*t*26, *Debrecen*, 4028, *Hungary*
[b]Analytical Minds Ltd.,
ΩΩimmediateÁrp
ΩΩimmediateád
ΩΩimmediateút 5, Beregsur
ΩΩimmediateány, 4933, Hungary

Abstract

Binary classification is a fundamental task in machine learning, with applications spanning various scientific domains. Whether scientists are conducting fundamental research or refining practical applications, they typically assess and rank classification techniques based on performance metrics such as *accuracy*, *sensitivity*, and *specificity*. However, reported performance scores may not always serve as a reliable basis for research ranking. This can be attributed to undisclosed or unconventional practices related to cross-validation, typographical errors, and other factors. In a given experimental setup, with a specific number of positive and negative test items, most performance scores can assume specific, interrelated values. In this paper, we introduce numerical techniques to assess the consistency of reported performance scores and the assumed experimental setup. Importantly, the proposed approach does not rely on statistical inference but uses numerical methods to identify inconsistencies with certainty. Through three different applications related to medicine, we demonstrate how the proposed techniques can effectively detect inconsistencies, thereby safeguarding the integrity of research fields. To benefit the scientific community, we have made the consistency tests available in an open-source Python package.

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

1. Introduction

Recently, various authors have warned the scientific community about the so-called *reproducibility crisis* in artificial intelligence and machine learning-based research [1, 2, 3, 4]. The crisis is characterized by a significant body of research results that prove challenging to replicate through independent experiments or are built upon flawed evaluation methodologies. Among several reasons, notable ones include the absence of shared code [1], improper use of statistics [1, 5], 'cosmetic' adjustments of findings [6] and the presence of typographical errors in reported figures.

To address the crisis, various recommendations have been proposed [3, 7] regarding the proper reporting of machine learning research results. Unfortunately, the adoption of these standards has been relatively slow, and they are unable to rectify existing issues in numerous fields. In practical terms, many domains within machine learning research operate as self-organized, ongoing competitions, with the primary goal of achieving the best outcomes for specific problems and datasets. Within these domains, research quality is often assessed solely based on reported performance scores, despite the proven difficulty of this task even in under controlled circumstances [8]. In such circumstances, reliability of reported performance scores becomes crucial. Once published, unrealistically high performance results can inadvertently validate flawed methodologies and may be amplified by publication bias [9], ultimately distorting entire research fields. For example, consider the reports of nearly perfect predictive results for premature delivery in pregnancy based on electrohysterograms (EHG) by multiple authors [10]. Subsequently, in [11], it was discovered that the overly optimistic results in 11 papers were due to a systematic data leakage in the evaluation methodology.

In general, most meta-analyses [12] related to reproducibility necessitate manual effort. One approach involves a thorough examination of the relevant papers, seeking deviations from the scientific method and best practices of statistics, as demonstrated in studies such as [13], [14], and [15]. Another approach entails attempting to replicate various results by re-implementing the techniques, as exemplified by the authors of the EHG report [11], which demands even more manual labor and becomes impractical at a larger scale. To facilitate meta-analysis, in this paper, we introduce numerical techniques to test the consistency of reported performance scores with the described experimental setups in binary classification.

Being one of the fundamental tasks in machine learning, binary classification [16] also suffers from the aforementioned problems resulting in the reporting of performance scores that are often incomparable and irreproducible. Whether in fundamental research or practical applications, binary classification performance is typically assessed by making predictions on a test set (sometimes in a cross-validation scheme [17]) and constructing the confusion matrix [18] tabulating the counts of correctly and incorrectly classified positive and negative instances. In practice, confusion matrices are rarely presented directly. Instead, these matrices are usually condensed into multiple numerical metrics (e.g., accuracy and f1-score [18]), often aggregated across multiple dataset folds or even datasets. The multitude of reported scores quantifies various aspects of performance.

Given the natural constraints imposed by the experimental setup on the confusion matrix (e.g., the sum of all entries must match the cardinality of the test set) and considering that the reported performance scores are interrelated (they cannot take arbitrary values independently), the question arises: *Can the reported performance scores be the result of the experiment?* Mathematically, this question relates to the existence of at least one compatible confusion matrix that aligns with the experimental conditions and yields the reported performance scores. Additional complexity is introduced by rounding and potential aggregations. If the problem proves to be infeasible, any attempts to reproduce the results are destined to fail with certainty.

