	Overall			
AdaBoost	SMOTE	~~71.57~~0.939	~~71.54~~0.548	~~71.57~~0.436	0.~~7155~~763	0.932	0.383	0.671	0.442
	K-Means SMOTE	~~71.03~~0.946	~~70.89~~0.583	~~71.03~~0.436	0.~~7091~~681	0.939	0.382	0.662	0.445

	Borderline SMOTE	~~77.65~~0.917	~~77.57~~0.595	~~77.65~~0.462	0.~~7757~~756	0.911	0.374	0.682	0.423
	ADASYN	~~72.83~~0.995	~~72.77~~0.226	~~72.83~~0.205	0.~~7263~~230	0.978	0.514	0.414	0.447
	Without Oversampling	~~97.59~~0.931	~~97.41~~0.524	~~97.59~~0.436	0.~~9749~~681	0.924	0.360	0.643	0.412
XGBoost ~~CatBoost~~	SMOTE	~~99.56~~0.991	~~99.56~~0.976	~~99.56~~0.974	0.~~9956~~837	0.989	0.774	0.945	0.846
	K-Means SMOTE	~~99.57~~0.993	~~99.57~~0.952	~~99.57~~0.949	0.~~9957~~785	0.990	0.787	0.920	0.842
	Borderline SMOTE	~~99.50~~0.994	~~99.50~~0.905	~~99.50~~0.769	0.~~9950~~733	0.990	0.803	0.850	0.823
	ADASYN	~~99.58~~0.990	~~99.58~~0.988	~~99.58~~0.974	0.~~9957~~830	0.988	0.761	0.946	0.837
	Without Oversampling	~~98.11~~0.996	~~97.85~~0.250	~~98.11~~0.026	0.~~9787~~333	0.980	0.553	0.401	0.447
Light GBM~~XGBoost~~	SMOTE	~~99.63~~0.983	~~99.63~~0.905	~~99.63~~0.974	0.~~9963~~748	0.980	0.656	0.903	0.750
	K-Means SMOTE	~~99.62~~0.984	~~99.62~~0.917	~~99.62~~0.872	0.~~9962~~704	0.980	0.654	0.869	0.737
	Borderline SMOTE	~~99.57~~0.990	~~99.57~~0.738	~~99.57~~0.667	0.~~9957~~637	0.983	0.695	0.758	0.720
	ADASYN	~~99.64~~0.981	~~99.64~~0.917	~~99.64~~0.974	0.~~9964~~741	0.978	0.638	0.903	0.735
	Without Oversampling	~~98.09~~0.996	~~97.63~~0.214	~~98.09~~0.205	0.~~9775~~326	0.980	0.547	0.435	0.472
CatBoost ~~Light GBM~~	SMOTE	~~99.55~~0.986	~~99.55~~0.964	~~99.55~~0.974	0.~~9955~~852	0.984	0.705	0.944	0.799
	K-Means SMOTE	~~99.60~~0.988	~~99.60~~0.952	~~99.60~~0.974	0.~~9960~~815	0.986	0.726	0.932	0.810
	Borderline SMOTE	~~99.54~~0.990	~~99.54~~0.798	~~99.54~~0.641	0.~~9954~~689	0.984	0.720	0.779	0.743
	ADASYN	~~99.58~~0.987	~~99.58~~0.988	~~99.58~~0.974	0.~~9958~~859	0.985	0.722	0.952	0.814
	Without Oversampling	~~98.00~~0.997	~~97.59~~0.226	~~98.00~~0.179	0.~~9770~~311	0.981	0.611	0.428	0.487
RF	SMOTE	~~99.52~~0.988	~~99.53~~0.988	~~99.52~~0.974	0.~~9952~~970	0.988	0.763	0.980	0.851
	K-Means SMOTE	~~99.64~~0.995	~~99.64~~0.917	~~99.64~~0.821	0.~~9964~~867	0.993	0.885	0.900	0.889
	Borderline SMOTE	~~99.58~~0.991	~~99.58~~0.976	~~99.58~~0.974	0.~~9958~~956	0.991	0.801	0.974	0.875
	ADASYN	~~99.54~~0.989	~~99.54~~0.988	~~99.54~~0.974	0.~~9953~~978	0.988	0.757	0.982	0.848
	Without Oversampling	~~98.09~~0.998	~~97.67~~0.190	~~98.09~~0.051	0.~~9763~~289	0.980	0.676	0.382	0.440
MLP	SMOTE	~~89.89~~0.958	~~90.07~~1.000	~~89.89~~1.000	0.~~8985~~948	0.958	0.554	0.977	0.671
	K-Means SMOTE	~~59.84~~0.965	~~73.31~~0.988	~~59.84~~0.974	0.~~5754~~830	0.964	0.578	0.939	0.689
	Borderline SMOTE	~~95.44~~0.937	~~95.53~~0.750	~~95.44~~0.641	0.~~9545~~659	0.932	0.444	0.747	0.518
	ADASYN	~~88.19~~0.927	~~88.44~~1.000	~~88.19~~0.974	0.~~8818~~963	0.928	0.554	0.966	0.652

