Empirical Software Engineering (2022) 27:185
https://doi.org/10.1007/s10664-022-10199-2

Comparing ϕ and the F-measure as performance metrics for software-related classifications

Luigi Lavazza[1] · **Sandro Morasca**[1]

Accepted: 2 July 2022

Abstract

Context The F-measure has been widely used as a performance metric when selecting binary classifiers for prediction, but it has also been widely criticized, especially given the availability of alternatives such as ϕ (also known as Matthews Correlation Coefficient).

Objectives Our goals are to (1) investigate possible issues related to the F-measure in depth and show how ϕ can address them, and (2) explore the relationships between the F-measure and ϕ.

Method Based on the definitions of ϕ and the F-measure, we derive a few mathematical properties of these two performance metrics and of the relationships between them. To demonstrate the practical effects of these mathematical properties, we illustrate the outcomes of an empirical study involving 70 Empirical Software Engineering datasets and 837 classifiers.

Results We show that ϕ can be defined as a function of *Precision* and *Recall*, which are the only two performance metrics used to define the F-measure, and the rate of actually positive software modules in a dataset. Also, ϕ can be expressed as a function of the F-measure and the rates of actual and estimated positive software modules. We derive the minimum and maximum value of ϕ for any given value of the F-measure, and the conditions under which both the F-measure and ϕ rank two classifiers in the same order.

Conclusions Our results show that ϕ is a sensible and useful metric for assessing the performance of binary classifiers. We also recommend that the F-measure should not be used by itself to assess the performance of a classifier, but that the rate of positives should always be specified as well, at least to assess if and to what extent a classifier performs better than random classification. The mathematical relationships described here can also be used to reinterpret the conclusions of previously published papers that relied mainly on the F-measure as a performance metric.

Communicated by: Martin Shepperd

This work was partly supported by the "Fondo di ricerca d'Ateneo" funded by the Università degli Studi dell'Insubria.

✉ Luigi Lavazza
luigi.lavazza@uninsubria.it

Extended author information available on the last page of the article.

Keywords Binary classification · Software defect prediction · Performance evaluation · Performance metrics · Matthews Correlation Coefficient · F-measure · F-score

1 Introduction

Classification problems are quite common in rather diverse application areas of software practice and research. Here are just a few examples:

- The classification of the words in requirements texts has been used to derive the semantic representation of functional software requirements (Sonbol et al. 2020).
- Software requirements have been classified into functional requirements and sub-classes of non-functional requirements via machine-learning techniques (Dias Canedo and Cordeiro Mendes 2020).
- Machine-learning techniques have also been used to recognize attacks to software-defined networks (Scaranti et al. 2020).
- The diffusion of news via Twitter was used to classify news articles pertaining to disinformation vs. mainstream news (Pierri et al. 2020).
- Defect prediction, which is probably the best known software classification activity (Hall et al. 2011), classifies software modules as faulty or non-faulty.

Many different classifiers are built and used to address these and other Empirical Software Engineering problems. Thus, it is important to assess how well classifiers perform, so the best can be selected.

In this paper, we focus on binary classifiers, which are the most widely used, and on the metrics that have been defined to evaluate their performance. Using one performance metric instead of another may lead to very different evaluations and ranking among competing classifiers. To select effective and practically useful classifiers, it is therefore crucial to use performance metrics that are sound and reliable. This requires carefully examining and comparing the properties and possible issues of performance metrics before adopting any of them.

The F-measure (also known as F-score or F1) is a performance metric that has been widely used in Empirical Software Engineering. For instance, it was used—along with other metrics—to evaluate the performance of the classifications obtained in all of the empirical studies mentioned above.

The F-measure combines two performance metrics, *Precision* and *Recall*, also widely used to measure specific aspects of performance. As such, the F-measure is often perceived as a convenient means for obtaining an overall performance metric. The F-measure was originally defined to evaluate the performance of information retrieval techniques (van Rijsbergen 1979). However, it has numerous serious drawbacks that spurred criticisms (Hernández-Orallo et al. 2012; Powers 2011; Sokolova and Lapalme 2009; Luque et al. 2019).

Several researchers favored using other performance metrics like ϕ (Cohen 1988) (also known as Matthews Correlation Coefficient (Matthews 1975)), which are generally considered sounder (Yao and Shepperd 2020).

Unfortunately, the F-measure and ϕ may rank competing classifiers in different ways. According to Yao and Shepperd's analysis of the literature, around 22% of the published results change when ϕ is used instead of the F-measure (Yao and Shepperd 2021).

The goal of this paper is to analyze and compare the issues, advantages, and relationships of F-measure and ϕ, to help decision-makers use the performance metric that allows them to select the classifiers that better suit their goals.

Thus, after introducing the basic notions and terminology in Section 2, the paper provides the following main contributions, which we list along with the section where they can be found.

- We provide an organized in-depth discussion and comparison of the characteristics of the F-measure and ϕ, by building on the criticisms of the literature and adding some more observations (Section 3).
- We show that ϕ is a mathematical function of *Precision*, *Recall*, and the rate of actual positive modules (Section 4).
- We show that ϕ can be mathematically expressed as a function of the F-measure and the rates of actual and estimated positive modules. We study the extent to which these rates influence the set of possible values of ϕ that correspond to a given value of the F-measure. We also derive the conditions under which both the F-measure and ϕ rank two classifiers in the same order (Section 5). Specifically, we proved that ϕ and the F-Measure tend to rank two classifiers in the same way when the rate of actual positives is quite small. This results explains why the F-Measure was originally proposed in the information retrieval domain, where the rate of actual negatives is generally very large. When that is not the case—as in many software engineering situations—even a seemingly high value of the F-Measure may correspond to a performance not better than that of a random classifier.
- The knowledge provided in this paper casts new light on some results published previously, allowing for a more rigorous and sound reinterpretation of such results, and in some cases leading to rejecting conclusions that are not based on reliable evaluations (Section 7).

Our mathematical approach, described and proved in Sections 4 and 5 (whose details can be found in the Appendices), provides a theoretical explanation for the findings of the previous literature (discussed in Section 8), which were based on empirical studies or simulations. In addition, it generalizes and extends them to new results and evidence. Our results are of mathematical nature and therefore do not need empirical confirmatory evidence. At any rate, for demonstrative illustration purposes only, we also carried out an empirical study with 70 real-life Empirical Software Engineering datasets and 837 classifiers (shown in Section 6), to show the practical relevance of the mathematical results.

As we remark in the conclusions in Section 9, our study indicates that i) the proportion of positive modules should always be reported, along with the performance metrics of choice, ii) the F-measure should be used only when the rate of positive modules is very small, iii) ϕ is always a useful alternative, as already observed by some other previous studies, iv) if possible, providing the raw measures that are used to compute performance metrics is the best choice, as it provides the most detailed view of performance.

As a final observation, the mathematical results reported in this paper depend only on the definitions of ϕ and F-measure. Therefore, they can be used in the evaluation of any binary classifier used in Software Engineering and any other domains. At any rate, in the Software Engineering domain, our results can be useful in software defect prediction, in which binary classifiers are used to estimate which software modules are likely to be defective and should

be treated as such. To this end, the illustration empirical study of Section 6 focuses on software defect prediction.

2 Background

A classifier is a function that partitions a set of n elements into equivalence classes, identified by different labels. We only deal with binary classifiers, hence we write "classifier" instead of "binary classifier" for conciseness in what follows. Also, since we are interested in software-related classifiers, instead of "element" we use "software module," or, for short, "module," by which term we denote any piece of software (e.g., routine, method, class). The modules of the set are therefore classified as "positive" or "negative," where the meaning of these labels depends on the specific application. For instance, when estimating whether software modules are defective, the label "positive" means "faulty module" and the label "negative" means "non-faulty module."

The performance of a classifier on a set of modules is usually assessed based on a 2×2 matrix called "confusion matrix" (also known as "contingency table") that shows how many of those n modules are correctly and incorrectly classified. As Table 1 shows, the cells of a confusion matrix contain the numbers of modules that are: correctly estimated negative (True Negatives *TN*); incorrectly estimated negative (False Negatives *FN*); incorrectly estimated positive (False Positives *FP*); and correctly estimated positive (True Positives *TP*).

In Table 1, we also reported *EN* and *EP*, the numbers of Estimated Negatives and Estimated Positives, and *AN* and *AP*, the numbers of Actual Negatives and Actual Positives. *AN* and *AP* are intrinsic characteristics of the dataset, as is the actual prevalence $\rho = \frac{AP}{n}$ (Yao and Shepperd 2021). Instead, *EN* and *EP* depend on the classifier, like the estimated prevalence $\sigma = \frac{EP}{n}$.

Note that prevalence, quantified via ρ, is closely related to the notion of class imbalance, as quantified by *IR* (Imbalance Ratio), which is the ratio of the number of elements of the majority class to number of the elements of the minority class. In several application areas, e.g., software defect prediction, there is a majority of negative elements, so, for instance, Song, Guo, and Shepperd (Song et al. 2019) take $IR = \frac{AN}{AP} = \frac{1}{\rho} - 1$. Because of the existence of this functional relationship between prevalence and imbalance, we take into account class imbalance via prevalence ρ in the paper. Unlike *IR*, ρ ranges between zero and one: according to ρ, a dataset is perfectly balanced when $\rho = 0.5$, while positive classes are prevalent when $\rho > 0.5$, and negative classes are prevalent when $\rho < 0.5$.

A perfect classifier has $FN = FP = 0$, but this is hardly ever the case for any real-life classifier, so, the closer *FN* and *FP* are to zero, the better. To evaluate the performance of a classifier with respect to *FP* or *FN*, two performance metrics have been defined and used, respectively, *Precision* and *Recall*. For brevity, and to shorten the length of the formulas,

Table 1 A confusion matrix

		Actual: Negative	Actual: Positive	
Estimated	Neg.	*TN*	*FN*	$EN = TN + FN$
	Pos.	*FP*	*TP*	$EP = FP + TP$
		$AN = TN + FP$	$AP = FN + TP$	$n = AN + AP$
				$n = EN + EP$

we denote *Precision* by *PPV* (Positive Predictive Value) and *Recall* by *TPR* (True Positive Rate), as defined in Formula (1)

$$Precision = PPV = \frac{TP}{EP} \qquad Recall = TPR = \frac{TP}{AP} \tag{1}$$

PPV is the proportion of estimated positives that have been correctly estimated, and can be used to quantify *FP*, since *FP* = *EP* (1–*PPV*). Maximizing *PPV* amounts to minimizing *FP*, regardless of the value of *FN*. Thus, maximizing *PPV* is important when the cost of dealing with an estimated positive is high, but the impact of having false negatives is low.

TPR is the proportion of correctly estimated actual positives, and it is related to *FN*, since *FN* = *AP* (1–*TPR*). Maximizing *TPR* amounts to minimizing *FN*, regardless of the value of *FP*. Maximizing *TPR* is important when the consequences of false negatives are substantial and the cost of dealing with a false positive instead is quite low.

So, using one of these performance metrics means dealing with only one between *FP* and *FN*.

Given two classifiers cl_1 and cl_2, it is easy to conclude that cl_1 is preferable to cl_2 if $TPR_1 > TPR_2$ and $PPV_1 > PPV_2$. However, it is not straightforward to draw any conclusions if $TPR_1 > TPR_2$ and $PPV_1 < PPV_2$, or if $TPR_1 < TPR_2$ and $PPV_1 > PPV_2$. This is a typical issue in multi-objective optimization, since the goal here is to minimize two figures of merit, i.e., *FN* and *FP*, or, equivalently, maximize *TPR* and *PPV*, which may not be possible at the same time. Multi-objective optimization is often reduced to single-objective optimization, by defining a single figure of merit (Serafini 1985). Based on the cells of the confusion matrix, several performance metrics have been defined and used to act as single figures of merit. Different performance metrics take into account different aspects of performance that can be of interest in different application cases.

2.1 The Definition of F-Measure

The purpose of the F-measure (*FM*) is to combine *PPV* and *TPR* into a single performance metric by taking their harmonic mean, as shown in Formula (2)

$$FM = \frac{2}{\frac{1}{PPV} + \frac{1}{TPR}} \tag{2}$$

Since *FM* was originally defined to evaluate the performance of information retrieval (van Rijsbergen 1979), the focus is on how well the true positives have been identified. It is important that, at the same time, (1) a high proportion of actual positives be correctly estimated as such, so *TPR* should be high, and (2) a high proportion of the estimated positives be positive indeed, so *PPV* should be high too. Instead, true negatives are not taken into account in the computation of the F-measure because (1) their number is usually very large, (2) it is generally unknown, and (3) in practice it is hardly relevant.

Strictly speaking, *FM* is not defined when $PPV = 0$ or $TPR = 0$, i.e., $TP = 0$, but we can safely assume that $FM = 0$ when $TP = 0$, since it can be easily shown that

$$FM = \frac{2TP}{2TP + FN + FP} \tag{3}$$

and the rightmost fraction is equal to 0 when $TP = 0$.

So, FM is in the [0,1] range, with $FM = 0$ if and only if $TP = 0$, i.e., no actual positives have been correctly estimated, and $FM = 1$ if and only if $FP = FN = 0$, i.e., in the perfect classification case. When interpreting FM, classifier cl_1 performs better than classifier cl_2 if $FM_1 > FM_2$. So, the higher FM, the better.

FM is a special case of a more general definition that includes a parameter β, used to weigh PPV and TPR differently, as shown in Formula (4)

$$F_\beta = (1 + \beta^2) \frac{PPV \cdot TPR}{\beta^2\ PPV + TPR} \tag{4}$$

However, β is set to 1 in the near totality of Empirical Software Engineering studies using FM. So, we use "F-measure" (or FM) instead of F_1.

