Testing the Consistency of Performance Scores Reported for Multiclass Classification Problems

György Kovács[a], Avi[b], Attila Fazekas[c]

[a]*Analytical Minds Ltd., Árpád út 5, Beregsurány, 4933, Hungary*
[b]*Avi Ltd., Avi street 1, Avi city, 4933, US*
[c]*Faculty of Informatics, University of Debrecen, Kassai út 26, Debrecen, 4028, Hungary*

Abstract

Multi-class classification is a cornerstone of modern machine learning, with widespread applications across scientific and engineering domains. Whether exploring theoretical models or developing practical tools, researchers commonly evaluate and compare classification methods using metrics such as overall accuracy, macro-averaged F1 score, and per-class precision and recall. However, these reported performance metrics may not always provide a reliable basis for comparing research findings. Potential sources of unreliability include unreported methodological choices, misinterpretations of multi-class evaluation protocols, or typographical and computational errors.

In any given experimental setup—defined by a specific number of test samples per class—performance metrics are constrained to a finite and interdependent set of values. Leveraging this property, we propose numerical techniques to verify the internal consistency of reported multi-class classification metrics with their implied experimental configuration. These methods do not rely on statistical inference; instead, they employ numerical tools such as interval analysis and integer linear programming to deterministically identify inconsistencies.

We illustrate the utility of our approach through case studies in diverse medical imaging and diagnostic applications. The proposed consistency tests successfully reveal discrepancies in reported metrics, offering a practical tool for safeguarding the integrity of empirical research. Power analyses indicate that the tests achieve at least 71% power when performance scores are reported with four decimal places. The tests have already identified inconsistencies in over 100 scientific publications. To support reproducible research practices, we have released our consistency checking tools as part of the open-source Python package mlscorecheck.

Keywords: binary classification, performance scores, ranking of binary classifiers, interval arithmetic, consistency

1. Introduction

Concerns over the *reproducibility crisis* in artificial intelligence and machine learning research have been voiced repeatedly in recent years [1, 2, 3, 4]. A substantial fraction of published findings prove difficult to replicate or rest on questionable methodological foundations. The causes are diverse: missing or inaccessible source code [1], inappropriate statistical methods [1, 5], selective reporting or "polishing" of results [6], and inadvertent errors in tabulated metrics all contribute to the problem.

To mitigate these issues, several best-practice guidelines have been proposed for the transparent and rigorous reporting of machine learning experiments [3, 7]. Yet despite growing awareness, adoption has been slow, and the

Email addresses: gyuriofkovacs@gmail.com (György Kovács), avi@gmail.com (Avi), attila.fazekas@inf.unideb.hu (Attila Fazekas)

existing literature remains replete with problematic results. In many subfields of machine learning, research advances are driven by informal benchmarking races, where algorithmic contributions are judged primarily by their reported performance on standard datasets. This dynamic reinforces a focus on numerical scores as the principal—or even sole—indicator of scientific merit. However, comparing performance scores is fraught with challenges, even when the evaluation protocol is fully specified [8]. Once published, inflated or implausible results can lend credibility to flawed procedures and, through citation and replication, propagate distortions across entire fields. Publication bias only exacerbates this effect [9].

Traditional approaches to scrutinizing the reliability of published machine learning results typically involve painstaking manual effort. Analysts may review methodological details for signs of misapplied statistical logic or implementation flaws [10, 11, 12], or they may attempt to reproduce results from scratch—an approach that is both time-consuming and unscalable. In light of these limitations, there is a need for tools that enable automated or semi-automated validation of published metrics.

In this paper, we propose numerical techniques for testing the internal consistency of performance scores reported in multi-class classification settings.

Multi-class classification, a core task in supervised learning [13], is not immune to the reproducibility challenges observed in binary problems. Evaluation is typically performed by computing predictions over a labeled test set, often using cross-validation or resampling schemes [14]. From these predictions, a confusion matrix of size $C \times C$ is constructed, where C denotes the number of classes. This matrix captures how often each true class was assigned to each predicted class. In published work, however, the full matrix is rarely presented. Instead, scalar metrics derived from it—such as overall accuracy, macro-averaged F1-score, or per-class recall—are reported, frequently averaged across folds or datasets [15].

Despite their prevalence, these metrics are not unconstrained: they are interdependent and bounded by the structure of the test data. For instance, the size of the confusion matrix is fixed by the number of test examples, and the scores themselves are mathematically tied to the counts in the matrix. This raises an important question: *Are the reported metrics consistent with any plausible confusion matrix under the described evaluation setup?* Put differently, does there exist a confusion matrix—subject to known constraints such as test set size and score definitions—that could give rise to the published performance values? Complicating this question further are effects such as rounding and aggregation across multiple folds. If no such matrix can exist, then the result is necessarily flawed and cannot be reproduced under the assumed conditions.

In earlier work, tools such as `DConfusion` [16] have attempted to infer partial properties of confusion matrices from reported scores, primarily in binary settings. However, these tools offer limited score coverage and do not account for score aggregation or numerical precision effects, making them prone to overestimating inconsistency. Our previous research introduced exact inference techniques in binary contexts, highlighting how consistent scoring claims can be deterministically verified or falsified using constraint-based reasoning [17, 18].

Building on these insights, the present paper extends the consistency checking framework to the general multi-class case. Unlike prior approaches, our method handles a broad range of evaluation metrics, accommodates cross-fold averaging, and rigorously incorporates rounding effects and numerical tolerances. In doing so, we enable deterministic inconsistency detection that is both precise and broadly applicable.

A key advantage of our method is that it is numerical rather than statistical. This means that the detection of an inconsistency implies a logical contradiction in the reported data—*type-I errors (false positives) are impossible*. As in hypothesis testing, there may be false negatives depending on the setup, but we show that in real-world cases the power of the method is often high enough to detect subtle inconsistencies with certainty.

We believe these techniques can serve as valuable tools for meta-research, enabling the community to audit published results and identify cases where evaluation procedures or numerical reporting may be flawed. This contributes to improving the transparency and reproducibility of multi-class classification studies in machine learning.

Contributions

The main contributions of this work are:

1. We present a general-purpose consistency test for performance scores computed from multi-class confusion matrices, supporting over 20 commonly used evaluation metrics.

2. For scenarios involving score aggregation across folds or datasets, we extend our method to handle averaging schemes while accounting for rounding and interval tolerances.

3. We demonstrate how the method enables reliable meta-analysis by uncovering incompatible results in real-world use cases.

4. The implementation is made available through the open-source Python package `mlscorecheck` [19], accessible via PyPI and GitHub: `https://github.com/FalseNegativeLab/mlscorecheck`.

Paper Structure

The remainder of this paper is structured as follows. Section 2 introduces the notation, problem setup, and performance metrics used in multi-class evaluation. Sections 3 and 4 present our proposed consistency checking methods for individual and aggregated scores, respectively. Section 5 showcases their application in real-world contexts. Finally, Section 6 concludes the paper.

2. Problem formulation

In this section, we formulate the problem we address and introduce the notations and concepts that we will reference throughout the rest of the paper.

In binary classification, typically there is a dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ consisting of N paired *feature vector*s ($\mathbf{x}_i \in \mathbb{R}^d$) and *class label*s ($y_i \in \{0, 1\}$). The classes labeled as 0 and 1 are commonly referred to as *negative* and *positive* classes, respectively. The primary objective of binary classification is to use this dataset to infer (train) a function h capable of making predictions about the class label of a previously unseen feature vector $\mathbf{x} \in \mathbb{R}^d$ as $h(\mathbf{x})$. All classification techniques predict class labels, but many of them (such as decision trees [13], neural networks [13], etc.) are designed to estimate class-membership probabilities $\mathbb{P}(\mathbf{x}|y = c)$, $c \in \{0, 1\}$, and derive class labels by assigning the label with the highest probability, thereby reducing the probability of misclassification [20]. The choice of which classifier outcome to favor depends on the specific application. For instance, image segmentation [21] requires hard labels indicating whether a pixel belongs to an object, whereas many medical applications prioritize ranking cases by the probability (risk) of having a condition [22]. Consequently, performance measurement varies by the field of application, with two predominant approaches: one quantifies how effectively the class-membership probabilities rank items, typically using the ROC-AUC score [23, 24]; the other assesses the quality of label assignment [15]. This paper focuses on evaluating the quality of labeling.

To eliminate bias, classifiers must be evaluated using feature vectors and corresponding class labels that were not used for training. In the rest of the paper, we refer to these as the *evaluation set*, denoted by $\mathcal{E}$. (We note that in many scenarios the terms *test set* and *validation set* are used synonymously.) For evaluation, the class label $\hat{y} \doteq h(\mathbf{x})$ is predicted for each $(\mathbf{x}, y) \in \mathcal{E}$ and by comparing the corresponding pairs y and $\hat{y}$, the *confusion matrix* (see Table 1) is constructed, tabulating the integer counts of true positive (tp), true negative (tn), false positive (fp), and false negative (fn) predictions. One can readily see that if the number of positive p and negative n instances of the evaluation set $\mathcal{E}$ is known, the binary confusion matrix has two degrees of freedom, since $p = tp + fn$ and $n = tn + fp$. Without loss of generality, we consider tp and tn as the independent components.

To facilitate the comparison of classification approaches, a confusion matrix is often summarized by scalar *performance score*s. (We use the term performance score to prevent confusion with the mathematical notions of *measure* and *metric*, which are used synonymously in various sources.) The literature has proposed numerous performance scores, each emphasizing different aspects of performance, some widely used (such as *accuracy*), while others tailored to specific fields (such as the *diagnostic odds ratio* in medical applications [25]). See Table 2 for a summary of the scores covered in this paper. In the rest of the paper, we always assume that there is a set $\mathcal{S} \subseteq \{acc, sens, spec, \ldots\}$ of scores reported (any subset of the abbreviations in Table 2). For a specific score $s \in \mathcal{S}$, the standardized functional form of the score is denoted by f_s, with $f_s(tp, tn, p, n)$ yielding the true, possibly infinite decimal value of the score for a confusion matrix characterized by tp, tn, p, and n.

In cases where predefined evaluation sets are not available, the common practice is adopting the *hold-out* approach: randomly partitioning the dataset into two disjoint sets (the training set $\mathcal{T}$ and the evaluation set $\mathcal{E}$). The random split makes the estimated performance scores uncertain, which can be mitigated by repeating the random partitioning and

Table 1: The potential outcomes in the evaluation of binary classification.

	predicted positive ($\hat{y}_i = 1$)	predicted negative ($\hat{y}_i = 0$)
positive ($y_i = 1$)	true positive (tp)	false negative (fn)
negative ($y_i = 0$)	false positive (fp)	true negative (tn)

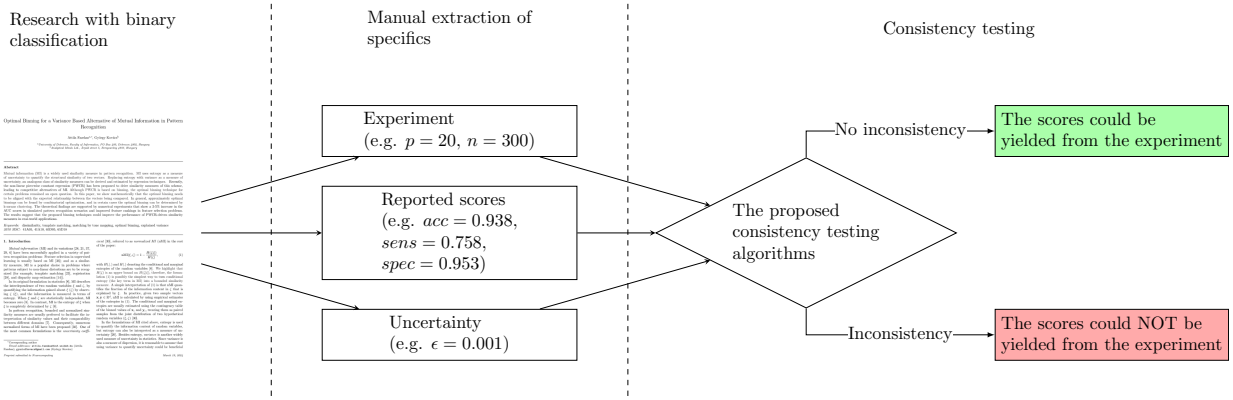


Figure 1: The intended use of the proposed methods.

evaluation multiple times and aggregating the results. To ensure that all data points are represented equally in the evaluation, usually *k-fold cross-validation* (kFCV) is used [14]. In a kFCV scheme, the dataset $\mathcal{D}$ is randomly divided into k disjoint subsets of equal size, referred to as *folds*. The evaluation process iteratively selects one fold as the evaluation set, trains the classifier on the remaining $k - 1$ folds, and evaluates it on the selected fold, using all data points for evaluation once. Finally, the results are aggregated over all folds. To improve stability further, kFCV can be repeated multiple times with different random partitionings of $\mathcal{D}$ (known as *repeated kFCV*). Another common enhancement to kFCV is applying *stratification* to ensure that the class distributions in each fold approximate that of the entire dataset. We also note that in some cases, classifiers are evaluated on and the results aggregated over multiple datasets.

