Received: 28 April 2021 | Revised: 31 May 2021 | Accepted: 14 June 2021

DOI: 10.1002/int.22557

RESEARCH ARTICLE

WILEY

An intelligent framework for end-to-end rockfall detection

Thanasis Zoumpekas[1] | **Anna Puig**[1] | **Maria Salamó**[1] | **David García-Sellés**[2] | **Laura Blanco Nuñez**[3,4] | **Marta Guinau**[2]

[1]Department of Mathematics and Computer Science, WAI Research Group, IMUB and UBICS Institutes, University of Barcelona, Barcelona, Spain

[2]Department of Earth and Ocean Dynamics, RISKNAT Research Group, Geomodels Institute, University of Barcelona, Barcelona, Spain

[3]Department of Earth and Ocean Dynamics, GGAC Research Group, Geomodels Institute, University of Barcelona, Barcelona, Spain

[4]Anufra—Soil and Water Consulting, Barcelona, Spain

Correspondence
Thanasis Zoumpekas, Department of Mathematics and Computer Science, WAI Research Group, IMUB and UBICS Institutes, University of Barcelona, Barcelona 08007, Spain.
Email: thanasis.zoumpekas@ub.edu

Funding information
Marie Skłodowska-Curie Actions, Grant/Award Number: 860843

Abstract

Rockfall detection is a crucial procedure in the field of geology, which helps to reduce the associated risks. Currently, geologists identify rockfall events almost manually utilizing point cloud and imagery data obtained from different caption devices such as Terrestrial Laser Scanner (TLS) or digital cameras. Multitemporal comparison of the point clouds obtained with these techniques requires a tedious visual inspection to identify rockfall events which implies inaccuracies that depend on several factors such as human expertize and the sensibility of the sensors. This paper addresses this issue and provides an intelligent framework for rockfall event detection for any individual working in the intersection of the geology domain and decision support systems. The development of such an analysis framework presents major research challenges and justifies exhaustive experimental analysis. In particular, we propose an intelligent system that utilizes multiple machine learning algorithms to detect rockfall clusters of point cloud data. Due to the extremely imbalanced nature of the problem, a plethora of state-of-the-art resampling techniques accompanied by multiple models and feature selection procedures are being investigated. Various machine

learning pipeline combinations have been examined and benchmarked applying well-known metrics to be incorporated into our system. Specifically, we developed machine learning techniques and applied them to analyze point cloud data extracted from TLS in two distinct case studies, involving different geological contexts: the basaltic cliff of Castellfollit de la Roca and the conglomerate Montserrat Massif, both located in Spain. Our experimental results indicate that some of the above-mentioned machine learning pipelines can be utilized to detect rockfall incidents on mountain walls, with experimentally validated accuracy.

KEYWORDS
geology, imbalanced classification, intelligent systems, machine learning, rockfall monitoring

1 | INTRODUCTION

In the field of geology, one crucial task is rockfall detection, which helps to reduce the risk of future hazards.[1] Recently, technologies that are able to characterize the geometrical properties of rock slopes and cliffs have emerged.[2] As a result, the possibility of detecting changes in a cliff with high precision has been increased greatly.

Currently, geoscientists utilize specific methodologies to detect changes in rock slopes mainly by comparing measurements at different points in time.[3] However, the problem with such techniques is that they rely greatly on the sensitivity of the sensor that captures the data or even the measurement tool. Consequently, due to potential measurement errors, the users need to examine case by case analyzing the nature of the change in the cliff, that is, vegetation, edge effect, noise or random objects. For instance, it is often difficult to distinguish rock detachments from other changes optically in point cloud or imagery data.

The issues presented above justify the need for a new automated intelligent system designed to detect rockfalls. Therefore, in this paper, we propose an intelligent framework for rockfall detection. At the time of writing, a thorough search for such a system has failed to yield results. In addition, although there are recent studies such as Reference [4] that develop rockfall identification frameworks using point cloud data, they do not address the data imbalance issues, which are significant in the geology field. We consider our study capable of elucidating the field towards the implementation of a general machine learning classification framework able to handle cases with extremely imbalanced data. The level of data imbalance in this study is significant. This is justified in the later sections of this article since there is a low number of rockfall labeled instances. Depending on the nature of the scenery, the vegetation or edge effects can be numerous compared to rockfall events.

In this article, we examine several of the intelligent methods dealing with rockfall and landslide detection, implement the relative machine learning models accompanied by various

resampling strategies to handle the imbalanced nature of the study and apply them on point cloud data to identify rockfall events. Two well differentiated geological environments have been selected as case studies. The first one is located at the Montserrat Massif (Barcelona, Spain), which corresponds to a fractured conglomerate cliff, called Degotalls. The second case study involves data from the basaltic lava flow cliff of Castellfollit de la Roca (Girona, Spain). The acquired data are preprocessed and used as input in the learning stage of our intelligent framework. We systematically compare various models to select the most accurate for each case study. Then, our prototype system utilizes the aforementioned selected models to perform effective rockfall event identifications.

Our framework provides a novel intelligent solution for the geoscientists, which may be incorporated as a stand-alone component in a geological decision support system targeted to rockfall monitoring. Its novelty is also derived from the fact that it is a ready-to-use geological point cloud machine learning software able to deal with imbalanced data, which is the only published work providing such a stand-alone rockfall detection system at the time of writing. Also, our results portray great robustness and generalizability performance, which is of utmost importance in rockfall detection.

Finally, the main contributions of this article are the following:

- We propose a full intelligent framework for rockfall detection handling highly imbalanced data.
- We conduct experiments with real data from two distinct case studies to validate the efficacy and effectiveness of our proposed intelligent system.
- We develop a web-based rockfall detection system.
- We provide a baseline methodology and a detection accuracy benchmark for future related experimental analyses.

The paper is organized as follows. Section 2 contains a synopsis of related efforts and Section 3 summarizes the design specifications and the implementation of the developed framework, while also provides the background information needed to understand the whole procedure of detecting rockfall events. In Section 4, we present the experimental analysis of the examined intelligent process, concerning common performance metrics for imbalanced classification, and the prototype web-based Rockfall Detection System. Section 5 includes the final observations of our experiments. Finally, Section 6 includes our concluding comments and potential issues for further investigation.

2 | RELATED WORK

Various research studies have been conducted on the intersection of the geology and machine learning domains. Over recent years, several machine learning applications have emerged in the geoscience field and there is a growing enthusiasm for intelligent methods.[5] However, the geoscience domain presents new and special challenges for machine learning algorithms and methodologies, because of the combinations of geoscience properties encountered in each specific case. In addition, there is an open need for novel machine learning research and automatic intelligent analysis, especially in the geosciences field as highlighted by Reference [6]. The automation of intelligent pipelines performing specific, special analysis will bring significant advances in both the geoscience and machine learning domains, especially in the

task of detecting objects and events, estimating and long-term predictions for geoscience variables and in extracting knowledge from geoscience data.

The detection of deformations and rockfall events have been studied in the literature for a couple of decades.[7,8] The number of scientific articles on the use of modern and advanced sensors such as Terrestrial Laser Scanner (TLS) in rockfall and landslide studies has escalated considerably in the last years. However, further research into the development of new methodologies using such sensors is required to address and improve the current time consuming analytical methods due to the high volume of data that these sensors produce.[9-13] Research studies on the *identification* of landslide and rockfall events and on general classification tasks in the geology domain present variations regarding their input data, feature inventories, methodologies, and in general their specific objectives.

Regarding the input data, we could categorize the related research studies into three broad categories: (i) the imagery data studies; (ii) point cloud studies; and (iii) other sensory data studies. Considering imagery data studies, remote-sensing techniques are considered to be among the most important for landslide event *detection* and *monitoring*. Multiple efforts by researchers working in the remote sensing field applied in geology appear in the literature, utilizing intelligent methods on high-resolution Earth Observation imagery data.[14] Imagery data seem to be a popular data source utilized in numerous studies. Recently published articles, such as References [15-21] use images as their main data source. In addition, researchers involved in Reference [22] constructed 3D images using a multiview stereo algorithm and consumer-grade cameras.

On the other hand, the use of TLS devices to extract point cloud data for the categorization of landslide kinematics is significant. Also, research studies show that it seems to be necessary to fuse point cloud data from various sources, namely TLS combined with Airborne Laser Scanner (ALS) or Structure from Motion (SfM) methods combined with TLS, to overcome the limitations of each individual technique.[23] Point clouds are heavily utilized by various researchers, mainly because they carry detailed, high quality information. Specifically, recent studies, including References [4,24-30] use point clouds as their main data source for their analysis. Moreover, advanced research studies use both images and point clouds portraying a speciality and differentiate themselves from the majority of the related studies.[31-33] Furthermore, Reference [34] use images and TLS point cloud data to model slopes. Last but not least, Reference [35] utilize data from wireless sensor networks and machine learning algorithms to monitor and forecast landslides in real time.

Several researchers have compiled a brief selection of the features used for landslide and rockfall identification. These features can be categorized in five groups, namely morphological, hydrological, geological, land cover features and features obtained from other sources, like rainfall intensity, according to Reference [14]. Pure geological features like the lithology of a studied field or geo-structural information appear in older studies such as Reference [36]. Recently, various scientists have utilized explanatory variables derived from basic cartographic operations on geological records. These variables are commonly used to form land cover features based on distances to faults, roads, and rivers, which are measured using simple spatial operations in GIS software.[37] In addition, Reference [20] use combined morphological predictors, namely elevation, aspect, curvature, slope, with hydrological ones like wetness index and rainfall intensity data. Similarly, Reference [38] utilize morphological and hydrological features adding multiple curvature measurements, such as plan and surface curvature to explain the mechanisms of landslides and to confirm the state of vegetation, roads and, in general, geometric deformations.

Furthermore, there are variations regarding the methodologies utilized in rockfall and landslide *identification* and *forecasting*, ranging from methods using traditional geology and statistics to more intelligent ones using machine learning approaches such as deep learning and neural networks. Multiple efforts have been presented introducing analysis frameworks able to deal with landslide and rockfall events.

Traditional geology is mainly utilized in research articles published more than 10 years ago, such as Reference [39] although recent papers still investigate the application of such methods.[2] Lague et al.[11] proposed a technique, which compares and combines two different sets of point clouds obtained from a TLS device. According to their methodology, two different time frames of point cloud data are combined and later compared to identify whether the input data form deformation or rockfall event. The aforementioned method uses traditional statistics and lacks generalization performance.

Machine learning-based methodological workflows are the most used techniques, as there are various research articles utilizing them. Recent research studies such as References [25-30,32,33,35] utilize mainly machine learning algorithms for rock-slope and landslide *monitoring* and *analysis*. More elaborate machine learning methods outperform the majority, examples include Reference [40], who developed a landslide monitoring approach for TLS point cloud data, which integrates a specialized machine learning classification involving topological rules in an object-based framework of analysis.

Although deep learning and neural networks are highly popular in other domains, such as the biomedical domain, achieving significant results in classification tasks,[41,42] are currently utilized in fewer studies in the geology domain, such as References [43,44]. However, advanced neural network architectures using specialized loss functions are being employed to tackle specific issues in the field such as data imbalance in landslide analysis, showing important advances in terms of performance.[14] Also, Xiao et al.[15] present a landslide susceptibility assessment framework based on deep learning algorithms using multisource imagery data. A wide assortment of recently published research studies combines and compares machine learning, neural networks, and ensemble-based techniques to achieve higher performance in terms of accuracy and generalizability.[4,15-21,31,45]

The data imbalanced issue has been addressed in multiple fields, for instance, in classification approaches in the biomedical domain.[46,47] Ijaz et al.[46] used Synthetic Minority Over-Sampling Technique (SMOTE) combined with Random Forest (RF) classifier to deal with imbalanced data and model diseases. In the geoscience domain, the landslide and rockfall *identification* are two tasks that present significant imbalanced data issues. Prakash et al.[14] approached the imbalanced learning task by utilizing sophisticated loss functions in the training phase and data augmentation techniques. Stumpf et al.[48] proposed a machine learning-based analysis framework with a repetitive strategy to handle class imbalance. Zhao et al.[49] tackled minor data imbalance issues in predicting landslide susceptibility by utilizing a voting system and the random processing of samples with a random forest algorithm.