2.2 The Definition of ϕ

The purpose of ϕ, defined in Formula (5), is to assess the strength of the association between estimated and actual values in a confusion matrix

$$\phi = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{EN \cdot EP \cdot AN \cdot AP}} \tag{5}$$

ϕ is not defined when $EN \cdot EP \cdot AN \cdot AP = 0$, i.e., when at least an entire row or column of the confusion matrix is null. As Chicco and Jurman (2020) observe, if exactly one among AP, AN, EP, or EN is null, i.e., when exactly one column or row of the confusion matrix is null, the value of ϕ can be set to 0. When a row and a column are null, ϕ can be set to 1 if the only nonnull cell in the confusion matrix is $TN = n$ or $TP = n$ (perfect classification) and instead set to -1 if the only nonnull cell in the confusion matrix is $FN = n$ or $FP = n$ (total misclassification). At any rate, these cases are quite peculiar, as Yao and Shepperd observe too (Yao and Shepperd 2021), since they apply to datasets composed exclusively of elements belonging to one class.

ϕ is in the $[-1, 1]$ range. Specifically, $\phi = 1$ if and only if $FP = FN = 0$, i.e., in the perfect classification case. $\phi = 0$ is the expected (i.e., average) performance of the random classifier that estimates a module positive with a probability equal to ρ, i.e., with the same probability as that of selecting a positive module totally at random from the set of modules.

$\phi = -1$ if and only if $TP = TN = 0$, i.e., in the perfect misclassification case, that is, with a "perverse" classifier. It is well-known that perfect misclassification can be transformed into perfect classification by simply inverting the estimations, which means swapping the rows, in terms of confusion matrices. More generally, when $\phi < 0$, a classifier appears to be better at misclassifying modules than at classifying them correctly, so one can invert the estimations to obtain a classifier that instead is better at classifying modules correctly.

The interpretation of $\phi \geq 0$ is that classifier cl_1 performs better than classifier cl_2 if $\phi_1 > \phi_2 \geq 0$, so the higher $\phi \geq 0$, the better.

Thus, ϕ is an effect size measure, which quantifies how far the estimation given by a classifier is far from being random i.e., from the random classifier that has $\phi = 0$. A commonly cited proposal (Cohen 1988) uses $\phi = 0.1$, $\phi = 0.3$, and $\phi = 0.5$ respectively to denote a weak, a medium, and a large effect size. ϕ is also related to the χ^2 statistic, since $|\phi| = \sqrt{\frac{\chi^2}{n}}$.

3 A Comparative Assessment of *FM* and ϕ

In Sections 3.1 and 3.2, we report and elaborate on some of the issues that have been found about *FM* in the past, and add another possible issue in Section 3.3. We show whether and how ϕ can address them. In Section 3.4, we introduce and discuss a possible advantage of *FM*, which seems to be more sensitive to false negatives than to false positives. We summarize the results of our comparative assessment in Section 3.5.

3.1 *FM* Does not Take into Account *TN*, while ϕ Does

Formula (3) clearly shows that *FM* does not depend on *TN*. So, let us consider the two confusion matrices CM_a and CM_b shown below, which concern different datasets. CM_a and CM_b only differ in the number of true negatives

CM_a

TN=100	FN=40
FP=10	TP=50

CM_b

TN=500	FN=40
FP=10	TP=50

Both have the same value $FM_a = FM_b = \frac{2 \cdot 50}{2 \cdot 50 + 10 + 40} \simeq 0.67$. However, $\phi_b \simeq 0.64$, while $\phi_a \simeq 0.5$: $\phi_b > \phi_a$ because ϕ accounts for the fact that in CM_b 400 more true negatives are correctly classified than in CM_a.

Take now a third confusion matrix CM_c, concerning a third dataset.

CM_c

TN=5	FN=39
FP=10	TP=51

It is $FM_c \simeq 0.68$, thus, according to *FM*, one should conclude that the performance represented by CM_c is slightly better than those represented in CM_a and CM_b. However, though one more actual positive is classified correctly in CM_c, when it comes to classifying actual negatives CM_c performs quite poorly. $\phi_c \simeq -0.07$ appears to account for the overall performance represented by CM_c more adequately.

Based on these examples and on Formula (3), it appears that *FM* is not always an adequate metric for quantifying the overall performance of a classifier, since it does not use all available information about the classification results. This is one of the main criticisms made to *FM* by previous studies (Powers 2011; Yao and Shepperd 2021).

Formula (5) instead shows that ϕ takes into account all of the cells of a confusion matrix, so ϕ is a better performance metric for the *overall* performance of a classifier. The above examples with CM_a, CM_b, and CM_c show typical cases in which ϕ agrees with intuition more than *FM* does.

Note, however, that CM_a, CM_b, and CM_c show results related to three different datasets. When it comes to comparing the performance of classifiers on the same dataset, things are a bit different. Let us rewrite *FM* as

$$FM = \frac{2\,TP}{n + TP - TN} = \frac{2\,AP - 2\,FN}{2\,AP + FP - FN} \tag{6}$$

where the first fraction contains only TP and TN, which are related to correct classifications, while the second only FP and FN, which are related to misclassifications. AP and AN are fixed when comparing classifiers on the same dataset. So, knowing the values of two cells in different columns equates to knowing the entire confusion matrix. Thus, *FM*

provides an overall performance evaluation of a classifier that can be used when comparing classifiers applied to the same dataset.

3.2 *FM* Does not Allow for Comparisons with Baseline Classifiers, while ϕ Does

The assessment of a model, like a classifier, is typically done by comparing its performance against the performance of a less complex baseline model. A classifier estimates the class of modules by taking into account information on their characteristics. For instance, a classifier may estimate a software module defective or not defective based on the module's number of Lines Of Code (*LOC*). However, how much performance do we gain by using that classifier, instead of using random estimation, i.e., a baseline classifier that does not require any knowledge of the modules? Recall that a *random classifier* behaves as described in Section 2.2, i.e., it estimates each module positive with probability ρ.

The expected values of *TPR* and *PPV* for the random classifier (i.e., the mean values obtained from a large number of random estimations) are both equal to ρ (Morasca and Lavazza 2020): by using these values in (3), we obtain $FM = \rho$ as well, so, when evaluating a classifier, we should compare its *FM* against ρ. Thus, the knowledge of *FM* by itself is not sufficient to tell whether a classifier performs better than even random estimation (Yao and Shepperd 2021). An example is given by confusion matrix CM_c above: it is $FM_c \simeq 0.68 < \rho_c = \frac{AP}{n} = \frac{90}{105} \simeq 0.86$, thus the performance represented by confusion matrix CM_c is worse than random, on average.

On the other hand, ϕ, by its very definition, quantifies how far a classifier is from the random classifier. This may lead to what might seem to be a paradox, especially if compared to what happens with *FM*. Take CM_d below. We have $FM_d \simeq 0.78$, which in general is considered a quite good result in terms of *FM*. Also, visual inspection of the confusion matrix shows that the classifier is able to correctly classify most of the majority class (the 90 actual positives), even though it does not fare well with the minority class (the 15 actual negatives).

CM_d

TN=4	FN=25
FP=11	TP=65

So, one may suggest that the classifier performs well, since, at any rate, the minority class is only one-sixth of the majority class, so its contribution to performance should be much less than that of the majority class anyway. However, $\phi_d \simeq -0.01$ casts some serious doubts on the performance level of the classifier, which appears to be rather poor. Which performance metric should we trust then? The answer is that the classifier is indeed good in itself at estimating the positive class, as indicated by the high value $FM_d \simeq 0.78$. However, even the random classifier would be better overall, since it has $FM_{random} \simeq 0.86$. This is also indicated by the value of ϕ_d, because ϕ is an effect size measure, which in this case shows that the classifier is quite close to the random classifier, just a bit worse. Since we can always perform random estimation without having to go through the pains of building and validating a classifier, we must conclude that the classifier at hand is "good," but nowhere nearly good enough.

To further explain why the comparison with a baseline classifier is a fundamental point in the evaluation of a classifier beyond Empirical Software Engineering, consider that using a classifier on a dataset is similar to administering a treatment to a set of subjects: in a way, it is like giving a treatment to a dataset. A classifier is worth using if it has greater beneficial effects than using another existing classifier or doing nothing, i.e., relying on randomness.

Likewise, it is worthwhile giving a certain treatment to subjects only if it is better than some existing treatment or better than providing no treatment.

Suppose therefore that we need to evaluate the effectiveness of a medication for a disease. Suppose that, in a clinical trial, 96 out of 100 diseased patients that take the medication fully recover, i.e., the treatment achieves a 96% recovery rate. This rate looks quite high, especially if the disease is a lethal one. However, by itself, this seemingly high value does not tell us much about the *real* effectiveness. Suppose that the rate of spontaneous recovery from the disease, i.e., without taking any medications, is 97%. Then, one may argue that the medication actually *worsens* the chances of recovery. If, instead, the spontaneous recovery rate was 54%, for instance, then the medication would appear to be very effective. Thus, when evaluating the performance of some treatment (i.e., classifier, in our case) we always need to compare its effect to those of some baseline treatment (i.e., baseline classifier, in our case).

Prediction in Empirical Software Engineering refers to totally different domains than medical treatments, but the consequences of misjudging classifier effectiveness can be quite serious too. Using a performance metric that leads to selecting a classifier that estimates too many false negatives results in, say, having too many vulnerability attacks in software security applications or, in software quality assurance, having too many faulty modules released to the final users. If the selected classifier estimates too many false positives, precious resources are wasted by unnecessarily maintaining software to make it supposedly more secure or less faulty. Note that these unnecessary software modifications may even lead to introducing more vulnerabilities or defects.

3.3 *FM* is Nonnegative, while ϕ Can Be Negative

However strange it may seem, another issue with *FM* is that $FM \geq 0$. Suppose that $FM \simeq 0$: does this really mean that there is no association between estimated and actual values? Though this is the usual interpretation, $FM \simeq 0$ should instead be interpreted as a lack of a *concordant* association between the estimated and the actual values, but not as a lack of a *discordant* one.

CM_e

TN=100	FN=40
FP=10	TP=5

CM_f

TN=5	FN=40
FP=10	TP=5

Take the confusion matrices CM_e and CM_f above, which differ only by *TN*. We have $FM_e = FM_f \simeq 0.167$, i.e., a fairly small value for *FM*. However, it is apparent that CM_e represents a much better situation than CM_f, in terms of correct module classification. It is also apparent that in CM_f more modules (50) are misclassified than correctly classified (just 10). Detecting discordant associations can be useful, since it is possible to obtain concordant associations by inverting the classifications.

With ϕ, we have $\phi_e \simeq -0.079$, which indicates a close-to-random classification, and $\phi_f \simeq -0.556$, which indicates a rather strong discordant association.

If we swap the rows of CM_f, we obtain a new confusion matrix $CM_{f'}$ having $FM_{f'} = 0.889$, so one may conclude that it is possible to use *FM* to detect discordant classifications anyway. However, swapping the rows of the confusion matrix basically equates to using a different performance metric on the original confusion matrix, defined as $\frac{2FN}{AP+EN}$. This would defeat the purpose of having a single performance metric to evaluate the overall performance of a classifier, while ϕ is actually able to detect both concordant and discordant associations.

3.4 *FM* Gives Different Relative Importance to False Positives and False Negatives, while ϕ does not

In practical applications, the cost of a false positive may be quite different from the cost of a false negative. For instance, suppose that a defective software module is not detected during the Verification & Validation phase in the development of a safety-critical application. That module—a false negative—is then released to the users as a part of the final product. The cost due to the damages it can do during operational use is typically much higher than the unnecessary Verification & Validation cost incurred by a false positive module.

FM is symmetrical with respect to *PPV* and *TPR*, but *not* with respect to *FN* and *FP*. Take the fraction in the rightmost member of Formula (6). Swapping *FN* and *FP* produces another fraction that is equal to the one in Formula (6) if and only if $FN = FP$.[1] Thus, the same extent of variation in *FN* and *FP* does not have the same impact on *FM*. We show in Appendix A that increasing or decreasing *FN* by some amount has more impact on *FM* than does increasing or decreasing *FP* by the same amount.

To provide even more relative importance to false negatives, one may set β of Formula (4) to specific values. Increasing β means giving more importance to *TPR* with respect to *PPV*. However, as we already noted, the vast majority of studies in the literature use the definition of *FM* of Formula (3).

ϕ is perfectly symmetrical with respect to *FN* and *FP*, whose variations therefore have the same impact on the value of ϕ.

3.5 Summary of Evaluations

FM does not fully capture the intrinsic characteristics of the underlying dataset, being defined as the harmonic mean of *PPV* and *TPR* (see Appendix C for some considerations on the usage of the harmonic mean). Therefore, *FM* quantifies an aspect of a classifier's performance related to the positive class that may be used only for comparing classifiers for the same dataset. *FM* cannot detect the existence and the extent of discordant associations between estimated and actual values either. Since it does not take into account ρ, which is also the rate with which a random classifier would successfully detect positives, *FM* cannot tell whether a classifier performs better than the random classifier, which can be taken as an inexpensive, default classifier a decision-maker can always fall back on to. The only advantage that *FM* seems to have over ϕ is that *FM* is more affected by variations of *FN* than of *FP*. Thus, software managers that rely on *FM* as performance metric may be encouraged to reduce false negatives more than false positives.

4 Defining ϕ in Terms of *PPV* and *TPR*

Since they use the same pieces of information, performance metrics computed based on the confusion matrix may be expected to be related to each other, to some extent, especially if the considered performance metrics have the goal of providing an overall performance evaluation.

[1] Notice that Formula (3) may deceivingly show that *FM* is symmetrical with respect to *FN* and *FP*, while it is not. The reason is that any change in *FN* also implies a change in *TP*, since $TP + FN = AP$. So, swapping *FN* and *FP* also induces a change in *TP*. Instead, changing *FP* implies changing *TN*, which is not used in the computation of *FM*.

In this section, we show how ϕ too can be defined as a function of *PPV* and *TPR*, like *FM*, so we can point out the structural differences between *FM* and ϕ, along with their consequences.

In the following Section 5, we directly investigate the relationship between *FM* and ϕ and how it can be influenced by ρ and σ.

From Formula (5), via a few mathematical computations (reported in Appendix B), we have

$$\phi = \frac{1}{\sqrt{1-\rho}} \frac{\sqrt{TPR}(PPV-\rho)}{\sqrt{PPV-\rho\, TPR}} \tag{7}$$

Unlike *FM*, ϕ is not a symmetrical function of *PPV* and *TPR*, so equal variations in *PPV* and *TPR* have different effects on ϕ.