	Without Oversampling	~~97.61~~0.995	~~96.00~~0.012	~~97.61~~0.026	0.~~9670~~015	0.974	0.380	0.262	0.270
LSTM	SMOTE	~~69.02~~0.878	~~69.09~~0.774	~~69.01~~0.897	0.~~6829~~815	0.877	0.451	0.841	0.522
	K-Means SMOTE	~~78.56~~0.981	~~78.98~~0.869	~~78.55~~0.923	0.~~7864~~763	0.977	0.693	0.884	0.766
	Borderline SMOTE	~~86.23~~0.948	~~86.52~~0.917	~~86.22~~1.000	0.~~8609~~919	0.948	0.527	0.946	0.636
	ADASYN	~~76.78~~0.953	~~77.06~~0.952	~~76.77~~1.000	0.~~7664~~911	0.953	0.552	0.954	0.661
	Without Oversampling	~~97.79~~0.996	~~95.63~~0.488	~~97.78~~0.667	0.~~9669~~415	0.985	0.804	0.642	0.704
Dynamic Ensemble	SMOTE	~~99.58~~0.978	~~99.58~~0.999	~~99.58~~0.999	0.~~9958~~997	0.994	0.994	0.994	0.994
	K-Means SMOTE	**~~99.68~~0.999**	**~~99.68~~1.000**	**~~99.68~~0.979**	**~~0.9968~~1.000**	**0.995**	**0.995**	**0.995**	**0.995**
	Borderline SMOTE	~~99.55~~0.982	~~99.55~~0.999	~~99.55~~0.999	0.~~9955~~997	0.995	0.995	0.995	0.995
	ADASYN	~~99.58~~0.979	~~99.58~~0.999	~~99.58~~0.999	0.~~9958~~997	0.994	0.994	0.994	0.994
	Without Oversampling	~~98.20~~0.998	~~97.83~~0.167	~~98.20~~0.128	0.~~9783~~296	0.981	0.646	0.397	0.462

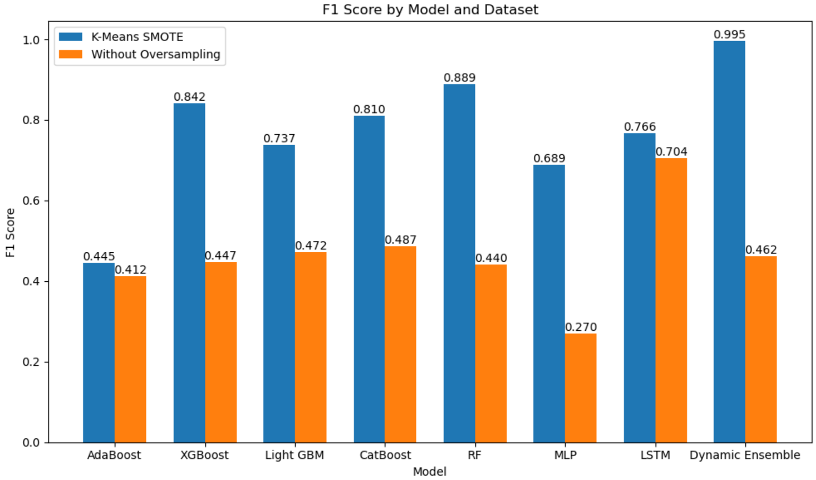


Figure 2: Comparison of F1 Score performance between K-Means SMOTE and without oversampling techniques across various ML models for soil movement prediction in testing. Blue bars represent F1 scores achieved with K-Means SMOTE, while orange bars represent F1 scores obtained without oversampling.