In a more recent framework,[4] proposed combining ALS and TLS data with GIS and introducing a hybrid ensemble model and a 3D kinematic rockfall forecasting model able to deal with rockfall hazard assessment, achieving promising accuracy. However, they did not address the imbalanced nature of the study or deal with imbalanced data. Our proposal considers a novel machine learning approach that also deals with the problem of imbalanced data. Specifically, we propose an intelligent analysis framework and rockfall detection decision support software that incorporate various processing stages, which are considered to be essential for such problems, namely clustering, resampling, model parameterization, and feature selection.

Regarding the input data sources, we consider only point cloud data acquired by a TLS device from two distinct geological contexts, with features derived mostly from the morphology of the terrain.

3 | END-TO-END MACHINE LEARNING FRAMEWORK: DESIGN AND IMPLEMENTATION

This section introduces our developed end-to-end machine learning framework for detecting rockfall events. First, we explain the basic procedure of detecting rockfalls. Then, we analytically present our proposed framework.

3.1 | Background on detecting rockfalls

One of the most common processes in the detection of rockfall events on mountain cliffs and slopes is explained in Reference [3] and depicted in Figure 1, specifically in Steps (A)–(E). The methodology implies capturing periodical measurements of the same cliff face at different time frames with a TLS, denoted as Step (A) in Figure 1. TLS is a measuring device which offers the ability to collect dense point-clouds of objects. It also provides high-precision and high-accuracy data and is widely used in the geology domain.[50] After the capture, the procedure continues with the detection of changes at the surface of the cliff from point cloud data comparison, which is mostly performed with a technique called *m3c2*,[11] denoted as Step (B). As a result, a new point cloud is obtained containing the metric distances between the compared dense clouds. A clustering algorithm is then applied in Step (C) following the change detection step outputting a set of clusters, which are aggregations of points with a significant distance value. These clusters allow for the management of subsets of point clouds with specific topological properties. A statistical analysis then generates a set of features (Step (D)) to be evaluated manually, through a visual inspection with expert criteria, to determine whether they are rockfalls or random noise (Step (E)). In most cases, samples, that is, the clusters, happen to be random noise or even measurement errors by the scanner itself.

The current semimanual classification task of the TLS point cloud data for the detection of rockfall events presents two main challenges. The first concerns the sensibility of the sensor because in some cases the detection of movements is smaller than the device's margin of error. The second concerns the process of distinguishing rock movement events from other kinds of events, such as the movement of the sensor between measurements, the appearance of

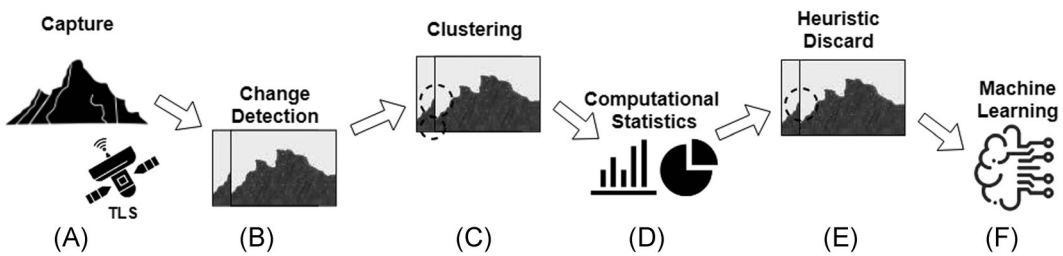

FIGURE 1 Procedure for detecting rockfalls. TLS, terrestrial laser scanner

vegetation or even random noise. The use of clustering techniques helps to mitigate the aforementioned issues.

Machine learning algorithms present an interesting and currently widely accepted solution for the automatic classification of TLS point cloud data for the detection of rockfall events. However, rockfall detection is considered to be a highly imbalanced classification task, due to the rarity of a rockfall event in a relative data set. Moreover, clustered point cloud data present patterns that are not easily distinguishable, while having high dimensionality, due to the considerable number of features. In addition, automating the above-mentioned process seems to be an interesting solution for an individual working in the geology domain. There is an open need for a consistent and concise intelligent rockfall identification framework, as detailed by Reference [6]. In Section 3.2, we propose a developed machine learning framework, which represents Step (F) in Figure 1.

3.2 | Framework overview

An outline of our proposal can be seen in Figure 3, where we synthesize our framework by providing its methodology workflow, and in Figure 2, where we present its component-based system architecture.

First, we collect point cloud data in different time frames (Figure 2A) using the TLS device and store it in a database (Figure 2B). Next, we preprocess it (Figure 2C) to be fed in our analysis framework. Then, we statistically analyse the input data, obtaining descriptive statistics to elucidate the field regarding the data. The obtained data set is then stored and normalized. Subsequently, to prevent issues of imbalance, we define our resampling module,

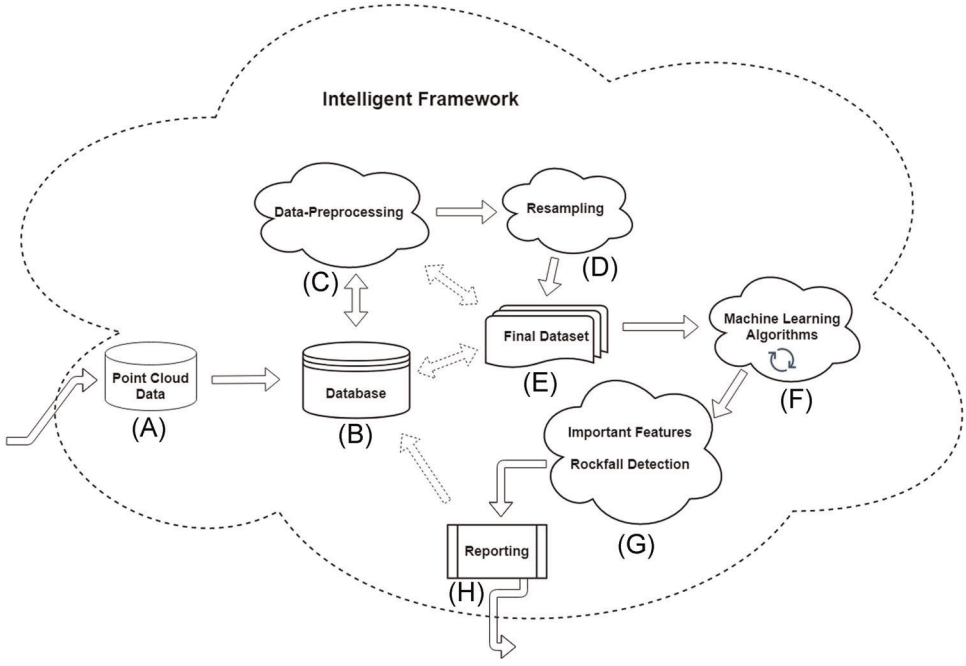

FIGURE 2 Intelligent framework—methodology workflow [Color figure can be viewed at wileyonlinelibrary.com]

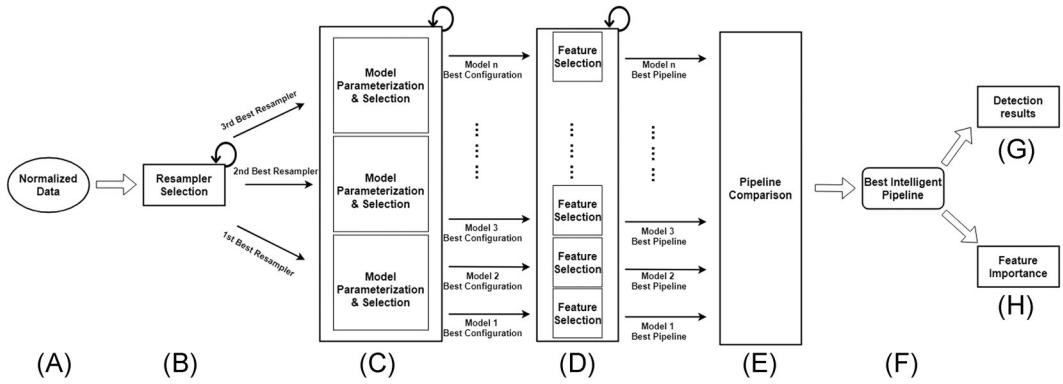

FIGURE 3 Component-based system architecture

portrayed in Figure 2D, which balances the normalized data (Figure 2E) to be used as input for machine learning algorithms (Figure 2F). Several intelligent machine learning pipelines are utilized to detect rockfall events and to identify the most significant features (Figure 2G). Finally, a reporting component is used to display the results in a convenient and illustrative manner, as depicted in Figure 2H.

Regarding our methodology, we initially feed normalized point cloud data in our intelligent framework as depicted in Figure 3A. Then, in the resampling process (Figure 3B), an assortment of several resampling strategies accompanied by multiple machine learning models are investigated to handle this imbalanced classification task. The three best resamplers are then selected and paired with a variety of models in the model selection and parametrization stage (Figure 3C). The completion of the hyper-parameterization stage follows the feature selection phase (Figure 3D), in which the best parameterized models of each model variant accompanied by each one of the three best resamplers are experimentally evaluated using different numbers of features. The output of the aforementioned stage consists of multiple properly parameterized machine learning pipelines (Figure 3E). These pipelines are then statistically compared to decide on the best intelligent pipeline to be used in our prototype system (Figure 3F), which is able to produce detection results (Figure 3G) and rank feature importance (Figure 3H) with promising accuracy. In the following sub-sections, we analytically present our framework and methodological approach.

3.3 | Data collection

The data used in this study has been freely provided by the RISKNAT* research group and the GEOMODELS** Research Institute, which belong to the University of Barcelona.

In this article, we utilize two distinct case studies. Regarding the first case study, we use point cloud data measured from the Degotalls cliff in Montserrat Massif, located in Barcelona, Spain. The data set consists of clustered TLS point cloud data aggregated from temporal point cloud measurements, from 2007 to 2020 in eight nearly regular time steps. The second case study includes clustered TLS point cloud data measured from the cliff of Castellfollit de la Roca, located in Girona, Spain, from 2008 to 2012. We would like to highlight that the two landscapes are very different in terms of their geological and imagery properties.

Actually, in each case study, the data is composed of a set of statistics computed from the point cloud data, using a technique based on the *m3c2* methodology,[11] which creates clusters of points, as described in Section 3.1, including several numerical features that are used in the proposed workflow to identify rockfalls. Then, the aforementioned statistical set is fed in our analysis framework. The Degotalls data consist of 6004 instances, that is, clusters, and has 37 distinct numerical features, while the Castellfollit data consist of 10,371 instances and has 31 numerical features. The aforementioned features include the coordinates of the point clouds and other statistically computed values. A list of these variables with a brief explanation can be found in Appendix S1.

3.4 | Data analysis

A data exploration analysis of the point cloud data is provided in this section. This analysis is critical because it elaborates the entire data collection process. Particularly, in Tables 4 and 5 in Appendix S2, we display the summary statistics for all the data features used in the Degotalls and Castellfollit case studies, respectively, supplying the foundation for our subsequent analysis. For clarification purposes, we chose to use an abbreviation for each variable of the data. The complete variable names followed by a brief explanation can be found in Appendix S1.

The classification labels of the clusters are determined by the event that causes the change at the surface of the cliff. Rockfall events are denoted as "Candidate." Also, we call "Precursor" the events in which the rock presents a small movement before rockfall. In addition, the vegetation of the cliffs is denoted as "Vegetation" and the unknown or human-based events as "Unknow." The artifacts due to the edge effects or the TLS noise are denoted as "Limit_effect."

Observing the graphs in Figure 4, it is clear that the data in each case study are considered highly imbalanced. In the Degotalls case study (Figure 4A), we have only 65 rockfall candidate samples compared to the total of 6004 samples. On the other hand, the Castellfollit case study (Figure 4C) includes only 38 compared to the total of 10,371 samples. Please note, that the labeling process is done by expert geoscientists with a visual inspection of the 6004 and the 10,371 clusters.

3.5 | Data preprocessing

Section 3.4 reveals that the initial data contains a wide range of values of significant size, indicating that normalization is essential to facilitate the faster convergence of the optimization algorithms and the machine learning models to achieve the best performance.[51] Thus, we first normalize the data using z-score normalization technique as proposed in Reference [51], which is defined in Equation (1).

$$X_{norm} = \frac{X - mean(X)}{std(X)}, \tag{1}$$

where X denotes the feature array and X_{norm} is the normalized X as resulting from the subtraction of its mean ($mean(X)$) and division by its SD ($std(X)$).