Formula (7) shows that, in addition to *PPV* and *TPR*, ϕ depends on ρ, which is an intrinsic characteristic of the dataset. Thus, for the same values of *PPV* and *TPR*, we can obtain different values of ϕ, depending on the imbalance degree of the dataset. However, as we show in Appendix B, given ρ, it is

$$\rho \leq \frac{PPV}{PPV + TPR - PPV \cdot TPR} \tag{8}$$

for $PPV \neq 0$ and $TPR \neq 0$. For completeness, Appendix B also shows what happens in the special case in which $PPV = TPR = TP = 0$. We also show in Appendix B that, for given values of *PPV* and *TPR*, ϕ is a monotonically decreasing function of ρ, which tends to $\sqrt{PPV \cdot TPR}$ when ρ tends to 0 and takes value $\frac{PPV+TPR-PPV\cdot TPR-1}{\sqrt{1-PPV}\sqrt{1-TPR}}$ when $\frac{PPV}{PPV+TPR-PPV\cdot TPR} = \rho$.

Figure 1 shows how ϕ varies depending on the value of ρ in three cases, depending on the values of *PPV* and *TPR* that satisfy (8). Note that $FM \approx 0.5$ for the three pairs of values of *PPV* and *TPR* used for the three curves.

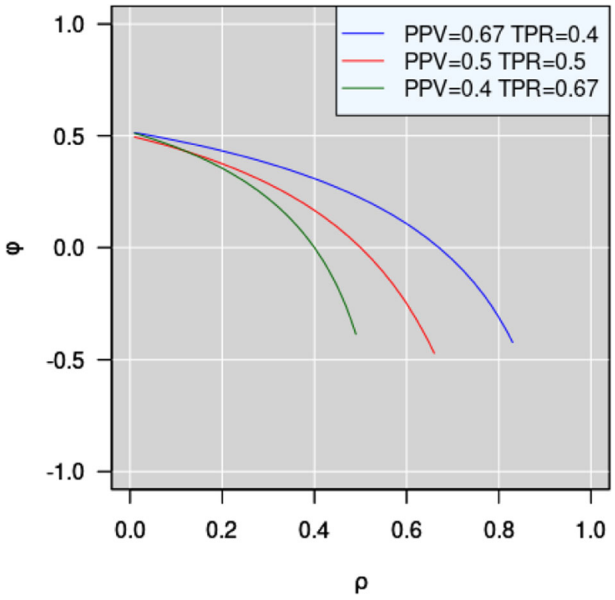

Fig. 1 ϕ vs. ρ, for different values of *PPV* and *TPR*

Figure 2 shows how ϕ varies depending on the value of $\rho \in [0, 1]$ in four cases, depending on the values of *PPV* and *TPR*. Note that $FM \approx 0.85$ for two pairs of *PPV* and *TPR*, and $FM \approx 0.24$ for the other two pairs.

In the special case of an unbiased classifier, i.e., when $EP = AP$, it is $PPV = TPR$ and $FP = FN$, so the confusion matrix is symmetric. In this case, as previously shown (see, e.g., Delgado and Tibau (2019)) ϕ coincides with Cohen's kappa (Cohen 1960), as follows

$$\phi = \frac{PPV - \rho}{1 - \rho} \tag{9}$$

Cohen's kappa is a measure of the extent to which two classifiers (the actual classifier and the estimated classifier, in our case) agree when classifying n items into a number of different categories (two categories, in our case).

It can be shown that Formula (9) as a function of ρ represents a rectangular hyperbola with asymptotes $\phi = 1$ and $\rho = 1$. Figure 3 shows how ϕ varies depending on the value of $\rho \in [0, 1]$ in three cases, i.e., for high (green line), medium (red line), and low (blue line) values of *PPV*.

Formula (9) also shows that ϕ, unlike *FM*, is not a central tendency indicator for *PPV* and *TPR*. Specifically, ϕ does not satisfy Cauchy's property (Cauchy 1821), according to which a central tendency indicator of a set of values must always be between the minimum and the maximum value in the set. In our case, for Cauchy's property to be satisfied, ϕ would have to be between *PPV* and *TPR*. Since *PPV* and *TPR* are equal for an unbiased classifier, this would mean having $\phi = PPV$ as a result of Formula (9), but this is not the case. *FM*, instead, satisfies Cauchy's property.

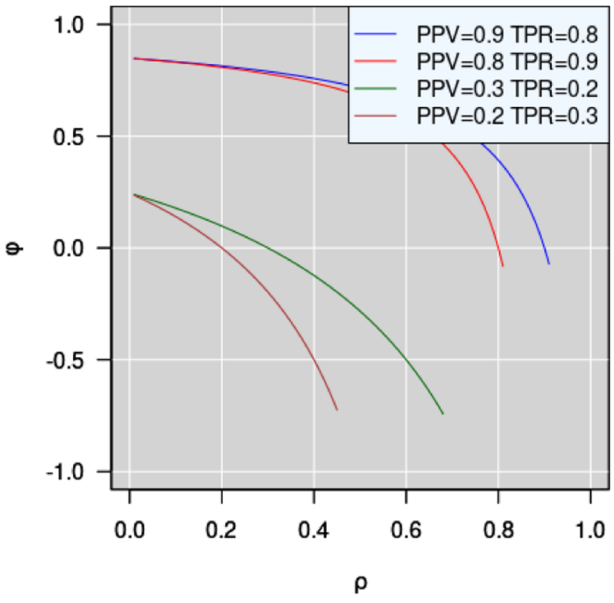

Fig. 2 ϕ vs. ρ, for different values of *PPV* and *TPR*

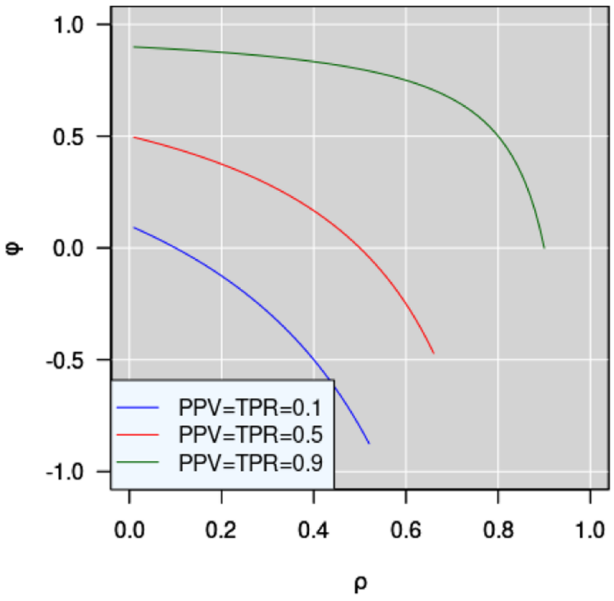

Fig. 3 ϕ vs. ρ in the unbiased case, i.e., for different values of $PPV = TPR$

5 The Relationships between ϕ and *FM*

Before presenting our mathematical study (described in Sections 5.2–5.5), we show some empirical evidence and simulation results about the relationship between ϕ and FM in Section 5.1.

5.1 Empirical Observations and Simulation Results

The scatterplot in Fig. 4 shows the values of ϕ and FM for all of the 837 classifiers that we obtained in our empirical study (more details in Section 6). The scatterplot, which shows only the part of the $FM \times \phi$ plane in which we obtained pairs of values (FM, ϕ) for our classifiers, is consistent with the findings by other researchers, like Yao and Shepperd (see Figure 4 in (Yao and Shepperd 2021)). It shows that the vast majority of the points are below the bisector. Also, FM and ϕ often provide discordant indications; while a high value of ϕ implies high values for FM, the converse is not true: when $\phi > 0.5$, it is also $FM > 0.5$, but when FM is close to 0.8, ϕ can be below 0.2 as well as above 0.6.

In a simulation analysis, Chicco and Jurman (Chicco and Jurman 2020) computed FM and ϕ for all confusion matrices (hence, for all ρ) with $n = 500$ and showed the results in a scatterplot. In Fig. 5, we show a similar scatterplot for illustration purposes, where we choose $n = 100$ because the dots corresponding to the confusion matrices are already dense enough that increasing the value of n would not change the graphical aspect of the figure. Figure 5 shows that, for a given value of FM, there is a wide range of possible values of ϕ, in general.

Sections 5.2–5.5 mathematically explain the scatterplots in Figs. 4 and 5. Specifically, in Section 5.2, we show how the relationship between FM and ϕ is influenced by ρ and σ. In Section 5.3, we derive the upper and lower bounds of ϕ when FM is known, based on a

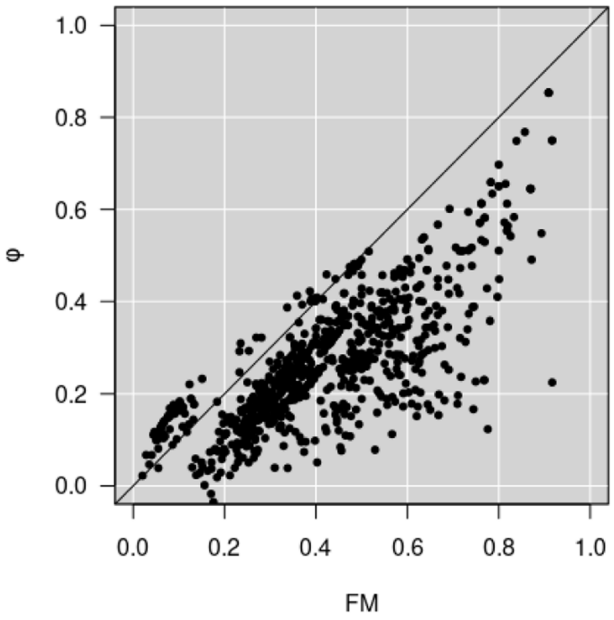

Fig. 4 FM vs. ϕ for all datasets, all classifiers

specified value of ρ. We show in Section 5.4 the conditions under which FM and ϕ provide the same ranking between two classifiers, given a value of ρ. Section 5.5 provides the upper and lower bounds of ϕ when FM is known, for all possible values of ρ.

5.2 The Mathematical Relationship Between ϕ and *FM*

The mathematical relationship between FM and ϕ can be expressed as in Formula (10) (the derivation can be found in Formula (38) of Appendix D)

$$\phi = \frac{1}{2\sqrt{\rho(1-\rho)}} \frac{(\rho+\sigma)FM - 2\,\rho\,\sigma}{\sqrt{\sigma\,(1-\sigma)}} \tag{10}$$

which shows how ϕ depends on FM, ρ, and σ.

In the special case of an unbiased classifier, i.e., when $EP = AP$ (hence $\sigma = \rho$)

$$\phi = \frac{FM - \rho}{1-\rho} \tag{11}$$

consistently with Formula (9), since $PPV = TPR = FM$ for unbiased classifiers. This relationship between FM and ϕ holds for all values of FM provided that $\rho \leq \frac{1}{2}$. When, instead, $\rho > \frac{1}{2}$, the relationship holds only for some values of FM, because it must be $\phi = \frac{FM-\rho}{1-\rho} \geq -1$, so $FM \geq 2\rho - 1$. For instance, when $\rho = 0.75$, it must be $FM \geq \frac{1}{2}$. This effect can also be seen in Fig. 3, in which the three lines also show ϕ vs. ρ for different values of $FM = PPV = TPR$ in the unbiased case. When $\rho = 0.75$, a line either coincides with the red line or is above it, i.e., it has a value of $FM = PPV = TPR \geq 0.5$.

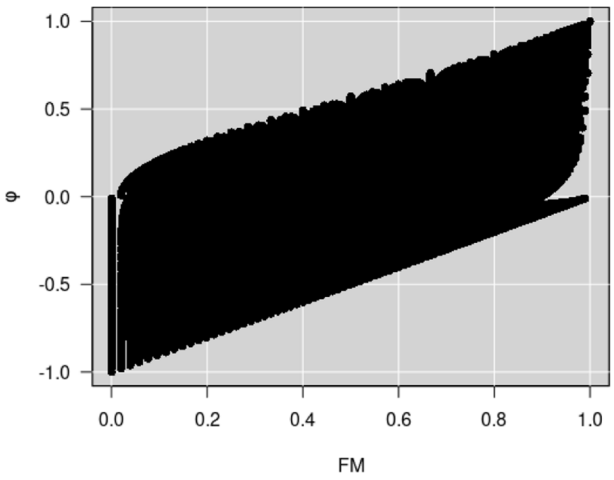

Fig. 5 *FM* vs. ϕ: scatterplot for all possible confusion matrices with $n = 100$

5.3 Variation Intervals of ϕ Depending on *FM* for Given Values of ρ

Formula (10) shows that, given a dataset (hence, given a value of ρ), the relationship between FM and ϕ is influenced by σ. To evaluate how tightly FM and ϕ are related to each other on a given dataset, it is useful to assess the extent of such influence, that is, how much ϕ can vary, depending on σ, for any given value of FM. Appendix E shows that ϕ belongs to an interval $[\phi_{\min}(FM;\rho), \phi_{\max}(FM;\rho)]$, where function $\phi_{\min}(FM;\rho)$ depends on the value of FM (for a given value of ρ, taken as a parameter), as shown in Formulas (12) and (13),

$$FM \leq \frac{2\rho}{1+\rho} \Rightarrow \phi_{\min}(FM;\rho) = -\sqrt{1 - \frac{FM}{2\rho - 2\rho^2 + \rho^2 FM}} \tag{12}$$

$$FM \geq \frac{2\rho}{1+\rho} \Rightarrow \phi_{\min}(FM;\rho) = \sqrt{\frac{FM}{1-\rho}}\sqrt{FM - 2\rho + \rho FM} \tag{13}$$

while the function that defines $\phi_{\max}(FM;\rho)$ is the same for all values of FM

$$\phi_{\max}(FM;\rho) = \sqrt{\frac{FM\,(1-\rho)}{2 - (1+\rho)\,FM}} \tag{14}$$

It can be shown that $\phi_{\min}(FM;\rho)$ is a continuous function and that it is zero for $FM = \frac{2\rho}{1+\rho}$.