Since many performance scores are ratios of integers, it is common practice in scientific writing to round them to a finite number of decimal places for presentation. For instance, introducing the notation $\hat{v}_s$ for the reported figure of the score s, $\hat{v}_s = 0.945$ implies that the original value was rounded to 3 decimal places. Consequently, the true value $v_s^* = f_s(tp^*, tn^*, p, n)$ is expected to fall within the interval $v_s^* \in [\hat{v}_s - \epsilon, \hat{v}_s + \epsilon] = [0.9445, 0.9455]$, where $\epsilon = 10^{-3}/2$ represents the *numerical uncertainty*. If flooring and ceiling are also allowed, the numerical uncertainty increases to $\epsilon = 10^{-3}$.

The problem we address in the rest of the paper can be phrased as follows: *Given a set of reported performance scores and the description of the experiment (the datasets involved, the evaluation scheme, the mode of aggregation), could the experiment have yielded the reported scores?*

3. Testing scores derived from one confusion matrix

In this section, we assume that there is a well-specified evaluation set available, with known numbers of positive (p) and negative (n) instances, and a set of scores $\mathcal{S} \subseteq \{acc, sens, \dots\}$ reported up to ϵ numerical uncertainty. It is worth noting that this experimental setup is common in various fields such as computer vision (when results for publicly available test images are shared [18, 31]); big data (where the hold-out strategy is used to reduce computational demand); and machine learning competitions, including those on platforms like Kaggle (`www.kaggle.com`), where a closed test set is withheld.

Table 2: A summary of all performance scores discussed in the paper, including their standardized forms that depend on tp, tn, p, and n, their original definitions, and descriptions that mention common synonyms and complements.

name	abbr.	standardized form	original definition	description
accuracy [15]	acc	$\frac{tn+tp}{n+p}$		The proportion of correctly classified items. Complement: error rate.
sensitivity [15]	sens	$\frac{tp}{p}$		The proportion of correctly classified positive items. Also known as: recall, true positive rate. Complement: false negative rate.
specificity [15]	spec	$\frac{tn}{n}$		The proportion of correctly classified negative items. Also known as: selectivity, true negative rate. Complement: false positive rate.
positive predictive value [15]	ppv	$\frac{tp}{n-tn+tp}$		The proportion of truly positive items among all items classified as positive. Also known as: precision. Complement: false discovery rate.
negative predictive value [15]	npv	$\frac{tn}{p+tn-tp}$		The proportion of truly negative items among all items classified as negative. Complement: false omission rate.
f_+^β [15]	fbp	$\frac{tp\left(\beta_+^2+1\right)}{\beta_+^2 p+n-tn+tp}$	$(1+\beta_+^2)\frac{ppv\cdot sens}{\beta_+^2 ppv+sens}$	The harmonic mean of positive predictive value and sensitivity, when sensitivity is β times more important than the positive predictive value. Usually referred to as the F-score, with $\beta=1$, the F1-score.
f_-^β [26]	fbn	$\frac{tn\left(\beta_-^2+1\right)}{\beta_-^2 n+p+tn-tp}$	$(1+\beta_-^2)\frac{npv\cdot spec}{\beta_-^2 npv+spec}$	The harmonic mean of negative predictive value and specificity, when specificity is β times more important than the negative predictive value.
unified performance measure [26]	upm	$\frac{4tn\cdot tp}{tn(n+p-tn+tp)+tp(n+p+tn-tp)}$	$2\frac{f_+^1\cdot f_-^1}{f_+^1+f_-^1}$	The harmonic mean of f_+^1 and f_-^1. Also known as: p4.
geometric mean [15]	gm	$\sqrt{\frac{tn\cdot tp}{np}}$	$\sqrt{sens\cdot spec}$	The geometric mean of sensitivity and specificity.
Fowlkes-Mallows index [27]	fm	$\frac{tp}{\sqrt{p(n-tn+tp)}}$	$\sqrt{ppv\cdot sens}$	The geometric mean of positive predictive value and sensitivity.
markedness [28]	mk	$\frac{tn}{p+tn-tp}+\frac{tp}{n-tn+tp}-1$	$npv+ppv-1$	Quantifies the probability that a condition is marked by the predictor (versus random chance). Also known as: Δp.
bookmaker informedness [28]	bm	$-1+\frac{tp}{p}+\frac{tn}{n}$	$sens+spec-1$	Quantifies how informed the classifier is for the specified condition. Also known as: informedness.
Matthews correlation coefficient [15]	mcc	$\frac{tn\cdot tp-(n-tn)(p-tp)}{\sqrt{np(n-tn+tp)(p+tn-tp)}}$	$\sqrt{bm\cdot mk}$	The Pearson correlation coefficient computed for the observed and predicted labels. Also known as: phi coefficient.
positive likelihood ratio [15]	lrp	$\frac{n\cdot tp}{p(n-tn)}$	$\frac{sens}{1-spec}$	Quantifies the probability of correct positive prediction relative to the probability of type I error.
negative likelihood ratio [15]	lrn	$\frac{n(p-tp)}{p\cdot tn}$	$\frac{1-sens}{spec}$	Quantifies the probability of type II error relative to the probability of correct negative prediction.
prevalence threshold [29]	pt	$-\frac{p\left(n\sqrt{\frac{tp(n-tn)}{np}}-n+tn\right)}{-n\cdot tp+p(n-tn)}$	$\frac{spec+\sqrt{sens(1-spec)}-1}{sens+spec-1}$	Estimates the threshold on prevalence under which the precision of classification declines rapidly.
diagnostic odds ratio [15]	dor	$\frac{tn\cdot tp}{(n-tn)(p-tp)}$	$\frac{lrp}{lrn}$	The ratio of the odds that tha classifier correctly predicts a positive label to the odds of incorrectly predicting the positive label.
Jaccard index [15]	ji	$\frac{tp}{n+p-tn}$		The intersection over the union respecting items predicted positive and observed positive. Also known as: threat score, ratio of verification, critical success index, Tanimoto coefficient.
balanced accuracy [15]	bacc	$\frac{1}{2}\left(\frac{tp}{p}+\frac{tn}{n}\right)$	$\frac{1}{2}(sens+spec)$	The mean of sensitivity and specificity.
Cohen's kappa [30]	kappa	$\frac{-2np+2n\cdot tp+2p\cdot tn}{n^2-n\cdot tn+n\cdot tp+p^2+p\cdot tn-p\cdot tp}$		Quantifies the agreement between the observed labeling and the labeling by the classifier, taking into account the probability of agreement by chance.

This section is organized as follows: In Subsection 3.1, we introduce the consistency test in terms of exhaustive search. In Subsection 3.2, we propose a more efficient implementation using interval computing. The method is illustrated through an example in Subsection 3.3, and finally, limitations are discussed in Subsection 3.4.

3.1. Testing by exhaustive search

The experimental setup (described by p and n) and the reported scores are consistent if there exist $tp^* \in \mathcal{P} \doteq \{0, \ldots, p\}$ and $tn^* \in \mathcal{N} \doteq \{0, \ldots, n\}$ integers such that

$$f_s(tp^*, tn^*, p, n) \in [\hat{v}_s - \epsilon, \hat{v}_s + \epsilon], \tag{1}$$

holds for each score $s \in \mathcal{S}$ simultaneously. This simple condition readily suggests an $O(p \cdot n|S|)$ time complexity algorithm based on exhaustive search. One can test each pair of $(tp, tn) \in \mathcal{P} \times \mathcal{N}$ to see if they satisfy the conditions. If there are no feasible pairs, the experimental setup and the reported scores are inconsistent. Although this brute-force algorithm is functional and can be applied to datasets of even medium sizes ($p + n \lessapprox 10K$), it is possible to reduce its time complexity, making it efficient for much larger datasets.

3.2. The improved time complexity test

The idea behind the improvement is that the double iteration through the potential values of both tp and tn could be reduced to one if we could determine for a given value of tp the values of tn leading to a desired performance score v_s^*. To do so, the analytical forms of the score functions need to be solved for the particular variables. Namely, from $v_s = f_s(tp, tn, p, n)$ solving

$$v_s - f_s(tp, tn, p, n) = 0 \tag{2}$$

for tn leads to the solution

$$tn = f_{s,tn}^{-1}(v_s, tp, p, n). \tag{3}$$

The solutions for all scores are provided in tables 3 and 4.

If we knew the exact values of the scores v_s^*, one would need to check if there exists any $tp \in \mathcal{P}$ such that $f_{s,tn}^{-1}(v_s^*, tp, p, n)$ gives the same integer result for each $s \in \mathcal{S}$.

In practice, we have only interval estimations for $v_s^* \in [\hat{v}_s - \epsilon, \hat{v}_s + \epsilon]$. Therefore, to evaluate (3) one needs to exploit interval arithmetic [32]. For example, given $p = 40$, $n = 70$ and $\hat{v}_{acc} = 0.927$, one wants to determine which tn values could lead to the reported score if $tp = 30$ (an arbitrary choice from $\mathcal{P}$). Selecting the solution $f_{acc,tn}^{-1}$ from Table 3 and evaluating it with interval arithmetic yields

$$f_{acc,tn}^{-1}([0.926, 0.928], 30, 40, 70) = [71.86, 72.08], \tag{4}$$

that is, the only integer tn can take to yield the reported accuracy score is $tn = 72$. We note that depending on the numerical uncertainty, the resulting intervals might contain multiple integers, and some scores have multiple solutions leading to the union of intervals (Table 4).

Ultimately, the consistency test can be carried out by iteratively testing if there exist at least one $tp \in \mathcal{P}$ such that the intersection $f_{s,tn}^{-1}([\hat{v}_s - \epsilon, \hat{v}_s + \epsilon], tp, p, n)$ for all $s \in \mathcal{S}$ contains at least one integer from the feasible set $\mathcal{N}$. If no such $tp \in \mathcal{P}$ exists, the experimental setup and the reported scores are inconsistent with each other. One can readily see that the choice of tp to iterate by is arbitrary; the consistency test could be implemented by iterating through $tn \in \mathcal{N}$ and using solutions for tp (also provided in Tables 3 and 4). Consequently, the time complexity can be reduced to $O(\min(p, n) \cdot |S|)$ if the figure with the smaller domain is chosen for iteration, leading to an efficient, linear time algorithm, which is tractable even when the evaluation set contains millions of records. Finally, since dynamic data structures are not utilized, the space complexity of the algorithm becomes $O(1)$. The detailed pseudo-code of the test is listed in Algorithm 1.

It is worth noting that the time complexity could be further reduced by solving pairs of performance scores as a system for tp and tn, eliminating the need for iteration by tp or tn. However, we found that solving pairs of scores with higher-order terms would require the involvement of advanced algebraic techniques, which falls beyond the scope of this paper.