The normalized clustered data of point clouds are used for the training of our machine learning models. Then, a vector (v) is constructed that models the target label naively as follows. If the cluster is considered a rockfall event, which can also be denoted as rockfall

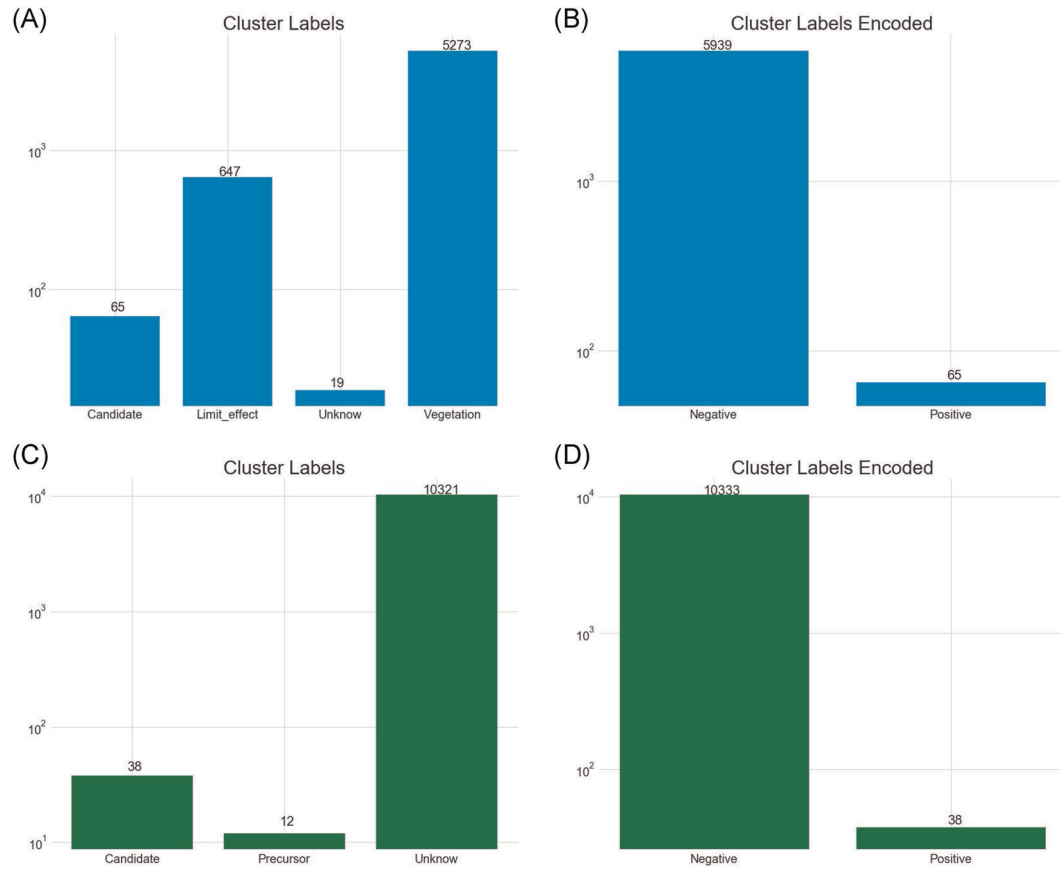

FIGURE 4 Target labels summary. *Y*-axis is in log-scale for comparison purposes. The actual values are included as text above each bar (A) Cluster labels—Degotalls case study, (B) encoded cluster labels for classification task—Degotalls case study, (C) cluster labels—Castellfollit case study, (D) encoded cluster labels for classification task—Castellfollit case study [Color figure can be viewed at wileyonlinelibrary.com]

candidate event, then we set a positive label ($v_i = 1$), and if not, we assign a negative label ($v_i = 0$). A summary of cluster labels is depicted in Figure 4, where subfigures (A) and (C) show the distribution of cluster labels before and subfigures (B) and (D) after encoding in the Degotalls and Castellfollit case studies, respectively.

Please note that the rockfall candidates belong in the class where $v_i = 1$ and the various "Limit_effect," "Unknow," "Vegetation," "Precursor" cases belong in the class where $v_i = 0$.

3.6 | Resampling techniques

Rockfall events are characterized by their rarity and thus our task is considered to be an imbalanced classification task. This is also justified by Figure 4, where the candidate clusters for rockfall represent roughly 1% and 0.4% of the total cases in Degotalls and Castellfollit, respectively. There are two well-known approaches to balance the data instances, namely *undersampling* and *oversampling* methods.[52] Undersampling removes elements from the majority class, while oversampling creates synthetic samples from the minority class. There are

also ensemble methods that oversample the minority class and undersample the majority one simultaneously.[53] Below, we briefly describe each one of the resamplers examined in this study. Finally, we conclude by utilizing the three best approaches in our subsequent analysis phases, according to their performance.

3.6.1 | Undersampling methods

The simplest way to perform undersampling is to do it at random, that is, by eliminating individual samples from the majority class without using any heuristics. More elaborate techniques employ some kind of heuristic processing to exclude elements that are informationally irrelevant to the overall data set. Some of the aforementioned methods, aggregate distinct samples to undersample the data using well-known clustering techniques, such as References [54].

In this study, we utilize two undersampling methodologies based on the K-means clustering algorithm. Specifically, we use two of the methods proposed by Reference [54] utilizing the K-means with the number of clusters proportional to the number of elements belonging to minority class. The first methodology swaps a cluster of samples belonging to the majority class with the cluster centroid obtained by the K-means algorithm (denoted as Cluster Centroids in this writing). The second one utilizes the same approach but uses the cluster center of the K-means instead of the cluster centroid (denoted as Cluster Representatives). Similar undersampling techniques utilizing clustering appear in recent studies such as Reference [55].

3.6.2 | Oversampling methods

Oversampling techniques perform the balancing of the data set by focusing on the minority classes. Various oversampling approaches are described in the literature.[56] SMOTE, one of the most widely used oversampling techniques, creates synthetic samples by combining features of the minority class's nearest neighbors.[57] Numerous variants of the aforementioned algorithm appear in the literature.[58] Kovacs et al.[56] presented a detailed empirical comparison of multiple variants of minority oversampling methods involving a wide assortment of imbalanced data sets. In this study, we focus on some of them, specifically the ones that we utilized for the purpose of this study.

The ADASYN oversampler determines the number of artificial samples to be produced for a given point by using a weighted density distribution for distinct minority class elements based on their level of difficulty in learning.[59] In contrast, SMOTE weights uniformly all minority points. The major difference between SMOTE and ADASYN lays in the generation of synthetic sample points for minority data points.

Barua et al.[60] introduced a "proximity weighted synthetic oversampling technique" (ProWSyn), which calculates weight values for the minority data samples based on their proximity information. Proximity, in this case, is portrayed as the distance between the sample and the boundary.[60] Another variant of SMOTE is the SMOTE-IPF, which is an extension through a new component, an "iterative ensemble-based noise filter called Iterative-Partitioning Filter" (IPF). SMOTE-IPF is considered to be a solution to the problems caused by noisy and borderline examples in unbalanced datasets as presented and explained in Reference [61].

SMOTE algorithm lacks distribution and density information of the data, both of which are critical for correctly synthesizing minority examples. The SMOBD algorithm addresses the above issues by effectively removing the effects of noise.[62] In addition, the Assembled-SMOTE approach implements oversampling by taking data distribution information into account to prevent the overlapping between classes.[63]

Other methods, such as the Lee algorithm, produce synthetic samples and determine whether to reject or accept them based on their location, that is, a synthetic sample's nearest neighbors.[64] Batista et al.[65] proposed an oversampling method call SMOTE-TomekLinks, which is essentially a SMOTE algorithm with the additional application of Tomek links to the oversampled training set as a data cleaning technique. Thus, the aforementioned technique improves the oversampling accuracy by removing examples from both classes. Tomek links are introduced in Reference [66] and can be defined as follows:

> "Given two examples E_i and E_j belonging to different classes, and $d(E_i, E_j)$ is the distance between E_i and E_j. A (E_i, E_j) pair is called a Tomek link if there is not an example E_l, such that $d(E_i, E_l) < d(E_i, E_j)$ or $d(E_j, E_l) < d(E_i, E_j)$."[65]

Furthermore, to effectively identify minority class instances, the CCR algorithm integrates the cleaning of the decision boundary around minority elements and controlled synthetic oversampling.[67] Moreover, the G-SMOTE approach pioneered hybrid oversampling that is driven by previously unknown patterns derived from the class with minor samples and randomization. The authors utilize concurrent oversampling and undersampling to deal with heavily skewed data distributions.[68] Additionally, LVQ-SMOTE integrates the SMOTE oversampler with feature codebooks learned by vector quantization to create synthetic samples that use more feature space than the other SMOTE variants.[69] Also, a comparable performance can be obtained using the Polynom-fit SMOTE, which oversamples the minority class using polynomial fitting functions.[70]

Moreover, the research study of Reference [71] proposed a method for selective preprocessing, called SPIDER, that combines filtering and oversampling of imbalanced data. Also, the SWIM algorithm creates synthetic elements with the same Mahalanobis distance from the majority class as currently available minority data samples.[72]

In this study, we examine and experiment with the 16 resamplers explained above, namely Cluster Centroids, Cluster Representatives, SMOTE, ProWSyn, SMOTE-IPF, SWIM, SMOBD, Lee, ADASYN, Assembled-SMOTE, SMOTE-TomekLinks, CCR, G-SMOTE, LVQ-SMOTE, Polynom-fit-SMOTE, and SPIDER.

3.7 | Models

This section presents the machine learning models trained and evaluated in the rockfall detection task. In addition, we provide basic information regarding the models utilized. However, a complete explanation and analysis of the algorithmic aspects of each model is out of the scope of this paper.

Our aim is to deal with rockfall detection with models belonging to two distinct and broad families of models, namely "single base learning" and "ensemble learning" algorithms, which are utilized in similar landslide classification tasks, as presented in Reference [73]. Single base learners are simpler models and learn by using a single algorithm, such as a decision tree

algorithm, while ensemble learners are comprised usually of multiple single base learners.[74] In this case, and following the notation of Ma et al.,[73] we use the following "single base learning" algorithms: **Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), K-Nearest Neighbors Classifier (KNN), Gaussian Naive Bayes (GNB), Decision Tree Classifier (DT), Support Vector Classifier (SVC) and Multilayer Perceptron (MLP) Classifier,** and the following "ensemble learning" algorithms: **AdaBoost Classifier (AdaB), Random Forest Classifier (RF), Extra Trees Classifier (ET), XGBoost Classifier (XGB)**. Following, there is a list containing all the models examined including their hyper-parameters.

Regarding the utilized "single base learning" approaches, we explain the models below:

Linear Discriminant Analysis (LDA) is a linear decision boundary classifier. The boundary of this classifier is created by fitting class conditional densities to the input data. In this model, each class is fitted with a Gaussian density function. The covariance matrix for all classes is considered to be the same.[75] We experiment with solvers such as singular value decomposition (SVD), eigenvalue decomposition (EIGEN) and least squares solution (LSQR), and with automatic or no shrinkage parameters.

Quadratic Discriminant Analysis (QDA) is a classifier that uses a quadratic decision boundary created by fitting class conditional densities to the input data to classify objects. The covariance matrix for all classes is assumed to be the same.[75] We utilize a regularization parameter of 0, that is, we do not regularize the per-class covariance approximations.

K-Nearest Neighbors Classifier (KNN) is a memory-based algorithm. Naively, given a query point x_0, we find the k training points $x_{(r)}$, $r = 1, ..., k$ nearest in distance to x_0, and then classify it using a majority vote policy among the k neighbors.[75] We experiment using various numbers of nearest neighbors, such as 1, 3, 5 and 9 neighbors.

Gaussian Naive Bayes (GNB) supposes that given a class $C = j$, the k number of features X_k are independent: $f_j(X) = \prod_{k=1}^{p} f_{jk}(X_k)$.[75] GNB considers continuous valued features and models them all as Gaussian (normal) distributions. To maintain measurement stability, a value of $1e-9$ of the highest variance of all features is appended to variances.