Earlier, a technique called *DConfusion* was proposed in [19] to reconstruct certain characteristics (but not the exact entries) of the confusion matrix from reported scores. *DConfusion* was then effectively applied to various studies involving binary classification [20] to assess the validity of reported scores. Independently, in a previous paper of ours, we introduced a similar approach to estimate the exact number of pixels used for evaluating retinal vessel seg-

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

mentation techniques [21]. Later on, refining the method, we developed a test that proved sensitive enough to identify a systematic and significant methodological inconsistency in the evaluation of retinal vessel segmentation: an analysis involving more than 100 papers led to the conclusion that the rankings of algorithms in the majority of studies are based on figures that cannot be compared directly [22], despite the authors using the same test set of images for evaluation. Drawing inspiration from the successful application of the concept, this paper generalizes the method further and develops consistency tests for many of the most commonly used performance scores and evaluation schemes in binary classification. The proposed approach differs from *DConfusion* [19] in several key ways: (a) while *DConfusion* supports only a limited set of performance scores typically encountered in the field of software fault prediction systems, the proposed technique supports the majority of performance scores used in the literature; (b) *DConfusion* neglects the impact of aggregations, whereas the proposed method addresses the aggregation of scores with mathematical rigor; (c) due to the omission of aggregations and the propagation of rounding errors, *DConfusion* might produce false alarms of inconsistency, whereas the inconsistencies identified by the proposed technique are certain.

We emphasize that the proposed consistency tests are numerical rather than statistical, devoid of any probability of type-I errors (false positives). In other words, inconsistencies are conclusively identified, implying that either the assumed experimental setup or the reported scores are incorrect. Given their robustness and ease of use, we believe that the proposed techniques can serve as effective tools for meta-analysis and contribute to enhancing the reproducibility of machine learning-based science.

The contributions of the paper to the field can be summarized as follows:

1. For performance scores calculated directly from confusion matrices, we introduce a consistency test that supports 20 of the most commonly used performance scores (without synonyms and complements). We highlight that the test can be easily extended to accommodate further score functions.

2. For experimental setups that involve the averaging of performance scores across dataset folds or multiple datasets, we propose a consistency test that supports four widely used scores: accuracy, sensitivity, specificity, and balanced accuracy.

3. We illustrate the practical application of the proposed techniques in three real-world problems, showcasing its effectiveness in identifying research with flawed evaluation methodologies and unreliable performance scores.

4. To benefit the research community, we have released the proposed tests as part of the open-source Python package `mlscorecheck`. This package is available in the standard PyPI repository or on GitHub at the: `https://github.com/FalseNegativeLab/mlscorecheck`.

The paper is organized as follows: Section 2 introduces the notation, the concept of binary classification, the performance scores and also outlines the problem of consistency testing. Sections 3 and 4 present the proposed techniques for various evaluation schemes and scenarios and Section 5 demonstrates how the techniques can be applied in practice. Finally, conclusions are drawn in Section 6.

2. Problem formulation

In this section, we formulate the problem we address and introduce the notations and concepts 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 vectors* ($\mathbf{x}_i \in \mathbb{R}^d$) and *class labels* ($y_i \in \{0, 1\}$), the classes labelled as 0 and 1 commonly referred 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 can predict class labels, but many of them (such as decision trees [16], neural networks [16], etc.) are designed to approximate the posterior distributions $\mathbb{P}(\mathbf{x}|y = c)$, $c \in \{0, 1\}$ and derive class labels by maximizing the posterior probability, thereby reducing the probability of misclassification [23]. The choice of which classifier outcome to favor depends on the specific application. For instance, image segmentation [24] requires hard labels indicating whether a pixel belongs to an object, whereas many medical applications prioritize ranking cases by the probability (risk) of a condition [25]. Consequently, performance measurement varies by the field of application, with two predominant approaches: one quantifies how effectively the posterior probabilities rank items, typically using the AUC score [26, 27]; the other assesses the quality of label assignment [18]. 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 them 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 **??**) 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 $n =$

Cohen's kappa [33]	kappa	$\frac{-2np + 2n \cdot tp + 2p \cdot tn}{n^2 - n \cdot tn + n \cdot tp + p^2 + p \cdot tn - p \cdot tp}$	label ing i by ch

$tn + fp$. 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 scores*. (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 a multitude of such 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 [28]). 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 $\mathcal{S} \subseteq \{acc, sens, spec, \dots\}$ (any of the score abbreviations in Table 2) of scores reported. 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.