Data: p, n, ϵ, the set of scores reported $\mathcal{S}$, the reported values $\hat{v}_s$, $s \in \mathcal{S}$
Result: *True* if inconsistency was found, *False* otherwise.
`/* Selecting the figure to solve for (`β`), the upper bound of the feasible set for` β `(B), and the upper bound of the integer set to iterate on (A) */`
if $p < n$ **then**
  $\beta \leftarrow$ 'tn';
else
  $\beta \leftarrow$ 'tp';
$A \leftarrow \min(p, n)$;
$B \leftarrow \max(p, n)$;
`/* Iterate through the possible values */`
for $\alpha = 0, 1, \ldots, A$ **do**
  $I \leftarrow \bigcap_{s \in \mathcal{S}} f^{-1}_{s,\beta}([\hat{v}_s - \epsilon, \hat{v}_s + \epsilon], \alpha, p, n)$;
  if $I \bigcap \{0, 1, \ldots, B\} \neq \emptyset$ **then**
    `/* Evidence found for feasibility */`
    return False;
`/* Inconsistency identified */`
return True;

Algorithm 1: Consistency testing for scores computed directly from the confusion matrix

Table 3: Scores with single solutions.

score (s)	tp formula ($f^{-1}_{s,tp}(s, tn, p, n)$)	tn formula ($f^{-1}_{s,tn}(s, tp, p, n)$)
acc	$acc \cdot n + acc \cdot p - tn$	$acc \cdot n + acc \cdot p - tp$
sens	$p \cdot sens$	
spec		$n \cdot spec$
ppv	$\frac{ppv\,(-n + tn)}{ppv - 1}$	$n + tp - \frac{tp}{ppv}$
npv	$p + tn - \frac{tn}{npv}$	$\frac{npv\,(-p + tp)}{npv - 1}$
fbp	$\frac{f_+^\beta \left(\beta_+^2 p + n - tn\right)}{\beta_+^2 - f_+^\beta + 1}$	$\frac{-\beta_+^2 tp + f_+^\beta \left(\beta_+^2 p + n + tp\right) - tp}{f_+^\beta}$
fbn	$\frac{-\beta_-^2 tn + f_-^\beta \left(\beta_-^2 n + p + tn\right) - tn}{f_-^\beta}$	$\frac{f_-^\beta \left(\beta_-^2 n + p - tp\right)}{\beta_-^2 - f_-^\beta + 1}$
gm	$\frac{gm^2 np}{tn}$	$\frac{gm^2 np}{tp}$
lrp	$\frac{lr_+ p\,(n - tn)}{n}$	$n - \frac{n \cdot tp}{lr_+ p}$
lrn	$\frac{p\,(-lr_- tn + n)}{n}$	$\frac{n\,(p - tp)}{lr_- p}$
bm	$\frac{p\,(n\,(bm + 1) - tn)}{n}$	$\frac{n\,(p\,(bm + 1) - tp)}{p}$
pt	$\frac{p\,(n - tn)}{n}$	$\frac{n\,(p - tp)}{p}$
dor	$\frac{dor \cdot p\,(n - tn)}{dor \cdot n - dor \cdot tn + tn}$	$\frac{dor \cdot n\,(p - tp)}{dor \cdot p - dor \cdot tp + tp}$
ji	$ji\,(n + p - tn)$	$n + p - \frac{tp}{ji}$
bacc	$\frac{p\,(2bacc \cdot n - tn)}{n}$	$\frac{n\,(2bacc \cdot p - tp)}{p}$
kappa	$\frac{\kappa\left(n^2 - n \cdot tn + p^2 + p \cdot tn\right) + 2np - 2p \cdot tn}{\kappa\,(-n + p) + 2n}$	$\frac{\kappa\left(n^2 + n \cdot tp + p^2 - p \cdot tp\right) + 2np - 2n \cdot tp}{\kappa\,(n - p) + 2p}$

3.3. Example

Suppose there is an evaluation set of $p = 1000$ positive and $n = 6000$ negative samples, and the reported scores are $\hat{v}_{acc} = 0.6801$, $\hat{v}_{npv} = 0.9401$ and $\hat{v}_{f_1} = 0.4004$. Being conservative and assuming the scores are floored or ceiled, the numerical uncertainty is $\epsilon = 0.0001$. Applying Algorithm 1, one finds that there are two pairs of (tp, tn) values compatible with the setup: (743, 4031) and (743, 4032). If the scores were slightly adjusted, for example, accuracy

Table 4: Scores with multiple solutions.

score	variable	formula
fm	$\mathrm{tp}_{1,2}$	$\frac{fm\left(fm\cdot p \pm \sqrt{p}\sqrt{fm^2p+4n-4tn}\right)}{2}$
	tn	$n+tp-\frac{tp^2}{fm^2p}$
mcc	$\mathrm{tp}_{1,2}$	$\frac{\pm mcc\sqrt{p}(n+p)\sqrt{mcc^2np+4n\cdot tn-4tn^2}+\sqrt{n}p\left(mcc^2(-n+p+2tn)+2n-2tn\right)}{2\sqrt{n}(mcc^2p+n)}$
	$\mathrm{tn}_{1,2}$	$\frac{\pm mcc\sqrt{n}(n+p)\sqrt{mcc^2np+4p\cdot tp-4tp^2}+n\sqrt{p}\left(mcc^2(n-p+2tp)+2p-2tp\right)}{2\sqrt{p}(mcc^2n+p)}$
mk	$\mathrm{tp}_{1,2}$	$\frac{mk(-n+p+2tn)-n\pm\sqrt{mk^2(n^2+2np+p^2)+mk(2n^2+2np-4n\cdot tn-4p\cdot tn)+n^2}}{2mk}$
	$\mathrm{tn}_{1,2}$	$\frac{mk(n-p+2tp)\pm p-\sqrt{mk^2(n^2+2np+p^2)+mk(2np-4n\cdot tp+2p^2-4p\cdot tp)+p^2}}{2mk}$
upm	$\mathrm{tp}_{1,2}$	$\frac{n}{2}+\frac{p}{2}+tn+\frac{-2tn\pm\frac{\sqrt{16tn^2+upm^2(n^2+2np+8n\cdot tn+p^2+8p\cdot tn)+upm(-8n\cdot tn-8p\cdot tn-16tn^2)}}{2}}{upm}$
	$\mathrm{tn}_{1,2}$	$\frac{n}{2}+\frac{p}{2}+tp+\frac{-2tp\pm\frac{\sqrt{16tp^2+upm^2(n^2+2np+8n\cdot tp+p^2+8p\cdot tp)+upm(-8n\cdot tp-8p\cdot tp-16tp^2)}}{2}}{upm}$

changed to 0.6811, Algorithm 1 concludes that there is no (tp, tn) pair fulfilling all conditions. Similarly, if the assumption for p is incorrect, for example, $p = 1100$, Algorithm 1 identifies the inconsistency.

3.4. Limitations and power analysis

The proposed test is linear in the size of the evaluation set as well as in the number of reported scores; therefore, there should be no computational limitations with reasonably sized datasets. As the examples illustrate, the test is capable of recognizing inconsistencies in reported performance scores and presumed experimental setups. Naturally, smaller numerical uncertainty and more scores being reported both increase the sensitivity of the test. However, it is still unclear how sensitive the test really is to deviations from the assumptions. To address this question – although the proposed method is numerical – we draw an analogy with statistical hypothesis testing and utilize the existing terminology.

In the proposed consistency tests, the *null hypothesis* is that the reported scores and the presumed experimental setup are consistent; the *alternative hypothesis* is that they are inconsistent in some sense, that is, the scores are not calculated in the presumed way. Inconsistencies are identified with certainty; therefore, the probability of a *type I error* (false rejection of the null hypothesis [33]) is zero. However, the test can still make *type II errors* [33], meaning that the null hypothesis is false (the scores are not calculated in the assumed way), but the test fails to recognize the inconsistency. The probability of making a type II error (β) is the complement of the *power* ($1-\beta$) of the test [33], which is assessed in *power analysis* [33] in the field of statistical hypothesis testing. Consequently, characterizing the sensitivity of the proposed test to recognize inconsistencies is equivalent to carrying out its power analysis.

Similar to statistical hypothesis testing, the power analysis requires making assumptions about the nature of the deviations (for example, typographical error in the last decimal place of a score; differing number of positive samples used to calculate the scores from those presumed, etc.) and also depends on the actual parameters (numerical uncertainty, the number of reported scores, etc.) of the problem. Therefore, similar to statistical hypothesis testing, a general power analysis cannot be carried out. However, it can be done in particular cases by simulations, similar to the one carried out in our previous paper addressing inconsistencies in the field of retinal vessel segmentation [18] and in Subsection 5.3 of this paper.

4. Testing scores derived by aggregations

In this section, we develop tests for those scenarios where the scores are aggregated over multiple evaluation sets (folds and/or datasets). The mode of aggregation (discussed in Subsection 4.1) leads to different tests, that we cover

in Subsections 4.2 and 4.3. The mapping of some common kFCV schemes to the representation used by the tests is discussed in Subsection 4.4. Finally, limitations are discussed in Subsection 4.5.

4.1. Mean of Scores and Score of Means aggregations

We assume an experiment involving N_e evaluation sets with p_i and n_i positive and negative samples, respectively, for $i = 1, \ldots, N_e$, each leading to a separate confusion matrix with entries $tp_i \in \{0, \ldots, p_i\}$ and $tn_i \in \{0, \ldots, n_i\}$. We are concerned about how these figures are summarized by scalar scores for the entire experiment.

A natural way of aggregation is to calculate a score for each evaluation set separately and take the average. Formally, for a particular score s, the overall score is calculated as

$$v_s^{MoS} = \frac{1}{N_e} \sum_{i=1}^{N_e} f_s(tp_i, tn_i, p_i, n_i), \tag{5}$$

where we introduced the notion of *Mean of Scores* (MoS) to indicate the way of aggregation. We note that the MoS mode of aggregation is extremely common in the evaluation of binary classifiers in cross-validation scenarios, with the benefit that the N_e-sized sample of scores enables the estimation of confidence intervals [34] and the use of hypothesis testing for the comparison of classification techniques [35].

Alternatively, one can calculate the averages of the tp, tn, fp, and fn figures first, for example, $\overline{tp} = \frac{1}{N_e} \sum_{i=1}^{N_e} tp_i$, and compute the scores from the mean figures as

$$v_s^{SoM} = f_s\left(\overline{tp}, \overline{tn}, \overline{p}, \overline{n}\right), \tag{6}$$

where we introduced the notion of *Score of Means* (SoM) to reflect the way of aggregation. One can readily see that the SoM aggregation is equivalent to a weighted MoS aggregation when the weights are defined as the denominators of the scores. SoM aggregation is beneficial when the scores for some individual evaluation sets might become undefined, typically with small and imbalanced data [36] (for example, if a fold has only a handful of positive samples, $tp = 0$ and $fp = 0$ lead to an undefined positive predictive value, which is a less likely scenario for $\overline{tp}$ and $\overline{fp}$).

The terms used for the aggregations are inspired by the analogous concepts of *Ratio of Means* (RoM) and *Mean of Ratios* (MoR) estimations for ratio statistics [37, 38], but generalized to accommodate the non-linearities in the numerators and denominators of some scores (such as Matthews correlation coefficient).

From the theoretical point of view, the goal of using multiple evaluation sets and aggregating the results is to get a more reliable estimation of performance for the population of problems represented by the evaluation sets. (We note that this concept leads to difficulties when multiple datasets are involved, as the population of classification problems represented by some datasets is usually not well-defined [39].) Nevertheless, estimation theory [40] can be expected to provide a guideline on which aggregation is more reasonable. Interestingly, already for the simplest scores (with linear terms in the numerator and denominator) it turns out that both aggregation schemes (5) and (6) are biased estimators of the population level statistics [38]. Moreover, even in the same experiment, different scores can lead to different interpretations regarding their meaning. For example, in kFCV, if the total number of samples ($p + n$) is divisible by the number of folds, the fold-level accuracy scores have the same constant denominator, the MoS and SoM aggregations become the same, and the aggregated accuracy becomes an unbiased estimator of the population level proportion of correctly classified items. In the same scenario, sensitivity has randomness in the denominator since the various folds can have varying number of positive samples; consequently, one can argue that weighting by the number of positive samples in a fold (using SoM) is a meaningful way to reduce noise. Finally, positive predictive value has correlated randomness in its numerator and denominator (through the presence of tp), leading to both the SoM and MoS aggregations becoming biased estimators [38]. Consequently, there is no consensus on the superiority of either mode of aggregation.

In practice, when the data distribution across evaluation sets is fairly uniform, the scores calculated by both aggregations are nearly identical (see Table 5). Therefore, authors often do not explicitly describe the method of aggregation, as it is not expected to significantly alter the qualitative outcome of the research. A particular choice could be motivated by multiple factors, for example: the best practices of a field, the available implementation, the need to estimate the uncertainty of the scores, small and imbalanced data, etc. Even in the same domain, with the

Table 5: Comparison of the MoS and SoM evaluations on sample data in a k-fold cross-validation scenario with $k = 5$: the folds and a sample evaluation are shown in Table 5a, and the scores in Table 5b. As expected, the more non-linearities are present in a score, the more the two aggregations deviate, but the most commonly used scores (accuracy, sensitivity, specificity, f_+^1) are very close to each other.

(a) The folds.

fold (i)	p_i	n_i	tp_i	tn_i
0	100	201	78	189
1	100	200	65	191
2	100	200	81	160
3	101	200	75	164
4	101	200	72	171

(b) The scores calculated by the two aggregation techniques.

score	MoS	SoM	score	MoS	SoM	score	MoS	SoM	score	MoS	SoM
acc	0.8290	0.8290	f_+^1	0.7443	0.7427	lrn	0.2975	0.2985	ppv	0.7606	0.7465
bacc	0.8066	0.8066	fm	0.7471	0.7428	lrp	8.1202	5.8713	pt	0.2795	0.2921
bm	0.6131	0.6132	gm	0.8021	0.8038	mcc	0.6215	0.6147	sens	0.7391	0.7390
dor	28.0174	19.6671	ji	0.5945	0.5908	mk	0.6312	0.6163	spec	0.8741	0.8741
f_-^1	0.8709	0.8719	kappa	0.6165	0.6147	npv	0.8706	0.8698	upm	0.8025	0.8022

same data, one can find examples of both aggregations [18]. Therefore, unless explicitly phrased, one cannot assume any aggregation as default.