The plots in Fig. 6 illustrate variation intervals for a few representative cases, in increasing order of ρ, namely $\rho = 0.01$, $\rho = 0.05$, $\rho = 0.1$, $\rho = 0.25$, $\rho = 0.5$, and $\rho = 0.75$. Thus, these plots are in decreasing order of dataset class imbalance IR, since $IR = \frac{1}{\rho} - 1$. The red and green lines respectively represent $\phi_{\max}(FM;\rho)$ and $\phi_{\min}(FM;\rho)$. For instance, in a dataset with $\rho = 0.05$, when $FM = 0.4$, ϕ may take a value between $\phi_{\min}(0.4; 0.05) \simeq 0.3671$ and $\phi_{\max}(0.4; 0.05) \simeq 0.4904$. The yellow straight line shows the relationship between FM and ϕ for unbiased classifiers defined by Formula (11). In this particular case, $\phi_{\max}(FM;\rho) = \phi_{\min}(FM;\rho)$, since there is no variation in σ, which is equal to ρ. For instance, when $\rho = 0.05$ and $FM = 0.4$, we have $\phi = 0.3684$.

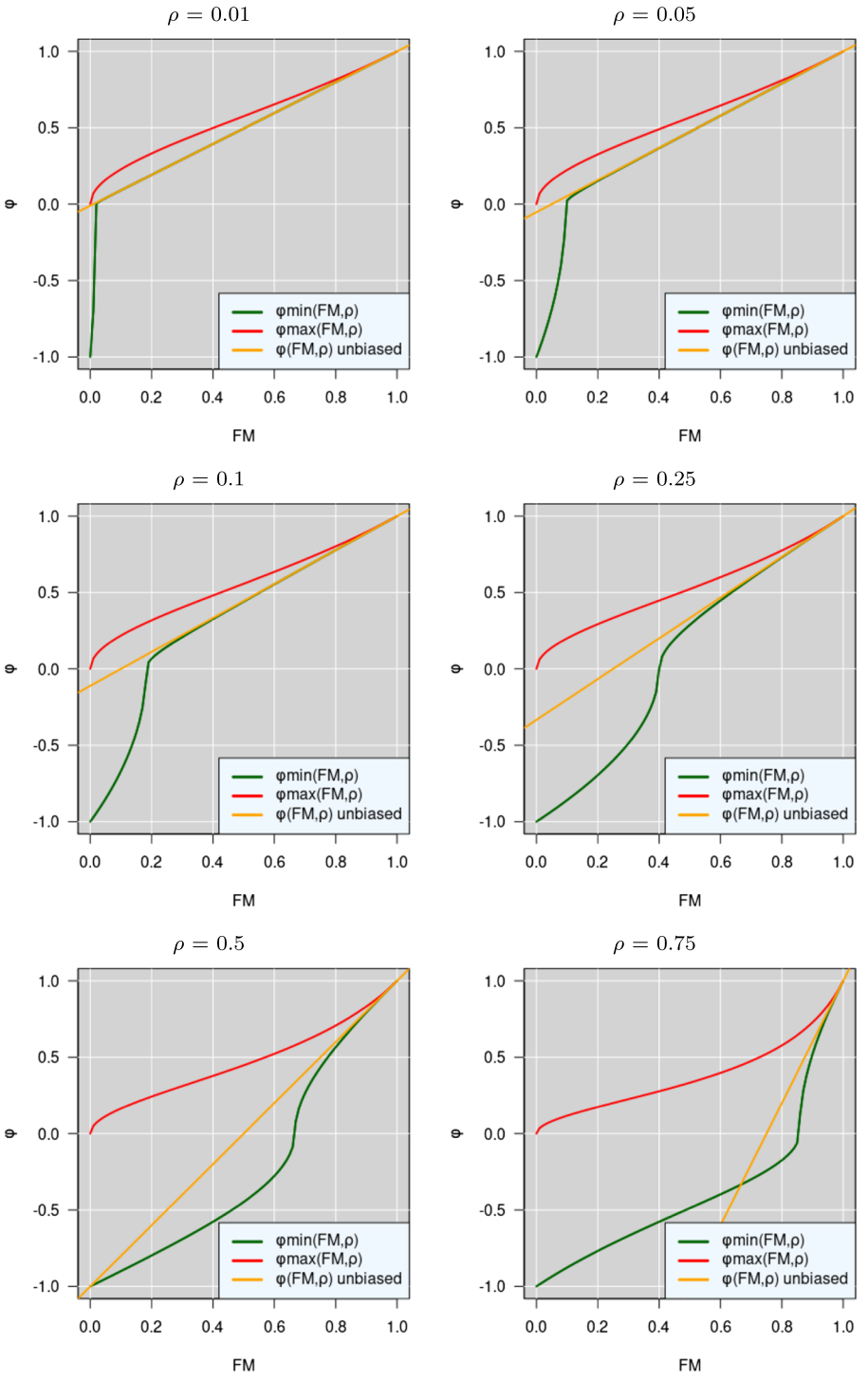

Fig. 6 Variation intervals of ϕ depending on *FM*, for various values of ρ

The plots graphically show how the width of the variation intervals depends on ρ and, therefore, imbalance. The region delimited by $\phi_{\min}(FM;\rho)$ and $\phi_{\max}(FM;\rho)$ is generally quite thin for small values of ρ and becomes thicker and thicker the larger ρ becomes. In other words, the uncertainty with which the value of ϕ can be known for a value of *FM* increases with ρ. Thus, the higher ρ, the easier it is to find both good and bad values of ϕ for any given value of *FM*. For instance, with $\rho = 0.5$, when $FM = 0.4$, ϕ may take a value between $\phi_{\min}(0.4; 0.5) \simeq -0.5773$ and $\phi_{\max}(0.4; 0.5) \simeq 0.378$.

The plots also show that, when $FM \geq \frac{2\rho}{1+\rho}$ and ρ is small, $\phi_{\min}(FM;\rho)$ approximates very well the straight line of Formula (11) that describes the relationship between *FM* and ϕ for unbiased classifiers. Note that, the smaller ρ, the larger is the interval $FM \in \left[\frac{2\rho}{1+\rho}, 1\right]$ in which this approximation can be used. For $FM \in \left[\frac{2\rho}{1+\rho}, 1\right]$, the difference between the value of ϕ of Formula (11) and $\phi_{\min}(FM;\rho)$ is maximum exactly when $FM = \frac{2\rho}{1+\rho}$. Since $\phi_{\min}\left(\frac{2\rho}{1+\rho}\right) = 0$, this maximum difference turns out to be equal to ρ.

5.4 Preserving Classifiers' Rankings with ϕ and *FM*

Let us now consider Fig. 6, specifically the plot for $\rho = 0.05$. Decision-makers relying on *FM* would not use classifiers with extremely low values of *FM*, so, let us focus on the region with $FM > 0.3$.[2] Figure 7 zooms in a part of that plot for $FM > 0.3$.

The variation interval of ϕ is quite narrow for $FM = 0.3$ (as $\phi_{\min}(0.3; 0.05) \simeq 0.26$ and $\phi_{\max}(0.3; 0.05) \simeq 0.41$), and it gets narrower and narrower as FM increases. As the variation interval gets narrower, the uncertainty about the values of ϕ for a given value of *FM* decreases, so it is more and more likely that two classifiers cl_a and cl_b are ranked in the same order by *FM* and ϕ. For instance, as shown in Fig. 7, suppose that $\rho = 0.05$. Take $FM_a = 0.4$, which has a variation interval $\phi \in [0.367, 0.490]$, and $FM_b = 0.5$, with variation interval $\phi \in [0.473, 0.567]$. These two intervals barely overlap, so it is quite unlikely that *FM* and ϕ provide two different orderings. Take now $FM_c = 0.6$, with variation interval $\phi \in [0.579, 0.645]$, which no longer overlaps with the ϕ variation interval for $FM_b = 0.5$. In this case, $FM_c > FM_b$ implies $\phi_c > \phi_b$ (and also $FM_c > FM_a$ implies $\phi_c > \phi_a$).

Also, the higher *FM*, the smaller the difference in *FM* to have complete separation between two variation intervals. With $\rho = 0.05$, suppose for instance that $FM_d = 0.65$, with variation interval $\phi \in [0.6313, 0.6846]$, and $FM_e = 0.7$, with variation interval $\phi \in [0.6840, 0.7250]$. These two intervals still minimally overlap, but it suffices to take $FM_f = 0.71$, which has a variation interval $\phi \in [0.6946, 0.7333]$ to have complete separation between the variation intervals related to FM_d and FM_f.

As shown in Appendix H, two intervals $[\phi_{\min}(FM_a;\rho), \phi_{\max}(FM_a;\rho)]$ and $[\phi_{\min}(FM_b;\rho), \phi_{\max}(FM_b;\rho)]$ with $FM_a < FM_b$ are completely separated if and only if

$$FM_b > \frac{\rho + \sqrt{\frac{2\rho^2(1-FM_a)+(1-\rho)\ FM_a}{2-(1+\rho)\ FM_a}}}{1+\rho} = sep(FM_a, \rho) \tag{15}$$

[2]There is no general consensus on acceptability thresholds for *FM*. As a matter of fact, a value $FM = 0.3$ would probably be considered too low for practical purposes, since generally it implies that either *PPV* or *TPR* is even below 0.3. However, we choose a value of *FM* low enough to show that the variation interval of ϕ is small for an interval of *FM* even larger than would be practically useful.

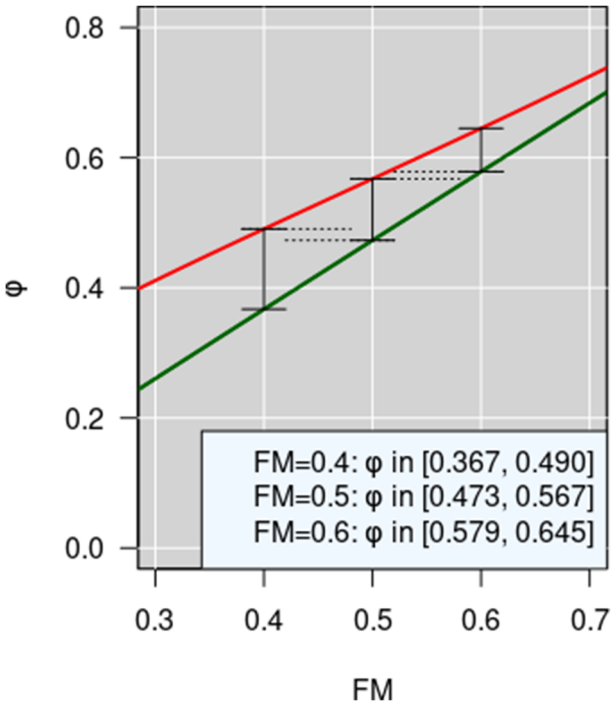

Fig. 7 Variation intervals of ϕ depending on FM ($\rho = 0.05$) may overlap or not

$sep(FM_a, \rho)$ is an increasing function of FM_a, as expected, i.e., the higher FM_a, the higher FM_b. It also is an increasing function of ρ. Figure 8 shows the behavior of $sep(FM_a, \rho)$ as a function of FM_a for a few values of ρ.

As suggested by Fig. 8, it can be shown that $sep(FM_a, \rho) \geq \rho$ for all values of FM_a and σ.

This inequality along with Formula (15) and Fig. 8 show that, given a value FM_a, the set of values of FM_b such that $FM_a < FM_b$ for which we have $\phi_a < \phi_b$ with certainty gets larger when ρ decreases. For instance, take $FM_a = 0.6$. When $\rho = 0.05$, any value of $FM_b > 0.663$ guarantees that $\phi_b > \phi_a$, while when $\rho = 0.5$, we need $FM > 0.783$, so that $\phi_b > \phi_a$. So, the ranking between two modules is more and more likely to be the same according to FM and to ϕ for smaller and smaller values of ρ.

Formula (15) is a special case of a more general formula that applies when using two classifiers cl_a and cl_b on two datasets with different actual prevalence values ρ_a and ρ_b, as discussed in Appendix F.

5.5 Variation Intervals of ϕ for All Values of ρ

We have so far supposed that ρ is given, so it is either known or it is assumed to be equal to some value. At any rate, we may want to delimit the variation interval $[\phi_{\min}(FM), \phi_{\max}(FM)]$ of ϕ for a given value of FM for all possible values of ρ. Appendix G shows that

$$\phi_{\min}(FM) = FM - 1 \tag{16}$$

$$\phi_{\max}(FM) = \sqrt{\frac{FM}{2 - FM}} \tag{17}$$

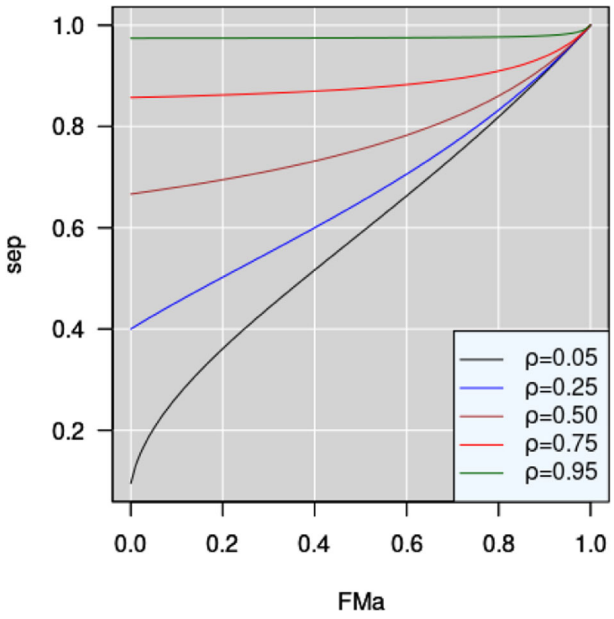

Fig. 8 $sep(FM_a, \rho)$ for a few values of ρ

Figure 9 shows how these two functions envelop the region in which all possible pairs $\langle FM, \phi \rangle$ appear. Thus, they analytically explain and confirm the simulation results by Chicco and Jurman (Chicco and Jurman 2020) and ours, as reported in Section 5.1.

