Computers & Geosciences 184 (2024) 105531

Contents lists available at ScienceDirect

Computers and Geosciences

journal homepage: www.elsevier.com/locate/cageo

Improving the quality evaluation process of machine learning algorithms applied to landslide time series analysis

Marco Conciatori *, Alessandro Valletta, Andrea Segalini

Department of Engineering and Architecture, University of Parma, Parco Area delle Scienze 181/a, 43124, Parma, Italy

ARTICLE INFO

Keywords:
Machine learning
Evaluation metrics
Landslide
Time series
Displacement forecasting

ABSTRACT

The introduction of Machine Learning (ML) in the geotechnical community has led to numerous applications for monitoring data elaboration. These techniques demonstrate promising performance in comparison to conventional methods aimed at determining the future behavior of a landslide. In this context, it is fundamental to have access to reliable methodologies and procedures to assess the quality of algorithms' predictions. This article proposes an improved method for evaluating ML algorithms applied to landslide time series analysis. The method relies on modified metrics that are sensible to biased classification due to imbalanced datasets, also enabling the evaluation of both regression and classification models using the same criteria. The calculated metrics include Accuracy, Precision, Recall, and F1-Score, each one representing a different aspect of the forecasting model effectiveness. Results obtained from the application of the proposed method to datasets collected by automated monitoring systems proved to be informative of the performance of the model and provides the means for objective comparison with other forecasting algorithms, making it a valuable tool to improve the prediction process reliability. In particular, the custom metrics allowed for a better evaluation of algorithms skewed in favor of the dominant class/classes, which are common occurrence in landslide displacement datasets. In these cases, the proposed approach highlighted the inability of the forecasting model in predicting critical events, presenting a more accurate representation of its performances compared to results obtained with standard approaches.

List of symbols and abbreviations.

Symbol/ Abbreviation	Meaning
ML	Machine Learning
TP	True Positives
TN	True Negatives
FP	False Positives
FN	False Negatives
F1	F1 Score/F1 Metric
MUMS	Modular Underground Monitoring System
MEMS	Micro Electro-Mechanical Systems
n	Number of (displacement) classes
$\mathbb{R}$	Set of all real numbers
$\mathbb{N}$	Set of all natural numbers
i,j	Indices to identify one element inside matrices: i is the row number, j is the column number
a_{ij}	Element of the confusion matrix identified by the coordinates row i, column j
T	Ordered set of thresholds (strictly ascending order). Used for the discretization of continuous displacement values
K	Index for thresholds set T
M	Class index
IP	Imbalance Percentage of the predictions
p_m	Proportion of correct predictions for class m
c_m	Number of correct predictions for class m
t_m	Cardinality of class m

(*continued on next column*)

(*continued*)

1. Introduction

Machine Learning (ML) is a field of artificial intelligence that focuses on the development of algorithms and models that can learn from data (Alzubi et al., 2018). Its history dates back to the mid-20th century, when researchers began developing rule-based systems and decision trees (Denes and Mathews, 1960). In the 1970s, Artificial Neural Networks (ANN) emerged as a powerful technique for solving complex problems such as image and speech recognition. The 2000s witnessed

* Corresponding author.
E-mail address: marco.conciatori@unipr.it (M. Conciatori).

https://doi.org/10.1016/j.cageo.2024.105531
Received 22 June 2023; Received in revised form 7 November 2023; Accepted 10 January 2024
Available online 17 January 2024

significant advances in ML, driven by the rise of big data and the development of deep learning algorithms (Bengio et al., 2021; Parvat et al., 2017; Schmidhuber, 2015).

At the time of writing, according to common search engines, the entirety of the articles available on this topic up to year 2012 is less than the papers published in the sole 2022. For example, papers available on Scopus up to 2012 are roughly 5,000, while over 40,000 studies were published in 2022 only. Fig. 1 shows a comparison according to different sources underlining the increasing number of scientific studies focused on Machine Learning.

Today, Machine Learning is a rapidly growing field that has a significant impact on many aspects of our lives, and applications spanning many industry sectors, including healthcare (Abugabah et al., 2022; Liu et al., 2022; Qayyum et al., 2021), autonomous vehicles (Aradi, 2022; Juyal et al., 2021), natural language understanding and generation (Bubeck et al., 2023; Feder et al., 2022; Wolf et al., 2020), decision making support and recommender systems (Adlung et al., 2021; Bell and Koren, 2007; Khosravi et al., 2019). In recent years, ML has expanded to many different fields, including geotechnics, where one of its promising applications is modeling and forecasting complex environments (Benbouras et al., 2021; Chang et al., 2022; Koopialipoor et al., 2022; Nava et al., 2023; Soranzo et al., 2022; Tokgozoglu et al., 2023).

Being able to verify the effectiveness and practicality of a ML-based model for prediction purposes plays a central role in the assessment process of the algorithm performance (Tilahun and Korus, 2023). In this context, ML has suffered for many years for the lack of standardized evaluation methods and benchmarks (Kotthoff et al., 2011; Reich and Barai, 1999). It is often hard to objectively compare different algorithms because many studies use their own arbitrary testing criteria and replicating others' experiments is not a trivial task (it may even be impossible if the relevant data is not public). This has been recognized as a real problem in ML research, which also includes ML applied to geotechnics (Olson et al., 2017). The two issues addressed in this study are the difficulty of correctly assessing the forecasting algorithms performances in the presence of imbalanced datasets and the problem of comparing different models, in particular regression and classification models.

The solution to both problems is to define metrics that are:

a) useful in evaluating landslide-forecasting algorithms (able to distinguish good and bad models in case of imbalanced predictions)
b) able to be used on any landslide-forecasting algorithm without modifications (must work on both regression and classification models)

This paper proposes the use of modified classification metrics for evaluating landslide forecasting algorithms (models). These metrics will provide evaluations that are both informative of the performance of the algorithm, and objectively comparable cross-work. There are many libraries that provide implementations of classification metrics (Abadi et al., 2016; Detlefsen et al., 2022; Pedregosa et al., 2011), none of them, however, interpret the inputs in the way required by the examined problem. The more sophisticated versions of the metrics allow for the input of continuous values along with the specification of thresholds for the conversion, but they expect vectors with one cell for each class, and they interpret the values inside the cells as probabilities.

Aside from the differences with existing metrics, the novelty comes from the specific domain of application, i.e., the evaluation of landslide forecasting algorithms. Such models can predict the raw displacement measurements (regression) or discretized class representations (classification). In the first case they are judged on quantitative errors (e.g., Mean Absolute Error, Mean Squared Error) in the second case they are evaluated on classification errors (e.g., Precision, Recall). With the proposed evaluation method, all models can be assessed using the exact same criteria and the results can be compared directly, even between classifiers and regression models. This is possible because classifiers are evaluated normally, while regression models are judged with the same metrics by transforming their output into classes.

2. Material and methods

2.1. Evaluation algorithm

In this research area, datasets are often very imbalanced (Li et al., 2021; Zhang et al., 2022), i.e., there are prolonged low-activity periods (small displacements) and rare bursts of high activity (large displacements). This presents a challenge for the evaluation of the performance of models, highlighted by the following extreme example. Using a dataset with 99% small displacements and 1% large displacements, a model which always predicts "small displacement" will get the correct answer 99% of the time. That, however, is not a good model, because it does not produce an accurate forecasting, being unable to identify the rare large displacements. For this reason, it is important to have more sophisticated metrics that can capture this phenomenon, and correctly assess the efficiency and accuracy of prediction models.