Decision Tree Classifier (DT) divides the feature space into rectangles, and then fits a simple model to each of them. Theoretically, tree-based classifiers are simple but efficient and accurate.[75] For this particular classifier, we experiment with two strategies for splitting each node, namely the best split and random split. Regarding the criterion for splitting or alternatively the function to calculate the quality of a split, we experiment with Gini impurity[76] and information gain metrics.

Support Vector Classifier (SVC) algorithm creates a hyperplane or set of hyperplanes in a high-dimensional space that can be used for classification. The margins of the hyperplanes are described by support vectors, which are discovered after an optimization process involving an objective function regularized by an error term and a constraint. SVMs may use kernel functions to create linear or nonlinear decision boundaries. Different kernel functions can be specified for the decision function.[77] In this particular case, we experiment with common kernels such as polynomial with multiple distinct degree values ranging from 2 to 8, a radial-basis function, and sigmoid. Regarding the regularization parameter, we have used multiple values, such as 0.1, 1, 10, and 100.

Multi-Layer Perceptron (MLP) Classifier is a class of feed-forward artificial neural networks.[78] Alternatively, the MLP classifier can be considered to be a shallow deep neural network. The investigated hyper-parameters are hidden layer sizes of 50, 100, 150,

and 200 and activation functions for the hidden layers, namely rectified linear unit (ReLu) $f(x) = max(0, x)$, hyperbolic tangent (TanH) $f(x) = tanh(x)$, and logistic sigmoid $f(x) = \frac{1}{1+exp(-x)}$. Besides, three optimizers are tested namely LBFGS, which is an optimizer belonging to the group of quasi-Newton methods, the stochastic gradient descent (SGD) and Adam, which is a stochastic gradient-based optimizer.

Following, we provide information on the utilized "ensemble learning" models:

AdaBoost Classifier (AdaB) is a meta-estimator and an ensemble machine learning algorithm that initially fits a classifier on the original input data and then fits additional copies of the classifier on the same data but conditionally. It fits the additional copies where the weights of wrongly classified instances are changed such that subsequent classifiers concentrate more on difficult cases.[79,80] Multiple numbers of estimators are investigated, such as 10, 50, 100, and 500. The base estimator is chosen to be a simple decision tree classifier with a max tree depth of 5.

Random Forest Classifier (RF) is a meta-estimator belonging to the ensemble group of algorithms that uses averaging methods to improve predictive accuracy. Also, it manages over-fitting by fitting multiple decision tree classifiers on various subsamples of the original input data.[75] Multiple numbers of trees are investigated, such as 10, 100, 500, and 1000. The quality measurement of a split of a tree-node is done by applying the Gini impurity and the information gain strategy. Both approaches are commonly used in the decision tree algorithms.[76]

Extra Trees Classifier (ET) is a meta estimator and an ensemble machine learning algorithm that uses averaging methods to boost predictive accuracy. Also, similarly to RF classifier, it manages over-fitting by fitting multiple randomized decision trees on different sub-samples of the original input data.[81] Different numbers of estimators are investigated, namely 10, 100, 500, and 1000. Also, the function to calculate the quality of a split of a tree-node is set to be the Gini impurity[76] or the information gain.

XGBoost Classifier (XGB) is a scalable end-to-end ensemble machine learning framework. It is actually a version of gradient boosted decision trees that has been engineered to produce cutting-edge results while staying fast and effective.[82] Different values of boosting learning rate are examined, ranging from 10^{-1} to 10^{-4}. In addition, various numbers of gradient boosted trees are used such as 10, 50, and 100, together with a variety of boosters such as *gbtree*, which is a version of a regression tree as a weak learner, *gblinear*, which uses generalized linear regression with L1 and L2 shrinkage, and *dart*, which drops trees in such way as to reduce the over-fitting issue.

Even though by using the "single base learning" algorithms, many pitfalls may be observed, such as overfitting and unreliable results,[73] in reality they are not always weak, as the correct utilization of them often leads to better outcomes.[74] For instance, DT is able to mine and identify significant relationships and key features, but it achieves lower accuracy values than its competitors.[83] Additionally, in simple rockfall and landslide classification applications, SVC algorithm achieves marginally better accuracy than the other methods, but the quality of its results is highly depended on the quality of the input data.[84] Besides, MLP achieved reasonably accurate results in similar studies.[85]

On the other hand, although "ensemble learning" algorithms, in general, are able to increase the robustness and generalizability of the "single base learning" ones,[73] they still need to be properly constructed and selected.[74] Essentially, in similar identification tasks an

RF model is considered to outperform other models and the stability of such ensemble models is ensured by the numerous base learners, that is, sets of decision trees.[86] Hong et al.[87] highlight that tree-based ensemble models could greatly improve the accuracy in landslide classification tasks.

Theoretically, GNB or SVC perform well in circumstances where limited and imbalanced data is available[88] although, practically, this phenomenon is not always witnessed.[89] LDA and QDA algorithms are utilized in classification studies, that involve geological features and phenomena.[90] While LDA can easily distinguish the data between two classes by applying linear combinations to input features, QDA enhances this ability and can also deal with classes having different covariance values, that is, imbalanced data.

Intuitively, facing the complex and challenging task of rockfall detection, which involves imbalanced point cloud data having a certain amount of features to handle, and examining studies such as Reference [91], where certain single base learners outperform some ensemble learners, lead us to use this selection of classifier models starting from simpler models like GNB, KNN, and so on, to more complex and elaborated ensemble methods like AdaB, ET, and XGB, to get strong performance proofs and to experimentally compare the ensemble methods with baseline methods. In addition, to further provide performance evidence of the utilized machine learning models, we use *10-fold cross-validation* procedures, as explained in the Section 4.2, which clearly facilitate the model selection by providing significant insights on the accuracy and robustness of each model.

3.8 | Feature selection

In this phase of the analysis framework, see Figure 3D, the best selected and properly parameterized models of each model variant accompanied with each one of the three best resamplers are used as inputs in a grid search algorithm aiming to select the exact number of features that achieve the highest performance. The criterion used for the univariate feature selection is mutual information between two random variables. Mutual information is a nonnegative value that measures the dependency between the variables.[92] According to the aforementioned methodology, we keep only the k highest scoring features, with k ranging from 5 to 37 (all features) in the Degotalls case study and from 5 to 31 (all features) in the Castellfollit case study.

3.9 | Pipeline selection

The input of this stage consists of the multiple machine learning pipelines, each properly parameterized with the best number of features, as shown in Figure 3E. These pipelines are statistically compared to conclude with the best intelligent pipeline to be used in our prototype system. The statistical comparison of the various pipelines is conducted by applying the Friedman and Nemenyi test, a nonparametric statistical test and a post-hoc test respectively, as described in Reference [93]. We run every pipeline $n = 10$ times using *10-fold cross-validation*. In each run, we rank the pipelines from 1 to n_{models}. By averaging the ranks table to get the averaged ranks for each pipeline, we can infer that two models are significantly different if their average ranks vary by at least the critical difference, as explained in Reference [94]. The critical difference is computed with the following formula[93]:

$$CD = q_\alpha \sqrt{\frac{n_{models} * (n_{models} - 1)}{6 * n}}, \quad (2)$$

where q_α denotes the critical value of the two tailed Nemenyi test, which depends on the chosen α level of statistical significance and the number of models or in this case the pipelines, n_{models}.

3.10 | Feature importance

In this phase of our framework, see Figure 3H, we calculate and visualize the feature importance of the input data set. This step provides significant insights in terms of the interpretability and explainability of the outcome produced by the utilized algorithm. Thus, we utilize a generalized feature importance measurement, the permutation feature importance as introduced and described in Reference [95]. The computation of permutation feature importance is done by first calculating a baseline metric, defined by a performance scoring function, and evaluated on a data set X. Next, a feature from the validation set is permuted and the metric is evaluated again. Finally, the permutation importance is defined as the difference between the baseline metric and the metric obtained from permutating the aforementioned feature.

In our study, we utilize an extensive permutation feature importance test to get robust results and insights regarding the most significant features utilized by the classifier. Specifically, we use the most statistical significant classifier as concluded from the computations in the Pipeline Selection stage of our framework, depicted in Figure 3E, with a *10-fold cross validation* technique on input data and 100 feature permutations in each fold. Then, we average all the values obtained and express them in percentage values for interpretability purposes. In addition, we use K-means clustering algorithm to group the feature importance obtained to get a better view of the most important features and their neighbors.

4 | EXPERIMENTAL ANALYSIS

This section presents the experimental analysis of our designed intelligent framework, based on well-known performance metrics. First, we explain the accuracy metric and then we present evaluation tables for each stage of the framework. Additionally, an ablation study is conducted to justify the addition of each component to the final prototype implementation.

4.1 | Performance metrics

Our aim is to design our models for a classification task, in which a false negative is usually more disastrous than a false-positive for preliminary rockfall detection. The nature of our problem is imbalanced as discussed in Section 3.1. For this reason, we utilize a special metric called balanced accuracy, as introduced and explained in Reference [96]. The authors of the aforementioned study define balanced accuracy as a performance metric for imbalanced classification tasks. The average accuracy obtained in either class can be naively characterized as balanced accuracy. The balanced accuracy is given by

$$Acc_b = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right), \tag{3}$$

where FP, TP, FN, and TN, are the false-positives, true-positives, false-negatives, and true-negatives, respectively, according to the confusion matrix, all of them clearly defined in Reference [96].

4.2 | Experimental evaluation

In this section, we evaluate the various machine learning models, which we proposed and concisely discussed in Section 3.7. We first exhibit the performance evaluation of the various resamplers in Table 1 and then a thorough evaluation of all the models used in the model selection and parameterization and feature selection phases in Table 2. The entire evaluation process is based on the balanced accuracy metric, denoted in Equation (3), using stratified *10-fold cross-validation*, which provides robust accuracy results.

4.2.1 | Resamplers

Table 1 displays the average performance of each resampler in the Degotalls (A) and Castellfollit (B) case studies. The depicted values are the average performances of all machine learning models evaluated with a stratified *10-fold cross validation* procedure, without prior parameterization, paired with each resampler. We average the balanced accuracy metric values resampler-wise, to select the three best resamplers to proceed to the model selection and parameterization stage. In both case studies, the Cluster Centroids resampler achieves the highest balanced accuracy while being the most robust method. It achieves a $\overline{Acc_b}$ of 0.89 with 3.83% error and 0.82 with 7.78% error in the Degotalls and Castellfollit case studies respectively. Observing the two subtables in Table 1, the results obtained in both cases do not seem to differ much in terms of ranking.

4.2.2 | Model selection

Table 2 shows the performance evaluation of the model selection and parameterization phase in the Degotalls (A) and Castellfollit (B) case studies. Specifically, it displays only the six best methods for each case study. For clarification purposes, the full tables are included in Tables 6 and 7, respectively in Appendix S2. For the selection of the best hyper-parameters of each model, a grid search algorithm is utilized using a stratified *10-fold cross-validation* procedure for each available combination of parameters. Please note that the depicted values in the above-mentioned table are the balanced accuracy measurements of another stratified *10-fold cross-validation* procedure on the data set utilizing only the best properly configured models that resulted from the aforementioned grid search. It is clear that in the Degotalls case study, the XGBoost classifier parameterized the best, accompanied by the SMOTE-IPF oversampler, noted as XGB-SMOTE_IPF in Table 2A, and performed better than all the other models while remaining robust. It achieves a $\overline{Acc_b}$ of 0.94 with a 3.83% error score. Regarding the best hyper-parameters of the XGBoost

TABLE 1 Acc_b metric summary for each resampling method. $\overline{Acc_b}$ denotes the average value of Acc_b of the *10-fold cross validation*

(a) Degotalls case study.

Resampling Method	$\overline{Acc_b}$	Error (%)
Cluster Centroids	0.89	3.83
ProWSyn	0.87	5.34
SMOTE-IPF	0.87	5.56
SWIM	0.86	5.65
SMOBD	0.86	6.29
Lee	0.86	5.48
ADASYN	0.86	6.28
Assembled-SMOTE	0.86	6.16
SMOTE	0.86	5.85
SMOTE-TomekLinks	0.86	6.56
CCR	0.85	7.91
G-SMOTE	0.85	7.15
LVQ-SMOTE	0.84	6.20
Polynom-fit-SMOTE	0.83	7.56
SPIDER	0.80	6.29
Cluster Representatives	0.78	7.67

(b) Castellfollit case study.