Œ

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 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 [17]. In a kFCV scheme, the dataset $\mathcal{D}$ is randomly divided into k disjoint subsets of equal sizes, 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, the 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 many cases, classifiers are evaluated on and the results aggregated over multiple datasets.

Since many performance scores are fractions 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 extends 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?*

Œ

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 [22, 34]); 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.

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

3.1. Testing by exhaustive search

The experimental setup (described by p and n) and the reported scores are consistent if there exist some $tp^* \in \mathcal{P} \doteq \{0, \dots, p\}$ and $tn^* \in \mathcal{N} \doteq \{0, \dots, n\}$ 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 $(tp, tn) \in \mathcal{P} \times \mathcal{N}$ 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 iteration through the potential values of both tp and tn could be eliminated 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 **??**.

If we knew the exact values of the scores v_s^*, one would need to check if there exist 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 arithmetics [35]. 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 arithmetics yields

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

that is, the only integer tn can take to result 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 the the union of intervals (Table **??**).

Table 3: Scores with single solutions.

score (s)	tp formula ($f_{s,tp}^{-1}(s, tn, p, n)$)	tn formula ($f_{s,tn}^{-1}(s, tp, p, n)$)
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	$\frac{f_+^\beta\,(\beta_+^2 p + n - tn)}{\beta_+^2 - f_+^\beta + 1}$	$\frac{-\beta_+^2 tp + f_+^\beta\,(\beta_+^2 p + n + tp) - tp}{f_+^\beta}$
fbn	$\frac{-\beta_-^2 tn + f_-^\beta\,(\beta_-^2 n + p + tn) - tn}{f_-^\beta}$	$\frac{f_-^\beta\,(\beta_-^2 n + p - tp)}{\beta_-^2 - f_-^\beta + 1}$
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	$\frac{\kappa\,(n^2 - n \cdot tn + \ldots) + 2np - 2p \cdot tn}{\kappa\,(-n + p) + 2n}$ [illegible]	$\frac{\kappa\,(\ldots tp + p^2 - p \cdot tp) + 2np - 2n \cdot tp}{\kappa\,(n - p) + 2p}$ [illegible]

Ultimately, the consistency test can be carried out by iteratively testing if there exist $tp \in \mathcal{P}$ 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 in 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 by $tn \in \mathcal{N}$ and using solutions for tp (also given in table 3 and **??**). 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. The pseudo-code of the test is listed in Algorithm **??**.

We note that the sensitivity of the test (ability to recognize inconsistencies) is dependent on the specifics of the experiment, and the number and precision of reported scores, however as we demonstrate in Section 5, it can be applied successfully in many typical scenarios.

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.

Œ

Œ

Œ

3.3. Example

Suppose, there is an evaluation set of $p = 1000$ positive and $n = 6000$ negative samples, and the reported score $\hat{v}_{acc} = 0.6801$, $\hat{v}_{npv} = 0.9401$ 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 **??**, one finds that there are two pairs of (tp, tn) values, compatible with the setup: (743, 4031); (743, 4032). If the scores were adjusted a little, for example, accuracy changed to 0.6811, there are no (tp, tn) pairs fulfilling all conditions. Similarly, if the assumption on p was different, for example, $p = 1100$, the scores turn to be inconsistent.

4. Testing scores derived by aggregations

In this section, we develop tests for those scenarios when the scores are aggregated over multiple evaluation sets (folds and/or datasets). The mode of aggregation (discussed in Section 4.1) leads to different tests that we cover in Sections 4.2 and 4.3. Finally, the mapping of some common kFCV schemes to the representation used by the tests is discussed in Section 4.4.

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, $i = 1, \ldots, N_e$ positive and negative samples, respectively, 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 describing the entire experiment.

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

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

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 [36] and the use of hypothesis testing for the comparison of classification techniques [37].

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

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

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

[38] (for example, if a fold has only a handful of positive samples, $tp = 0$ and $fp = 0$ leads 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 [39, 40], but generalized to accommodate the non-linearities in the numerators and denominators of some scores (like 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 [41].) Nevertheless, estimation theory [42] 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 [40]. 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 [40]. 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 same data, one can find examples of both aggregations [22], 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 rigour. Therefore, in Sections 4.2 and 4.3 we develop consistency tests for the two types of aggregations separately.