Since we develop sharp tests to check the consistency of reported scores, even minor differences must be handled with mathematical rigor. Therefore, in Subsections 4.2 and 4.3, we develop consistency tests for the two types of aggregations separately.

4.2. Testing scores aggregated by the Score of Means *approach*

While we introduced the term *Score of Means* to reflect the analogy with the concept of *Ratio of Means* in statistics, taking the mean of the figures tp, tn, p, and n (equation 6) is unnecessary. It can be readily seen that all scores covered in the paper (see Table 2) are invariant to scaling. In other words, for any score s, the equation $f_s(tp, tn, p, n) = f_s(\alpha \cdot tp, \alpha \cdot tn, \alpha \cdot p, \alpha \cdot n)$ holds for $\alpha \in \mathbb{R}^+$, and consequently,

$$f_s(\overline{tp}, \overline{tn}, \overline{p}, \overline{n}) = f_s\left(\sum_{i=1}^{N_e} tp_i, \sum_{i=1}^{N_e} tn_i, \sum_{i=1}^{N_e} p_i, \sum_{i=1}^{N_e} n_i\right). \tag{7}$$

Therefore, any score calculated using the SoM approach can be treated as if it were calculated from the confusion matrix of a problem with $p' = \sum_{i=1}^{N_e} p_i$ positive and $n' = \sum_{i=1}^{N_e} n_i$ negative samples. Consequently, when scores aggregated in the SoM manner are reported, the consistency tests developed in Section 3 are applicable using the total number of positives p' and negatives n'.

4.3. Testing scores aggregated by the Mean of Scores *approach*

In this section, we develop consistency tests for the MoS aggregation. First, we formulate the problem mathematically in Subsection 4.3.1, then propose a tractable algorithm based on integer linear programming in Subsection 4.3.2, and finally, illustrate the use of the technique in Subsection 4.3.3.

4.3.1. Mathematical formulation

According to the definition of MoS aggregation (equation (5)), we are concerned with the simultaneous feasibility of the inequalities:

$$\hat{v}_s^{MoS} - \epsilon \leq \frac{1}{N_e} \sum_{i=1}^{N_e} f_s(tp_i, tn_i, p_i, n_i) \leq \hat{v}_s^{MoS} + \epsilon, \text{ for } s \in \mathcal{S}, \tag{8}$$

where $tp_i \in \{0, \ldots, p_i\}$ and $tn_i \in \{0, \ldots, n_i\}$ for $i = 1, \ldots, N_e$. Due to the non-linearities and the presence of raw figures tp_i and tn_i in both the numerators and denominators of most scores, the averaging cannot be simplified, resulting in a total of $2N_e$ degrees of freedom in the general case.

Similarly to the approach introduced in Subsection 3.1, one could enumerate all possible combinations of the $0 \leq tp_i \leq p_i$ and $0 \leq tn_i \leq n_i$, $i = 1, \ldots, N_e$ figures and check if any of them leads to the reported scores $\hat{v}_s^{MoS}$, $s \in \mathcal{S}$

within the numerical uncertainty ϵ. However, the time complexity $O\left(\prod_{i=1}^{N_e} p_i n_i\right)$ of this brute-force approach renders it intractable in practice, even in the simplest cases: a 5-fold evaluation with $p_i \sim 10$ positive and $n_i \sim 10$ negative records in each fold lead to approximately 10^{10} different combinations of the free parameters.

4.3.2. Feasibility by integer linear programming

The condition set (8) can be interpreted as the definition of the feasibility region of a non-linear integer programming problem (with any dummy objective function). In general, non-linear integer programming is NP-complete [41], with no efficient algorithms for exact solutions, and approximations are not suitable for sharp consistency tests requiring exact decisions regarding feasibility.

However, for that subset of scores which leads to linear conditions, integer linear programming can be exploited, which is solvable by numerous techniques [41]. Consequently, the proposed consistency test for MoS aggregations supports only those scores which are linear functions of tp and tn, namely, *accuracy*, *sensitivity*, *specificity*, and *balanced accuracy*. Although this test is limited to these four scores only, we note that these scores are among the most commonly reported ones. Expanding (8) for the linear scores leads to the condition set

$$\begin{aligned}
\hat{v}_{acc}^{MoS} - \epsilon &\leq \frac{1}{N_e}\sum_{i=1}^{N_e}\frac{tp_i + tn_i}{p_i + n_i} \leq \hat{v}_{acc}^{MoS} + \epsilon, \\
\hat{v}_{sens}^{MoS} - \epsilon &\leq \frac{1}{N_e}\sum_{i=1}^{N_e}\frac{tp_i}{p_i} \leq \hat{v}_{sens}^{MoS} + \epsilon, \\
\hat{v}_{spec}^{MoS} - \epsilon &\leq \frac{1}{N_e}\sum_{i=1}^{N_e}\frac{tn_i}{n_i} \leq \hat{v}_{spec}^{MoS} + \epsilon, \\
\hat{v}_{bacc}^{MoS} - \epsilon &\leq \frac{1}{N_e}\sum_{i=1}^{N_e}\frac{tp_i}{2p_i} + \frac{tn_i}{2n_i} \leq \hat{v}_{bacc}^{MoS} + \epsilon, \\
tp_i \in \{0,\ldots,p_i\}, &\quad tn_i \in \{0,\ldots,n_i\},
\end{aligned} \tag{9}$$

which is the most general set of conditions that is compatible with integer integer programming. The consistency test operates by specifying the conditions (9) for the available scores $\mathcal{S} \bigcap \{acc, sens, spec, bacc\}$, and using any integer linear programming solver to check the feasibility of the condition set. If the conditions are feasible, there is no inconsistency between the scores and the experimental setup; if the feasibility region is empty, the reported scores and the assumptions on the experimental setup are inconsistent.

We mention that there is one more piece of information that is sometimes reported and can strengthen the test: the minimum and maximum scores across folds. As noted in Subsection 4.1, one benefit of using MoS aggregation is that one gets a distribution of scores, and sometimes authors report the minimum and maximum values achieved across the folds. Adding these constraints shrinks the feasibility region and improves the sensitivity of the test. For example, if minimum ($\hat{v}_{min(acc)}^{MoS}$) and maximum ($\hat{v}_{max(acc)}^{MoS}$) scores are reported for accuracy, additional N_e pieces of constraints can be added to the linear programming problem:

$$\hat{v}_{min(acc)}^{MoS} - \epsilon \leq \frac{tp_i + tn_i}{p_i + n_i} \leq \hat{v}_{max(acc)}^{MoS} + \epsilon, i = 1,\ldots,N_e \tag{10}$$

We note that under special circumstances (stratified kFCV, both p and n divisible by the number of folds), for some subsets of the scores possibly fractional or convex programming with certain relaxation techniques could be exploited [42]. However, the exploration of these special cases is beyond the scope of the paper.

Finally, the time complexity of the test is equivalent to that of integer linear programming, which is known to be NP-hard [41]. However, due to the relatively small number of folds used in typical experiments, the problems to be solved are typically small, and the test remains tractable in practice. The space complexity of integer linear programming depends on the algorithm implemented by a particular solver.

4.3.3. Example

The usage of the test is illustrated through the sample problem shared in Table 5. Suppose the scores

$$\hat{v}_{acc}^{MoS} = 0.8290, \quad \hat{v}_{sens}^{MoS} = 0.7391, \quad \hat{v}_{spec}^{MoS} = 0.8741 \tag{11}$$

are reported. With a conservative choice of numerical uncertainty set at $\epsilon = 0.0001$ (allowing ceiling or flooring), after substituting the scores and the fold specifications from Table 5a into (9), and testing the feasibility using an integer linear programming solver (specifically, we used the Python package `pulp` [43]), the solver confirms that the problem is feasible. This result suggests that the reported scores could indeed have been obtained from the experiment. However, if accuracy is incorrectly reported as $\hat{v}_{acc}^{MoS} = 0.8280$, the solver returns that the configuration is infeasible, indicating that the scores are incompatible with the assumptions on the experiment.

4.4. Application of the tests in various experimental setups

In the previous sections, we introduced consistency tests for the SoM and MoS modes of aggregation. In this section, we discuss the assessment of various kFCV schemes using these tests. In a kFCV experiment with k folds on a dataset comprising p positive and n negative entries, we define the *fold configuration* as the distribution of positives and negatives across the k folds, denoted as $(p_i, n_i)_{i=1}^{k}$. As usual in kFCV, we assume $p_i + n_i$ is either $\lfloor (p+n)/k \rfloor$ or $\lfloor (p+n)/k \rfloor + 1$. Additionally, we require that at least two folds contain at least one negative and at least two folds contain at least one positive sample. This ensures that all training sets in the folding process contain samples of both classes, and at least accuracy is calculable in each iteration. We treat fold configurations as multisets, where a certain pair of positive and negative counts can appear multiple times. The order of the pairs is irrelevant, as they lead to the same linear programming problem regardless of order.

4.4.1. Testing the SoM scores of kFCV experiments

In the case of SoM aggregation, as discussed in Subsection 4.2, only the overall counts of positive and negative samples are necessary for testing. Therefore, in a standard kFCV experiment (where each entry of a dataset is evaluated once), the parameters p and n of the dataset should be used. Generally, in a repeated kFCV scenario involving N_d datasets with N_r repetitions, since the fold configurations are irrelevant, the parameters $p' = N_r \cdot \sum_{i=1}^{N_d} p_i$ and $n' = N_r \cdot \sum_{i=1}^{N_d} n_i$ need to be used for testing.

4.4.2. Testing the MoS scores of kFCV experiments

For MoS aggregations, knowing the fold configuration is crucial to formulate the linear programming problem (9). While some data providers, such as the KEEL data repository [44], supply foldings of the datasets, the fold configuration is typically unknown in general kFCV scenarios. However, there are special cases where the fold configuration can be inferred. Particularly, when *stratified KFCV* is used, proper stratification methods (such as the technique implemented in the *sklearn* [45] Python package) ensure that each fold differs by at most one sample in terms of the overall count of items, as well as the counts of positives and negatives. This characteristic implies a unique fold configuration that can be determined from p, n and k as shown in Table 6. In repeated kFCV experiments or when using multiple datasets, the joint fold configuration needs to be used.

4.4.3. Testing in the lack of knowing the fold configuration

When stratification is not used or its use is not explicitly indicated in a paper, any fold configuration can be assumed. Addressing these scenarios requires enumerating and testing all possible fold configurations. Although this might seem intractable, it is feasible even in real-life applications, as demonstrated in Subsection 5.2, especially when the number of folds falls in the usual range (5-10) and the dataset is imbalanced or relatively small. If all configurations prove inconsistent with the reported scores, it can be concluded that the scores could not have been yielded from the assumed experiment. Enumerating all possible k-fold configurations given p, n, and k is a non-trivial combinatorial problem. In the remainder of this section, we develop an algorithm tailored specifically for this task.

The proposed algorithm is based on the observation that enumerating all fold configurations is closely related to the problem of integer partitioning in combinatorics and number theory [46]. Specifically, we are interested in the various ways p (or n) can be decomposed as $p = p_1 + \cdots + p_k$. Given a particular partition, it can be complemented

Table 6: Fold configurations by the stratified kFCV implemented in *sklearn*, with $p_{mod} = p \mod k$, $p_{div} = \lfloor p/k \rfloor$, $n_{mod} = n \mod k$, $n_{div} = \lfloor n/k \rfloor$. The *count of folds* column indicates how many times folds with p_i and n_i counts appear in the configuration, when $p_{mod} + n_{mod} > k$ (a) and $p_{mod} + n_{mod} \leq k$ (b).

(a) If $p_{mod} + n_{mod} > k$

count of folds	positives (p_i)	negatives (n_i)
$p_{mod} + n_{mod} - k$	$p_{div} + 1$	$n_{div} + 1$
$k - n_{mod}$	$p_{div} + 1$	n_{div}
$k - p_{mod}$	p_{div}	$n_{div} + 1$

(b) If $p_{mod} + n_{mod} \leq k$

count of folds	positives (p_i)	negatives (n_i)
$k - p_{mod} - n_{mod}$	p_{div}	n_{div}
p_{mod}	$p_{div} + 1$	n_{div}
n_{mod}	p_{div}	$n_{div} + 1$

with negatives to achieve the desired cardinality of folds ($n_i \sim (p+n)/k - p_i$), resulting in one fold configuration. There are algorithms proposed for the enumeration of all unique partitions of an integer to m positive parts (for example, the algorithm on page 343 in [46]). The main complexity lies in addressing the varying cardinalities of folds when the total number of elements ($p + n$) is not divisible by the number of folds (k).