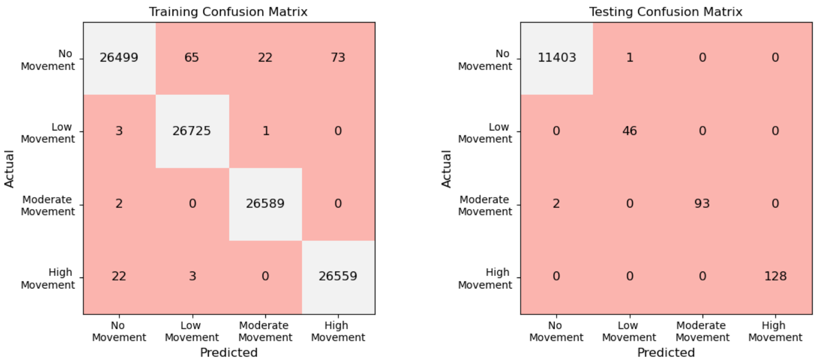


Figure 3: Confusion matrix depicting the performance of the Dynamic Ensemble model on the training and testing datasets using K-Mean SMOTE oversampling technique.

7. Discussion and Conclusions

In ~~conclusion,~~summary, the threat posed by landslides ~~pose significant threats to lives and properties, necessitating~~requires the development of effective ~~landslide~~ prediction frameworks~~. While IoT devices have been used to notify people in advance about potential landslides, modeling~~, although modelling the chaotic nature of natural data remains ~~a challenge~~challenging. The analyzed dataset ~~used in the analysis described above shows~~exhibited a significant class imbalance, with the ~~minority classes representing only 2% of the data while the~~ majority class ~~accounts for 98%.~~dominating the samples. This ~~substantial disparity in sample~~ distribution ~~can pose challenges in analysis and modeling, requiring~~imbalance necessitated careful consideration and appropriate techniques to address the ~~class imbalance~~issue.

~~To address the class imbalance,~~ Various oversampling techniques ~~such as~~ were employed to tackle the class imbalance, including SMOTE and its extensions (~~KMeans~~K-Means SMOTE, Borderline SMOTE, and ~~SVM SMOTE) were employed.~~ADASYN). ADASYN, which focuses on the ~~classification~~ minority class boundary ~~of the minority class, was particularly effective in generating~~, effectively generated synthetic data points and ~~improving the balance of~~ improved the class distribution balance.

~~Various~~Multiple classification models ~~were evaluated for predicting soil movement, including~~ , such as ADABoost, XGBoost, Light GBM, CatBoost, RF, MLP, LSTM, and a dynamic ensemble ~~of ADABoost, XGBoost, Light GBM, CatBoost, and RF.~~, were evaluated to predict soil movement. The ~~hyperparameters of each model were optimized using a~~ grid search approach~~. The dynamic ensemble with K-Mean SMOTE and RF with K-Mean SMOTE emerged as the top performing models, with dynamic ensemble achieving slightly higher accuracy.~~

~~The combination of K-Means SMOTE oversampling with the dynamic ensemble model yielded the highest accuracy, precision, and recall of 99.68% and an F1-score of 0.9968 in predicting soil movement. These results demonstrate the effectiveness of oversampling techniques and the dynamic ensemble model for addressing class imbalance and improving the accuracy of landslide prediction.~~

~~This study highlights the importance of preprocessing, class labeling, addressing class imbalance, and selecting appropriate classification models in predicting soil movement. The findings can contribute to a better understanding and mitigation of landslide risks, ultimately aiding in developing effective preventive measures.~~

~~There are several limitations to consider in this study. Firstly, the findings may not be directly applicable to other regions or geological conditions due to the specific dataset used. Secondly, while oversampling techniques~~ and 5-CV were employed to ~~address class imbalance, the synthetic data points may not fully capture the complexity of real-world landslide occurrences. Additionally, the choice of classification models and hyperparameter settings may introduce bias and alternative configurations could yield~~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 ~~results. The study also relied on historical data, potentially limiting its ability to account for future changes. Finally, factors such as rainfall intensity, seismic activity, and human influences were not fully captured, suggesting the need for further research to enhance landslide prediction accuracy~~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 of parameter values, thereby enhancing our understanding of how the models generalize to unseen data.

~~In future work, we plan to evaluate the~~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 an 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 ~~encoder-decoder~~ 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 affected both training and testing performances. The models faced 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, Light GBM, 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 Light GBM and CatBoost, and the robustness of RF creates a robust and versatile ensemble that excels 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 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 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 ML models to learn these generalized patterns more effectively. As evident in the parameter analysis results, oversampling techniques lead to smaller stdev 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 on the class-imbalanced movement dataset is planned. These models ~~have demonstrated~~, known for their success in sequence-to-sequence tasks ~~and could potentially~~, may improve classification accuracy and address class imbalance challenges. This avenue of experimentation ~~will~~aims to provide valuable insights into ~~their~~the suitability of advanced models for analyzing and ~~modeling~~modelling 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 modelling, 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.