Œ

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 (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,

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

Therefore, any score calculated using the SoM approach can be treated as if it was 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 Section 4.3.1, then propose a tractable algorithm based on linear integer programming in Section 4.3.2, finally, the use of the technique is illustrated in Section 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:

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

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 Section 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 leads to approximately 10^{10} different combinations of the free parameters.

4.3.2. Feasibility by linear integer 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 [43], with no efficient algorithms for exact solutions and approximations are not suitable for sharp consistency tests requiring exact decisions regarding feasibility.

On the other hand, for that subset of scores which leads to linear conditions, linear integer programming can be exploited, which is solvable by numerous techniques [43]. 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 linear 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 linear integer 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.

The sensitivity of the test (the ability to recognize inconsistencies) highly depends on the structure of the evaluation sets and the numerical uncertainty of the reported scores. However, as we illustrate in Section 5, it is able recognize inconsistencies in real scenarios. We also mention that there is one more piece of information that is sometimes reported and can strenghten the test: the minimum and maximum scores across folds. As noted in Section 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 [44]. However, the exploration of these special cases is beyond the scope of the paper.

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 being $\epsilon = 0.0001$ (allowing ceiling or flooring), substituting the scores and the specifications of the folds from Table 5a into (9), and checking the feasibility by a linear integer programming solver (we used the Python package `pulp` [45]), the solver returns that the problem is feasible, indicating that the scores could be the outcome of 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 preceding sections, we introduced consistency tests for two modes of aggregations in terms of N_e evaluation sets. In this section, we discuss how various kFCV experiments can be assessed using these tests. In a kFCV experiment with k folds on a dataset comprising p positive and n negative entries, we define *fold configuration* as the distribution of positives and negatives across the k folds, denoted as (p_i, n_i), $i = 1,\ldots,k$. We assume $p_i + n_i$ is either $\lfloor (p+n)/k \rfloor$ or $\lfloor (p+n)/k \rfloor + 1$ and there are at least two folds containing at least one negative and two folds containing at least one positive sample – this is a necessary requirement to ensure all training sets in the folding process contain samples of both classes and at least the accuracy scores is computable on each fold. We treat fold configurations as multisets, a certain pair of positive and negative counts can appear multiple times, but the order of the pairs is irrelevant as the 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 Section 4.2, only the overall counts of positive and negative samples are needed for the testing. Therefore, in an ordinary kFCV experiment (evaluating each entry of a dataset once), the parameters of the dataset need to be used. In a repeated kFCV scenario, involving N_d datasets with N_r repetitions, $p' = N_r \cdot \sum_{i=1}^{N_d} p_i$ and $n' = N_r \cdot \sum_{i=1}^{N_d} n_i$ need to be used, the fold configurations are irrelevant.

4.4.2. Testing the MoS scores of kFCV experiments

In the case of MoS aggregations, the fold configuration needs to be known to formulate the linear programming problem (9). Although there are some data providers that also supply foldings of the data (for example, the KEEL data repository [46]), in a general kFCV scenario, the fold configuration is unknown. There are however special cases, when the fold configuration can be inferred. If kFCV is carried out in a stratified manner, the stratification technique can be used to infer the configuration of folds. For example, the stratification implemented in the `sklearn` [47] Python package ensures that folds differ at most in 1 sample regarding the overall count of items, as well as the counts of positives and negatives. This requirement leads to a unique configuration that can be inferred from p, n and k which is shared in Table **??**. In the case of repeated kFCV experiments or the use of multiple datasets, the union of all fold configurations need to be used.

Œ

4.4.3. Testing in the lack of knowing the fold structure

When stratification is not used or its use is not indicated in a paper explicitly, any fold configuration can be assumed that satisfies the minimum requirements we formulated before. To address these scenarios, we note that if the number of folds is in the usual range (5-10) and the dataset is imbalanced or relatively small, it is feasible to enumerate all possible fold configurations and test each one of them. If all fold configurations are inconsistent with the reported scores, one can conclude that the experimental setup could not have yielded the reported scores. Although it might seem intractable to test all possible configurations in a real-life scenario, in Section 5.2, we show real applications of this scenario. Enumerating all possible k-fold configurations given p, n and k is a non-trivial combinatorial problem. In the rest of this section, an algorithm is developed 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 [48]. 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 be complemented 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 [48]). The only complexity that needs to be addressed is that the folds have varying cardinalities if 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, that we denote by the 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 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 p^a positives can be 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, further configurations can be removed, for instance, if sensitivity is reported, configurations with folds having zero positives could not drive the evaluation, since they lead to undefined sensitivity scores which contradicts the fact that the average sensitivity is reported.