Let $k_{div} = \lfloor (p + n)/k \rfloor$ and $k_{mod} = (p + n) \mod k$. There are two types of folds regarding cardinalities, denoted by superscripts a and b: $k^a = k_{mod}$ folds, each with $c^a = k_{div} + 1$ elements, and $k^b =:= k - k_{mod}$ folds each with $c^b = k_{div}$ elements. Without the loss of generality, we choose the number of positives to drive the enumeration. Suppose there are overall p^a positive samples in the folds of type a, implying $p^b = p - p^a$ positive samples in the folds of type b. The various configurations of p^a positives distributed in the folds of type a can be determined by enumerating all integer partitions of p^a into at most k^a parts. Similarly, for folds of type b, one can determine all integer partitions of p^b into at most k^b parts. One combination of the partitions of positives in folds of type a and b can be complemented by negative samples to match the cardinality of the respective folds, resulting in a unique fold configuration. By iterating through all possible ways to split the total number of positives p between the two types of folds, all fold configurations can be generated. A precise algorithm (with all technical details such as the removal of configurations not having a certain class present in at least two folds) is provided in Algorithm 2. We note that in practice, depending on the scores reported, some configurations can be skipped. For instance, if sensitivity is reported, configurations with folds having zero positives lead to undefined sensitivity scores, contradicting the fact that the average sensitivity is reported.

For instance, in the case of $p = 30$, $n = 300$, and $k = 5$ (which is comparable in size to many small and imbalanced medical datasets), the total number of fold configurations is 673. As an example, one particular output yielded by the generator in Algorithm 2 is a pair of vectors $\mathbf{p} = [1, 2, 6, 9, 12]$, $\mathbf{n} = [65, 64, 60, 57, 54]$ representing the fold configuration $[(p_1 = 1, n_1 = 65), (p_2 = 2, n_2 = 64), (p_3 = 6, n_3 = 60), (p_4 = 9, n_4 = 57), (p_5 = 12, n_5 = 54)]$.

4.4.4. Testing in the lack of knowing the mode of aggregation

If mode of aggregation (MoS or SoM) is unknown, one can still apply the proposed consistency tests to identify inconsistencies: the scores can be tested under both assumptions. If both tests lead to inconsistencies, it can be concluded that the reported scores could not have resulted from the experimental setup under either of these two reasonable modes of aggregation.

4.5. Limitations and power analysis

The tests for aggregated scores rely on integer linear programming and inherit the limitations of the available solvers. Since the number of free variables is twice the number of evaluation sets (see eq. (9)), the integer linear programs implied by typical 5- or 10-fold cross-validation schemes (10-20 free variables) are usually tractable. However, using many more evaluation sets can deteriorate the solvability of the system. Naturally, smaller numerical uncertainty and more reported scores both improve the ability of the test to recognize inconsistencies.

The analogy drawn between the proposed tests and statistical hypothesis testing in Subsection 3.4 is equally valid for testing aggregated scores. Assessing the sensitivity of the test to recognize inconsistencies requires a power analysis, which cannot be conducted without making assumptions about the nature of the deviations. However, in specific cases, it can be done through numerical simulations, similar to the analyses carried out in our previous paper [18] addressing inconsistencies in the field of retinal vessel segmentation, and also in Subsection 5.2 of this paper.

Generator `KFoldConfigurations(`*p, n, k*`)`**:**
  Description: *Generates all fold configurations.*
  Args: *the number of positives (p), negatives (n), and folds ($k \geq 2$)*
  Yields: $(\mathbf{p}, \mathbf{n}) \in \mathbb{N}^{k\times k}$, $(\mathbf{p}_i, \mathbf{n}_i)$ *representing the number of positives and negatives in the ith fold, respectively.*
  $k_{div}, k_{mod} \leftarrow \lfloor (p+n)/k \rfloor, (p+n) \mod k$;
  $k^a, k^b \leftarrow k_{mod}, k - k_{mod}$;
  $c^a, c^b \leftarrow k_{div} + 1, k_{div}$;
  for $p^a = 0, \ldots, \min\{p, k^a \cdot c^a\}$ **do**
    for $\mathbf{p}^a$ *in* `Partition(`p^a, k^a, c^a`)` **do**
      for $\mathbf{p}^b$ *in* `Partition(`$p - p^a$, k^b, c^b`)` **do**
        $\mathbf{p} \leftarrow [\mathbf{p}^a, \mathbf{p}^b]$;
        $\mathbf{n} \leftarrow [\mathbf{1}_{k^a} \cdot c^a - \mathbf{p}^a, \mathbf{1}_{k^b} \cdot c^b - \mathbf{p}^b]$;
        if $\sum_{i=1}^{k} \mathbb{I}_{\mathbf{p}_i>0} \geq 2 \wedge \sum_{i=1}^{k} \mathbb{I}_{\mathbf{n}_i>0} \geq 2$ **then**
          Yield: $\mathbf{p}, \mathbf{n}$

Generator `Partition(`*q, k,* q_{max}`)`**:**
  Description: *Generates all partitions of q with at most k parts, no part greater than* q_{max}.
  if $q = 0$ **then**
    Yield: $\mathbf{0}_k$
  for $m = 1, \ldots, \min(q, k)$ **do**
    for $\mathbf{q}$ *in* `MPartition(`*q, m*`)` **do**
      if $\sum_{i=1}^{m} \mathbb{I}_{\mathbf{q}_i > q_{max}} = 0$ **then**
        Yield: $[\mathbf{0}_{k-m}, \mathbf{q}]$

Generator `MPartition(`*q, m*`)`**:**
  Description: *Implements the algorithm on page 343 in [46], iteratively yielding one unique partitioning of q to m parts:* $\mathbf{q} \in \mathbb{Z}_+^m$.

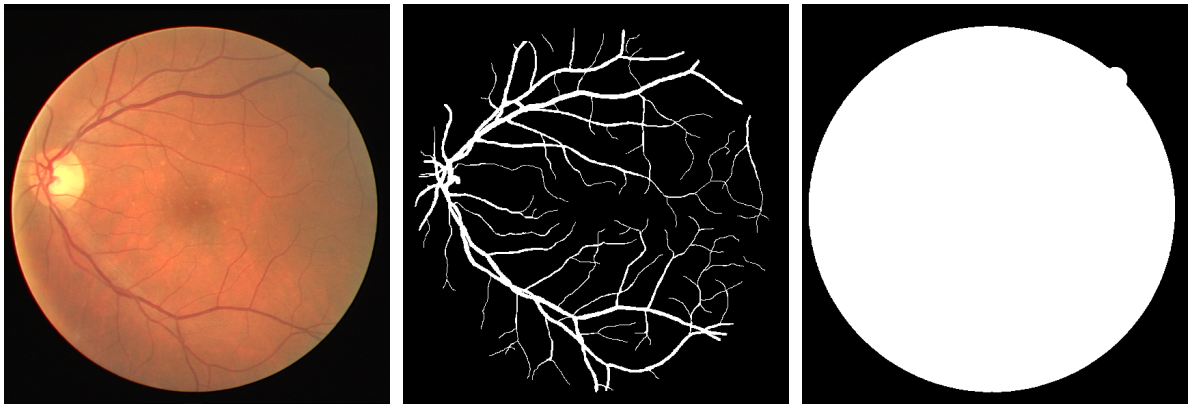
Algorithm 2: The algorithm generates all possible fold configurations, ensuring that for each class there at least 2 folds containing at least one sample, thereby ensuring a valid training set in all configurations. $\mathbf{0}_x$ and $\mathbf{1}_x$ denote the x dimensional 0-vector and 1-vector, respectively, $[\mathbf{x}, \mathbf{y}]$ stands for the concatenation of vectors $\mathbf{x}$ and $\mathbf{y}$.

5. Applications

In this section, we demonstrate the application of the proposed consistency tests in three real problems related to the use of machine learning in medicine and also discuss potential further applications.

5.1. Retinal vessel segmentation and further applications in retina image processing

The techniques proposed in this paper are generalizations of those used in our previous work [18] in the field of retinal vessel segmentation. In this subsection, we provide a concise overview of that scenario, including the results and potential applications in related fields. The field of retinal vessel segmentation has been a popular research area for nearly two decades. The most widely used dataset for evaluation (DRIVE [47]) offers 20 training and 20 test images with manual annotations. Due to the well-defined test set, the reported performance scores – typically accuracy, sensitivity, and specificity – serve as the basis for ranking algorithms in most papers. Due to the specialized image acquisition techniques, the useful image content resides in a disk-shaped area in the center of a rectangular image, referred to as the *Field of View* (FoV) (see Figure 2 for one entry of the DRIVE dataset). Textual evidence suggests that some authors evaluate the segmentation performance within the FoV region only, while others use all pixels in the images. However, in most papers, the region of evaluation is not specified explicitly. The problem arises from the fact that the pixels outside the FoV region can easily be identified as non-vessel pixels, and account for about 30% of all pixels. Including these pixels as true negatives boosts the accuracy and specificity scores compared to evaluating only the pixels within the FoV region.



(a) Retina image (b) Vessel mask (c) Field of View mask

Figure 2: One entry ("21") of the DRIVE retinal vessel segmentation dataset: a retina image (2a); the vessel mask (white pixels indicate the vasculature to be segmented) (2b); and the FoV mask, white pixels indicating the FoV region (2c).

For a fair comparison and ranking of algorithms, it is desired to identify the evaluation methodology used for calculating a certain set of scores. This can be achieved by the proposed methods (Sections 3, 4.2 and 4.3) by testing the consistency of the scores under two assumptions:

- The authors used only the pixels in the FoV region for evaluation;
- The authors used all pixels of the images for evaluation (note that the overall number of negatives, n, is different).

In our meta-analysis [18], we selected the 100 most cited papers in the field and assessed the consistency of both image-level and aggregated scores under the two assumptions, uncovering the following key insights into the state of the art in the field:

1. Approximately 30% of the papers reported scores that are inconsistent with both assumptions, casting doubt on their validity.
2. For the remaining papers, we identified the evaluation region, and revealed a systematic bias: authors using all pixels of the images reported higher performance scores – an artifact of the undisclosed evaluation methodology – resulting in skewed algorithm rankings favoring those using all pixels for evaluation.
3. In 100 of the most cited papers in the field, incomparable performance scores were compared and ranked regardless of the evaluation region, leading to biased conclusions about the capabilities of certain algorithms.

Numerous other lesions and anatomical structures in retinal images, such as *exudates* [48] and the *optic disk* [49], are targeted by segmentation and detection algorithms. The possibility of evaluating their performance in the FoV region or using all pixels in the images exists in all these problems. Building on the successful application of the proposed techniques for retinal vessel segmentation, it is reasonable to assume that these methods can be applied to validate and rectify the reported results in other problems of retinal image processing.

5.2. Preterm delivery prediction from electrohysterogram signals

In recent years, there has been growing interest in predicting preterm delivery from electrohysterogram (EHG) signals [50], particularly with the availability of the TPEHG dataset [51], containing 38 positive and 262 negative records. At some point, multiple authors reported almost perfect prediction scores in kFCV scenarios. However, in the study [52], it was revealed that these exceptionally high performance scores could not be replicated. The root cause of these overly optimistic results was traced to a methodological flaw: the improper use of minority oversampling.

The Synthetic Minority Oversampling TEchnique (SMOTE) [53] and its variations are commonly employed techniques to enhance the performance of binary classification on highly imbalanced data [54]. These techniques involve artificially generating additional training samples for the minority class to address the asymmetric degeneracy in the learning process. When employing minority oversampling in a kFCV scenario, it is critical to apply oversampling to each training set separately, excluding any elements of the fold designated for testing. Applying oversampling to the entire dataset prior to kFCV adds highly correlated samples to the dataset, leading to a significant data leakage. The authors of [52] reproduced all of 11 studies and concluded that the most likely cause for the overly optimistic results is the application of minority oversampling prior to kFCV.

When minority oversampling is applied prior to kFCV, it increases the number of samples used for evaluation. Therefore, the scores derived from the augmented dataset can be expected to exhibit inconsistencies with the correct experimental setup. Consequently, the time-consuming and error-prone task of reimplementing algorithms to verify the results (as the authors did in [52]) could be replaced by employing the techniques developed in this paper. For example, one study with the methodological flaw [55] reported the scores $\hat{v}_{acc}^{MoS} = 0.9447$, $\hat{v}_{sens}^{MoS} = 0.9139$ and $\hat{v}_{spec}^{MoS} = 0.9733$ in a 5-fold cross-validation scenario. Although the authors mentioned using minority oversampling to increase the overall number of positive samples to $p' = 244$, it remains unclear whether they used the additional samples solely for training or also for evaluation. The authors did not mention using stratification, hence testing all possible fold configurations is necessary for conclusive results. With the parameters of the correct experimental setup ($p = 38$, $n = 262$, and $k = 5$), the number of possible fold configurations becomes 918 (determined by the exhaustive enumeration algorithm presented in Subsection 4.4.3). Applying the MoS test (Subsection 4.3) reveals inconsistencies between the reported scores and each of the 918 configurations, suggesting that the reported scores could not have been obtained through 5-fold cross-validation on the original dataset. Under the assumption of using $p' = 244$ positive samples (i.e., including the highly correlated generated samples for evaluation), the total number of possible fold configurations expands to approximately $\sim 2.6M$. Notably, the 962nd fold configuration $[(1, 101), (4, 97), (40, 61), (99, 2), (100, 1)]$ already provides evidence that the reported scores could be achieved with the p' and n counts and 5-fold cross-validation, with the corresponding (tp_i, tn_i) counts of $[(1, 96), (3, 92), (38, 59), (90, 2), (96, 1)]$. With this example, we have provided numerical confirmation of the findings in [52] regarding the improper use of minority oversampling in [55], without the need to reimplement [55].