The structure of the proposed algorithm consists of a series of operations and controls, summarized in the flowchart reported in Fig. 2, while the following subsections present additional details regarding every step of the elaboration process. For each input data (the pair "true value-prediction") deriving from a regression model, a conversion procedure is applied in order to change the real values into class indices. Then, the confusion matrix is updated accordingly. Once all input pairs have been examined, all the implemented metrics can be computed. The custom metrics extracted from the confusion matrix are first calculated on a by-class basis, and subsequently averaged to get a single value. With this procedure, the performance in each class has the same weight on the

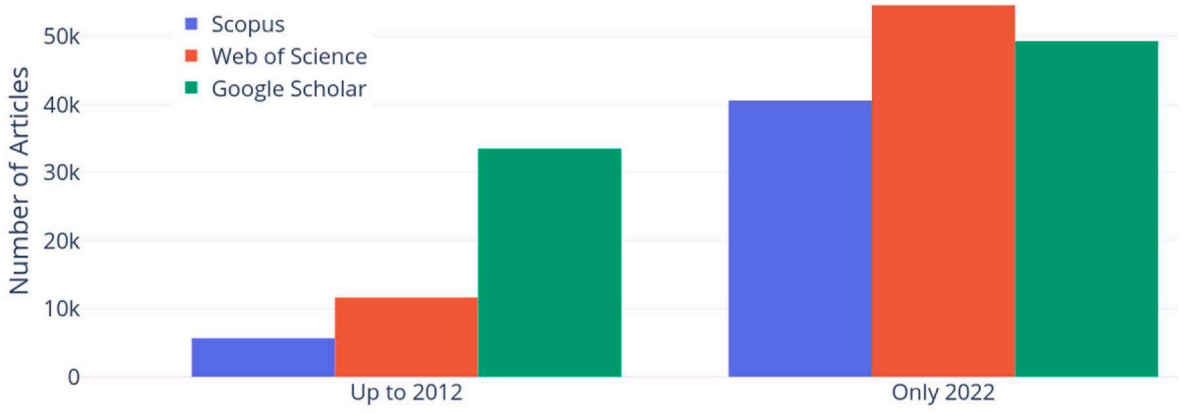

Fig. 1. ML-related articles published.

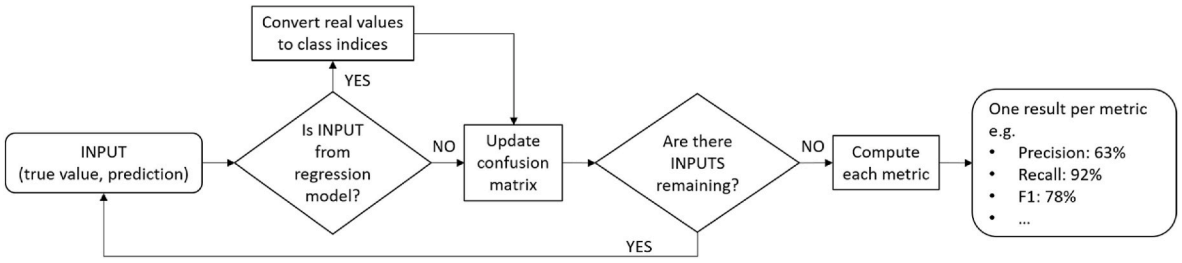

Fig. 2. Structure of the algorithm.

final result, regardless of the number of elements in each class.

2.2. Confusion matrix

The proposed algorithm starts by constructing a confusion matrix for the metrics evaluation procedure. The confusion matrix is typically used to represent the results of classification algorithms; it is a two-dimensional square table with one row and one column for each class of the problem domain (Ting, 2017a). Table 1 and Table 2 present two basic examples of the structure of a confusion matrix for the metrics currently discussed, referring to 2-class and 3-class cases, respectively.

The rows represent the actual classes, while the columns are the predicted classes. It should be noted that the proposed algorithm accepts any number of classes $n \in \mathbb{N}, n \geq 2$. To clarify, in the above 3-class case, the toy dataset is composed of 100 measurements, shown formally by equation (1):

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} = 100 \tag{1}$$

with a_{ij} representing a generic element of the confusion matrix $\forall i,j \in \{1, 2, \ldots n\}$, where $n =$ number of classes.

The measurements are divided in 75 small displacements, 20 medium displacements, 5 large displacements, as can be extracted with equation (2):

$$\sum_{j=1}^{n} a_{1j} = 75 \quad \sum_{j=1}^{n} a_{2j} = 20 \quad \sum_{j=1}^{n} a_{3j} = 5 \tag{2}$$

The values on the main diagonal of the confusion matrix correspond to correct predictions, and rows contain the true displacements. For example, in the last row there are a total of five elements (this implies they are all large displacements) and all of them are placed inside the "large displacements" column, which means the model predicted them correctly. For a negative case, in the first row (small displacements) the model wrongly predicted one of them as "medium displacement", that is why there is a single "1" in the "medium" column of the first ("small") row. Additionally, the confusion matrix contains more than just information about correct and wrong predictions. In fact, it also allows to distinguish between type I (false positive) and type II (false negative) errors for each class (Banerjee et al., 2009).

To assess the quality of a model, the metrics need the output of said model (prediction), along with the correct answer ("target" or "true value"), that must be known from the start. The algorithm can evaluate categorial models, which predict displacement class indices, and regression models, which predict numeric (real) displacements. In the previous examples the indices would correspond to.

- 2-class case:
  - o "1" for small displacements
  - o "2" for large displacements
- 3-class case:
  - o "1" for small displacements
  - o "2" for medium displacements
  - o "3" for large displacements

Table 1
Confusion matrix with a 2-class configuration.

		Displacements	
		Small	Large
Displacements	**Small**	92	3
	Large	0	5

Table 2
Confusion matrix with a 3-class configuration.

		Displacements		
		Small	Medium	Large
Displacements	**Small**	68	1	6
	Medium	2	15	3
	Large	0	0	5

In order to be able to evaluate both types of models, the algorithm automatically converts each input (the pair "true value-prediction") of regression models from continuous values to the corresponding class indices with a threshold-based procedure. This passage describes the generation of integer indices in both cases, which can be exploited to populate the confusion matrix.

Equation (3) describes the discretization function, used to transform true values and outputs of regression models:

$$Discretize(x, T) = \begin{cases} 1, & x < T_1 \\ k, & T_{k-1} \leq x < T_k \\ n, & x \geq T_{n-1} \end{cases} \tag{3}$$

with $x \in \mathbb{R}$ real-valued displacement (both measured and predicted), $T = \{T_1, T_2, \ldots T_{n-1}\} \in \mathbb{R}^{n-1}$ $s.t.$ $\forall k \in \{2, 3, \ldots n-1\} \Rightarrow T_{k-1} < T_k$, $n =$ number of classes, $k \in \{1, 2, \ldots n-1\} =$ threshold index. T is a set of thresholds that divide $\mathbb{R}$ into n intervals. This operation ensures that, from this point onward, true values and predictions will be in the form of class indices, the same type of output of a classifier.