5.6 Consequences of the Analytical Relationship Between ϕ and *FM*

Sections 5.2–5.5 show that the relationship between ϕ and *FM* depends on ρ and σ. Providing *FM* by itself, without specifying ρ, provides at best an incomplete view of a classifier's performance. In some cases, notably when ρ is quite small, ϕ and *FM* basically provide the same information, e.g., they tend to rank classifiers in the same order. The value ρ may be quite small in some software-related application cases. For instance, the actual prevalence of vulnerable software modules in a software system is typically quite low. In other application areas, however, the range of ρ can be quite wide, e.g., in software defect prediction. For instance, the real-life datasets that we used in the empirical study of Section 6 have ρ ranging between 0.007 and 0.988 (see also Fig. 10).

So, it is not possible to tell whether a classifier is an effective and useful one by simply looking at *FM*, and a statement like "classifier X achieves $FM = 0.8$, therefore it is very accurate," the likes of which have sometimes appeared in the literature, may be misleading. Therefore, the *FM* achieved by a classifier on a dataset should *always* be accompanied by the actual prevalence ρ of the dataset, also because ρ provides the F-measure value of a totally random classifier.

As a final observation, we note that our results confirm the validity of *FM* as a performance metric in the domain in which it was originally proposed, i.e., information retrieval. In fact, ρ is generally very small in information retrieval situations. Consider for instance the case of a search on `google.scholar.com`: you are typically interested in no more than a few hundred papers out of the 10^8 indexed papers, hence ρ is in the order of 10^{-5}.

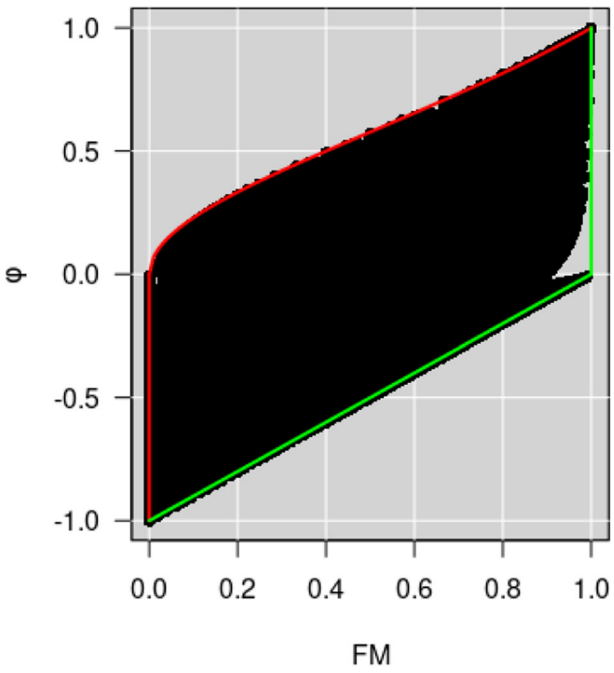

Fig. 9 *FM* vs. ϕ: variation intervals for all values of ρ

6 An Empirical Demonstration of the ϕ vs. *FM* Relationship

In this section, we show how the analytical results of Section 5 can explain empirical data, which were obtained from real-life projects. Note that this empirical demonstration is not meant to confirm the validity or correctness of the relationship between ϕ and *FM* or of any mathematical results introduced in Section 5. Those results were derived analytically and therefore do not need any empirical validation.

6.1 The Datasets

We use two sets of datasets that are publicly available from the SEACRAFT repository (2017) and are reported among the most widely used (Singh et al. 2015). The first set was collected by Jureczko and Madeyski (2010) from real-life projects of different types and has been used in several defect prediction studies (e.g., Bowes et al. (2018) and Zhang et al. (2017)). The second set is the NASA Metrics Data Program defect dataset (Menzies and Di Stefano 2004); it has also been used in several defect prediction studies (e.g., Gray et al. (2011)). Therefore, in this section, a positive module is a defective one and a negative module a non-defective one. Some descriptive statistics concerning the datasets are given in Appendix I.

The data from the aforementioned datasets were used to derive models of module defectiveness. The technique used to derive defect predictors is immaterial for the purpose of this work; nonetheless, we provide some details in Appendix I.

Given the importance of ρ, the distribution of ρ in the considered datasets is illustrated by the boxplot in Fig. 10. This distribution contains a fairly large and varied set of values

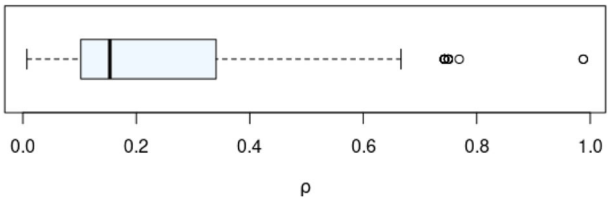

Fig. 10 Boxplot illustrating the distribution of ρ in the datasets used for the experimental demonstration

of ρ, though it is clearly skewed, with most datasets having a quite small actual prevalence. Specifically, ρ is in the [0.007, 0.988] range, with mean 0.23, median 0.15 and standard deviation 0.19.

6.2 Analysis of *FM* vs. ϕ with Different Values of ρ

First, we plot *FM* vs. ϕ when ρ is small. Figure 11 (which shows only the part of the $FM \times \phi$ plane in which we obtained pairs of values (FM, ϕ)) illustrates the situation when ρ is close to 0.05, namely when $\rho \in [0.025, 0.075]$. We select the data that correspond to a range, rather than to a specific value of ρ, because in the latter case we would end up selecting data from a single dataset. In the [0.025, 0.075] range, we have 2 datasets and 44 classifiers.

The yellow line has equation (11) with $\rho = 0.05$ (the mean value for these datasets).

FM and ϕ tend to provide practically equivalent information, regardless of σ, especially when $FM > 0.4$, and the relationship between *FM* and ϕ is well represented by (11). These results are practically relevant for application areas such as vulnerability prediction or defect prediction, in which low values of actual prevalence ρ can be found.

For higher values of ρ, the correspondence between *FM* and ϕ is less clear: Fig. 12 shows *FM* vs. ϕ when $\rho \in [0.74, 0.77]$ (like with Figs. 4 and 11, we only show the relevant part of the $FM \times \phi$ plane). We could not observe higher values of ρ, because no dataset with higher values of ρ supported enough classifiers. In the [0.74, 0.77] range, we have 16 classifiers from 3 datasets (`xerces-1.4`, with $\rho = 0.743$, `pbeans1`, with $\rho = 0.769$, and `velocity 1.4` with $\rho = 0.750$). The value of ρ used to draw the yellow line having equation (11) is the mean of the three datasets' ρ, i.e., 0.754.

Figure 12 shows that the different values of σ of different classifiers blur the relationship between *FM* and ϕ. Also, some predictors with high *FM* (close to 0.8) actually have a rather poor value of ϕ (around 0.2). Namely, we have a model that features $FM = 0.77$ and $\phi = 0.23$. This is coherent with (12), (13), and (14), according to which ϕ is expected to be in the $[-0.22, 0.54]$ range, when $\rho = 0.754$ and $FM = 0.77$.

In practice, it is apparent that, with high values of ρ, *FM* can be deceiving, showing high values that correspond to rather low ϕ.

6.3 On the Threats to Validity of the Demonstration Study

Even though our results are of an analytical nature, let us here explore the possible threats to validity that would derive from an empirical study like the demonstration study that we described.

Construct validity. Our demonstration study is about analyzing the relationships between two specific variables, i.e., ϕ and *FM*, so there is no real threat to construct validity. Instead,

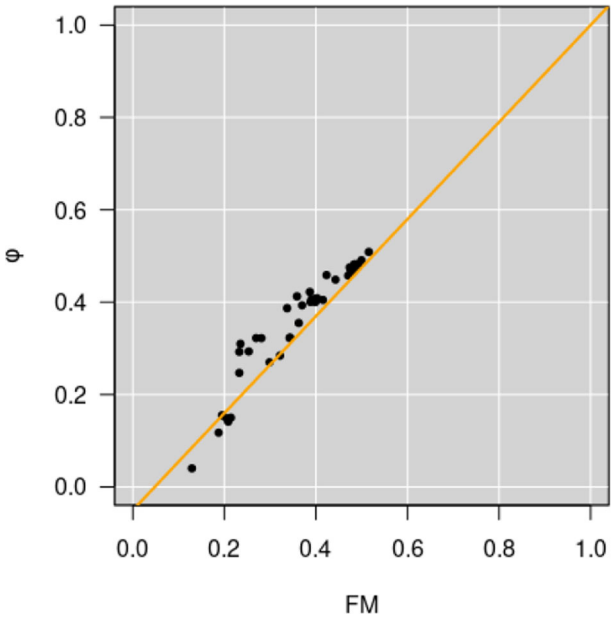

Fig. 11 *FM* vs. ϕ of defect classifiers for datasets with $\rho =\in [0.025, 0.075]$

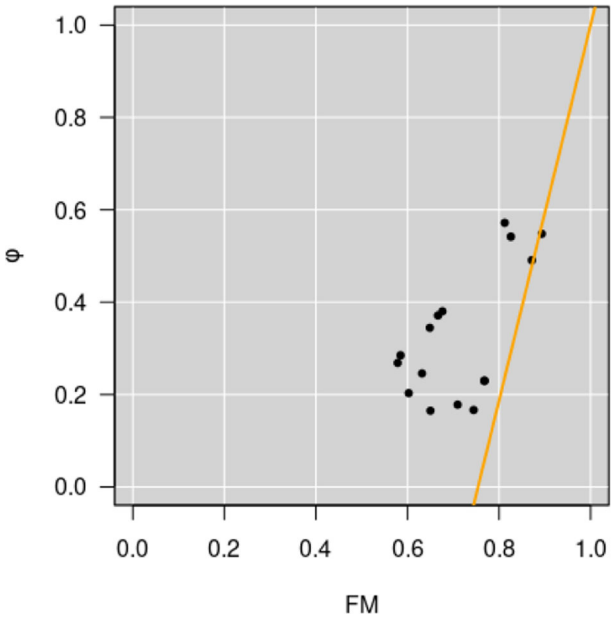

Fig. 12 *FM* vs. ϕ of defect classifiers for datasets with $\rho \in [0.74, 0.77]$

the debate is not over on whether these two variables adequately represent the overall performance of a classifier, even though *FM* has been heavily criticized in the last few years (Hernández-Orallo et al. 2012; Powers 2011; Sokolova and Lapalme 2009; Luque et al. 2019). Our analytical results provide researchers and practitioners with more information to make a more informed decision on the one that they would like to use.

Internal Validity. Our goal was of a descriptive kind, i.e., we wanted to show how ϕ varied for any given value of *FM* depending on ρ. We did not look for possible associations or correlations between them. This would likely be the goal of an empirical study, by using statistical or machine-learning techniques. An association/correlation could be found or not depending on the specific sample. However, if ρ and σ were included as additional independent variables, a statistical or machine-learning technique may indicate perfect correlation in all cases, even though this is not guaranteed, because of the nature of the technique used. For instance, suppose that a linear model that combines FM, ρ, and σ were used to estimate ϕ. Since the relationship between these variables is not linear, even having all the information needed to determine ϕ would not suffice. At any rate, even if a technique were able to find a perfect correlation, this would simply be an empirical way of finding the relationship that we analytically describe in Formula (10). In addition, the empirical approach would only provide strong evidence about perfect correlation, but not certainty.

External Validity. Like with any empirical study, we took a sample of possible subjects (the software projects) and we showed results about it. Thus, in an empirical study like the ones we show here, it is very possible that the results have limited external validity. In our demonstration study, we took projects from different application domains, of different sizes and with different prevalence values, so the results may be applicable to a fairly large set of projects. However, the analytical results are applicable to all projects and are valid beyond software defect prediction and Software Engineering.

7 Revisiting Previous Empirical Studies

We here show how our analytical study can be used to reinterpret previous defect prediction analyses that reported results via *FM* (and possibly other performance metrics, like *PPV* and *TPR*), but did not report ϕ values.

7.1 Case 1

Li et al. used Binary Logistic Regression (BLR), Naive Bayes (NB), Decision Trees (DT), CoForest (CF), and ACoForest (ACF) to build defect classifiers. They performed within-project defect predictions, training predictors with data from a variable number of modules from the software system that was the object of predictions (Li et al. 2012).

As an example of the outcomes of the study by Li et al., Table 2 (taken from Table 5 in Li et al. (2012)) shows the values of *FM* for the classifiers built with CF, BLR, NB, and DT. For each row, i.e., for each dataset, the highest *FM* value is in bold.

Li et al. conclude that "*It can be easily observed from the table that CoForest achieves the best performance among the compared methods except that on SWT NaiveBayes performs the best.*"

Of the considered datasets, all but one have $\rho \geq 0.3$. Based on the considerations illustrated in Section 5.3, conclusions based on *FM* alone should not be trusted when ρ is that high. Since Li et al. reported the values of *PPV* and *TPR* in Tables 8–13 of their paper, we

Table 2 *FM* of CoForest and the compared methods in predicting defects when only 10% modules are sampled, from Table 5 in Li et al. (2012)

Project	CF	BLR	NB	DT
JDT.Core	**0.73**	0.63	0.68	0.70
SWT	0.57	0.45	**0.64**	0.54
ECLIPSE 2.0	**0.57**	0.53	0.44	0.52
ECLIPSE 3.0	**0.74**	0.66	0.62	0.70
XALAN	**0.60**	0.57	0.55	0.58
LUCENE	**0.69**	0.65	0.64	0.67

could compute ϕ via (7). The results are in Table 3, where the highest ϕ of each row is in bold.

Table 3 shows clearly that 1) the performance of the CF classifier is unacceptably low when quantified via ϕ, with $\phi \leq 0.23$ for all datasets, 2) NB classifiers have the best ϕ for all datasets, and 3) NB classifiers are the only ones with acceptable performance, featuring $\phi \geq 0.3$ in 4 datasets out of 6.