Œ

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 assuming both aggregations. If both tests lead to inconsistencies, one can safely conclude that the reported scores could not be the outcome of the experimental setup under either of the two reasonable modes of aggregation.

5. Applications

In this section, we illustrate 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 the ones we used in our previous work [22] in the field of retinal vessel segmentation. In this subsection, we provide a concise overview of that scenario, 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 popular dataset for evaluation (DRIVE [49]) 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 of ranking algorithms in most papers. Due to the specialities of the image acquisition techniques, the useful image content resides in a disk shaped area in the center of a rectangular image, referred as the *Field of View* (FoV) (see Figure **??** plotting one entry of the DRIVE dataset). Textual evidence suggested that some authors evaluate the performance of segmentation in the FoV region only, while others using all pixels of the images, but usually the region of evaluation is not specified explicitly. The problem is that the pixels outside the FoV region can easily be identified as non-vessel pixels, and account for about 30% of all pixels: adding them to the true negatives boosts the accuracy and specificity scores compared to the case when the pixels covered by the FoV mask are used for evaluation.

Some authors report performance scores at the image level, and almost all authors provide average accuracy, sensitivity and specificity scores based on the 20 test images. In [22], we used specified versions of the tests proposed in Sections 3, 4.2 and 4.3 to assess the consistency of scores reported in 100 papers with two different assumptions: using only the pixels in the FoV region or using all pixels in the images for evaluation (note that the overall number of negatives n is different). We found that about 30% of the scores were inconsistent with any of the assumptions. The remaining scores were consistent with one of the assumptions and inconsistent with the other. A careful analysis of the rankings in the papers revealed that the rankings are based on incomparable scores in 100 papers. Beyond this insight, we also managed to derive a new baseline ranking by applying a correction to the scores derived from all pixels in the images to make them comparable with the ones calculated in the FoV region only.

Numerous further lesions and anatomical structures in retinal images, such as *exudates* [50] and the *optic disk* [51], are targeted by segmentation and detection algorithms. The possibility of evaluating their performance in the FoV region or using all pixels in the images is present in all related problems. Building on the successful application of the proposed techniques for vessel segmentation, it is reasonable to assume that these methods can be applied to validate and rectify the outcomes of other problems in retinal image processing.

Œ

5.2. Preterm delivery prediction from electrohysterogram signals

In recent years, there has been growing interest in predicting preterm delivery from electrohysterogram (EHG) signals [10], particularly with the availability of the TPEHG dataset [52], which comprises 38 positive and 262 negative records. At some point, multiple authors reported almost perfect prediction scores in kFCV scenarios. However, in the study [11], 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. 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 [11] reproduced all of the 11 papers and concluded that the most likely cause for the overly optimistic results is the application of minority oversampling prior to kFCV.

Since minority oversampling prior to kFCV increases the number of samples used for evaluation, one can expect inconsistencies between the reported scores and the correct experimental setup. Consequently, the cumbersome, time-consuming and error-prone work of reimplementing the algorithms could be replaced by employing the techniques developed in this paper. For example, one of the papers identified with the methodological flaw was [54], reporting $\hat{v}_{acc}^{MoS} = 0.9447$, $\hat{v}_{sens}^{MoS} = 0.9139$ and $\hat{v}_{spec}^{MoS} = 0.9733$ by 5-fold cross-validation. Although the authors mention that they used minority oversampling to increase the overall number of positives to $p' = 244$, it is unclear if it was used only for training or for evaluation, as well. The authors did not refer to using stratification, therefore, the evaluation of all fold configurations is needed for conclusive results. The number of possible fold configurations (with the correct $p = 38$, $n = 262$ counts and $k = 5$) becomes 918 (by the exhaustive enumeration in Section 4.4.3). The MoS test (Section sec:mor) reveals the inconsistency of scores with each configuration, leading to the conclusion that the reported scores are inconsistent with assumption of using 5-fold cross-validation

on the original dataset. However, with the assumption of using $p' = 244$ positive samples (that is, using the highly correlated generated samples for both training and evaluation), the overall number of possible fold configurations becomes $2.6M$, but already the 962th configuration $[(1, 101), (4, 97), (40, 61), (99, 2), (100, 1)]$ serves as evidence that the reported scores could be yielded with p' and n counts, 5-fold cross validation and the actual (tp_i, tn_i) counts $[(1, 96), (3, 92), (38, 59), (90, 2), (96, 1)]$. Consequently, the proposed method serves a numerical confirmation of the findings in [11]: the reported scores could not be the outcome of a proper cross-validation, but they could be the outcome of the improper use of minority oversampling.