Resampling Method	$\overline{Acc_b}$	Error (%)
Cluster Centroids	0.82	7.78
ProWSyn	0.71	16.94
CCR	0.70	15.88
SWIM	0.70	14.90
Lee	0.68	17.52
SMOTE	0.68	18.13
LVQ-SMOTE	0.68	18.33
ADASYN	0.68	18.50
SMOBD	0.68	18.86
SMOTE-IPF	0.68	17.76
SMOTE-TomekLinks	0.68	17.33
Assembled-SMOTE	0.67	18.08
G_SMOTE	0.66	19.83
Polynom-fit-SMOTE	0.65	18.00
SPIDER	0.58	17.29
Cluster Representatives	0.56	18.28

Note: With dark grey we denote the best, with lighter grey the second best and with pale grey the third best.

classifier, a linear booster, namely gblinear, with a learning rate of 0.1 with 50 estimators, was chosen. On the other hand, in the Castellfollit case study, the Linear Discriminant Analysis classifier paired with the ProWSyn oversampler, noted as LDA-ProWSyn in Table 2B, appears to be the most robust technique in terms of balanced accuracy, achieving a $\overline{Acc_b}$ of 0.93 with 0.36% of error, using Eigenvalue Decomposition as the solver and with an automatic shrinkage parameter.

4.2.3 | Feature selection

In addition, Table 2 also displays the performance results of the feature selection phase in the Degotalls (A) and Castellfollit (B) case studies, in which each properly parameterized model

TABLE 2 Acc_b metric summary. $\overline{Acc_b}$ denotes the average value of Acc_b of the *10-fold cross validation*

(a) Degotalls case study.

Method	Model Parameterization		Feature Selection		
	$\overline{Acc_b}$	Error (%)	$\overline{Acc_b}$	Error (%)	Features
XGB-SMOTE_IPF	0.94	3.83	0.95	3.78	35
MLP-SMOTE_IPF	0.94	5.54	0.95	5.64	35
KNN-ClusterCentroids	0.94	4.29	0.95	4.09	36
XGB-ProWSyn	0.94	3.84	0.95	3.68	35
SVC-ProWSyn	0.94	5.68	0.95	4.14	17
LDA-SMOTE_IPF	0.94	4.13	0.95	4.07	31

(b) Castellfollit case study.

Method	Model Parameterization		Feature Selection		
	$\overline{Acc_b}$	Error (%)	$\overline{Acc_b}$	Error (%)	Features
XGB-ProWSyn	0.93	8.11	0.94	6.92	30
LDA-ProWSyn	0.93	0.36	0.93	3.98	30
MLP-CCR	0.93	8.21	0.93	8.18	30
LDA-CCR	0.93	0.54	0.93	4.06	29
XGB-CCR	0.92	8.11	0.92	8.17	30
SVC-ClusterCentroids	0.91	8.18	0.91	8.26	30

Note: With dark grey we denote the best model in terms of accuracy and then robustness and with lighter grey the second best for each phase. "Features" column displays the number of features utilized by each algorithm to achieve this score.

accompanied by a resampler is evaluated using the best number of features over a stratified *10-fold cross-validation* process. In the first case study, the XGBoost classifier accompanied by the SMOTE-IPF oversampler with 35 features seems to be the best performing model, while in the second best model is the XGBoost classifier paired with ProWSyn with 30 features, as displayed in Table 2.

Generalizing, we could say that in both case studies the top performing methods are approximately the same utilizing around the same percentage of features compared to the total available number of features in each case.

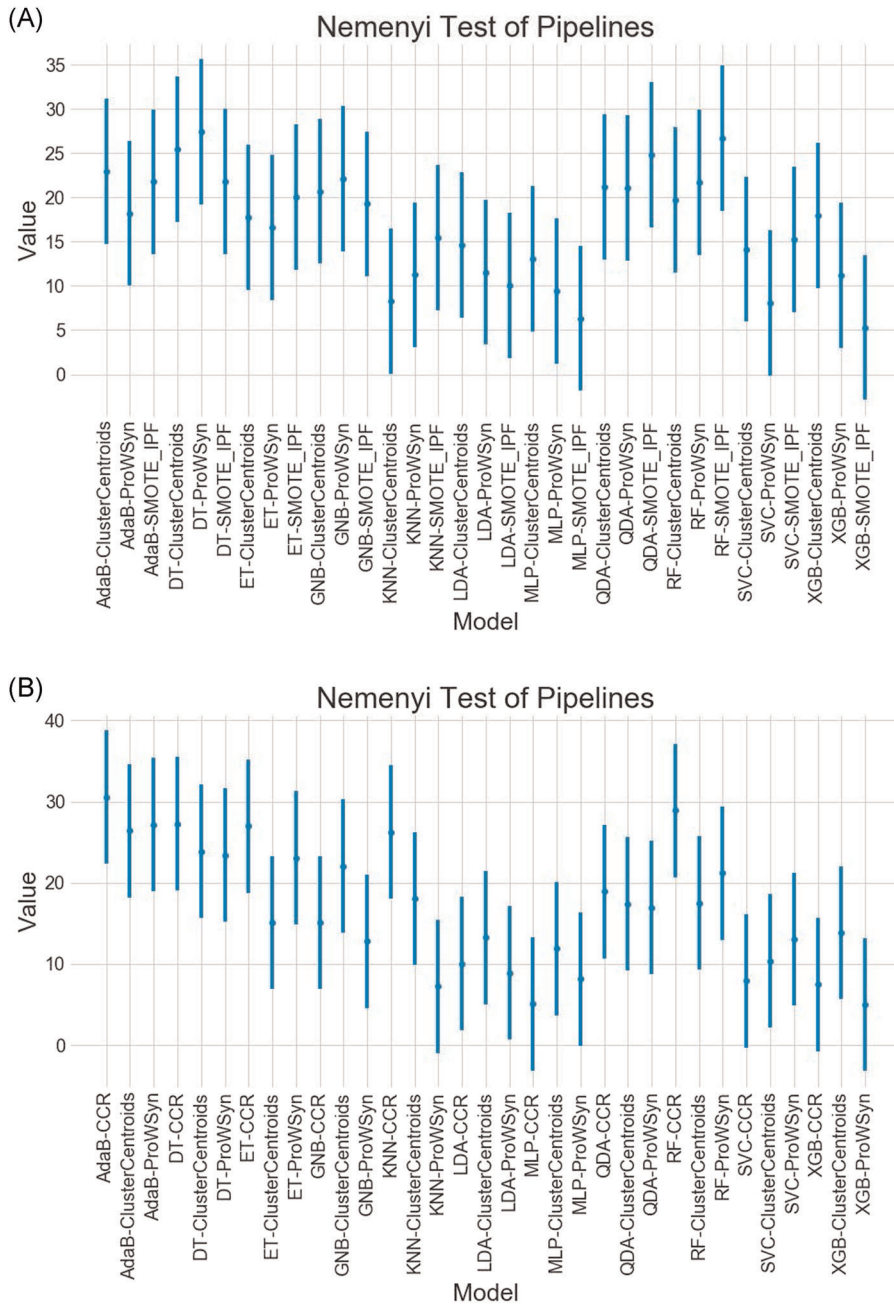

FIGURE 5 Nemenyi test—scores of model pipelines (A) Degotalls case study, (B) Castellfollit case study [Color figure can be viewed at wileyonlinelibrary.com]

4.2.4 | Pipeline selection

In Figure 5, there is an illustration of the Friedman and Nemenyi techniques, which are computed as described in Section 3.9, for the Degotalls (A) and Castellfollit (B) case studies. This process provides enough evidence to identify which of the methods are more statistically

significant than the others. We use a critical distance corresponding to 95% statistical significance. Please note that between two algorithmic approaches there is a statistical significance when the lines do not overlap. The best algorithm is the one with the minimum rank value.

Figure 5A shows that the XGBoost classifier properly parameterized and paired with the SMOTE-IPF oversampler with 35 features performs significantly better than the majority of the machine learning pipelines examined. In addition, the MLP classifier paired with the SMOTE-IPF undersampler with 35 features could also be an acceptable solution.

Regarding the second case study in Figure 5B, the XGBoost classifier properly parameterized and paired with the ProWSyn oversampler with 30 features is the best machine learning pipeline by a significant margin. Furthermore, the MLP classifier accompanied by the CCR oversampler with 30 features could also be an alternative acceptable solution.

4.2.5 | Feature importance

In Figure 6, there is a visualization of permutation feature importance technique, as described and explained in Section 3.10. Please note that the colorization refers to the distinct clusters acquired from the K-means algorithm. For the calculation of the importance of feature in the Degotalls case study, we use the XGB classifier paired with the SMOTE-IPF oversampler with 35 features, which we concluded performed better than the majority of all other methods tested, while for the Castellfollit case study, we utilize the XGB classifier paired with the ProWSyn oversampler with 30 features.

Observing Figure 6A,B, we can conclude that the most important features for each case study appear to be the same. Moreover, the highly important features for rockfall event detection are the coordinates of the point clouds.

4.3 | Ablation study

An ablation study is crucial for the development of an intelligent system because it helps to understand the contribution of each component to the overall system. Table 3 portrays how the balanced accuracy metric is increased with the addition of each proposed component in our analysis framework, in the Degotalls (A) and Castellfollit (B) case studies. It shows the increment of the aforementioned metric across all models ($\overline{Acc_b}$) and the best performing model (Acc_b^{best}) in each phase. The baseline is considered a state in which all the models are trained and evaluated without prior resampling of data, parameterization and feature selection. It is clear in both case studies that the addition of each component provides an interesting performance increment, improving the accuracy and robustness of the utilized method.

4.4 | Prototype implementation

The implementation of a web-based system for rockfall detection is presented in this section. Informatively, the end system incorporates the best intelligent pipeline according to our performance evaluation. To showcase the effectiveness of our intelligent system, we have designed a simple application that detects whether the input point cloud data are considered candidates for a rockfall event. Figure 7 illustrates the architecture of the developed prototype system for

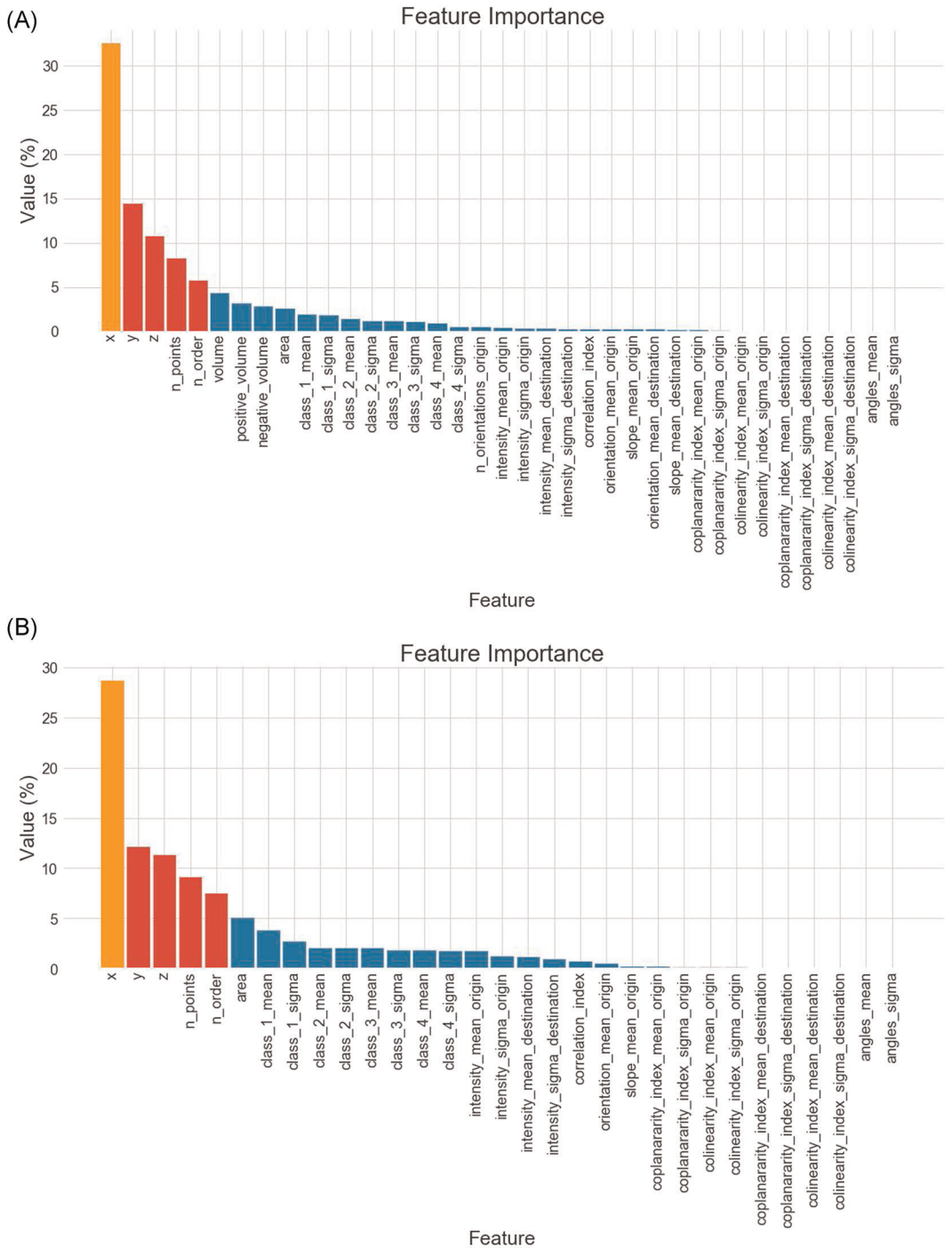

FIGURE 6 Feature Importance. Orange color denotes high importance, blue and red denote mid and low importance respectively (A) Degotalls case study and (B) Castellfollit case study [Color figure can be viewed at wileyonlinelibrary.com]