The input vector needed to fill the confusion matrix is composed of the sequence of all single inputs. After the conditional discretization phase, each input is defined as an array of two values (true-class index, predicted-class index). Both indices refer to the ranges of displacements identified by the thresholds.

- The predicted-class index is the output of the model to be evaluated, given its expected input (distinct from the metrics' input discussed up to now), which typically is the combination of various information about the state of a landslide.
- The true-class index is the class index of the actual displacement occurred after the landslide state fed to the model. This means that, to test the model, it is necessary to use past measurements, for which the following evolution is already known.

The confusion matrix is populated using the vector of inputs and applying the following algorithm.

```
PROCEDURE update_confusion_matrix (CM, input):
    FOR EACH element IN input DO
    i <- element₀
    j <- element₁
    CMᵢ,ⱼ <- CMᵢ,ⱼ + 1
    RETURN CM
```

With i = true class index, j = predicted class index. This means that the value inside row i and column j of the confusion matrix is increased by the number of inputs that have true class index = i and predicted class index = j. As an example, in the previous 3-class case, among the 100 inputs exactly 3 were medium displacements (true class index = row = 2) that were also predicted as large displacements (predicted class index = column = 3). That is why in position (2, 3), the confusion matrix has value 3.

2.3. Metrics

From the completed confusion matrix, it is then possible to calculate a wide range of classification metrics; the present version of the algorithm integrates four metrics: Accuracy, Precision, Recall, and F1 Score (Powers, 2008; Sasaki, 2007; Ting, 2017b).

$$\text{Accuracy} = \frac{1}{n}\sum_{m=1}^{n}\frac{\text{TP}_m + \text{TN}_m}{\text{TP}_m + \text{TN}_m + \text{FN}_m + \text{FP}_m} \tag{4}$$

$$\text{Precision} = \frac{1}{n}\sum_{m=1}^{n}\frac{\text{TP}_m}{\text{TP}_m + \text{FP}_m} \tag{5}$$

$$\text{Recall} = \frac{1}{n}\sum_{m=1}^{n}\frac{\text{TP}_m}{\text{TP}_m + \text{FN}_m} \tag{6}$$

$$F_1 = \frac{1}{n}\sum_{m=1}^{n}\frac{2 * \text{TP}_m}{2 * \text{TP}_m + \text{FP}_m + \text{FN}_m} \tag{7}$$

with n = number of classes, $m \in \{1, 2, \ldots n\}$ = class index. They are calculated with the set of equations (4)–(7), once for each class, and then the results are averaged.

The abbreviations used previously are.

- TP_m = true positives, elements of class m that are rightfully classified by the model.
- FN_m = false negatives, elements of class m that are not recognized by the model (they are predicted as a different class).
- TN_m = true negatives, elements not of class m, correctly not predicted as class m data.
- FP_m = false positives, elements not of class m, incorrectly predicted as belonging to class m.

These variables are used extensively in statistics, and are at the base of many classification metrics, because they carry much of the relevant information in a very compact format. Since all the metrics here discussed (and many others) are proportions with values bound in the interval [0,1], they are often expressed as percentage values.

Accuracy is the percentage of right answers, and it gives a general sense of the quality of the model, potentially overlooking many nuances of the situation. Precision is the percentage of non-false alarms: a low value means that the model is giving many false alarms. Recall is the percentage of alarms missed by the model. Precision and Recall are competing metrics, difficult to optimize simultaneously. In fact, to increase the latter, the model should be made more sensitive. This change, however, would cause the model to also generate more false alarms, which would, by definition, reduce the Precision and vice versa. This motivates the introduction of the last metric, F1 Score (or simply F1). As can be seen from its first formulation, F1 is a combination (namely, the harmonic mean) of Precision and Recall, which makes it an informative summary of the quality of the model.

There are, however, many more metrics – for example, the well-known *TorchMetrics* library contains about a hundred different implementations, with their own advantages and drawbacks (Detlefsen et al., 2022). Each one of them measures different aspects of the inputs, has different properties, and is more suitable for certain types of problems (Lui et al., 2022; Tharwat, 2020).

The algorithm developed differs only in the creation of the confusion matrix from other standard versions and the resulting confusion matrix, format-wise, is perfectly standard. This allows the implementation of any existing standard metric (or even new custom metrics) based on confusion matrix or the set of variables *TP*, *FP*, *TN*, *FN*.

3. Landslide monitoring data

In subsections 3.1 and 3.2, the two test sites used for data collection are described. The two datasets are the results of years of continuous monitoring activity performed with automatic devices.

The measurement instrumentation installed in both sites is named Vertical Array (Fig. 3a), an automatic inclinometer developed and patented by ASE S.r.l. (IT) and based on MUMS (Modular Underground Monitoring System) technologies. Each Vertical Array is composed of a series of epoxy resin nodes, known as Links (Fig. 3b), which are connected by an aramid fiber cable and a single quadrupole electrical cable to create an array of sensors (Segalini et al., 2014). The MUMS tools can be customized in terms of the number, distance, and type of sensors used, and their length can be decided according to the monitoring needs. Available sensors include 3D MEMS (Micro Electro-Mechanical Systems), electrolytic tilt cells, piezometers, barometers, and high-resolution thermometers. As a result, the MUMS is a multi-parametric device that can measure displacements, pore pressure, and temperatures at various depths depending on the situation. Finally, a local data logger with SD card and Internet connection allows for customizable (and potentially high) sample frequency.

Since each Vertical Array is customized to specific needs, they have different sensor composition between the two sites, and even among Arrays installed in the same site. For this reason, in each case, the complete list of equipped sensors is given.

Moreover, it is worth noting that the metrics require true values paired with the values predicted by a forecasting model as their input. In the experiments the predictions needed are not the output of an actual model. They are generated by adding random gaussian noise to the true values, with zero mean and standard deviation calculated from the data. This is useful because it allows us to control the output and set criteria for the predictions, necessary conditions to conduct the tests described in Section 4.