The power analysis of the method in this particular problem can be conducted as follows. Firstly, we specify the type of inconsistencies for which we are performing the power analysis. In this case, a reasonable choice is inconsistencies arising from the application of minority oversampling prior to 5-fold cross-validation. Specifically, the reported accuracy, sensitivity and specificity scores are calculated from a dataset with $p = 262$ and $n = 262$ using 5-fold cross-validation. We simulate such scenarios by randomly drawing the number of true positives and true negatives for each fold, calculating the performance scores, averaging them, and rounding to k decimal places. Subsequently, we apply the proposed consistency test assuming the proper experimental setup ($p = 38$) and record whether inconsistencies are detected. Based on 1000 such test scenarios, we estimate the probability of detecting this specific type of inconsistency when the performance scores are rounded to 2, 3, or 4 decimal places. The results are summarized in Figure 3, where the vertical axis represents the power of the test (probability of recognizing the inconsistency). As depicted in the figure, when performance scores are reported with 4 decimal places, which is typical in the field, the probability of detecting inconsistencies due to the improper use of minority oversampling is 0.71. While there remains a 29% chance of false negatives, the outcome suggests that the proposed technique effectively identifies this specific type of methodological flaw. As anticipated, the power declines when the scores are reported with fewer digits. It is worth noting that similar power analyses could be conducted for various other types of inconsistencies, such as assuming typographical errors in the reported scores.

Beyond eliminating the need to reimplement algorithms to identify methodological flaws in this specific application, the example also highlights that the proposed consistency tests are effective for detecting similar methodological flaws in various other fields where minority oversampling [53] is used.

5.3. Classification of skin lesions

In this subsection, we illustrate the application of the proposed techniques in a field where – to the best of our knowledge – no meta-analysis aimed at validating reported results has been conducted before: the classification of skin lesion images. An analysis as detailed as the one we conducted in retinal vessel segmentation [18] is clearly beyond the scope of this paper. However, we can test the reported scores in some highly influential papers to see

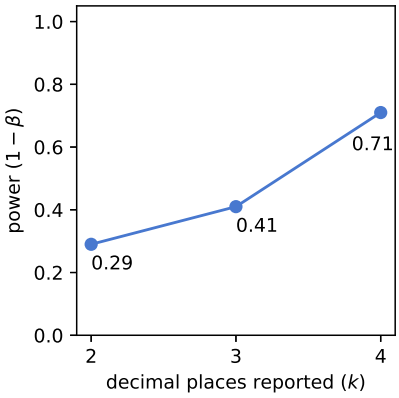


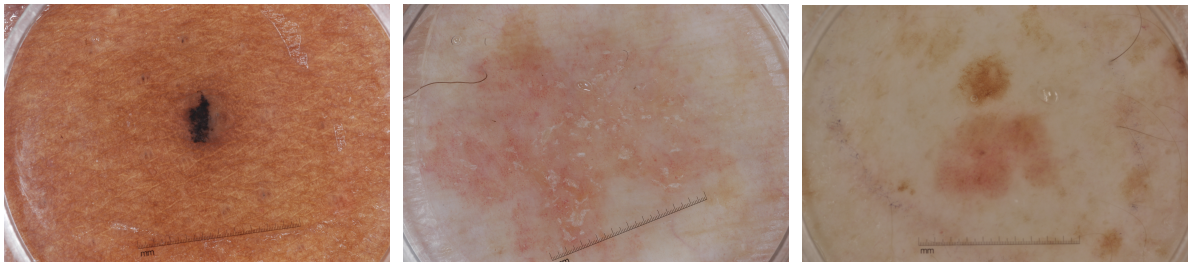
Figure 3: The results of the power analysis for the TPEHG preterm delivery dataset. The assumed inconsistency is that minority oversampling is carried out prior to 5-fold cross-validation. The vertical axis represents the power of the test $(1-\beta)$, as a function of the number of decimal places reported.

if ambiguities are present. The analysis is based on the highly cited survey [56] providing a systematic overview of research up until 2021. From this survey, we selected the 10 papers (listed in Table 7) with the most citations according to Google Scholar at the time of writing.

Unlike the applications discussed in Subsections 5.1 and 5.2, this field is centered around multiple datasets (see Table 7 for a summary) compiled for the classification of skin lesion images into two classes (malignant and non-malignant, ISIC2016 [31]) or multiple, more specific categories (ISIC2017 [57], see Figure 4 for an illustration). After carefully analyzing the selected papers, we found that [58], [59], and [60] are not suitable for consistency testing (refer to the 'conclusion' column of Table 7 for details). In the remaining papers, the authors provide sufficient details to apply the consistency tests. (Note that the 3-class problem ISIC2017 [57] was treated by the authors as two one-vs-all binary classification problems targeting the recognition of the classes *Melanoma* (M) and *Seborrheic Keratosis* (SK)). In most papers, multiple sets of performance scores are shared, illustrating the operation of certain algorithmic steps, with the most commonly reported scores being accuracy, sensitivity, and specificity. Given that the datasets either include a designated test set of images or the authors specify one and share its details, the consistency test developed in Section 3 was applied in each case. The number of reported score sets ($N_{sc.}$) and the number of inconsistent sets ($N_{inc.}$) are provided in the corresponding columns of Table 7. No inconsistencies were detected in [61] and [62]. In [63], [64], and [65], only a handful of score sets showed inconsistencies, likely due to typographical errors. However, the scores reported in papers [66] and [67] were inconsistent with the assumptions of the experiment, prompting a more detailed analysis of these papers.

Regarding [66], we tested multiple assumptions, such as the possibility that scores like M_{ACC} represent the accuracy of the *Melanoma* (M) class against the *Nevus* class instead of against both the *Nevus* and *Seborrheic Keratosis* classes. We also assumed that the accuracy and specificity figures might have been interchanged in the paper, as accuracy cannot be higher than both sensitivity and specificity simultaneously. However, all assumptions led to inconsistencies with the claimed number and distribution of test images. Therefore, we concluded that the reported scores are not comparable with those reported in other papers using the same dataset (such as [65]). Regarding [67], the authors mention that they report the performance scores of binary classification in a weighted manner, from the perspective of both classes treated as positive and using the number of samples in a certain positive class as weights. This uncommon weighting makes sensitivity equivalent to accuracy and turns specificity into a figure with no common interpretation. Although this reinterpretation resolves the inconsistencies, we still consider the scores inconsistent with the commonly accepted definitions of the terms.

The final conclusion of the analysis is that, in the field of skin lesion classification, there are highly influential papers (such as [66] and [67]) with inconsistent scores that are often cited for comparison and ranking in other research (e.g., [68]).



(a) *Nevus* (b) *Seborrheic Keratosis* (c) *Melanoma*

Figure 4: Entries of the ISIC 2017 [57] dataset: *Nevus* (4a); *Seborrheic Keratosis* (4b); *Melanoma* (4c).

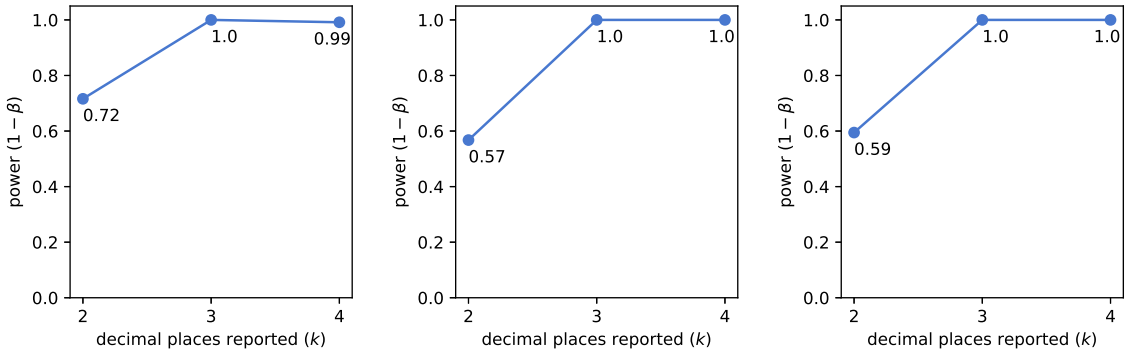
Table 7: The summary of consistency testing in skin lesion classification

ref.	cit.	dataset	digits	scores	$N_{sc.}$	$N_{inc.}$	conclusion
[63]	991	ISIC2016 [31]	3	acc, sens, spec	18	1	Potentially typos present.
[58]	603	ISIC2016 [31]	-	-	-	-	The paper is about image segmentation performance.
[64]	574	ISIC2016 [31]	3	acc, sens, spec	27	3	Potentially typos present.
[65]	389	ISIC2017 [57] m/sk	3	acc, sens, spec	32	2	Potentially typos present.
[59]	389	ISIC2016 [31] (custom selection)	-	-	-	-	Not enough details shared.
[61]	322	Argenziano's [69]	3	ppv, sens, spec	16	0	No inconsistency identified.
[60]	313	custom	-	-	-	-	Not enough details shared.
[66]	312	ISIC2017 [57] m/sk	3	acc, sens, spec	20	20	All accuracy, sensitivity and specificity scores reported for the two binary classification tasks M and SK are inconsistent.
[62]	259	custom	4	acc, sens, spec	18	0	No inconsistency identified.
[67]	238	ISIC2016 [31]	4	acc, sens, spec, f1	8	4	Unorthodox weighting of the scores.

Similar to Subsection 5.2, we carried out a power analysis to estimate how effective the proposed test is in recognizing inconsistencies in the skin lesion classification problems ISIC2016 and the two binary classification aspects of ISIC2017: recognizing the Melanoma class against Nevus and Seborrheic Keratosis, and recognizing Seborrheic Keratosis against Nevus and Melanoma. In this analysis, we are not aware of potential systematic methodological flaws (such as the improper use of minority oversampling in the case of preterm delivery prediction in Subsection 5.2). Therefore, we investigated the ability of the test to recognize even the slightest typographical error in one of the reported scores, specifically when the last digit of accuracy is altered by 1. For example, instead of the true accuracy 0.932, accidentally 0.933 is reported. The power analysis was carried out in a similar manner as described in Subsection 5.2, and the results are summarized in Figure 5. As the results suggest, when at least 3 decimal places are reported, the proposed test will recognize even the slightest typographical error as an inconsistency with a probability of 1, indicating its effectiveness in testing the consistency of reported scores in this field. It is worth highlighting that the scope of the power analysis does not limit the ability of the test to identify various other types of inconsistencies, such as adjustments to the evaluation set by changing the number of positive or negative samples (e.g., accidentally removing samples), or when performance scores are calculated using incorrect formulas. The power analyses for these types of inconsistencies can be carried out in a similar manner.

6. Conclusions

The meta-analysis of research is crucial to address the reproducibility crisis in artificial intelligence research and applications. Nevertheless, without numerical techniques, it demands enormous manual labor. To facilitate meta-analysis and enable the numerical identification of methodological flaws and inconsistencies, we have introduced various numerical tests to assess the consistency of reported performance scores and experimental setups in binary



(a) ISIC2016 (b) ISIC2017 Melanoma (c) ISIC2017 Seborrheic Keratosis

Figure 5: The results of the power analysis for the ISIC2016 dataset (a) and the two binary aspects of the ISIC2017 dataset, recognizing Melanoma (b) and Seborrheic Keratosis (c). The assumed deviation is a typographical error with the smallest possible effect: a 10^{-k} difference in the kth digit of the accuracy score, when the results are reported to k decimal places. The vertical axis represents the power of the test $(1-\beta)$ as a function of the number of decimal places reported. In these problems, when at least 3 digits are reported, the test recognizes the slightest deviation with a probability of 1. The slight decrease in power in the case of ISIC2016 at 4 decimal places is due to the fact that the size of the effect decreases with the increasing number of reported digits.

classification. The tests are based on the fact that whenever multiple performance scores are reported, their values are interrelated, and these interrelations can be verified by numerical techniques.