This case demonstrates that considering *FM* without taking ρ into consideration is risky, as it can easily lead to untrustworthy conclusions.

However, for papers that published not only the values of *FM*, but also those of ρ and *TPR* or *PPV* it is possible to derive reliable indications based on ϕ. For instance, the conclusions by Li et al. concerning ACF appear reliable, according to our computation of ϕ (not reported here).

7.2 Case 2

Deng et al. addressed cross-project defect prediction via a method that adopts a better abstract syntax tree node granularity and proposes and uses multi-kernel transfer convolutional neural networks (Deng et al. 2020).

They evaluated their approach on 110 cross-project defect prediction tasks formed by 11 open-source projects. As an example of their evaluations, we report in Table 4 the *FM* values from Table 7 in Deng et al. (2020) concerning the proposed method MK-TCNN-mix and the ρ of the projects used to evaluate method MK-TCNN-mix (from Table 3 in (Deng et al. 2020)).

Based on the ρ and *FM* columns of Table 4, we computed the range to which ϕ must belong, via (12), (13), and (14). It turns out that only for the `Xerces` dataset and, to some

Table 3 Performance, expressed via both *FM* and ϕ, of CoForest and the compared methods in predicting defects when only 10% modules are sampled

		CF		BLR		NB		DT	
Project	ρ	*FM*	ϕ	*FM*	ϕ	*FM*	ϕ	*FM*	ϕ
JDT.Core	0.535	0.73	0.21	0.63	0.03	0.68	**0.41**	0.70	0.10
SWT	0.247	0.57	0.23	0.45	0.10	0.64	**0.51**	0.54	0.20
ECLIPSE 2.0	0.388	0.57	0.13	0.53	0.04	0.44	**0.25**	0.52	0.06
ECLIPSE 3.0	0.628	0.74	0.12	0.66	0.00	0.62	**0.25**	0.70	0.03
XALAN	0.464	0.60	0.12	0.57	0.03	0.55	**0.35**	0.58	0.06
LUCENE	0.597	0.69	0.09	0.65	0.01	0.64	**0.30**	0.67	0.03

Table 4 Possible values of ϕ for Deng et alii's data

Project	ρ	FM	$\phi_{\min}$	$\phi_{\max}$
Camel	0.201	0.343	0.07	0.41
Forrest	0.063	0.151	0.07	0.28
Ivy	0.114	0.275	0.16	0.38
Jedit	0.135	0.322	0.19	0.41
Log4J	0.959	0.672	− 0.19	0.20
Lucene	0.616	0.638	− 0.33	0.50
Poi	0.653	0.668	− 0.31	0.51
Synapse	0.336	0.516	0.17	0.51
Velocity	0.664	0.519	− 0.48	0.39
Xalan	0.469	0.693	0.32	0.61
Xerces	0.153	0.638	0.57	0.61

extent, for the `Xalan` dataset, the performance is surely good. For all the other datasets, ϕ belongs to too large ranges to allow for reliable conclusions (in 4 cases ϕ might even indicate perverse performance).

The *FM* values in Table 4 were used by Deng et al. to draw conclusions concerning the proposed method's performance; however, as Table 4 clearly shows, no reliable conclusion (i.e, neither in favor nor against Deng et alii's proposal) is supported.

In conclusion, the paper by Deng et al. shows that *FM*, even with ρ, does not let readers appreciate the actual performance of classifiers. This is because *FM* is a reliable metric only for small values of ρ (see Fig. 6). Take for instance the result on project `Velocity` in Table 4: the *FM* obtained (0.519) could correspond to $\phi = 0$, indicating that the proposed method MK-TCNN-mix is equivalent to random estimation. Unfortunately, Deng et al. do not provide in the paper additional data that can be used to compute ϕ more precisely than done in Table 4; hence, solving the doubts concerning the validity of Deng et al's conclusions is not possible.

7.3 Case 3

In paper "Slope-based fault-proneness thresholds for software engineering measures" (Morasca and Lavazza 2016), we also used *FM* to evaluate classifications. Specifically, we proposed a method to set thresholds for defect estimation based on the slope of Binary Logistic Regression (BLR) and Probit Regression (PBR) functions (Morasca and Lavazza 2016). The performance of the classifiers built with the proposed method was evaluated via an empirical study that used data from several projects from the SEACRAFT repository, including project `berek`, which has $n = 43$ software modules, of which $AP = 16$ defective, so $\rho = \frac{16}{43} \simeq 0.372$. Performance was quantified and reported via FM and TPR. $\rho \simeq 0.372$ is too large a value to assure that a reliable value of ϕ can be derived from FM alone. Nonetheless, we can derive the value of ϕ for all the models presented in Morasca and Lavazza (2016), by means of the following procedure:

1. Derive *PPV* from *FM* and *TPR*:

$$PPV = \frac{FM \cdot TPR}{2TPR - FM}$$

2. Compute $TP = AP \cdot TPR$; then, compute TN as follows:

$$TN = AN - AP \cdot TPR \frac{1 - PPV}{PPV}$$

(these equations can be derived via simple transformations of the definitions of PPV and TPR in Formula (1)).
3. Compute FP, FN, EP and EN based on their definitions (Table 1).
4. Compute ϕ based on its definition (Formula (5)).

For instance, the model that uses RFC to predict faultiness via BLR is reported to have $FM = 0.88$ and $TPR = 0.94$. Thus, $PPV = 0.83$, $TP = 15$, $TN = 24$, $FP = 3$, $FN = 1$, $EP = 18$, $EN = 25$. Finally, $\phi = 0.81$. In this case, we get a quite high value for ϕ, which confirms the good performance reported by $FM = 0.88$.

Noticeably, in an extended version of the paper, being aware of the limitations of FM, we reported ϕ in addition to FM (Morasca and Lavazza 2017). The values of ϕ that can be computed as shown above match exactly the values of ϕ that were computed based on the confusion matrices and were reported in Morasca and Lavazza (2017).

8 Related Work

Yao and Shepperd investigated the relationship between FM and ϕ (Yao and Shepperd 2020; 2021) from an empirical point of view. Via a systematic literature review, they identified 38 refereed primary studies in which FM and ϕ were used, to evaluate the effects of using FM instead of ϕ. In this sense, the work by Yao and Shepperd provides a solid background and a strong justification for our analytical study. In fact, they found that around 22% of all results found in the 38 primary studies would be reversed if ϕ is selected as a performance metric instead of FM. Based on the empirical results and a comparison of the properties of ϕ and FM, they strongly recommend that FM should no longer be used and that ϕ should be used instead.

In a simulation analysis, Chicco and Jurman (2020) computed FM and ϕ for all confusion matrices with $n = 500$ and showed the results in a scatterplot. A similar scatterplot is in Figure 5, which shows that, for a given value of FM, there is a wide range of possible values of ϕ, in general. Our work (see Section 5.5) provides the theoretical explanation for their simulation results. Chicco and Jurman also illustrated via representative numerical cases how the imbalance between the actual negatives and actual positives affects the ability of FM and ϕ to assess classifier performance. When ρ is quite low, they find that both FM and ϕ provide the same kind of evaluation. Our study (see Section 5.4) provides the general mathematical bases and explanations for their numerical results.

Bowes et al. (2012) observed that a variety of different performance metric are used in empirical studies. Since these measures are not directly comparable, comparing different results is often difficult. Also, decision-makers may be interested in different measures than those reported in a specific study. Therefore, Bowes et al. proposed an approach to reconstruct a frequency confusion matrix based on the values of the performance measures provided in empirical studies. The proposal by Bowes et al. can therefore be used to compute FM or ϕ when an empirical study does not provide them, but provides instead a suitable set of metrics, as specified in Table 5 in Bowes et al. (2012).

9 Conclusions and Future Work

9.1 Findings

Different performance metrics provide different evaluations and rankings for a set of classifiers. We focused on two performance metrics that have been extensively used in the Empirical Software Engineering literature, namely *FM* and ϕ.

Previous research found that imbalanced data can significantly affect performance metrics. However, to the best of our knowledge, this is the first time that the role of imbalance (via prevalence ρ) in the relationship between ϕ and *FM* is made explicit.

Our study provides the mathematical explanations for some phenomena that have been detected empirically or via simulations. Specifically, we show the mathematical relationships between *FM* and ϕ, and how they are influenced by the values of actual and estimated prevalence. Though *FM* and ϕ are based on different formulas, we show the conditions under which both *FM* and ϕ provide the same ranking between two classifiers. Specifically, it appears that *FM* and ϕ tend to agree on the ranking more when the actual prevalence ρ is low, as is the case for several datasets used for software defect prediction. In addition, we review existing analyses about the validity and usefulness of *FM* and ϕ, and add two more observations. The mathematical relationships between *FM* and ϕ can be used also to get a more rigorous and sound interpretation of the results published in papers that used *FM* alone.

9.2 Recommendations

Based on the considerations reported through the paper, we can formulate a few recommendations about the performance metrics to be used to evaluate classifiers.

It is not advisable to evaluate the performance of a classifier based exclusively on *FM*. Also, if using *FM*, the value of ρ needs to be specified, at least to know if a classifier performs better than the random one. Unfortunately, this practice has not been followed in many cases, which led to many questionable evaluations (Yao and Shepperd 2021).

At any rate, we recommend that, even when the value of ϕ is reported, FM should not be used without also providing the value of ϕ, to have at least a more complete evaluation of a classifier. For instance, take the data in Table 4: for dataset `Synapse`, we have $FM = 0.516$ and $\rho = 0.336$. *FM* is sufficiently larger than ρ to suggest that the classifier performs better than the random one. However, our mathematical results show that in this case ϕ is between 0.17 (which would be rather bad) and 0.51 (which would be fairly good). Thus, in this case, the knowledge of *FM* and ρ is not sufficient to establish how good the binary classifier is.

Unlike *FM*, ϕ takes into account all of the cells in a confusion matrix. Thus, ϕ seems to be more adequate to be used as an overall metric for the performance of a classifier. It is true that *FM* and ϕ are likely to provide the same ranking when the actual prevalence is small. However, one may as well use ϕ without using *FM*.

In addition, it would be useful that, whenever possible, authors of scientific articles provide the entire confusion matrices for the classifiers. Based on the confusion matrices, any performance metric of interest to decision-makers and researchers can be computed.

9.3 Dealing with Other Performance Metrics

In this paper, we investigated *FM* and ϕ. As already mentioned, many performance metrics have been proposed. Of these, several are used in common practice. Therefore, it could be useful to explore the relationship among these metrics (including their relationships with *FM* and ϕ).

To this end, we note that in a previous paper (Morasca and Lavazza 2020) we provided the mathematical basis for comparing some performance metrics. Some comparisons have also been performed already, although not systematically. For instance, in Morasca and Lavazza (2020) and Lavazza and Morasca (2022) we showed how ϕ, *FM* and Youden's J can be expressed in terms of *TPR* and *FPR* (i.e, the axis of the ROC space) and ρ. The systematic investigation of additional relationships is part of our research agenda, from a mathematical and an empirical point of view.

In this respect, an important topic that we plan to investigate further is the impact of data imbalance on the indications provided by the various performance metrics, some of which, like Youden's J, do not suffer from imbalance effects while others appear to be largely influenced by imbalance.

Appendix A: Comparing the Variations of *FM* Depending on *FP* and *FN*

Formula (6) shows that *FM* can be written as follows

$$FM = 2\,\frac{AP - FN}{2\,AP - FN + FP} \tag{18}$$

Suppose that we would like to compare the performance of *CM* against the performance of another confusion matrix CM_Δ defined by "difference" from *CM*, as below

CM

TN	FN
FP	TP

CM_Δ

$TN - \Delta_N$	$FN + \Delta_P$
$FP + \Delta_N$	$TP - \Delta_P$

Δ_P and Δ_N are the variations on the numbers of false negatives and false positives with respect to *CM*. We here study how FM_Δ varies depending on the values of Δ_P and Δ_N.

Let us first deal with a few trivial cases:

$$\Delta_P = \Delta_N = 0 \Rightarrow FM_\Delta = FM$$
$$\Delta_P \cdot \Delta_N \neq 0 \wedge \Delta_P \leq 0 \wedge \Delta_N \leq 0 \Rightarrow FM_\Delta > FM$$
$$\Delta_P \cdot \Delta_N \neq 0 \wedge \Delta_P \geq 0 \wedge \Delta_N \geq 0 \Rightarrow FM_\Delta < FM$$

So, let us now assume that Δ_P and Δ_N are nonzero and have opposite signs.

First, take $\Delta_P = A > 0$ and $\Delta_N = -R < 0$, to obtain a new confusion matrix CM' in which, in comparison to CM, R units are removed from *FP* at the price of adding A units to *FN*. The value of FM' for CM' is

$$FM' = 2\,\frac{AP - FN - A}{2\,AP - FN + FP - A - R} \tag{19}$$

Via mathematical computations, we obtain

$$FM' > FM \Leftrightarrow \frac{R}{A} > \frac{AP + FP}{AP - FN} \tag{20}$$

It is easy to show that

$$\frac{AP + FP}{AP - FN} = \frac{2}{FM} - 1 \tag{21}$$

Considering that $\frac{1}{FM} \geq 1$, it is also $\frac{2}{FM} \geq 2$, hence $\frac{2}{FM} - 1 \geq 1$, therefore

$$FM' > FM \Leftrightarrow \frac{R}{A} > \frac{AP + FP}{AP - FN} = \frac{2}{FM} - 1 \geq 1 \tag{22}$$

Formula (22) shows that, to have $FM' > FM$, the number R of units removed from FN must be at least as large as the number of units added to FP.