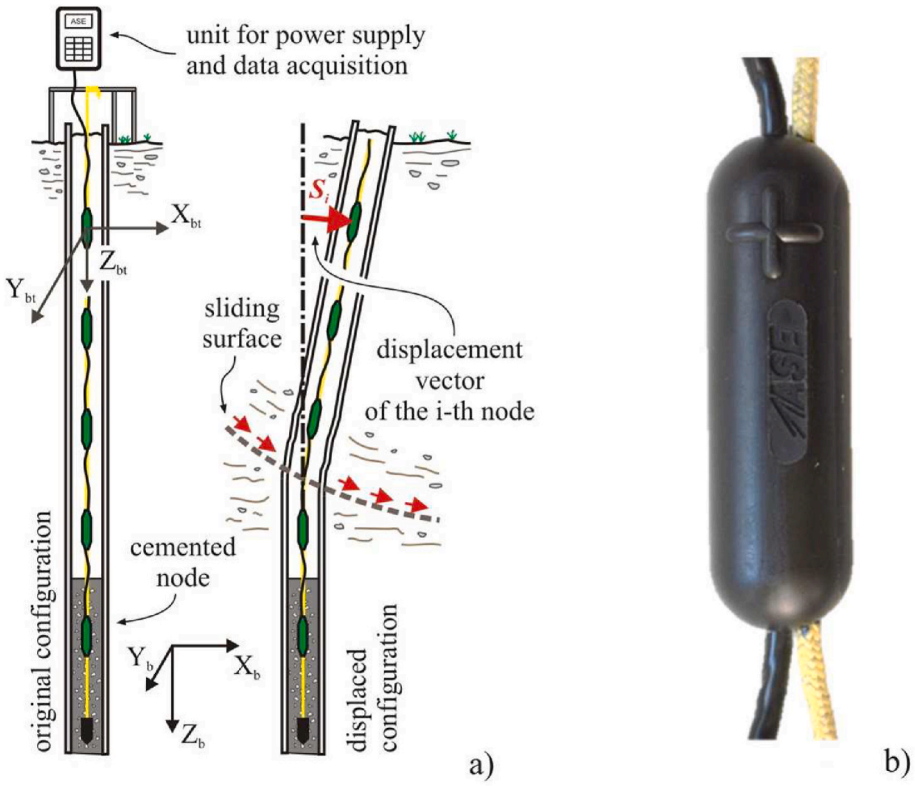

Fig. 3. (a) Vertical Array structure and working principle; (b) Epoxy resin node (Link) (modified after Valletta et al., 2023).

Table 3
Sensor composition of Test Site 1.

Array ID	Array typology	Sensors number and typology	Array length [m]	Installation date [dd/mm/yyyy]
DT0080	Vertical Array	15x Tilt Link HR 3D V 1x Piezo Link 1x Baro Link	15.00	August 29, 2017
DT0081	Vertical Array	15x Tilt Link HR 3D V 1x Piezo Link 1x Therm Link	15.00	August 29, 2017

3.1. Test site 1

Dataset 1 derives from the monitoring activity performed on a geogrids reinforced earth retaining wall, with a height of 12 m. The structure was located on the French Alps more than 1200 m above sea level, protecting a road that gave access to a tunnel nearby (Segalini et al., 2019). Topographic surveys were conducted after the detection of signs of instability and unexpected deformations. Results confirmed that the retaining wall was subject to relevant instabilities.

This prompted the installation of two MUMS Vertical Arrays (Table 3) on the retaining wall, 3 m apart along the maximum slope direction. Each Array was equipped with 15 Tilt Link HR 3D V nodes with 1 m spacing between one another. They also included a piezometer (Piezo Link) located at a depth of 13 m. The first Array (DT0080) featured a barometer while the second one (DT0081) integrated a high-resolution thermometer (Therm Link) 1 m below the top margin of the wall. Vertical Arrays were installed on August 29, 2017, and the sensors sampled a new value every 12 h. Monitoring activities lasted until the retaining wall was demolished for safety reasons in January 2019.

Table 4
Sensor composition of Test Site 2.

Array ID	Array typology	Sensors number and typology	Array length [m]	Installation date [dd/mm/yyyy]
DT0099	Vertical Array	20x Tilt Link HR 3D V 2x Piezo Link	20.00	March 06, 2019
DT0100	Vertical Array	20x Tilt Link HR 3D V 2x Piezo Link 1x Baro Link	20.00	December 05, 2018
DT0101	Vertical Array	20x Tilt Link HR 3D V 1x Piezo Link 1x Baro Link	20.00	March 06, 2019
DT0102	Vertical Array	20x Tilt Link HR 3D V 1x Piezo Link	20.00	March 06, 2019

Table 5
Description of each data configuration analyzed in this paper, with the corresponding relevance in the framework of landslide monitoring activities.

Subsection name	Subsection objective	Relevance in landslide monitoring activities
Optimal case	To evidence the advantage brought by the custom metrics in a specific scenario, characterized by imbalanced datasets	Similar configurations are observed in landslides featuring sudden increases in displacement rates, typically associated to critical events
General case	To compare custom and standard metrics in general cases, showing that the domain of application is not limited to the specific scenario previously discussed	This subsection takes into consideration a more general case, which could be associated with a generic dataset collected by monitoring instrumentation
Deterministic behavior	To validate the claim that multiple evaluations of the same inputs will lead to identical results	The positive outcome of this analysis confirms that the custom metrics are deterministic, therefore only one elaboration is required for each configuration
Data invariance	To highlight that the particular choice of a dataset does not influence the results in significant ways	The custom metrics are not site-specific and are independent of the dataset dimension if the imbalance percentage is the same

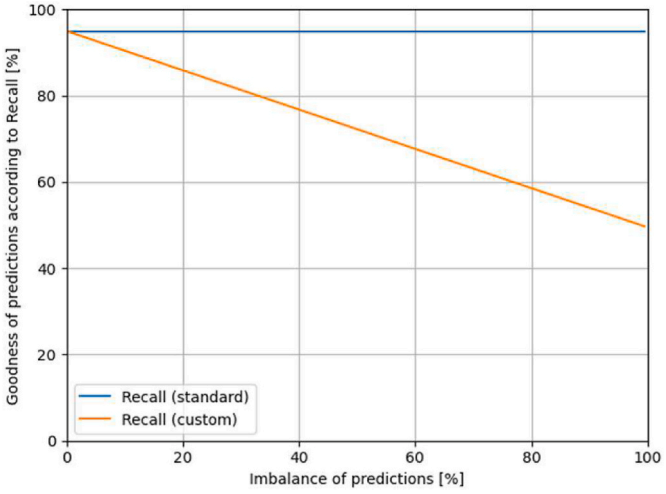

Fig. 4. Standard and custom Recall values for varying degrees of imbalance.

3.2. Test site 2

Dataset 2 originates from the monitoring of the construction site of a transport infrastructure crossing a mountainous and hilly area in Northern Italy (Valletta, 2022). The system consists of 4 MUMS Vertical Arrays.

They are equipped with 20 Tilt Link HR 3D V, and 1 piezometer. Arrays DT0099 and DT0100 have an additional piezometer at a different height. Finally, Arrays DT0100 and DT0101 are both equipped with a barometer. Table 4 reports the features of each device installed on site. The monitoring activity is still ongoing, with a sampling frequency of 6 readings every day.

4. Results and discussion

Data configurations discussed in this study are divided into four separate subsections, based on their purpose. In this phase, it is especially important to analyze scenarios that could be directly related to monitoring activities, in order to have a positive indication of the methodology effectiveness for the assigned objective. Table 5 summarizes the features of each analysis performed, in particular underlining the significance of the chosen configuration in the context of landslide monitoring.

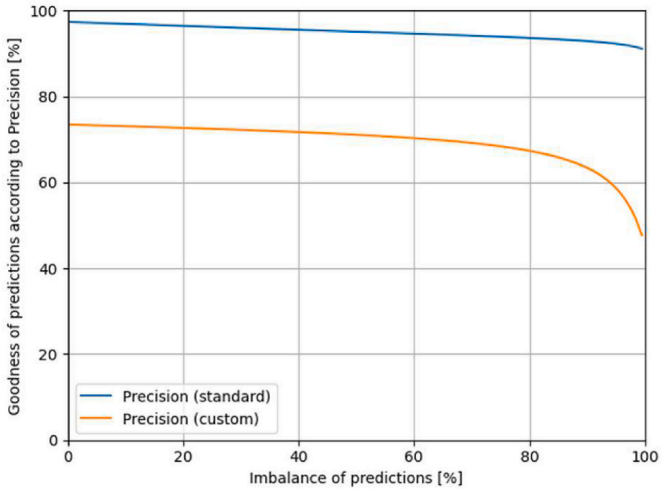

Fig. 5. Standard and custom Precision values for varying degrees of imbalance.