The proposed tests cover numerous evaluation scenarios. The test developed for performance scores derived from a single evaluation set supports 20 different scores (Section 3). Regarding scores obtained by aggregations, we showed that in the case of the *Score of Means* strategy, the testing falls back to the methods developed for scores derived from a single evaluation set (Subsection 4.2). For *Mean of Scores* aggregations, we developed a test supporting four of the most commonly reported scores (Subsection 4.3), and we also proposed the enumeration of all fold configurations when stratification is not used and the specifics of the folds are unknown (Subsection 4.4.3). Across Sections 3 and 4, we highlighted multiple opportunities to improve the efficiency or extend the coverage of the tests. Regarding the computational limitations, the test proposed for scores derived from a single confusion matrix is applicable to any reasonably sized dataset, while the tests developed for aggregated scores are limited by the capabilities of the integer linear programming solver being used.

In terms of applications, we briefly discussed the prior application of simplified forms of the proposed methods in the field of retinal vessel segmentation (Subsection 5.1) and explored potential further applications related to retinal image processing. We also demonstrated that the proposed techniques are suitable for replacing the manual labor required to verify the reported results by reimplementation in situations similar to preterm delivery prediction from EHG signals (Subsection 5.2). This application also illustrated that the proposed methods can be used to identify a common methodological pitfall when synthetic minority oversampling is employed. The power analysis of the method in this application reveals that the proposed test can recognize inconsistencies with a probability of 71%. Lastly, in Subsection 5.3, through a small meta-analysis employing the proposed techniques in the field of skin lesion classification, we uncovered the presence of irreproducible results systematically cited in the literature. The power analysis focusing on typographical errors showed that, in this problem, the proposed method is capable of recognizing the slightest typographical errors with 100% probability.

Given the reproducibility crisis in machine learning and artificial intelligence research, along with the vast number of scientific papers and the limited capacity for reimplementation, validation, and verification, the proposed tests – as illustrated by the applications – offer effective tools to assess and safeguard the integrity of research.

For the benefit of the community, a reference implementation of the proposed consistency tests has been released as an open-source Python package with an intuitive interface. The package is available in the standard Python repository (PyPI) under the name `mlscorecheck` [19] and on GitHub at the following URL: `https://github.com/`

FalseNegativeLab/mlscorecheck.

References

[1] S. Kapoor, A. Narayanan, Leakage and the reproducibility crisis in machine-learning-based science, PatternsPublisher: Elsevier (Aug. 2023). doi:10.1016/j.patter.2023.100804.
URL https://www.cell.com/patterns/abstract/S2666-3899(23)00159-9

[2] M. Hutson, Artificial intelligence faces reproducibility crisis, Science 359 (6377) (2018) 725–726. doi:10.1126/science.359.6377.725.
URL https://doi.org/10.1126/science.359.6377.725

[3] M. McDermott, S. Wang, N. Marinsek, R. Ranganath, L. Foschini, M. Ghassemi, Reproducibility in machine learning for health research: Still a ways to go, Science Translational Medicine 13 (2021) eabb1655. doi:10.1126/scitranslmed.abb1655.

[4] B. Haibe-Kains, G. A. Adam, A. Hosny, F. Khodakarami, T. Shraddha, R. Kusko, S.-A. Sansone, W. Tong, R. D. Wolfinger, C. E. Mason, W. Jones, J. Dopazo, C. Furlanello, L. Waldron, B. Wang, C. McIntosh, A. Goldenberg, A. Kundaje, C. S. Greene, T. Broderick, M. M. Hoffman, J. T. Leek, K. Korthauer, W. Huber, A. Brazma, J. Pineau, R. Tibshirani, T. Hastie, J. P. A. Ioannidis, J. Quackenbush, H. J. W. L. A. and, Transparency and reproducibility in artificial intelligence, Nature 586 (7829) (2020) E14–E16. doi:10.1038/s41586-020-2766-y.
URL https://doi.org/10.1038/s41586-020-2766-y

[5] D. Slutsky, Statistical errors in clinical studies, Journal of Wrist Surgery 02 (04) (2013) 285–287. doi:10.1055/s-0033-1359421.
URL https://doi.org/10.1055/s-0033-1359421

[6] D. Fanelli, How many scientists fabricate and falsify research? a systematic review and meta-analysis of survey data, PLOS ONE 4 (5) (2009) 1–11. doi:10.1371/journal.pone.0005738.
URL https://doi.org/10.1371/journal.pone.0005738

[7] B. J. Heil, M. M. Hoffman, F. Markowetz, S.-I. Lee, C. S. Greene, S. C. Hicks, Reproducibility standards for machine learning in the life sciences, Nature Methods 18 (10) (2021) 1132–1135. doi:10.1038/s41592-021-01256-7.
URL https://doi.org/10.1038/s41592-021-01256-7

[8] L. Maier-Hein, M. Eisenmann, A. Reinke, S. Onogur, M. Stankovic, P. Scholz, T. Arbel, H. Bogunovic, A. P. Bradley, A. Carass, C. Feldmann, A. F. Frangi, P. M. Full, B. van Ginneken, A. Hanbury, K. Honauer, M. Kozubek, B. A. Landman, K. März, O. Maier, K. Maier-Hein, B. H. Menze, H. Müller, P. F. Neher, W. Niessen, N. Rajpoot, G. C. Sharp, K. Sirinukunwattana, S. Speidel, C. Stock, D. Stoyanov, A. A. Taha, F. van der Sommen, C.-W. Wang, M.-A. Weber, G. Zheng, P. Jannin, A. Kopp-Schneider, Why rankings of biomedical image analysis competitions should be interpreted with care, Nature Communications 9 (1) (Dec. 2018). doi:10.1038/s41467-018-07619-7.
URL https://doi.org/10.1038/s41467-018-07619-7

[9] S. B. Nissen, T. Magidson, K. Gross, C. T. Bergstrom, Publication bias and the canonization of false facts, eLife 5 (Dec. 2016). doi:10.7554/elife.21451.
URL https://doi.org/10.7554/elife.21451

[10] R. A. Poldrack, G. Huckins, G. Varoquaux, Establishment of best practices for evidence for prediction, JAMA Psychiatry 77 (5) (2020) 534. doi:10.1001/jamapsychiatry.2019.3671.
URL https://doi.org/10.1001/jamapsychiatry.2019.3671

[11] D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, K. Rieck, Dos and don'ts of machine learning in computer security, in: 31st USENIX Security Symposium (USENIX Security 22), USENIX Association, Boston, MA, 2022, pp. 3971–3988.
URL https://www.usenix.org/conference/usenixsecurity22/presentation/arp

[12] J. Nalepa, M. Myller, M. Kawulok, Validating hyperspectral image segmentation, IEEE Geoscience and Remote Sensing Letters 16 (8) (2019) 1264–1268. doi:10.1109/lgrs.2019.2895697.
URL https://doi.org/10.1109/lgrs.2019.2895697

[13] S. Shalev-Shwartz, S. Ben-David, Understanding Machine Learning, Cambridge University Press, 2014. doi:10.1017/cbo9781107298019.
URL https://doi.org/10.1017/cbo9781107298019

[14] S. Bates, T. Hastie, R. Tibshirani, Cross-validation: What does it estimate and how well does it do it?, Journal of the American Statistical Association (2023) 1–12doi:10.1080/01621459.2023.2197686.
URL https://doi.org/10.1080/01621459.2023.2197686

[15] A. Tharwat, Classification assessment methods, Applied Computing and Informatics 17 (1) (2020) 168–192. doi:10.1016/j.aci.2018.08.003.
URL https://doi.org/10.1016/j.aci.2018.08.003

[16] D. Bowes, T. Hall, D. Gray, DConfusion: a technique to allow cross study performance evaluation of fault prediction studies, Automated Software Engineering 21 (2) (2013) 287–313. doi:10.1007/s10515-013-0129-8.
URL https://doi.org/10.1007/s10515-013-0129-8

[17] G. Kovács, A. Hajdu, A self-calibrating approach for the segmentation of retinal vessels by template matching and contour reconstruction, Medical Image Analysis 29(4) (2016) 24–46. doi:10.1016/j.media.2015.12.003.

[18] G. Kovács, A. Fazekas, A new baseline for retinal vessel segmentation: Numerical identification and correction of methodological inconsistencies affecting 100+ papers, Medical Image Analysis 75 (2022) 102300. doi:https://doi.org/10.1016/j.media.2021.102300.

[19] G. Kovács, A. Fazekas, mlscorecheck: Testing the consistency of reported performance scores and experiments in machine learning, Neurocomputing 583 (2024) 127556. doi:https://doi.org/10.1016/j.neucom.2024.127556.
URL https://www.sciencedirect.com/science/article/pii/S0925231224003278

[20] L. Devroye, L. Györfi, G. Lugosi, A Probabilistic Theory of Pattern Recognition, Springer New York, 1996. doi:10.1007/978-1-4612-0711-5.
URL https://doi.org/10.1007/978-1-4612-0711-5

[21] Y. Wang, U. Ahsan, H. Li, M. Hagen, A comprehensive review of modern object segmentation approaches, Foundations and Trends® in Computer Graphics and Vision 13 (2-3) (2022) 111–283. doi:10.1561/0600000097.
URL https://doi.org/10.1561/0600000097
[22] M. E. Shipe, S. A. Deppen, F. Farjah, E. L. Grogan, Developing prediction models for clinical use using logistic regression: an overview, Journal of Thoracic Disease 11 (S4) (2019) S574–S584. doi:10.21037/jtd.2019.01.25.
URL https://doi.org/10.21037/jtd.2019.01.25
[23] T. Yang, Y. Ying, Auc maximization in the era of big data and ai: A survey, ACM Comput. Surv. 55 (8) (dec 2022). doi:10.1145/3554729.
URL https://doi.org/10.1145/3554729
[24] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer New York, 2009. doi:10.1007/978-0-387-84858-7.
URL https://doi.org/10.1007/978-0-387-84858-7
[25] J. A. Gallis, E. L. Turner, Relative measures of association for binary outcomes: Challenges and recommendations for the global health researcher, Annals of Global Health 85 (1) (2019). doi:10.5334/aogh.2581.
URL https://doi.org/10.5334/aogh.2581
[26] A. R. Redondo, J. Navarro, R. R. Fernández, I. M. de Diego, J. M. Moguerza, J. J. Fernández-Muñoz, Unified performance measure for binary classification problems, in: Lecture Notes in Computer Science, Springer International Publishing, 2020, pp. 104–112. doi:10.1007/978-3-030-62365-4_10.
URL https://doi.org/10.1007/978-3-030-62365-4_10
[27] E. B. Fowlkes, C. L. Mallows, A method for comparing two hierarchical clusterings, Journal of the American Statistical Association 78 (383) (1983) 553–569. doi:10.1080/01621459.1983.10478008.
URL https://doi.org/10.1080/01621459.1983.10478008
[28] D. Powers, Evaluation: From precision, recall and f-factor to roc, informedness, markedness & correlation, Mach. Learn. Technol. 2 (01 2008).
[29] J. Balayla, Prevalence threshold (phi(e)) and the geometry of screening curves, PLoS One 15 (10) (2020) e0240215.
[30] M. L. McHugh, Interrater reliability: the kappa statistic, Biochemia Medica (2012) 276–282doi:10.11613/bm.2012.031.
URL https://doi.org/10.11613/bm.2012.031
[31] D. Gutman, N. Codella, M. E. Celebi, B. Helba, M. Marchetti, N. K. Mishra, A. Halpern, Skin lesion analysis toward melanoma detection: A challenge at the international symposium on biomedical imaging (isbi) 2016, hosted by the international skin imaging collaboration (isic), arXiv:1605.01397 [cs.CV] (05 2016).
[32] R. E. Moore, Introduction to interval analysis, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2009.
[33] B. Everitt, The Cambridge dictionary of statistics, Cambridge University Press, Cambridge, UK; New York, 2002.
URL http://www.worldcat.org/search?qt=worldcat_org_all&q=052181099X
[34] A. Piryatinska, B. Darkhovsky, A. Kaplan, Binary classification of multichannel-EEG records based on the ϵ-complexity of continuous vector functions, Computer Methods and Programs in Biomedicine 152 (2017) 131–139. doi:10.1016/j.cmpb.2017.09.001.
URL https://doi.org/10.1016/j.cmpb.2017.09.001
[35] A. Occhipinti, L. Rogers, C. Angione, A pipeline and comparative study of 12 machine learning models for text classification, Expert Systems with Applications 201 (2022) 117193. doi:10.1016/j.eswa.2022.117193.
URL https://doi.org/10.1016/j.eswa.2022.117193
[36] R. Kubinski, J.-Y. Djamen-Kepaou, T. Zhanabaev, A. Hernandez-Garcia, S. Bauer, F. Hildebrand, T. Korcsmaros, S. Karam, P. Jantchou, K. Kafi, R. D. Martin, Benchmark of data processing methods and machine learning models for gut microbiome-based diagnosis of inflammatory bowel disease, Frontiers in Genetics 13 (Feb. 2022). doi:10.3389/fgene.2022.784397.
URL https://doi.org/10.3389/fgene.2022.784397
[37] T. Rao, Mean of ratios or ratio of means or both?, Journal of Statistical Planning and Inference 102 (1) (2002) 129–138. doi:https://doi.org/10.1016/S0378-3758(01)00181-1.
[38] C. Salas, T. G. Gregoire, Statistical analysis of ratio estimators and their estimators of variances when the auxiliary variate is measured with error, European Journal of Forest Research 129 (5) (2009) 847–861. doi:10.1007/s10342-009-0277-3.
URL https://doi.org/10.1007/s10342-009-0277-3
[39] T. Lattimore, M. Hutter, No free lunch versus occam's razor in supervised learning, in: Algorithmic Probability and Friends. Bayesian Prediction and Artificial Intelligence, Springer Berlin Heidelberg, 2013, pp. 223–235. doi:10.1007/978-3-642-44958-1_17.
URL https://doi.org/10.1007/978-3-642-44958-1_17
[40] B. C. Levy, Principles of Signal Detection and Parameter Estimation, 1st Edition, Springer, 2008.
[41] M. Conforti, G. Cornuéjols, G. Zambelli, Integer Programming, Springer International Publishing, 2014. doi:10.1007/978-3-319-11008-0.
URL https://doi.org/10.1007/978-3-319-11008-0
[42] D. Bertsekas, Nonlinear Programming, Athena scientific optimization and computation series, Athena Scientific, 2016.
URL https://books.google.hu/books?id=rC1EEAAAQBAJ
[43] J. P. A. Pereira, S. Kroon, D. Vandenbussche, Pulp, https://github.com/coin-or/pulp (2022).
[44] J. Alcala-Fdez, A. Fernández, J. Luengo, J. Derrac, S. Garc'ia, L. Sanchez, F. Herrera, Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework, Journal of Multiple-Valued Logic and Soft Computing 17 (2010) 255–287.
[45] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, Journal of machine learning research 12 (Oct) (2011) 2825–2830.
[46] J. Arndt, Matters Computational, Springer Berlin Heidelberg, 2011. doi:10.1007/978-3-642-14764-7.
URL https://doi.org/10.1007/978-3-642-14764-7
[47] J. Staal, M. D. Abr 'amoff, M. Niemeijer, M. A. Viergever, B. Van Ginneken, Ridge-based vessel segmentation in color images of the retina, IEEE transactions on medical imaging 23 (4) (2004) 501–509.
[48] S. Joshi, P. Karule, A review on exudates detection methods for diabetic retinopathy, Biomedicine & Pharmacotherapy 97 (2018) 1454–1460.