Conversely, let us now be take $\Delta_P = -R < 0$ and $\Delta_N = A > 0$, i.e., we have a new confusion matrix CM'' in which, in comparison to CM, R units are removed from FN while A units are added to FP. The value of FM' for this CM'' is

$$FM'' = 2\,\frac{AP - FN + R}{2\,AP - FN + FP + A + R} \tag{23}$$

Via mathematical computations, we obtain

$$FM'' > FM \Leftrightarrow \frac{A}{R} < \frac{AP + FP}{AP - FN} = \frac{2}{FM} - 1 \tag{24}$$

We now compare FM'' against FM', i.e., we check whether removing R units from FN and adding A units to FP (like in CM') is more advantageous than removing R units from FP and adding A units to FN (like in CM''). Mathematical computations show that

$$FM'' > FM' \Leftrightarrow FP + FN > R - A \tag{25}$$

The rightmost inequality in Formula (25) is always satisfied, since $\min\{FP, FN\} \geq R$ and $A > 0$. So, given a classifier cl whose performance is represented by CM, if there is an alternative classifier cl_x that removes R units form FN and adds A units to FP, cl_x is preferable to cl if $\frac{R}{A}$ satisfies inequality (20). In addition, cl_x is preferable to another classifier cl_y that removes R units form FP and adds A units to FN.

It can also be shown that $FM'' > FM'$ also when exactly one between A and R is zero.

Summarizing, reducing or increasing FN by some amount has more impact on FM than does reducing or increasing FP by the same amount.

Appendix B: Defining ϕ in Terms of *PPV* and *TPR*

Starting from Formula (5), which defines ϕ, we apply a few mathematical computations, as follows.

$$\begin{aligned}
\phi &= \frac{TP \cdot TN - FP \cdot FN}{\sqrt{AP \cdot AN \cdot EP \cdot EN}} = \frac{TP\,(AN - FP) - FP\,(AP - TP)}{\sqrt{AP \cdot AN \cdot EP \cdot EN}} \\
&= \frac{AN \cdot TP - TP \cdot FP - AP \cdot FP + TP \cdot FP}{\sqrt{AP \cdot AN \cdot EP \cdot EN}} = \frac{AN \cdot TP - AP \cdot FP}{\sqrt{AP \cdot AN \cdot EP \cdot EN}} \\
&= \frac{AN \cdot TP + AP \cdot TP - AP \cdot FP - AP \cdot TP}{\sqrt{AP \cdot AN \cdot EP \cdot EN}} = \frac{n \cdot TP - AP \cdot EP}{\sqrt{AP \cdot AN \cdot EP \cdot EN}} \\
&= \frac{n \cdot TP - \rho n \cdot EP}{\sqrt{\rho n\,(1-\rho) n\,EP \cdot EN}} = \frac{1}{\sqrt{\rho(1-\rho)}}\,\frac{TP - \rho EP}{\sqrt{EP \cdot EN}}
\end{aligned} \tag{26}$$

By dividing both numerator and denominator by EP, we have

$$\begin{aligned}\phi &= \frac{1}{\sqrt{\rho(1-\rho)}} \frac{\frac{TP}{EP} - \rho}{\sqrt{\frac{EP}{EP}\left(\frac{n-EP}{EP}\right)}} = \frac{1}{\sqrt{\rho(1-\rho)}} \frac{PPV - \rho}{\sqrt{\frac{n}{AP}\frac{AP}{TP}\frac{TP}{EP} - 1}} \\ &= \frac{1}{\sqrt{\rho(1-\rho)}} \frac{PPV - \rho}{\sqrt{\frac{PPV}{\rho\, TPR} - 1}} = \frac{1}{\sqrt{1-\rho}} \frac{\sqrt{TPR}(PPV - \rho)}{\sqrt{PPV - \rho\, TPR}} \end{aligned} \tag{27}$$

We now show the possible values of PPV and TPR, given ρ.

$$\begin{aligned}\rho = \frac{AP}{n} = \frac{AP}{AP + FP + TN} \leq \frac{AP}{AP + FP} = \frac{AP}{AP + EP - TP} \\ = \frac{TP \cdot AP}{TP(AP + EP - TP)} = \frac{\frac{TP}{EP}}{\frac{TP}{EP} + \frac{TP}{AP} - \frac{TP^2}{EP \cdot AP}} = \\ = \frac{PPV}{PPV + TPR - PPV \cdot TPR} = constr_\rho(PPV, TPR)\end{aligned} \tag{28}$$

It is immediate to note that $constr_\rho(PPV, TPR)$ in (28) is 1 if and only if $PPV = 1$.

For completeness, let us now study the special case in which $PPV = TPR = TP = 0$. We have

$$\rho = \frac{FN}{AN + FN} \tag{29}$$

which can take any value between the minimum 0 and the supremum 1.

It can be shown that, given PPV and TPR, ϕ is a monotonically decreasing function of ρ, except when $PPV = TPR = 1$. Based on Formula (27), the first derivative of ϕ with respect to ρ is

$$\frac{d\phi}{d\rho} = \frac{(PPV + TPR - 2PPV \cdot TPR)\rho + PPV(PPV + TPR - 2)}{2(PPV - (PPV + TPR)\rho + TPR\rho^2)^{\frac{3}{2}}} \tag{30}$$

The denominator of the right-hand fraction in Formula (30) is always positive, as it is the cube of the denominator of the right-hand fraction in Formula (27). Thus, the sign of the derivative is the same as the sign of the numerator of the right-hand fraction in Formula (30). The numerator is a linear function of ρ. The value of the numerator for $\rho = 0$ is $PPV(PPV + TPR - 2)$, which is a negative value unless $PPV = TPR = 1$ (in which case $\phi = 1$ for all values of ρ). Thus, the numerator is negative for all values in the interval $\rho \in [0, \overline{\rho}(PPV, TPR))$, where

$$\overline{\rho}(PPV, TPR) = \frac{PPV(2 - PPV - TPR)}{PPV + TPR - 2\, PPV \cdot TPR} \tag{31}$$

We now prove that $constr_\rho(PPV, TPR) \leq \overline{\rho}(PPV, TPR)$, so ϕ is a decreasing function for all values of ρ. To prove that $constr_\rho(PPV, TPR) \leq \overline{\rho}(PPV, TPR)$, i.e.,

$$\frac{PPV}{PPV + TPR - PPV \cdot TPR} \leq \frac{PPV(2 - PPV - TPR)}{PPV + TPR - 2\, PPV \cdot TPR} \tag{32}$$

consider that the numerator in the left side fraction is never greater than the numerator in the right side fraction, and, at the same time the denominator in the left side fraction is never

smaller than the denominator in the right side fraction. Thus, $constr_\rho(PPV, TPR) \leq \overline{\rho}(PPV, TPR)$.

Therefore, ϕ is minimum when $\rho = constr_\rho(PPV, TPR)$, with value

$$\phi_{\min} = -\sqrt{1 - PPV}\sqrt{1 - TPR} \tag{33}$$

It is $\phi_{min} \leq 0$, since the numerator in (33) is $(1 - TPR)(PPV - 1) \leq 0$.

Finally, for completeness, the supremum of ϕ is

$$\phi_{sup} = \sqrt{PPV \cdot TPR} \tag{34}$$

based on Formula (27).

Appendix C: *FM* is the *Harmonic* Mean of *TPR* and *PPV*

A characteristic of *FM* that is not fully explained (Yao and Shepperd 2021) is that it is defined as the *harmonic* mean of *TPR* and *PPV*. The often suggested rationale is that the harmonic mean is a "low" mean: it is never higher than the geometric mean, which, in turn, is never higher than the arithmetic mean (which would probably be a more easily interpretable choice (Yao and Shepperd 2021)). It is immediate to show that the harmonic mean of two values is never greater than twice the lesser of the two, i.e., it never gets "too far" from the lower value. As an example, if $PPV = 0.04$, $FM \leq 0.08$ no matter how high the value of *TPR*.

It is not clear, however, what is to be gained by choosing a "low" mean instead of a "high" mean. If it were all that important to have low values, one could then use $\min\{PPV, TPR\}$ as a performance metric. What matters more, instead, is that choosing a different mean implies choosing a different ordering among performances and, therefore, a different preference ranking among classifiers. For instance, a classifier cl_f with $TPR_f = 0.2$ and $PPV_f = 0.8$ would rank better than a classifier cl_g with $TPR_g = 0.3$ and $PPV_g = 0.5$ if taking the geometric mean of *TPR* and *PPV* as a performance metric (in fact, $\sqrt{0.2\ 0.8} = 0.4 > \sqrt{0.3\ 0.5} = 0.387$), but worse with the harmonic mean, i.e., *FM* (in fact, $\frac{2\ 0.8\ 0.2}{0.8+0.2} = 0.32 < \frac{2\ 0.3\ 0.5}{0.3+0.5} = 0.375$).

As we show in Section 4, ϕ is not defined as a mean (of any type) of *TPR* and *PPV*, but, rather, as an effect size metric.

Appendix D: The Relationship between ϕ and *FM*

We take the definition formula for *FM*

$$FM = \frac{2\,TP}{AP + EP} \tag{35}$$

we solve it for *TP*, thereby obtaining

$$TP = \frac{AP + EP}{2} FM \tag{36}$$

i.e., *TP* can be seen as a function of *FM* and *EP*. We replace the value of *TP* in the rightmost member of Formula (26) and carry out a few computations

$$\begin{aligned}\phi &= \frac{1}{\sqrt{\rho(1-\rho)}} \frac{\frac{AP+EP}{2} FM - \rho EP}{\sqrt{EP \cdot EN}} \\ &= \frac{1}{2\sqrt{\rho(1-\rho)}} \frac{(\rho n + EP)FM - 2\,\rho\, EP}{\sqrt{EP\,(n-EP)}} \qquad (37) \\ &= \frac{1}{2\sqrt{\rho(1-\rho)}} \frac{(\rho+\sigma)FM - 2\,\rho\,\sigma}{\sqrt{\sigma\,(1-\sigma)}} \qquad (38)\end{aligned}$$

Formula (38) shows how ϕ depends on *FM*, *EP*, and ρ. It is the basis for studying the relationship between ϕ and *FM*, which we do in Section 5.

Appendix E: Variation Interval of ϕ as a Function of *FM*

The range of values of σ in Formula (10) is constrained because of the natural constraints on the cells of the confusion matrix, as we now detail. For illustration purposes, we set the constraints in terms of *EP*, which can always be immediately translated in terms of $\sigma = \frac{EP}{n}$.

$$TP \leq AP \Leftrightarrow \frac{AP+EP}{2} FM \leq AP \Leftrightarrow EP \leq \frac{2-FM}{FM} AP \qquad (39)$$

$$TP \leq EP \Leftrightarrow \frac{AP+EP}{2} FM \leq EP \Leftrightarrow \frac{FM}{2-FM} AP \leq EP \qquad (40)$$

$$FP \leq AN \Leftrightarrow EP - \frac{AP+EP}{2} FM \leq AN \Leftrightarrow EP \leq \frac{2AN + AP \cdot FM}{2-FM} \qquad (41)$$

It can be shown that all other natural constraints (e.g., $FN \leq EN$, or $EP \leq n$) are satisfied when the above constraints are satisfied and all cells of the confusion matrix are nonnegative.

The rightmost inequality in Formula (40) shows that there is a lower bound on the values of EP, i.e., $\frac{FM}{2-FM} AP$. As for the upper bound, we have that $EP \leq \frac{2-FM}{FM} AP$ (per Formula (39)), and $EP \leq \frac{2AN+AP \cdot FM}{2-FM}$ (*per* Formula (41)), and $EP \leq n$. It can be immediately proven that the lower bound $\frac{FM}{2-FM}$ of Formula (40) is never greater than the upper bounds of Formulas (39) and (41).

We need to find under what conditions these upper bounds are stricter than the others. Let us first compare the upper bounds of Formulas (39) and (41). We have

$$\frac{2AN + AP \cdot FM}{2-FM} \leq \frac{2-FM}{FM} AP \Leftrightarrow \frac{FM}{2-FM} \leq \rho \Leftrightarrow FM \leq \frac{2\rho}{1+\rho} \qquad (42)$$

Formula (42) shows that, for *FM* up to $\frac{2\rho}{1+\rho}$, the upper bound of Formula (41) is stricter or as strict as the upper bound of Formula (39), while the reverse is true for *FM* equal to $\frac{2\rho}{1+\rho}$ or higher.

We now compute the interval $[\phi_{\min}(FM;\rho), \phi_{\max}(FM;\rho)]$ of values of ϕ for each value of *FM*. Based on Formula (37), $\phi_{\min}(FM;\rho)$ and $\phi_{\max}(FM;\rho)$ are, respectively, the minimum and the maximum value of ϕ as *EP* varies in its interval. ϕ is a continuous function of

EP, so we use the first derivative of ϕ with respect to EP to identify minima and maxima. The derivative is

$$\phi' = \frac{n}{4\sqrt{\rho(1-\rho)}} \frac{(FM - 2\rho + 2\rho FM)EP - \rho FM\, n}{(EP\,(n - EP))^{\frac{3}{2}}} \tag{43}$$

Clearly,

$$\phi' \geq 0 \Leftrightarrow (FM - 2\rho + 2\rho FM)EP \geq \rho\ FM\ n \tag{44}$$

So, we need to study for which values of EP the inequality in Formula (44) holds when EP varies between its lower and upper bound. Now, the lower bound for EP is always $\frac{FM}{2-FM}AP$. As we showed with Formula (42), depending on whether $FM \leq \frac{2\rho}{1+\rho}$ or $FM \geq \frac{2\rho}{1+\rho}$, two cases are possible for the upper bound, which we now investigate separately.

E.1 *FM* up to $\frac{2\rho}{1+\rho}$

When $FM \leq \frac{2\rho}{1+\rho}$, we have $EP \in \left[\frac{FM}{2-FM}AP, \frac{2AN+AP\cdot FM}{2-FM}\right]$.

When $FM - 2\rho + 2\rho FM \leq 0$, inequality (44) never holds, so the first derivative ϕ' is always negative and ϕ is a decreasing function of EP. This inequality $FM - 2\rho + 2\rho FM \leq 0$ can be rewritten as $FM \leq \frac{2\rho}{1+2\rho}$, so this is what happens for all values of FM up to $\frac{2\rho}{1+2\rho}$.