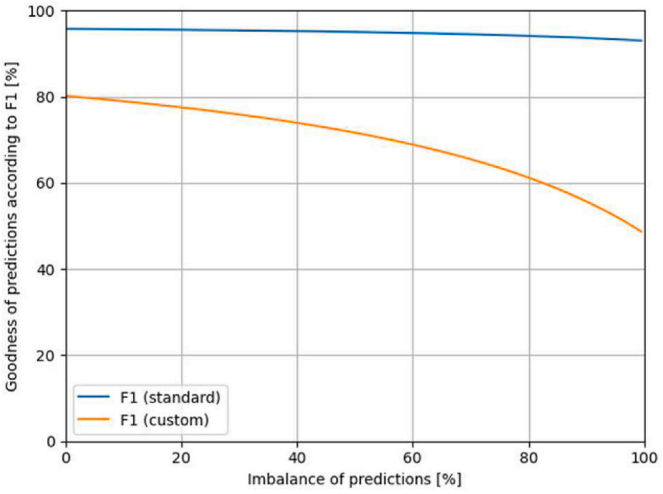

Fig. 6. Standard and custom F1 values for varying degrees of imbalance.

4.1. The optimal case

Figs. 4–6 illustrate the relevance of the metrics calculated with the process described in subsection 2.1. It compares the variation of standard and custom Recall of a fictitious model, depending on the imbalance of the predictions. The dataset is from Test Site 1, and it is composed of 95% small displacements and 5% large displacements, and the model always predicts correctly 95% and misses 5% of the total predictions. The imbalance in calculated as a percentage with equations (8) and (9):

$$IP = |p_0 - p_1| * 100 \tag{8}$$

$$p_m = \frac{c_m}{t_m} \tag{9}$$

with IP = Imbalance Percentage of the predictions, p_m = proportion of correct predictions for class m, c_m = number of correct predictions for class m, t_m = cardinality of class m, m = class index. Thus, there is 0% imbalance when the model has the same percentage of exact guesses for

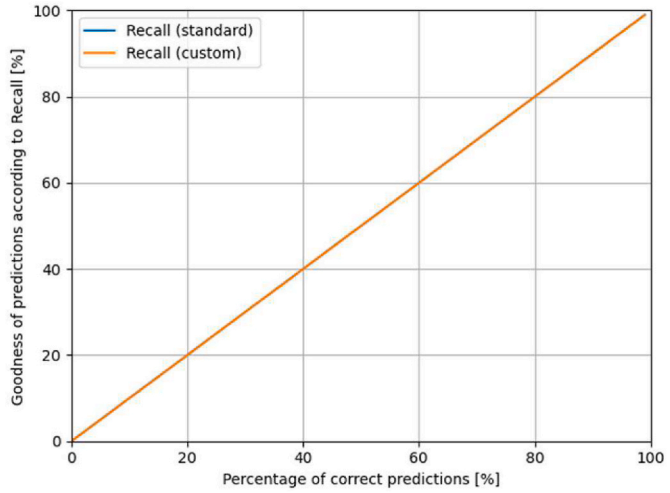

Fig. 7. Standard and custom Recall values for varying degrees of correct predictions.

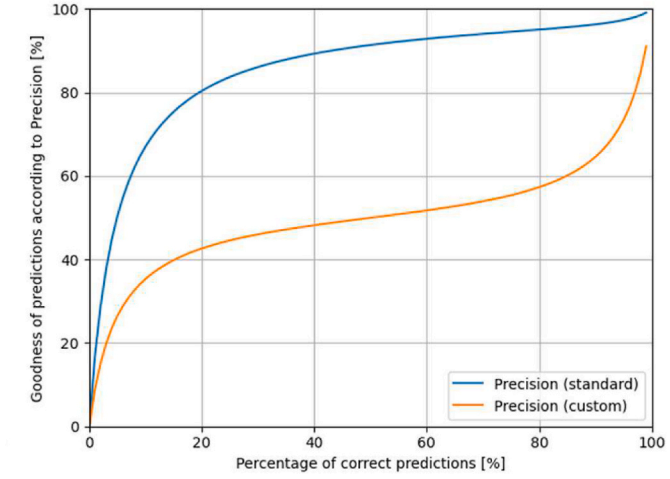

Fig. 8. Standard and custom Precision values for varying degrees of correct predictions.

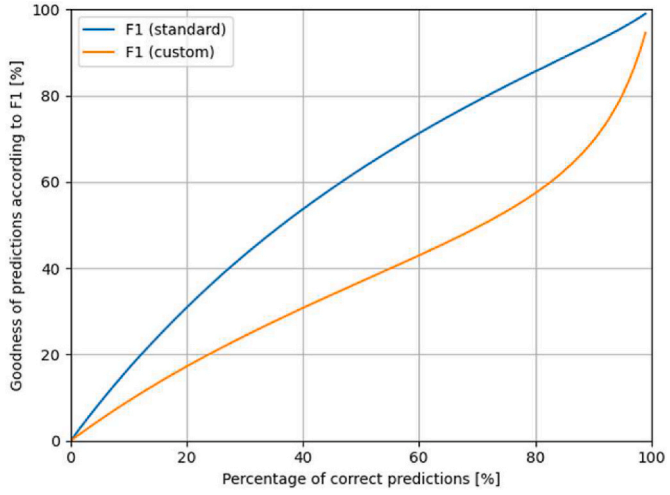

Fig. 9. Standard and custom F1 values for varying degrees of correct prediction.

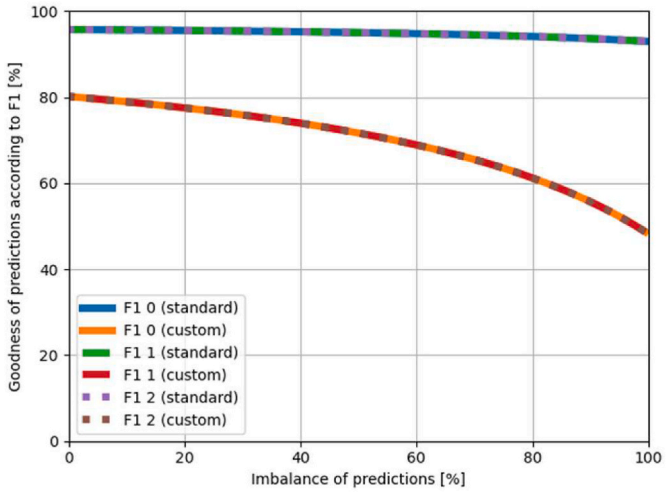

Fig. 10. Three sets of standard and custom F1 values for varying degrees of imbalance. Each set uses different predictions.

each class, while the 100% value corresponds to a situation where the model predicts a class completely correctly while getting each prediction of the other class wrong.