doi:10.1016/j.biopha.2017.11.009.
URL https://doi.org/10.1016/j.biopha.2017.11.009

[49] M. Alawad, A. Aljouie, S. Alamri, M. Alghamdi, B. Alabdulkader, N. Alkanhal, A. Almazroa, Machine learning and deep learning techniques for optic disc and cup segmentation – a review, Clinical Ophthalmology Volume 16 (2022) 747–764. doi:10.2147/opth.s348479.
URL https://doi.org/10.2147/opth.s348479

[50] J. Garcia-Casado, Y. Ye-Lin, G. Prats-Boluda, J. Mas-Cabo, J. Alberola-Rubio, A. Perales, Electrohysterography in the diagnosis of preterm birth: a review, Physiological Measurement 39 (2) (2018) 02TR01. doi:10.1088/1361-6579/aaad56.
URL https://doi.org/10.1088/1361-6579/aaad56

[51] G. Fele-Žorž, G. Kavšek, Ž. Novak-Antolič, F. Jager, A comparison of various linear and non-linear signal processing techniques to separate uterine EMG records of term and pre-term delivery groups, Medical & Biological Engineering & Computing 46 (9) (2008) 911–922. doi:10.1007/s11517-008-0350-y.
URL https://doi.org/10.1007/s11517-008-0350-y

[52] G. Vandewiele, I. Dehaene, G. Kovács, L. Sterckx, O. Janssens, F. Ongenae, F. D. Backere, F. D. Turck, K. Roelens, J. Decruyenaere, S. V. Hoecke, T. Demeester, Overly optimistic prediction results on imbalanced data: a case study of flaws and benefits when applying over-sampling, Artificial Intelligence in Medicine 111 (2021) 101987. doi:10.1016/j.artmed.2020.101987.

[53] A. Fernández, S. Garcia, F. Herrera, N. Chawla, Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary, Journal of Artificial Intelligence Research 61 (2018) 863–905. doi:10.1613/jair.1.11192.

[54] S. Rezvani, X. Wang, A broad review on class imbalance learning techniques, Applied Soft Computing 143 (2023) 110415. doi:https://doi.org/10.1016/j.asoc.2023.110415.
URL https://www.sciencedirect.com/science/article/pii/S1568494623004337

[55] U. R. Acharya, V. K. Sudarshan, S. Q. Rong, Z. Tan, C. M. Lim, J. E. Koh, S. Nayak, S. V. Bhandary, Automated detection of premature delivery using empirical mode and wavelet packet decomposition techniques with uterine electromyogram signals, Computers in Biology and Medicine 85 (2017) 33–42. doi:10.1016/j.compbiomed.2017.04.013.
URL https://doi.org/10.1016/j.compbiomed.2017.04.013

[56] M. A. Kassem, K. M. Hosny, R. Damaševičius, M. M. Eltoukhy, Machine learning and deep learning methods for skin lesion classification and diagnosis: A systematic review, Diagnostics 11 (8) (2021) 1390. doi:10.3390/diagnostics11081390.
URL https://doi.org/10.3390/diagnostics11081390

[57] N. C. F. Codella, D. Gutman, M. E. Celebi, B. Helba, M. A. Marchetti, S. W. Dusza, A. Kalloo, K. Liopyris, N. Mishra, H. Kittler, A. Halpern, Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (ISBI), hosted by the international skin imaging collaboration (ISIC), in: 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), IEEE, 2018, pp. 168–172. doi:10.1109/isbi.2018.8363547.
URL https://doi.org/10.1109/isbi.2018.8363547

[58] Y. Yuan, M. Chao, Y.-C. Lo, Automatic skin lesion segmentation using deep fully convolutional networks with jaccard distance, IEEE Transactions on Medical Imaging 36 (9) (2017) 1876–1886. doi:10.1109/tmi.2017.2695227.
URL https://doi.org/10.1109/tmi.2017.2695227

[59] T. J. Brinker, A. Hekler, A. H. Enk, J. Klode, A. Hauschild, C. Berking, B. Schilling, S. Haferkamp, D. Schadendorf, T. Holland-Letz, J. S. Utikal, C. von Kalle, W. Ludwig-Peitsch, J. Sirokay, L. Heinzerling, M. Albrecht, K. Baratella, L. Bischof, E. Chorti, A. Dith, C. Drusio, N. Giese, E. Gratsias, K. Griewank, S. Hallasch, Z. Hanhart, S. Herz, K. Hohaus, P. Jansen, F. Jockenhöfer, T. Kanaki, S. Knispel, K. Leonhard, A. Martaki, L. Matei, J. Matull, A. Olischewski, M. Petri, J.-M. Placke, S. Raub, K. Salva, S. Schlott, E. Sody, N. Steingrube, I. Stoffels, S. Ugurel, A. Zaremba, C. Gebhardt, N. Booken, M. Christolouka, K. Buder-Bakhaya, T. Bokor-Billmann, A. Enk, P. Gholam, H. Hänßle, M. Salzmann, S. Schäfer, K. Schäkel, T. Schank, A.-S. Bohne, S. Deffaa, K. Drerup, F. Egberts, A.-S. Erkens, B. Ewald, S. Falkvoll, S. Gerdes, V. Harde, A. Hauschild, M. Jost, K. Kosova, L. Messinger, M. Metzner, K. Morrison, R. Motamedi, A. Pinczker, A. Rosenthal, N. Scheller, T. Schwarz, D. Stölzl, F. Thielking, E. Tomaschewski, U. Wehkamp, M. Weichenthal, O. Wiedow, C. M. Bär, S. Bender-Säbelkampf, M. Horbrügger, A. Karoglan, L. Kraas, J. Faulhaber, C. Geraud, Z. Guo, P. Koch, M. Linke, N. Maurier, V. Müller, B. Thomas, J. S. Utikal, A. S. M. Alamri, A. Baczako, C. Berking, M. Betke, C. Haas, D. Hartmann, M. V. Heppt, K. Kilian, S. Krammer, N. L. Lapczynski, S. Mastnik, S. Nasifoglu, C. Ruini, E. Sattler, M. Schlaak, H. Wolff, B. Achatz, A. Bergbreiter, K. Drexler, M. Ettinger, S. Haferkamp, A. Halupczok, M. Hegemann, V. Dinauer, M. Maagk, M. Mickler, B. Philipp, A. Wilm, C. Wittmann, A. Gesierich, V. Glutsch, K. Kahlert, A. Kerstan, B. Schilling, P. Schrüfer, Deep learning outperformed 136 of 157 dermatologists in a head-to-head dermoscopic melanoma image classification task, European Journal of Cancer 113 (2019) 47–54. doi:10.1016/j.ejca.2019.04.001.
URL https://doi.org/10.1016/j.ejca.2019.04.001

[60] U.-O. Dorj, K.-K. Lee, J.-Y. Choi, M. Lee, The skin cancer classification using deep convolutional neural network, Multimedia Tools and Applications 77 (8) (2018) 9909–9924. doi:10.1007/s11042-018-5714-1.
URL https://doi.org/10.1007/s11042-018-5714-1

[61] J. Kawahara, S. Daneshvar, G. Argenziano, G. Hamarneh, Seven-point checklist and skin lesion classification using multitask multimodal neural nets, IEEE Journal of Biomedical and Health Informatics 23 (2) (2019) 538–546. doi:10.1109/jbhi.2018.2824327.
URL https://doi.org/10.1109/jbhi.2018.2824327

[62] F. Xie, H. Fan, Y. Li, Z. Jiang, R. Meng, A. Bovik, Melanoma classification on dermoscopy images using a neural network ensemble model, IEEE Transactions on Medical Imaging 36 (3) (2017) 849–858. doi:10.1109/tmi.2016.2633551.
URL https://doi.org/10.1109/tmi.2016.2633551

[63] L. Yu, H. Chen, Q. Dou, J. Qin, P.-A. Heng, Automated melanoma recognition in dermoscopy images via very deep residual networks, IEEE Transactions on Medical Imaging 36 (4) (2017) 994–1004. doi:10.1109/tmi.2016.2642839.
URL https://doi.org/10.1109/tmi.2016.2642839

[64] N. C. F. Codella, Q.-B. Nguyen, S. Pankanti, D. A. Gutman, B. Helba, A. C. Halpern, J. R. Smith, Deep learning ensembles for melanoma recognition in dermoscopy images, IBM Journal of Research and Development 61 (4/5) (2017) 5:1–5:15. doi:10.1147/jrd.2017.2708299.

URL https://doi.org/10.1147/jrd.2017.2708299
[65] J. Zhang, Y. Xie, Y. Xia, C. Shen, Attention residual learning for skin lesion classification, IEEE Transactions on Medical Imaging 38 (9) (2019) 2092–2103. doi:10.1109/tmi.2019.2893944.
URL https://doi.org/10.1109/tmi.2019.2893944
[66] B. Harangi, Skin lesion classification with ensembles of deep convolutional neural networks, Journal of Biomedical Informatics 86 (2018) 25–32. doi:10.1016/j.jbi.2018.08.006.
URL https://doi.org/10.1016/j.jbi.2018.08.006
[67] M. A. Al-masni, D.-H. Kim, T.-S. Kim, Multiple skin lesions diagnostics via integrated deep convolutional networks for segmentation and classification, Computer Methods and Programs in Biomedicine 190 (2020) 105351. doi:10.1016/j.cmpb.2020.105351.
URL https://doi.org/10.1016/j.cmpb.2020.105351
[68] M. F. J. Acosta, L. Y. C. Tovar, M. B. Garcia-Zapirain, W. S. Percybrooks, Melanoma diagnosis using deep learning techniques on dermatoscopic images, BMC Medical Imaging 21 (1) (Jan. 2021). doi:10.1186/s12880-020-00534-8.
URL https://doi.org/10.1186/s12880-020-00534-8
[69] G. Argenziano, P. Soyer, V. De Giorgi, D. Piccolo, P. Carli, M. Delfino, A. Ferrari, R. Hofmann-Wellenhof, D. Massi, G. Mazzocchetti, M. Scalvenzi, I. Wolf, Interactive atlas of dermoscopy. Dermoscopy: a tutorial (Book) and CD-ROM., Edra, 2000.