Beyond eliminating the need of reimplementation in this particular application, the example also highlights that the proposed techniques are suitable for detecting methodological flaws related to the increasingly prevalent use of minority oversampling in various fields [53].

5.3. Classification of skin lesions

In this section, we illustrate the application of the proposed techniques in a field where – to our best knowledge – no meta-analysis with the purpose of validating the reported results has been conducted before: the classification of skin lesion images. An analysis as detailed as the one we did in retinal vessel segmentation [22] is clearly beyond the scope of the paper, however, we can test the reported scores in some highly influential papers to see if ambiguities are present. The analysis is based on the highly cited survey [55] providing a systematic overview of research up until 2021. From the survey, we have selected the 10 papers (listed in Table **??**) with the most citations according to Google Scholar at the time of writing.

Unlike the problems in Sections 5.1 and 5.2, this field is centered around multiple datasets (see Table **??**) with the task of classifying skin lesion images into two (malignant and non-malignant) or multiple, more specific classes. After a careful analysis of the selected papers, we found that [56], [57] and [58] are not suitable for testing (for the reason see the 'conclusion' column of Table **??**). In the remaining papers, the authors share enough details to apply the consistency tests (the multiclass problem of ISIC2017 [59] 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, there are multiple sets of performance scores shared (illustrating the operation of certain steps of the algorithm), the most commonly reported ones being accuracy, sensitivity and specificity. Given that the datasets are either supplied with a test set of images, or the authors designate a test set and share its details, in each case the consistency test developed in Section 3 was applied. The number of reported score sets ($N_{sc.}$) and the number of inconsistent sets ($N_{inc.}$) is shared in the corresponding columns of Table **??**. In [60] and [61] no inconsistency was detected. In [62], [63], and [64], only a handful of score sets show inconsistencies, which might be caused by typos. However, the scores reported in papers [65] and [66] were found to be inconsistent with the assumption regarding the experiment, therefore, a more detailed analysis was carried out on these papers.

Regarding [65], we tested multiple assumptions, such as scores like M_{ACC} represent the accuracy of the *melanoma* (M) class against the *nevus* class instead of both the *nevus* and *seborrheic keratosis* classes; and we also assumed that the accuracy and specificity figures are interchanged in the paper, since accuracy cannot be higher than both sensitivity and specificity the same time. All assumptions lead to inconsistencies with the claimed number and distribution of test images, therefore, we concluded that the reported scores are incomparable with the scores reported in other papers for the same dataset (such as [64]). Regarding [66], 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 the certain classes as weights. This uncommon weighting makes sensitivity the same as accuracy, and turns specificity into 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 [65] and [66]) with inconsistent scores reported and often recited for comparison and ranking in other research (e.g., [67]).

Œ
Œ

6. Conclusions

The meta-analysis of research is a crucial tool for addressing the reproducibility crisis in AI research and applications. Nevertheless, without numerical techniques, it demands enormous manual labor. To facilitate meta-analysis and enable the numerical identification of methodological inconsistencies, we have introduced various tests assessing the consistency of reported performance scores and experimental setups in binary classification.

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). We showed that the testing of scores aggregated by the *Score of Means* strategy falls back to the case of scores derived from a single evaluation set (Section 4.2). For *Mean of Scores* aggregations we developed a test supporting four of the most commonly reported scores (Section 4.3), and we also proposed the enumeration of all fold configurations when stratification is not used the specifics of the folds are unknown (Section 4.4.3). Across Sections 3 and 4, we highlighted multiple opportunities to improve the efficiency or extend the coverage of the tests.

In terms of applications, we briefly discussed the prior application of simplified forms of the proposed method in the field of retinal vessel segmentation (Section 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 for reimplementation in situations similar to the problem of preterm delivery prediction from EHG