the visualization of the predictions. A basic but well-designed table is used to source input feature values in the front-end, as shown in Figure 8, and a button that triggers the initialization of the detection procedure. In the back-end, we normalize the data entered by the users, if needed, and generate a prediction using our intelligent framework, which is the pretrained and loaded machine learning pipeline. Finally, the results are displayed to the user in a fast and accurate manner.

Our web application is placed at this GitHub repository. In addition, we used the Python programming language to develop our models and run our experiments. Particularly, for the development of the models, we mainly use the Scikit-learn Python library.[97] The Matplotlib library is used for visualization.[98] Also, for developing the web application, we utilize a microweb framework, which is called Flask.[99]

5 | FINAL OBSERVATIONS

In this section, we provide the final observations of our experiments and analysis. In the particular case studies, we conclude that some of the resampling methods and machine learning models examined could be used to form intelligent pipelines to detect rockfall events. In the case of the Degotalls mountain wall TLS data, we come to the conclusion that the XGBoost classifier using a linear booster, namely *gblinear*, with a learning rate of 0.1 and 50 estimators utilizing 35 features of the input data set accompanied by the SMOTE-IPF resampler, achieved the best outcome, with a robust balanced accuracy score of 95%. Furthermore, in the Castellfollit case study, we conclude that the XGBoost classifier with a linear booster, a learning rate of 0.1 and 100 estimators utilizing 30 features accompanied by the ProWSyn resampler achieved the best balanced accuracy score of 94%.

Intuitively, we could say that the second-order gradients and the advanced regularization of the XGBoost algorithm during the learning phase, help to identify better the data relations. In addition, using the *gblinear* booster, the algorithm builds multiple regularized

TABLE 3 Ablation study—results based on Acc_b metric

(A) Degotalls case study			
Method	$\overline{Acc_b}$	**Error (%)**	Acc_b^{best}
Baseline	0.78	7.65	0.84
+Resampling	0.85	3.25	0.89
+Model parameterization	0.89	4.65	0.94
+Feature selection	0.91	4.45	0.95
(B) Castellfollit case study			
Method	$\overline{Acc_b}$	**Error (%)**	Acc_b^{best}
Baseline	0.56	18.04	0.80
+Resampling	0.68	8.21	0.82
+Model parameterization	0.79	15.93	0.93
+Feature selection	0.82	11.75	0.94

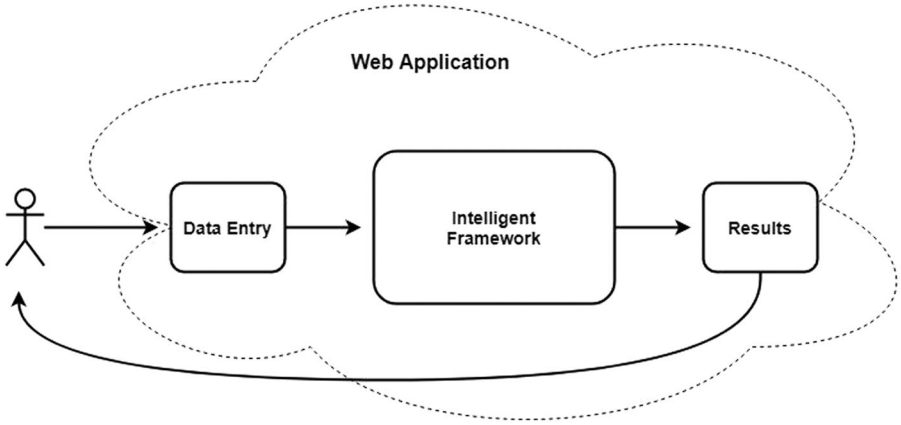

FIGURE 7 Rockfall detection—web application architecture [Color figure can be viewed at wileyonlinelibrary.com]

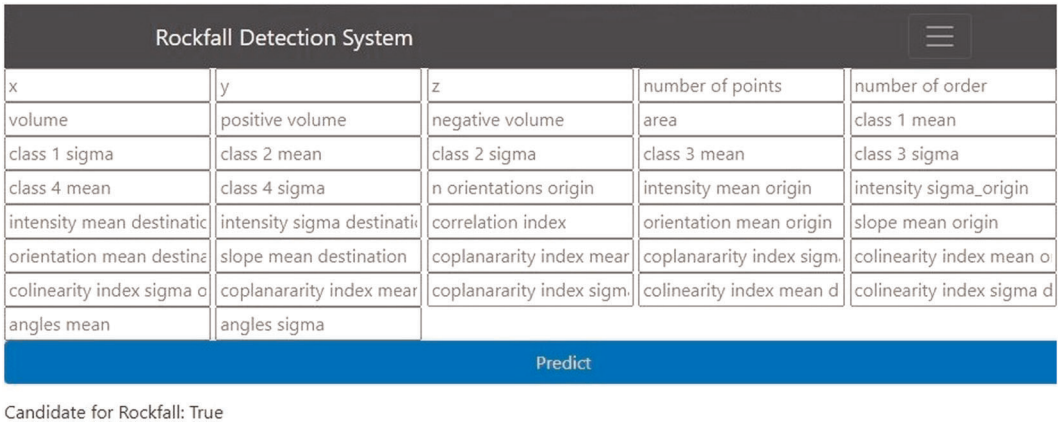

FIGURE 8 Rockfall detection—web application [Color figure can be viewed at wileyonlinelibrary.com]

linear models to later incorporate them in a generalized linear model with advanced regularization. It seems that learning inner linear models and then additively producing a generalized linear model facilitates the achievement of a higher accuracy than the other algorithms in both case studies while remaining fast. Additionally, the Castellfollit case study appears to be more difficult than the Degotalls case study in terms of model learning, because of its data imbalance of 0.4% compared to 1% in the Degotalls study. This can also be justified by the fact that the XGBoost algorithm utilized twice as many estimators in Castellfollit than Degotalls.

Moreover, the ablation study, portrayed in Table 3 justifies the need for each component of our system. The increment in balanced accuracy metric seems to be significant with the addition of the special framework stages, namely resampling, model selection and parameterization and feature selection. Specifically, the most significant addition appears to be the resampling module of our system, displaying an increment in accuracy from the baseline of approximately 9% and 21% in the Degotalls and Castellfollit case studies, respectively. Additionally, in general, an ablation study provides an interesting view of how the developed

intelligent system behaves by adding additional components. This is a good practice to identify the performance gains of a developed system.

To provide further evidence for our developed framework, we visualize the predicted point cloud data along with the original data of the Degotalls case study, as displayed in Figure 9. We chose to display the former visualization because it includes four incorrectly classified rockfall events representing roughly 6% of the total rockfall events in the data set, as depicted in Figure 4A. For clarification purposes, in the Castellfollit case study, all rockfall events in the initial data set were correctly classified by the XGBoost classifier using 30 features paired with the ProWSyn resampler, which is the best selected pipeline, as discussed above.

5.1 | Practical implications

Point cloud data of a 3D scan may contain millions of points with highly detailed information, but they also contain scattered, disjointed information and in most cases with a lot of noise.[100] Geoscientists, to properly deal with such huge data and correctly identify rockfall candidates, go through a semimanual repetitive process, as explained in Section 3.1. The proposed intelligent system will be a key tool in the hands of geology engineers, scientists, and any individual working at the intersection of the geology domain and decision support systems. It aims to provide meaningful and accurate insights into the task of rockfall detection in distinct geological contexts. Our learning methodology provides a robust and accurate way to identify rockfalls using point clouds. Indeed, our automatic intelligent system advances the accuracy and efficiency of the geoscientists' work, saving a lot of time from their work-routines on identifying such events.

Moreover, the final intelligent system could easily be incorporated as a stand-alone component in a rockfall monitoring system providing support and warning alarms. The web application that incorporates our intelligent system could automate the whole rockfall detection process by providing a user-friendly front-end, explained in Section 4.4 and shown in Figure 8.

Particular practical implications of our proposed intelligent system, which enhance the accuracy and efficiency of the geoscientists' work, are further explained below with examples. For instance, having identified the best intelligent pipeline for a specific mountain cliff could aid the rockfall detection in other similar geological contexts, that is, similar mountain cliff types. This is especially relevant for geologists as they regularly capture point clouds from the same mountains to observe changes in their surface structure. Therefore, having a pretrained model on such mountains is essential to observe and detect the evolution of changes automatically. This means, that in a run of our web-based application, using the pretrained best intelligent pipeline derived by analysing a mountain cliff, an individual could identify rockfall candidates in a similar mountain. In addition, taking a closer look at the outcome of our intelligent system, that is, a predicted rockfall candidate cluster of points, geoscientists could later focus on this specific region of the mountain cliff and take additional 3D scans of only this region to be further processed and analysed. To sum up, our system accelerates both the process and the analysis of identifying rockfall events.

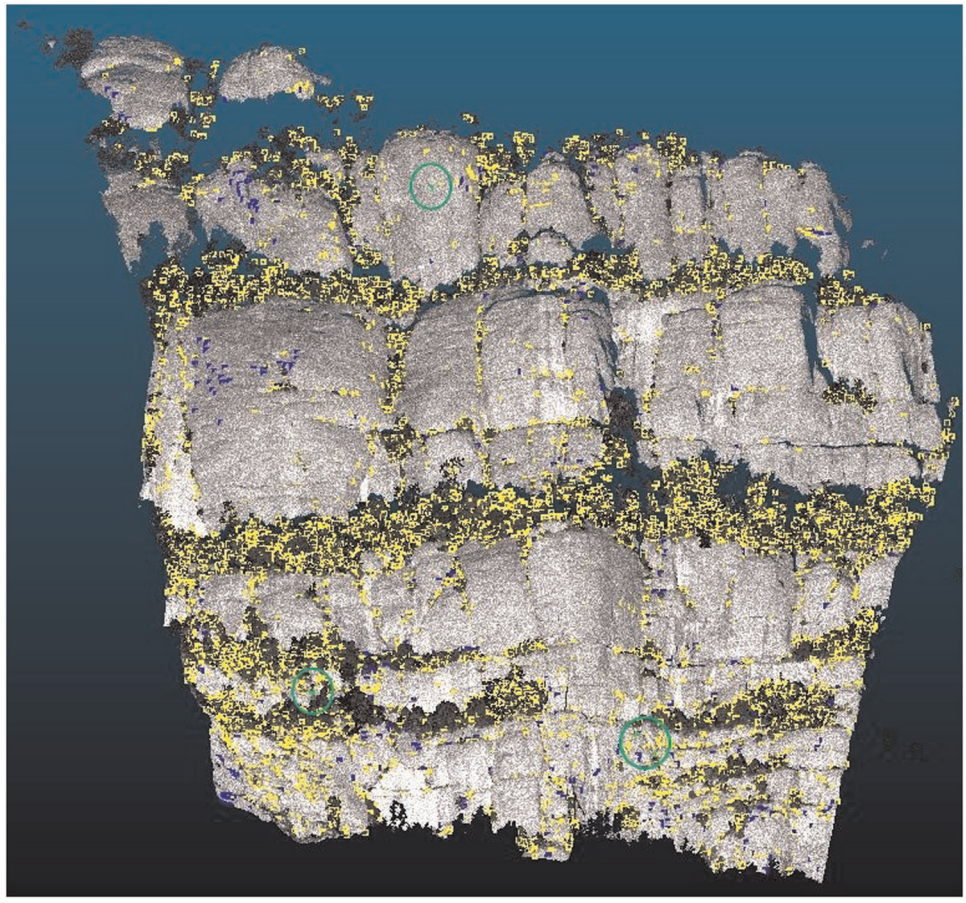

FIGURE 9 Point clouds visualization—Degotalls case study. We denote the correctly and incorrectly classified instances with yellow and blue colors, respectively. We further highlight them in green and circle, the events that are initially rockfall events but were incorrectly classified as not rockfall [Color figure can be viewed at wileyonlinelibrary.com]

5.2 | Limitations

Currently, the proposed intelligent system is analyzed and applied in two distinct geological contexts, that is, two types of mountain cliffs. The potential next steps are to enrich our database with more point cloud data coming from various geological contexts. Additionally, the use of clustering techniques on geological point cloud data increases resource efficiency by decreasing the processing time and memory footprint of the data but in some cases it eliminates some detailed geometrical and topological properties of them. Thus, this can be considered another limitation of the current study, which, however, could be lifted by processing and analysing clustered with raw point cloud data simultaneously.