Acknowledgements

We ~~are grateful to the~~ want to acknowledge and express our sincere gratitude to the DST, India, and the DDMA ~~Mandi,~~Kangra (IITM/DDMA-Kan/KVU/357), Kinnaur~~, and Kangra for providing the fund for~~ (IITM/DDMA-Kinn/VD/345), and Mandi (IITM/DDMA-M/VD/325 and IITM/DDMA-M/VD/358) for their invaluable financial support towards this research ~~grant~~project. We are also ~~thankful~~immensely grateful to the IIT Mandi for ~~providing the space and computing~~ generously facilitating us with the necessary infrastructure, including research facilities ~~for~~and computational resources, that have been instrumental in the successful execution of this ~~research work~~study.

References

Breiman, L.: Random forests. *Machine learning*, 45, 5-32, 2001.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P.: SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 321-357, 2002.

Chen, T., & Guestrin, C.: Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pp. 785-794, 2016.

Crosta, G.: Regionalization of rainfall thresholds: an aid to landslide hazard evaluation. *Environmental Geology*, 35(2), 131-145, 1998.

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

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

Han, H., Wang, W. Y., & 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, August 23-26, 2005, Proceedings, Part I 1, pp. 878-887, 2005.

He, H., Bai, Y., Garcia, E. A., & 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)*, pp. 1322-1328, 2008.

Hochreiter, S., & Schmidhuber, J.: Long short-term memory. *Neural computation*, 9(8), 1735-1780, 1997.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y.: Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

Ko, A. H., Sabourin, R., & Britto Jr, A. S.: From dynamic classifier selection to dynamic ensemble selection. *Pattern recognition*, 41(5), 1718-1731, 2008.

Kumar, P., Sihag, P., Sharma, A., Pathania, A., Singh, R., Chaturvedi, P., & 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(4), 788-810 (2021a).

Kumar, P., Sihag, P., Chaturvedi, P., Uday, K. V., & Dutt, V.: BS-LSTM: an ensemble recurrent approach to forecasting soil movements in the real world. *Frontiers in Earth Science*, 9, 696-792 (2021b).

Kumar, P., Sihag, P., Pathania, A., Agarwal, S., Mali, N. C. P., Singh, R., ... & Dutt, V.: Landslide debris-flow prediction using ensemble and non-ensemble machine-learning methods. In *International Conference on Time Series and Forecasting* (Vol. 1), 2019.

Kumar, P., Priyanka, Pathania, A., Agarwal, S., Mali, N., Singh, R., ... & 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, Volume 2*, pp. 67-81, 2020.

Kumar, P., Priyanka, P., Dhanya, J., Uday, K. V., & 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(2), 177-204, 2011.

Pathania, A., Kumar, P., Priyanka, P., Maurya, A., Uday, K. V., & 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*, pp. 365-379, 2021.

Pathania, A., Kumar, P., Sihag, P., Chaturvedi, P., Singh, R., Uday, K. V., & 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*, 2020.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A.: CatBoost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

Ray, R. L., Lazzari, M., & Olutimehin, T.: Remote sensing approaches and related techniques to map and study landslides. *Landslides Investig. Monit.*, 2020.

Rosenblatt, F.: Principles of neurodynamics. perceptrons and the theory of brain mechanisms. *Cornell Aeronautical Lab Inc Buffalo NY*, 1961.

Sahin, E. K.: Comparative analysis of gradient boosting algorithms for landslide susceptibility mapping. *Geocarto International*, 37(9), 2441-2465, 2022.

Semwal, T., Priyanka, P., Kumar, P., Dutt, V., & 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*, pp. 87-100, 2022.

~~Tang, Y., Zhang, Y. Q., Chawla, N. V., & Krasser, S.: SVMs modeling for highly imbalanced classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(1), 281-288, 2008.~~

Wu, Y., Ke, Y., Chen, Z., Liang, S., Zhao, H., & Hong, H.: Application of alternating decision tree with AdaBoost and bagging ensembles for landslide susceptibility mapping. *Catena*, 187, 104396, 2020.

Zhang, S., Wang, Y., & Wu, G.: Earthquake-Induced Landslide Susceptibility Assessment Using a Novel Model Based on Gradient Boosting Machine Learning and Class Balancing Methods. *Remote Sensing*, 14(23), 5945, 2022.