Since the total percentage of correct predictions is 95% in all the cases, the standard Recall remains fixed exactly at that value. The custom Recall is instead sensible to the imbalance in the predictions and is able to determine the relevance of the forecasting error in the context of its application, i.e., identifying potentially critical behaviors. Consequently, the outcome allows us to distinguish between the better models with low imbalance, and the worse models with high imbalance.

The custom Precision is more punishing than the standard Precision and that is reflected, in part, on the F1 metric, which is a combination of Precision and Recall. F1 can be viewed as a single comprehensive summarization of the performance of the model, and in this case, it exhibits the intended properties, highlighting the poor performances of the algorithm as the imbalance of predictions increases in percentage.

4.2. *The general case*

Subsection 4.1 highlighted the usefulness of the proposed metrics in the case that they are specifically designed for, i.e., forecasting operations involving heavily unbalanced datasets. To acquire additional information regarding their reliability and applicability to different cases, we compare their behavior with their standard counterparts in more general scenarios. Figs. 7–9 show the results obtained for the same dataset (Test Site 1), while correct predictions are kept perfectly balanced among the two classes ($IP = 0$). The variable quantity this time is the total percentage of correct predictions (represented on the x-axis in place of the imbalance in the corresponding graphs).

The standard and custom Recall functions are identical, both following a pattern where the percentage of correct precision is always equal to the goodness of the predictions. Meanwhile, for the two remaining parameters, it is possible to evidence how the custom metrics apply a more conservative evaluation of the algorithm performance compared to the standard ones. Specifically, the Precision has lower results than its counterpart for intermediate values of x, but they are coherent on the boundaries. The F1 metric displays a behavior similar to the Precision, but it follows more closely the standard F1.

All the examples refer to a two-class configuration for its relative simplicity, so that relevant properties and results can be clearly presented. The proposed metrics, however, are applicable to any forecasting algorithm that has the appropriate input and output values, and for any number of classes involved in the process.

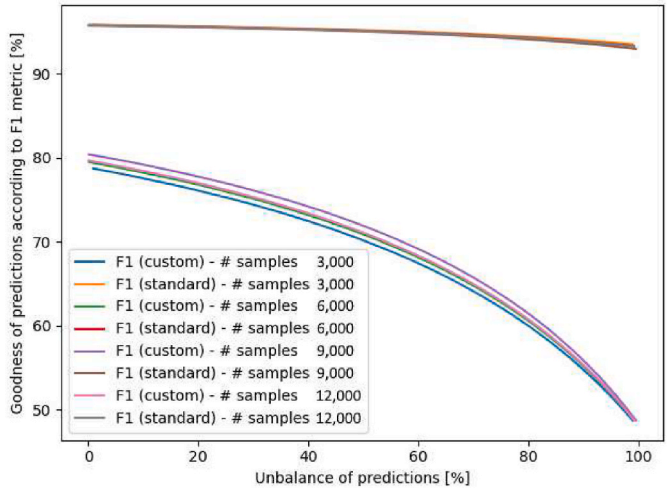

Fig. 11. Four sets of standard and custom F1 values for varying degrees of imbalance. Each set uses different groups of data from Test Site 1.

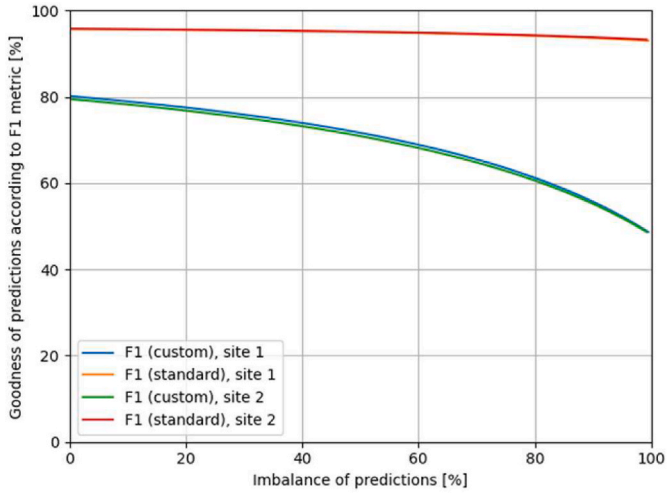

Fig. 12. Two sets of standard and custom F1 values for varying degrees of imbalance. Each set uses data from one of the two sites.

4.3. Deterministic behavior

Even though the predictions are generated randomly each time, the results of the evaluations will be the same as long as the same criteria are applied. It is possible to show this behavior by conducting the same evaluation multiple times with the same dataset and different sets of predictions produced with fixed conditions. In this example, the F1 metric is used to evaluate the quality of three sets of predictions generated with the same characteristics used in subsection 4.1 over the data from Test Site 1 (see Fig. 10). The internal operations performed by standard and custom metrics alike are completely deterministic, that is why the examples and calculations are performed one time only, and not averaged form multiple experiments.

4.4. Data invariance

Data invariance is the property that allowed all the experiments to be conducted with a single dataset (from Test Site 1), except for the one performed to verify this hypothesis. The metrics do not depend directly on the chosen dataset, they only measure the characteristics of models' predictions with respect to the dataset. To demonstrate the invariance with respect to data quantity, four different subsets are extracted from Test Site 1 data. Each subset has a different number of observations, respectively 3,000, 6,000, 9,000, and 12,000.

With the same objective, for the next experiment, data from the two different sites (Test Sites 1 and 2) is used. They are both partitioned as 95% small displacements and 5% large displacements. In both cases the fictitious model generates predictions with the same relevant properties: 95% correct and 5% wrong (the same characteristics used in subsection 4.1). The slight differences that can be observed in Figs. 11 and 12 are mainly due to the fact that many groups of measurements have the same values. This causes a small and unpredictable deviation from the expected split of targets 95% and 5% (respectively small and large displacements).

5. Conclusions

The recent introduction, in the geotechnical field, of algorithms and elaboration processes based on Machine Learning principles has presented several challenges related to the evaluation of the performances on these models (Dahal and Lombardo, 2023). Specifically, the assessment of the quality of forecasting techniques applied to landslide monitoring data plays an essential part in the creation of reliable early warning procedures (Xing et al., 2020).

The custom metrics discussed and applied in this study are significantly better than their standard counterparts in a specific but common scenario, i.e., a dataset with an imbalanced class distribution. In particular, the outcomes obtained from each configuration allow to draw the following conclusions.

- The custom metrics are able to distinguish between models with the same error rate by weighing some errors differently: the estimate of the relevance of the errors, based on their class, allows for a more pertinent and nuanced evaluation of the models.
- The assessment given by the custom metrics decreases for increasing imbalance in the predictions, linearly for Recall and non-linearly for Precision and F1. Standard metrics in the same scenario shows no – or very low – correlation to the imbalance. This behavior makes the proposed metrics particularly appropriate for models applied to slope movements presenting sudden displacement increments.
- For fixed imbalance, custom metrics tend to be more conservative than their standard counterparts, giving lower evaluations especially for models with high error rate. It can be seen clearly in the comparison between custom and standard Precision, and custom and standard F1.
- Even though data quantity and quality are extremely important elements in the development of a ML forecasting algorithm, this is not the case for the proposed evaluation procedures. As a consequence, the metrics can be applied to models trained on very different datasets and are not dependent on the monitoring activity duration.
- The custom metrics are modified so that they can seamlessly work on predictions produced by both classifiers and regression models, by treating both types as classifiers. They are useful because they provide a meaningful standalone evaluation of a model, while also giving the means for direct comparison with other forecasting algorithms.