Although deep learning models are popular in a vast majority of challenging engineering tasks, they are not analysed in this study. On the other hand, even though advanced neural network architectures may produce higher accuracy with lower error rates in a great variety of engineering tasks, they are not the panacea for every specific task they are employed to. The

case of analyzing imbalanced geological point cloud data with deep learning needs to be carefully investigated to properly select the right methods and algorithms to deal with the challenging task of rockfall and landslide detection.

6 | CONCLUSION AND FUTURE WORK

Rockfall events are considered to be a great hazard in multiple regions across the world. People working in the geology domain put a lot of effort into identifying such events. However, rockfall event detection utilizing data from remote sensors such as LiDAR or commercial cameras still need specialized analysis tools that can improve the accuracy and efficiency of the distinction of such events from other changes triggered by edge effects, data noise, vegetation, and so on. We believe that an intelligent decision assisting tool would facilitate geologists' work. However, the development of such a tool seems to be a complex process, due to the inherent nature of the data.

This study presents a thorough examination of several computational approaches for detecting rockfall events. In particular, we focus on the development of an end-to-end machine learning framework able to analyze clustered point cloud data from a TLS device. In the presented framework, due to the imbalanced nature of the rockfall event detection, we implement various resampling methodologies, parameterize several machine learning models and create intelligent pipelines to tackle the abovementioned task.

We experimentally evaluate our framework on two case studies, involving data from TLS measurements of a cliff at the Degotalls, Montserrat Massif in Barcelona and the cliff at the Castellfollit de la Roca in Girona, both located in Spain. We introduce and examine our design analytically and validate it experimentally. These actions clarify the components of the overall process required to set up the pillars of a reliable and intelligent rockfall detection system. We demonstrate the feasibility of creating a practical system for detecting rockfalls.

Our study provides an opportunity to clarify some key issues and provide concrete scientific and technical solutions at the intersection of geology and machine learning. There is a major potential for further additional research on the identification of rockfalls. Additionally, raw point cloud data carry an enormous amount of information and utilizing such information effectively and efficiently remains a challenging task. For this, a system utilizing raw point cloud data as input is being developed, in which the proposed methodology is used in conjunction with various added machine learning and neural network based methodologies to detect rockfall events directly from temporal point cloud data, showing good potential. Analyzing raw point cloud data without any grouping techniques using sophisticated deep learning architectures may bring advances in rockfall detection, because of the complete utilization of the inner geometrical and topological features of the data.

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 860843. In addition, the authors want to acknowledge the support from PROMONTEC (CGL2017-84720-R AEI/FEDER, UE) and SALTEC CGL2017-85532-P (AEI/FEDER, UE) projects, founded by the Spanish MINEICO, and AGAUR project 2016 DI 069 (Agència de Gestió d'Ajuts Universitaris i de Recerca). Data from Castellfollit de la Roca were acquired with the support of the Spanish Ministry of Science and Education (predoctoral grant 2004-1852) and funded by the Natural

Park of the Garrotxa Volcanic Field (PNZVG) and the following projects: MEC CGL2006-06596 (DALMASA), TopoIberia CSD2006-0004/Consolider-Ingenio2010, MEC CGL2010-18609 (NUTESA). Data from Montserrat Massif were funded by the Institut Cartogràfic i Geològic de Catalunya (ICGC). Anna Puig and Maria Salamó also acknowledge Generalitat de Catalunya, for its support under project 2017-SGR-341. Moreover, we would like to thank Nicolas Pascual González for his contribution in this study.

ENDNOTES

*http://www.ub.edu/risknat/

**http://www.ub.edu/geomodels/

ORCID

Thanasis Zoumpekas http://orcid.org/0000-0002-3736-1155
Anna Puig https://orcid.org/0000-0002-2184-2800
Maria Salamó https://orcid.org/0000-0003-1939-8963
David García-Sellés https://orcid.org/0000-0002-5995-5712
Marta Guinau https://orcid.org/0000-0002-2898-4625

REFERENCES

1. Robbins BA, Stephens IJ, Marcuson WF. Geotechnical engineering. In: *Encyclopedia of Geology*. Elsevier; 2021:377-392.
2. Robiati E, Vanneschi F, Venn C. Application of remote sensing data for evaluation of rockfall potential within a quarry slope. *ISPRS Int J Geo-Inform*. 2019;8:367.
3. DiFrancesco PM, Bonneau D, Hutchinson DJ. The implications of M3C2 projection diameter on 3D semi-automated rockfall extraction from sequential terrestrial laser scanning point clouds. *Remote Sensing*. 2020;12:1885.
4. Fanos AM, Pradhan B, Alamri A, Lee C-W. Machine learning-based and 3D kinematic models for rockfall hazard assessment using LiDAR data and GIS. *Remote Sensing*. 2020;12:1755.
5. Dramsch JS. 70 years of machine learning in geoscience in review. In: *Advances in Geophysics*. Vol 61. Academic Press Inc.; 2020:1-55.
6. Karpatne A, Ebert-Uphoff I, Ravela S, Babaie HA, Kumar V. Machine learning for the geosciences: challenges and opportunities. *IEEE Trans Knowl Data Eng*. 2019;31:1544-1554.
7. Crosta GB, Agliardi F. Failure forecast for large rock slides by surface displacement measurements. *Canadian Geotech J*. 2003;40:176-191.
8. Carrara A. Multivariate models for landslide hazard evaluation. *J Int Assoc Math Geol*. 1983;15:403-426.
9. Abellán A, Calvet J, Vilaplana JM, Blanchard J. Detection and spatial prediction of rockfalls by means of terrestrial laser scanner monitoring. *Geomorphology*. 2010;119:162-171.
10. Abellán A, Oppikofer T, Jaboyedoff M, Rosser NJ, Lim M, Lato MJ. Terrestrial laser scanning of rock slope instabilities. *Earth Surface Process Landforms*. 2014;39:80-97.
11. Lague D, Brodu N, Leroux J. Accurate 3D comparison of complex topography with terrestrial laser scanner: application to the Rangitikei canyon (N-Z). *ISPRS J Photogrammetry Remote Sensing*. 2013;82:10-26.
12. Oppikofer T, Jaboyedoff M, Blikra L, Derron M-H, Metzger R. Characterization and monitoring of the Åknes rockslide using terrestrial laser scanning. *Natural Hazards Earth Syst Sci*. 2009;9:1003-1019.
13. Jaboyedoff M, Oppikofer T, Abellán A, et al. Use of LIDAR in landslide investigations: a review. *Natural Hazards*.2012;61:5-28.
14. Prakash N, Manconi A, Loew S. Mapping landslides on EO data: performance of deep learning models vs. traditional machine learning models. *Remote Sensing*.2020;12:346.

15. Xiao L, Zhang Y, Peng G. Landslide susceptibility assessment using integrated deep learning algorithm along the China-Nepal highway. *Sensors*. 2018;18.
16. Fanos AM, Pradhan B, Mansor S, Yusoff ZMd, Abdullah AFbin. A hybrid model using machine learning methods and GIS for potential rockfall source identification from airborne laser scanning data. In: *Landslides*. Vol 15, 2018:1833-1850.
17. Pham BT, Shirzadi A, Tien BD, Prakash I, Dholakia MB. A hybrid machine learning ensemble approach based on a Radial Basis Function neural network and Rotation Forest for landslide susceptibility modeling: a case study in the Himalayan area, India. *Int J Sediment Res*. 2018;33:157-170.
18. Bandura L, Halpert AD, Zhang Z. Machine learning in the interpreter's toolbox: unsupervised, supervised, and deep learning applications. In: *2018 SEG International Exposition and Annual Meeting (SEG 2018)*. Society of Exploration Geophysicists; 2019:4633-4637.
19. Ghorbanzadeh O, Blaschke T, Gholamnia K, Meena SR, Tiede D, Aryal J. Evaluation of different machine learning methods and deep-learning convolutional neural networks for landslide detection. *Remote Sensing*. 2019;11:196.
20. Wang H, Zhang L, Yin K, Luo H, Li J. Landslide identification using machine learning. *Geosci Front*. 2020.
21. Bui DT, Tsangaratos P, Nguyen VT, Liem NV, Trinh PT. Comparing the prediction performance of a Deep Learning Neural Network model with conventional machine learning models in landslide susceptibility assessment. *Catena*. 2020;188:104426.
22. James MR, Robson S. Straightforward reconstruction of 3D surfaces and topography with a camera: accuracy and geoscience application. *J Geophys Res: Earth Surface*. 2012;117.
23. Lissak C, Bartsch A, De Michele M, et al. Remote sensing for assessing landslides and associated hazards. *Survey Geophys*. 2020;41:1391-1435.
24. Mayr A, Rutzinger M, Geitner C. Multitemporal analysis of objects in 3D point clouds for landslide monitoring. *Int Arch Photogrammetry Remote Sensing Spatial Inform Sci—ISPRS Archives*. 2018;42:691-697.
25. Weidner L, Walton G, Kromer R. Classification methods for point clouds in rock slope monitoring: a novel machine learning approach and comparative analysis. *Eng Geol*. 2019;263.
26. Weidner L, Walton G, Kromer R. Generalization considerations and solutions for point cloud hillslope classifiers. *Geomorphology*. 2020;354:107039.
27. Kong D, Wu F, Saroglou C. Automatic identification and characterization of discontinuities in rock masses from 3D point clouds. *Eng Geol*. 2020;265:105442.
28. Huu PNT. Lidar point cloud classification using expectation maximization algorithm. *Int J Comput Sci Inform Technol*. 2020;12:1-13.
29. Weidner L, Walton G, Kromer R. Automated rock slope material classification using machine learning. In: *54th U.S. Rock Mechanics/Geomechanics Symposium*. American Rock Mechanics Association; 2020.
30. Wang Y, Huang J, Tang H. Automatic identification of the critical slip surface of slopes. *Eng Geol*. 2020;273:105672.
31. Fanos AM, Pradhan B. A novel hybrid machine learning-based model for rockfall source identification in presence of other landslide types using LiDAR and GIS. *Earth Syst Environ*. 2019;3:491-506.
32. Bernsteiner H, BroÅová N, Eischeid I, et al. Machine learning for classification of an eroding scarp surface using terrestrial photogrammetry with nir and rgb imagery. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol 5. Copernicus GmbH; 2020:431-437.
33. Loghin AM, Pfeifer N, Otepka-Schremmer J. Supervised classification and its repeatability for point clouds from dense Vhr tri-stereo satellite image matching using machine learning. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol 5, 2020:525-532.
34. Salvini R, Francioni M, Riccucci S, Bonciani F, Callegari I. Photogrammetry and laser scanning for analyzing slope stability and rock fall runout along the Domodossola-Iselle railway, the Italian Alps. *Geomorphology*. 2013185:110-122.
35. Hemalatha T, Ramesh MV, Rangan VP. Effective and accelerated forewarning of landslides using wireless sensor networks and machine learning. *IEEE Sensors J*. 2019;19:9964-9975.
36. Agliardi F, Crosta G, Zanchi A. Structural constraints on deep-seated slope deformation kinematics. *Eng Geol*. 2001;59:83-102.

37. Reichenbach P, Rossi M, Malamud BD, Mihir M, Guzzetti F. A review of statistically-based landslide susceptibility models. *Earth-Sci Rev*. 2018;180:60-91.
38. Micheletti N, Foresti L, Robert S, et al. Machine learning feature selection methods for landslide susceptibility mapping. *Mathe Geosci*. 2014;46:33-57.
39. Abellán A, Jaboyedoff M, Oppikofer T, Vilaplana JM. Detection of millimetric deformation using a terrestrial laser scanner: experiment and application to a rockfall event. *Natural Hazards Earth Syst Sci*. 2009;9:365-372.
40. Mayr A, Rutzinger M, Bremer M, Oude ES, Stumpf F, Geitner C. Object-based classification of terrestrial laser scanning point clouds for landslide monitoring. *Photogramm Record*. 2017;32:377-397.
41. Alfian G, Syafrudin M, Ijaz MF, Syaekhoni MA, Fitriyani NL, Rhee J. A personalized healthcare monitoring system for diabetic patients by utilizing BLE-based sensors and real-time data processing. *Sensors*. 2018;18.
42. Srinivasu PN, SivaSai JG, Ijaz MF, Bhoi AK, Kim W, Kang JJ. Classification of skin disease using deep learning neural networks with mobilenet V2 and LSTM. *Sensors*. 2021;21:2852.
43. Shi M, Sun W, Zhang T, Liu Y, Wang S, Song X. Geology prediction based on operation data of tbm: comparison between deep neural network and soft computing methods. In: *1st International Conference on Industrial Artificial Intelligence, IAI 2019*. Institute of Electrical and Electronics Engineers Inc.; 2019.
44. Chen J, Zeng Z, Jiang P, Tang H. Deformation prediction of landslide based on functional network. *Neurocomputing*. 2015;149:151-157.
45. Wang Z, Jia K. Frustum ConvNet: sliding frustums to aggregate local point-wise features for amodal 3D object detection. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE; 2019:1742-1749.
46. Ijaz MF, Alfian G, Syafrudin M, Rhee J. Hybrid prediction model for type 2 diabetes and hypertension using DBSCAN-based outlier detection, synthetic minority over sampling technique (SMOTE), and random forest. *Appl Sci*. 2018;8.
47. Ijaz MF, Attique M, Son Y. Data-driven cervical cancer prediction model with outlier detection and oversampling methods. *Sensors*. 2020;20:2809.
48. Stumpf A, Kerle N. Object-oriented mapping of landslides using random forests. *Remote Sens Environ*. 2011;115:2564-2577.
49. Zhao L, Wu X, Niu R, Wang Y, Zhang K. Using the rotation and random forest models of ensemble learning to predict landslide susceptibility. *Geomatics Nat Hazards Risk*. 2020;11:1542-1564.
50. Williams R, Brasington J, Vericat D, Hicks M, Labrosse F, Neal M. Monitoring braided river change using terrestrial laser scanning and optical bathymetric mapping. In: *Developments in Earth Surface Processes*. Vol 15. Elsevier B.V.; 2011:507-532.
51. Singh D, Singh B. Investigating the impact of data normalization on classification performance. *Appl Soft Comput J*. 2019;97:105524.
52. Hernandez J, Carrasco-Ochoa JA, Martínez-Trinidad JF. An empirical study of oversampling and undersampling for instance selection methods on imbalance datasets. In: *Lecture Notes in Computer Science* (*including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*). Vol 8258. Berlin, Heidelberg: Springer; 2013:262-269.
53. Junsomboon N, Phienthrakul T. Combining over-sampling and under-sampling techniques for imbalance dataset. In: *ACM International Conference Proceeding Series; Part F1283*. New York, NY: Association for Computing Machinery; 2017:243-247.
54. Yen SJ, Lee YS. Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Syst Appli*. 2009;36:5718-5727.
55. Wang N, Zhao S, Cui S, Fan W. A hybrid ensemble learning method for the identification of gang-related arson cases. *Knowl-Based Syst*. 2021;218:106875.
56. Kovács G. An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets. *Appl Soft Comput J*. 2019;83:105662.
57. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. *J Artif Intell Res*. 2002;16:321-357.
58. Fernandez A, Garcia S, Herrera F, Chawla NV. SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *J Artif Intell Res*. 2018;61:863-905.

59. He H, Bai Y, Garcia EA, Li S. ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: *Proceedings of the International Joint Conference on Neural Networks*. 2008:1322-1328.
60. Barua S, Islam MdM, Murase K. ProWSyn: proximity weighted synthetic oversampling technique for imbalanced data set learning. In: *Lecture Notes in Computer Science* (*including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*). Vol 7819. Berlin, Heidelberg: Springer; 2013: 317-328.
61. Sáez JA, Luengo J, Stefanowski J, Herrera F. SMOTE-IPF: addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Inform Sci*. 2015;291:184-203.
62. Cao Q, Wang S. Applying over-sampling technique based on data density and cost-sensitive SVM to imbalanced learning. In: *Proceedings—2011 4th International Conference on Information Management, Innovation Management and Industrial Engineering, ICIII 2011*. Vol 2; 2011:543-548.
63. Zhou B, Yang C, Guo H, Hu J. A quasi-linear SVM combined with assembled SMOTE for imbalanced data classification. In: *Proceedings of the International Joint Conference on Neural Networks*; 2013.
64. Lee J, Kim NR, Lee JH. An over-sampling technique with rejection for imbalanced class learning. In: *ACM IMCOM 2015—Proceedings*. Vol 2015. New York, NY: Association for Computing Machinery, Inc, 1-6.
65. Batista GEAPA, Prati RC, Monard MC. MC. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explor Newsletter*. 2004;6:20-29.
66. Tomek I. Two modifications of CNN. *IEEE Trans Syst Man Cybernet*. 1976;SMC-6:769-772.
67. Koziarski M, Wozniak M. CCR: a combined cleaning and resampling algorithm for imbalanced data classification. *Int J Appl Math Comput Sci*. 2017;27:727-736.
68. Sandhan T, Choi JY. Handling imbalanced datasets by partially guided hybrid sampling for pattern recognition. In: *Proceedings—International Conference on Pattern Recognition*. Institute of Electrical and Electronics Engineers Inc.; 2014:1449-1453.
69. Nakamura M, Kajiwara Y, Otsuka A, Kimura H. LVQ-SMOTE - Learning vector quantization based synthetic minority over-sampling technique for biomedical data. *BioData Mining*. 2013;6:16.
70. Gazzah S, Amara NEB. New oversampling approaches based on polynomial fitting for imbalanced data sets. In: *DAS 2008—Proceedings of the 8th IAPR International Workshop on Document Analysis Systems*. 2008:677-684.
71. Stefanowski J, Wilk S. Selective pre-processing of imbalanced data for improving classification performance. In: *Lecture Notes in Computer Science* (*including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*). Vol 5182. Berlin, Heidelberg: Springer; 2008:283-292.
72. Sharma S, Bellinger C, Krawczyk B, Zaiane O, Japkowicz N. Synthetic oversampling with the majority class: a new perspective on handling extreme imbalance. In: *Proceedings—IEEE International Conference on Data Mining, ICDM*. Vol 2018-Novem. Institute of Electrical and Electronics Engineers Inc.; 2018: 447-456.
73. Ma Z, Mei G, Piccialli F. Machine learning for landslides prevention: a survey. *Neural Comput Appl*. 2020.
74. Zhou Z-H. Ensemble learning. In: *Encyclopedia of Biometrics*. Boston, MA: Springer US; 2009:270-273.
75. Hastie T, Tibshirani R, Friedman J. *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY: Springer; 2009.
76. D'Ambrosio A, Tutore VA. Conditional classification trees by weighting the gini impurity measure. In: *Studies in Classification, Data Analysis, and Knowledge Organization*. Berlin, Heidelberg: Springer; 2011: 273-280.
77. Awad M, Khanna R, Awad M, Khanna R. Support vector machines for classification. In: *Efficient Learning Machines*. Apress; 2015:39-66.
78. Murtagh F. Multilayer perceptrons for classification and regression. *Neurocomputing*. 1991;2:183-197.
79. Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. *J Comput Syst Sci*. 1997;55:119-139.
80. Zhu J, Zou H, Rosset S, Hastie T. *Multi-class AdaBoost,* Technical Report; 2009.
81. Geurts P, Ernst D, Wehenkel L. Extremely randomized trees. *Mach Learn*. 2006;63:3-42.
82. Chen T, Guestrin C. XGBoost: a scalable tree boosting system. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. August 13–17. Association for Computing Machinery; 2016:785-794.

83. Hwang S, Guevarra IF, Yu B. Slope failure prediction using a decision tree: a case of engineered slopes in South Korea. *Eng Geol*. 2009;104:126-134.
84. Marjanović M, Bajat B, Abolmasov B, Kovačević M. Machine learning and landslide assessment in a GIS environment. In: *Advances in Geographic Information Science*. Cham: Springer; 2018:191-213.
85. Lee S, Ryu J-H, Won J-S, Park H-J. Determination and application of the weights for landslide susceptibility mapping using an artificial neural network. *Eng Geol*. 2004;71:289-302.
86. Chen W, Xie X, Wang J, et al. A comparative study of logistic model tree, random forest, and classification and regression tree models for spatial prediction of landslide susceptibility. *CATENA*. 2017;151:147-160.
87. Hong H, Liu J, Bui DT, et al. Landslide susceptibility mapping using J48 decision tree with AdaBoost, bagging and rotation forest ensembles in the Guangchang area (China). *CATENA*. 2018;163:399-413.
88. Forman G, Cohen I. Learning from little: comparison of classifiers given little training. In: *Lecture Notes in Computer Science* (*including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*). Vol 3202. Springer Verlag; 2004:161-172.
89. Klein D, Manning CD. Conditional structure versus conditional estimation in NLP models. In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing—EMNLP '02*. Vol 10. Morristown, NJ: Association for Computational Linguistics; 2002:9-16.
90. Hong H, Naghibi SA, Moradi DM, Pourghasemi HR, Chen W. A comparative assessment between linear and quadratic discriminant analyses (LDA-QDA) with frequency ratio and weights-of-evidence models for forest fire susceptibility mapping in China. *Arabian J Geosci*. 2017;10:167.
91. Li Y, Chen W. A comparative performance assessment of ensemble learning for credit scoring. *Mathematics*. 2020;8:1756.
92. Kraskov A, Stögbauer H, Grassberger P. Estimating mutual information. *Phys Rev E—Stat Phys Plasmas Fluids Related Interdiscip Topics*. 2004;69:16.
93. Demšar J. *Statistical Comparisons of Classifiers over Multiple Data Sets*, Technical Report; 2006.
94. Mozetič I, Torgo L, Cerqueira V, Smailović J. *How to Evaluate Sentiment Classifiers for Twitter Time-Ordered Data?*; 2018.
95. Breiman L. Random forests. *Mach Learn*. 2001;45:5-32.
96. Brodersen KH, Ong CS, Stephan KE, Buhmann JM. The balanced accuracy and its posterior distribution. In: *Proceedings—International Conference on Pattern Recognition*. 2010:3121-3124.
97. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: machine learning in Python. *J Mach Learn Res*. 2011;12:2825-2830.
98. Hunter JD. Matplotlib: a 2D graphics environment. *Comput Sci Eng*. 2007;9:90-95.
99. The Pallets Projects. *Flask|The Pallets Projects*; 2010.
100. Bello SA, Yu S, Wang C, Adam JM, Li J. Review: deep learning on 3D point clouds. *Remote Sensing*. 2020; 12:1729.

SUPPORTING INFORMATION

Additional Supporting Information may be found online in the supporting information tab for this article.

How to cite this article: Zoumpekas T, Puig A, Salamó M, García-Sellés D, Blanco Nuñez L, Guinau M. An intelligent framework for end-to-end rockfall detection. *Int J Intell Syst*. 2021;1-32. https://doi.org/10.1002/int.22557