This study implements a small subset of metrics (Accuracy, Precision, Recall, and F1-Score) and analyzes the efficacy of the method for them. Since there is no single metric that can capture all the information regarding the performance of a model, having many different evaluation options is important. The change proposed in this work only concerns the calculation of the confusion matrix. The consequence is that any metric derivable from the confusion matrix can be computed very easily using their standard equation. The value of our modification applied to each new metric, however, should be studied individually before employing it.

The modifications of the standard metrics are designed specifically

for predictive models that work on the displacement time series, the pertinence of the custom metrics for other algorithms in the same field (es landslide identification from satellite images) has not been tested.

Computer code availability

Name of the code/library: ML algorithms evaluation.

Contact: Marco Conciatori, email: marco.conciatori@unipr.it.

Hardware requirements: the code was tested on ASUS Zenbook UX535LH with Intel Core i7 CPU, 16 GB RAM, NVIDIA GeForce GTX 1650 Max-Q. The code should be able to run on different and slower machines.

Program language: Python.

Software required: the algorithm was executed on Windows 11 but should run on any OS. The requirements are Python, Numpy, Pytorch, Torchmetrics. For a full list of libraries and dependencies, including their versions, please refer to the environment files automatically generated with *Conda*.

Program size: 28 kB.

The source code is available for downloading at the link: https://github.com/marco-conciatori-public/ml_algorithms_evaluation (Code available under the GNU General Public License v3.0).

Authorship contribution statement

Marco Conciatori: Conceptualization, Investigation, Methodology, Software, Validation, Formal analysis, Data Curation, Writing - Original Draft, Visualization. Alessandro Valletta: Conceptualization, Investigation, Methodology, Validation, Resources, Writing - Review & Editing, Visualization. Andrea Segalini: Conceptualization, Resources, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

Project partially funded under the National Recovery and Resilience Plan (NRRP), Mission 4 Component 2 Investment 1.5 - Call for tender No. 3277 of December 30, 2021 of Italian Ministry of University and Research funded by the European Union – NextGenerationEU. Award Number: Project code ECS00000033, Concession Decree No. 1052 of June 23, 2022 adopted by the Italian Ministry of University and Research, CUP D93C22000460001, "Ecosystem for Sustainable Transition in Emilia-Romagna" (Ecosister), Spoke 4.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., 2016. TensorFlow: a system for large-scale machine learning. In: Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation. OSDI '16, pp. 265–283. https://doi.org/10.48550/arXiv.1605.08695.

Abugabah, A., Mehmood, A., Almotairi, S., Smadi, A.A.L., 2022. Health care intelligent system: a neural network based method for early diagnosis of Alzheimer's disease using MRI images. Expert Systems Volume 39 (9), e13003. https://doi.org/10.1111/exsy.13003.

Adlung, L., Cohen, Y., Mor, U., Elinav, E., 2021. Machine learning in clinical decision making. Méd. 2, 642–665. https://doi.org/10.1016/j.medj.2021.04.006.

Alzubi, J., Nayyar, A., Kumar, A., 2018. Machine learning from theory to algorithms: an overview. J. Phys. Conf. 1142, 012012 https://doi.org/10.1088/1742-6596/1142/1/012012.

Aradi, S., 2022. Survey of deep reinforcement learning for motion planning of autonomous vehicles. IEEE Trans. Intell. Transport. Syst. 23, 740–759. https://doi.org/10.1109/TITS.2020.3024655.

Banerjee, A., Chitnis, U.B., Jadhav, S.L., Bhawalkar, J.S., Chaudhury, S., 2009. Hypothesis testing, type I and type II errors. Ind. Psychiatr. J. 18, 127–131. https://doi.org/10.4103/0972-6748.62274.

Bell, R.M., Koren, Y., 2007. Lessons from the Netflix prize challenge. ACM SIGKDD Explorations Newsletter 9, 75–79. https://doi.org/10.1145/1345448.1345465.

Benbouras, M.A., Petrişor, A.-I., Zedira, H., Ghelani, L., Lefilef, L., 2021. Forecasting the bearing capacity of the driven piles using advanced machine-learning techniques. Appl. Sci. 11, 10908 https://doi.org/10.3390/app112210908.

Bengio, Y., Lecun, Y., Hinton, G., 2021. Deep learning for AI. Commun. ACM 64, 58–65. https://doi.org/10.1145/3448250.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y.T., Li, Y., Lundberg, S., Nori, H., Palangi, H., Ribeiro, M.T., Zhang, Y., 2023. Sparks of Artificial General Intelligence: Early Experiments with GPT-4. https://doi.org/10.48550/arXiv.2303.12712.

Chang, Z., Catani, F., Huang, F., Liu, G., Meena, S.R., Huang, J., Zhou, C., 2022. Landslide susceptibility prediction using slope unit-based machine learning models considering the heterogeneity of conditioning factors. J. Rock Mech. Geotech. Eng. 15 (5), 1127–1143. https://doi.org/10.1016/j.jrmge.2022.07.009.

Dahal, A., Lombardo, L., 2023. Explainable artificial intelligence in geoscience: a glimpse into the future of landslide susceptibility modeling. Comput. Geosci. 176, 105364 https://doi.org/10.1016/j.cageo.2023.105364.

Denes, P., Mathews, M.V., 1960. Spoken digit recognition using time-frequency pattern matching. J. Acoust. Soc. Am. 32, 1450–1455. https://doi.org/10.1121/1.1907936.

Detlefsen, N.S., Borovec, J., Schock, J., Harsh, A., Koker, T., Di Liello, L., Stancl, D., Quan, C., Grechkin, M., Falcon, W., 2022. TorchMetrics - measuring reproducibility in PyTorch. J. Open Source Softw. 7 (70), 4101. https://doi.org/10.21105/joss.04101.

Feder, A., Keith, K.A., Manzoor, E., Pryzant, R., Sridhar, D., Wood-Doughty, Z., Eisenstein, J., Grimmer, J., Reichart, R., Roberts, M.E., Stewart, B.M., Veitch, V., Yang, D., 2022. Causal inference in natural language processing: estimation, prediction, interpretation and beyond. Transactions of the Association for Computational Linguistics 10, 1138–1158. https://doi.org/10.1162/tacl_a_00511.

Juyal, A., Sharma, S., Matta, P., 2021. Deep learning methods for object detection in autonomous vehicles. In: Proceedings of the 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), pp. 751–755. https://doi.org/10.1109/ICOEI51242.2021.9452932, 3-5 June 2021, Tirunelveli, Tamilnadu, India.

Khosravi, K., Shahabi, H., Pham, B.T., Adamowski, J., Shirzadi, A., Pradhan, B., Dou, J., Ly, H.-B., Gróf, G., Ho, H.L., Hong, H., Chapi, K., Prakash, I., 2019. A comparative assessment of flood susceptibility modeling using Multi-Criteria Decision-Making Analysis and Machine Learning Methods. J. Hydrol. 573, 311–323. https://doi.org/10.1016/j.jhydrol.2019.03.073.

Koopialipoor, M., Asteris, P.G., Salih Mohammed, A., Alexakis, D.E., Mamou, A., Armaghani, D.J., 2022. Introducing stacking machine learning approaches for the prediction of rock deformation. Transportation Geotechnics 34, 100756. https://doi.org/10.1016/j.trgeo.2022.100756.

Kotthoff, L., Gent, I., Miguel, I., 2011. A preliminary evaluation of machine learning in algorithm selection for search problems. Proceedings of the Fourth Annual Symposium on Combinatorial Search 2, 84–91. https://doi.org/10.1609/socs.v2i1.18184.

Li, L., Wu, Y., Miao, F., Xue, Y., Huang, Y., 2021. A hybrid interval displacement forecasting model for reservoir colluvial landslides with step-like deformation characteristics considering dynamic switching of deformation states. Stoch. Environ. Res. Risk Assess. 35, 1089–1112. https://doi.org/10.1007/s00477-020-01914-w.

Liu, T., Siegel, E., Shen, D., 2022. Deep learning and medical image analysis for COVID-19 diagnosis and prediction. Annu. Rev. Biomed. Eng. 24, 179–201. https://doi.org/10.1146/annurev-bioeng-110220-012203.

Lui, T.C.C., Gregory, D.D., Anderson, M., Lee, W.-S., Cowling, S.A., 2022. Applying machine learning methods to predict geology using soil sample geochemistry. Applied Computing and Geosciences 16, 100094. https://doi.org/10.1016/j.acags.2022.100094.

Nava, L., Carraro, E., Reyes-Carmona, C., Puliero, S., Bhuyan, K., Rosi, A., Monserrat, O., Floris, M., Meena, S.R., Galve, J.P., Catani, F., 2023. Landslide displacement forecasting using deep learning and monitoring data across selected sites. Landslides. https://doi.org/10.1007/s10346-023-02104-9.

Olson, R.S., La Cava, W., Orzechowski, P., Urbanowicz, R.J., Moore, J.H., 2017. PMLB: a large benchmark suite for machine learning evaluation and comparison. BioData Min. 10, 36. https://doi.org/10.1186/s13040-017-0154-4.

Parvat, A., Chavan, J., Kadam, S., Dev, S., Pathak, V., 2017. A survey of deep-learning frameworks. In: Proceedings of the 2017 International Conference on Inventive Systems and Control (ICISC), pp. 1–7. https://doi.org/10.1109/ICISC.2017.8068684.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É., 2011. Scikit-learn: machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830, 2011.

Powers, D., 2008. Evaluation: from precision, Recall and F-factor to ROC, informedness, markedness & correlation. International Journal of Machine Learning Technology 2 (1), 37–63. https://doi.org/10.48550/arXiv.2010.16061, 2011.

Qayyum, A., Qadir, J., Bilal, M., Al-Fuqaha, A., 2021. Secure and robust machine learning for healthcare: a survey. IEEE Reviews in Biomedical Engineering 14, 156–180. https://doi.org/10.1109/RBME.2020.3013489.

Reich, Y., Barai, S.V., 1999. Evaluating machine learning models for engineering problems. Artif. Intell. Eng. 13 (3), 257–272. https://doi.org/10.1016/S0954-1810(98)00021-1.

Sasaki, Y., 2007. The truth of the F-measure. Teach Tutor Mater 1, 5.

Schmidhuber, J., 2015. Deep learning in neural networks: an overview. Neural Network. 61, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003.

Segalini, A., Chiapponi, L., Pastarini, B., Carini, C., 2014. Automated inclinometer monitoring based on micro electro-mechanical system technology: applications and verification. In: Sassa, K., Canuti, P., Yin, Y. (Eds.), Landslide Science for a Safer Geoenvironment. Springer, Cham. https://doi.org/10.1007/978-3-319-05050-8_92.

Segalini, A., Valletta, A., Carri, A., Cavalca, E., 2019. Monitoring of a retaining wall with innovative multi-parameter tools. In: Proceedings of the 4th Regional Symposium on Landslides in the Adriatic - Balkan Region. Presented at the 4th Regional Symposium on Landslides in the Adriatic - Balkan Region. Društvo za geotehniku u Bosni i Hercegovini, pp. 31–36. https://doi.org/10.35123/ReSyLAB_2019_5.

Soranzo, E., Guardiani, C., Chen, Y., Wang, Y., Wu, W., 2022. Convolutional neural networks prediction of the factor of safety of random layered slopes by the strength reduction method. Acta Geotechnica. https://doi.org/10.1007/s11440-022-01783-3.

Tharwat, A., 2020. Classification assessment methods. Appl. Comput. Inform. 17, 168–192. https://doi.org/10.1016/j.aci.2018.08.003.

Tilahun, T., Korus, J., 2023. 3D hydrostratigraphic and hydraulic conductivity modelling using supervised machine learning. Applied Computing and Geosciences 19, 100122. https://doi.org/10.1016/j.acags.2023.100122.

Ting, K.M., 2017a. Confusion matrix. In: Sammut, C., Webb, G.I. (Eds.), Encyclopedia of Machine Learning and Data Mining. Springer US, Boston, MA. https://doi.org/10.1007/978-1-4899-7687-1_50, 260–260.

Ting, K.M., 2017b. Precision and Recall. In: Sammut, C., Webb, G.I. (Eds.), Encyclopedia of Machine Learning and Data Mining. Springer, Boston, MA. https://doi.org/10.1007/978-1-4899-7687-1_659.

Tokgozoglu, K., Aladag, C.H., Gokceoglu, C., 2023. Artificial neural networks to predict deformation modulus of rock masses considering overburden stress. Geomechanics Geoengin. 18, 48–64. https://doi.org/10.1080/17486025.2021.2008518.

Valletta, A., 2022. Automatic Detection of Landslide Events for Risk Management and Early Warning Procedures. Doctoral thesis, University of Parma, Department of Engineering and Architecture. https://hdl.handle.net/1889/4822.

Valletta, A., Carri, A., Savi, R., Segalini, A., 2023. Algorithms for the near-real time identification and classification of landslide events detected by automatic monitoring tools. In: Zembaty, Z., Perkowski, Z., Beben, D., Massimino, M.R., Lavan, O. (Eds.), Environmental Challenges in Civil Engineering II. ECCE 2022. Lecture Notes in Civil Engineering, vol. 322. Springer, Cham. https://doi.org/10.1007/978-3-031-26879-3_6.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., Rush, A., 2020. Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Association for Computational Linguistics, Online, pp. 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6.

Xing, Y., Yue, J., Chen, C., Qin, Y., Hu, J., 2020. A hybrid prediction model of landslide displacement with risk-averse adaptation. Comput. Geosci. 141, 104527 https://doi.org/10.1016/j.cageo.2020.104527.

Zhang, H., Song, Y., Xu, S., He, Y., Li, Z., Yu, X., Liang, Y., Wu, W., Wang, Y., 2022. Combining a class-weighted algorithm and machine learning models in landslide susceptibility mapping: a case study of Wanzhou section of the Three Gorges Reservoir, China. Comput. Geosci. 158, 104966 https://doi.org/10.1016/j.cageo.2021.104966.