Let us now suppose that $FM - 2\rho + 2\rho FM > 0$, i.e., that $\frac{2\rho}{1+2\rho} < FM \leq \frac{2\rho}{1+\rho}$. In this interval of FM, inequality (44) can therefore be rewritten as $EP \geq \frac{\rho FM\, n}{FM - 2\rho + 2\rho FM}$, which shows for which values of EP the first derivative of ϕ is nonnegative. However, these values of EP must also be below the upper bound $\frac{2AN+AP\cdot FM}{2-FM}$. We can show that this is not the case, i.e., $\frac{\rho FM\, n}{FM-2\rho+2\rho FM} \geq \frac{2AN+AP\cdot FM}{2-FM}$ when $\frac{2\rho}{1+2\rho} < FM \leq \frac{2\rho}{1+\rho}$.

The mathematical computations are as follows

$$\begin{aligned} \frac{\rho FM\, n}{FM - 2\rho + 2\rho FM} &\geq \frac{2AN + AP \cdot FM}{2 - FM} \\ &\Leftrightarrow (\rho + \rho^2)FM^2 + (1 - 3\rho^2)FM - 2\rho(1-\rho) \leq 0 \end{aligned} \tag{45}$$

The second-degree polynomial of FM in Formula (45) has roots $FM_1 = -\frac{1-\rho}{\rho}$ and $FM_2 = \frac{2\rho}{1+\rho}$, so it is less than or equal to 0 between those two roots. Root FM_1 is negative, while root FM_2 coincides with the upper bound of the interval of FM we are currently investigating, so we can conclude that $\phi' < 0$ when $FM \leq \frac{2\rho}{1+\rho}$.

This implies that, for all $FM \leq \frac{2\rho}{1+\rho}$, $\phi_{min}(FM)$ is achieved with the highest value possible for EP, i.e., $EP = \frac{2AN+AP\cdot FM}{2-FM}$, and $\phi_{max}(FM)$ is achieved with the lowest value possible for EP, i.e., $EP = \frac{FM}{2-FM}AP$. Thus, we have

$$\phi_{\min}(FM) = -\sqrt{1 - \frac{FM}{2\rho - 2\rho^2 + \rho^2 FM}} \tag{46}$$

$$\phi_{\max}(FM) = \sqrt{\frac{FM(1-\rho)}{2 - (1+\rho)FM}} \tag{47}$$

E.2 *FM* Greater than or Equal to $\frac{2\rho}{1+\rho}$

When $FM \geq \frac{2\rho}{1+\rho}$, we have $EP \in \left[\frac{FM}{2-FM}AP, \frac{2-FM}{FM}AP\right]$.

The coefficient $(FM - 2\rho + 2\rho FM)\ EP$ in inequality (44) is greater than 0, so the first derivative is negative for $EP < \frac{\rho FM\ n}{FM-2\rho+2\rho FM}$ and, conversely, positive for $EP > \frac{\rho FM\ n}{FM-2\rho+2\rho FM}$. It can be shown that the value $EP = \frac{\rho FM\ n}{FM-2\rho+2\rho FM}$ in which the first derivative is null belongs to the interval $\left[\frac{FM}{2-FM}AP, \frac{2-FM}{FM}AP\right]$ of admissible values for EP.

Thus, the minimum value $\phi_{\min}$ is obtained for $EP = \frac{\rho FM\ n}{FM-2\rho+2\rho FM}$.

The maximum value ϕ_{max} is therefore obtained at the lower bound $\frac{FM}{2-FM}AP$ or at the upper bound $\frac{2-FM}{FM}AP$ of the interval of EP. Via computations, we have

$$\phi_{lb}(FM) = \sqrt{1-\rho}\sqrt{\frac{FM}{2-FM-\rho FM}} \tag{48}$$

$$\phi_{ub}(FM) = \frac{1}{\sqrt{1-\rho}}\sqrt{\frac{FM-2\rho+\rho FM}{2-FM}} \tag{49}$$

Furthermore, it can be shown that $\phi_{lb}(FM) \geq \phi_{ub}(FM)$, so $\phi_{\max} = \phi_{lb}(FM)$. Summarizing, we have

$$\phi_{\min}(FM) = \sqrt{\frac{FM}{1-\rho}}\sqrt{FM-2\rho+\rho FM} \tag{50}$$

$$\phi_{\max}(FM) = \sqrt{\frac{FM(1-\rho)}{2-(1+\rho)FM}} \tag{51}$$

Note $\phi_{\max}(FM)$ is defined by the same function in Formulas (47) and (51).

Appendix F: Preserving Classifiers' Rankings with ϕ and *FM* for Datasets with Different Actual Prevalence

Formula (52) shows the inequality that must hold when using two classifiers cl_a and cl_b on two datasets with different actual prevalence values ρ_a and ρ_b.

$$FM_b > \frac{\rho_b + \sqrt{\frac{2\rho_b^2(1-FM_a)+(1-\rho_a)\ FM_a}{2-(1+\rho_a)\ FM_a}}}{1+\rho_b} \tag{52}$$

Formula (52) reduces to Formula (15) when $\rho_a = \rho_b$. It can be shown that, for every value of FM_a, there always exists FM_b such that there is complete separation between the variation intervals, regardless of the values of ρ_a and ρ_b. The right-hand side of the inequality in Formula (52) is an increasing function of ρ_b and a decreasing function of ρ_a: the constraint on FM_b becomes stricter when ρ_b increases and easier to satisfy when ρ_a decreases.

Figure 13 shows the minimum value of FM_b as a function of FM_a, for a few pairs $\langle \rho_a, \rho_b \rangle$.

Appendix G: Variation Intervals of ϕ for All Values of ρ

Let us first compute $\phi_{\min}(FM)$, the minimum value of $\phi_{\min}(FM; \rho)$ for a given value of FM. The two functions in Formulas (12) and (13) are used, depending on whether $FM \leq$

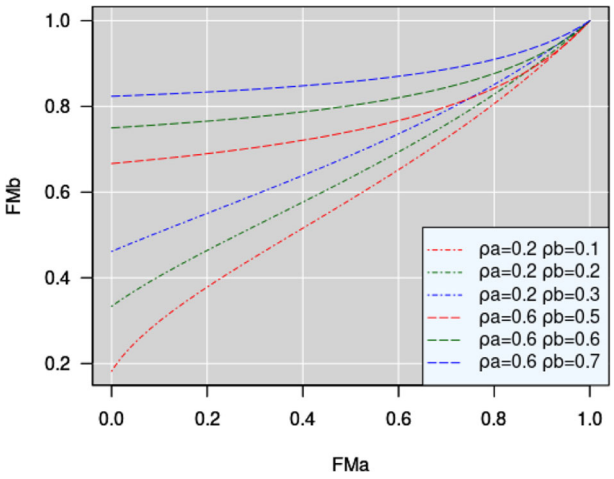

Fig. 13 Interval separation: FM_b as a function of FM_a for different $\langle \rho_a, \rho_b \rangle$ pairs

$\frac{2\rho}{1+\rho}$ or $FM \geq \frac{2\rho}{1+\rho}$. The function in Formula (12) is never positive, while the one in Formula (13) is nonnegative, so, the value of ρ that minimizes $\phi_{min}(FM; \rho)$ for a given value of FM is the one that minimizes the function in Formula (12). Thus, let us take the first derivative of the function in the square root in the right-hand side of the equality in Formula (12)

$$\frac{\partial}{\partial \rho}\left(1 - \frac{FM}{2\rho - 2\rho^2 + \rho^2 FM}\right) = \frac{2FM(1-(2-FM)\rho)}{(2\rho - 2\rho^2 + \rho^2 FM)^2} \tag{53}$$

This derivative is positive for $\rho < \frac{1}{2-FM}$, null in $\rho = \frac{1}{2-FM}$, and negative for $\rho > \frac{1}{2-FM}$, so the function in the square root has a maximum in $\rho = \frac{1}{2-FM}$, and $\phi_{\min}(FM; \rho)$ has a minimum there. By setting $\rho = \frac{1}{2-FM}$ in Formula (12), we obtain $\phi_{\min}(FM)$ as in Formula (16).

Let us now compute $\phi_{\max}(FM)$, the maximum value of $\phi_{\max}(FM; \rho)$ for a given value of FM. To this end, let us compute the first derivative of the term in the square root sign of Formula (14)

$$\frac{\partial \phi^2_{\max}(FM; \rho)}{\partial \rho} = \frac{2FM(FM-1)}{(2-(1+\rho)FM)^2} \leq 0 \tag{54}$$

For all values of $FM < 1$, this derivative is negative, so $\phi_{\max}(FM; \rho)$ attains its maximum when $\rho = 0$. Thus, we obtain $\phi_{\max}(FM)$ as in Formula (17). For completeness, $\phi_{\max}(FM; \rho) = 1$ when $FM = 1$, which coincides with the value obtained in Formula (17) for $FM = 1$ anyway.

Appendix H: Mathematical Details for Section 5.4

Here we demonstrate inequality (15), skipping a few mathematical passages for conciseness.

It can be proved that two intervals $[\phi_{\min}(FM_a; \rho), \phi_{\max}(FM_a; \rho)]$ and $[\phi_{\min}(FM_b; \rho), \phi_{\max}(FM_b; \rho)]$ with $FM_a < FM_b$ are completely separated if and only if $\phi_{\max}(FM_a;$

$\rho) < \phi_{\min}(FM_b; \rho)$. Since $\phi_{\max}(FM_a; \rho) \geq 0$, it must $FM_b \geq \frac{2\rho}{1+\rho}$, since $\phi_{\min}(FM_b; \rho) < 0$ for $FM_b < \frac{2\rho}{1+\rho}$ and $\phi_{\min}(FM_b; \rho) \geq 0$ for $FM_b \geq \frac{2\rho}{1+\rho}$. Therefore, we have

$$FM_b > FM_a \Leftrightarrow \phi_{\min}(FM_b; \rho) > \phi_{\max}(FM_a; \rho) \Leftrightarrow$$

$$\sqrt{\frac{FM_b}{1-\rho}}\sqrt{FM_b - 2\rho + \rho FM_b} > \sqrt{\frac{FM_a\,(1-\rho)}{2-(1+\rho)\,FM_a}} \Leftrightarrow$$

$$(1+\rho)FM_b^2 - 2\rho FM_b > \frac{FM_a\,(1-\rho)^2}{2-(1+\rho)\,FM_a} \Leftrightarrow$$

$$(1+\rho)FM_b^2 - 2\rho FM_b - \frac{FM_a\,(1-\rho)^2}{2-(1+\rho)\,FM_a} > 0 \tag{55}$$

This second-degree inequality is satisfied for values of FM_b outside an interval $[FM_{b1}, FM_{b2}]$, where FM_{b1} and FM_{b2} are the two roots of the left hand side of the inequality. Since the three coefficients of the left hand side of the inequality are, respectively, positive, negative, and negative, $FM_{b1} < 0$ and $FM_{b2} > 0$. Thus, the second-degree inequality must be satisfied for $FM_b > FM_{b2}$. The value of FM_{b2} is

$$FM_{b2} = \frac{2\rho + \sqrt{4\rho^2 + 4(1+\rho)\frac{(1-\rho)^2 FM_a}{2-(1+\rho)FM_a}}}{2(1+\rho)} = \frac{\rho + \sqrt{\frac{2\rho^2(1-FM_a)+(1-\rho)\,FM_a}{2-(1+\rho)\,FM_a}}}{1+\rho} \tag{56}$$

where the last equality is obtained via algebraic manipulations. Thus, inequality (15) holds.

Appendix I: On the Construction of Defect Predictors

The original datasets provide data at the class level. For every class of every project it is known:

- if the class is defective or not;
- a set of code measures.

Using these data we built Binary Logistic Regression models of fault-proneness (i.e., the probability that a class is faulty) based on code measures. Using these data we built scoring classifiers with Binary Logistic Regression that estimate the probability that a class is defective based on code measures. These scoring classifiers were then used to build binary classifiers by using the actual positive ratio $\rho = \frac{AP}{n}$ as the threshold: classes whose estimated probability to be defective is greater or equal to ρ are classified defective (i.e., positive) and the others non-defective (i.e., negative).

A few descriptive statistics of the analyzed datasets are given in Table 5.

Table 5 Descriptive statistics of the analyzed datasets

	n (num. modules)	AP	ρ	LoC
min	18	4	0.007	1910
max	23014	2738	0.988	3816692
mean	1823	240	0.286	249283
median	283	65	0.198	56220
st.dev.	4469	516	0.223	674951

A file describing the 837 models from 70 datasets that we analyzed is available from http://www.dista.uninsubria.it/supplemental_material/PhiFM/binclass.csv

The file specifies also the value of ρ for each dataset, the confusion matrix of each binary classifier, and a set of performance metrics.

Funding Open access funding provided by Università degli Studi dell'Insubria within the CRUI-CARE Agreement.

Declarations

Conflict of interest The authors have no conflict of interest to declare.

References

The SEACRAFT repository of empirical software engineering data. https://zenodo.org/communities/seacraft (2017)

Bowes D, Hall T, Gray D (2012) Comparing the performance of fault prediction models which report multiple performance measures: recomputing the confusion matrix. In: Proceedings of the 8th international conference on predictive models in software engineering, pp 109–118

Bowes D, Hall T, Petrić J (2018) Software defect prediction: do different classifiers find the same defects? Softw Qual J 26(2):525–552

Cauchy A (1821) Cours d'analyse de l'école royale polytéchnique, Vol. I. Analyse analyse. International Centre for Mechanical Sciences. Debure

Chicco D, Jurman G (2020) The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC genomics 21(1):1–13

Cohen J (1960) A coefficient of agreement for nominal scales. Educ Psychol Meas 20(1):37–46. https://doi.org/10.1177/001316446002000104

Cohen J (1988) Statistical power analysis for the behavioral sciences lawrence earlbaum associates. Routledge, New York

Delgado R, Tibau XA (2019) Why Cohen's Kappa should be avoided as performance measure in classification. PloS one 14(9):e0222916

Deng J, Lu L, Qiu S, Ou Y (2020) A suitable AST node granularity and multi-kernel transfer convolutional neural network for cross-project defect prediction. IEEE Access 8:66647–66661

Dias Canedo E, Cordeiro Mendes B (2020) Software requirements classification using machine learning algorithms. Entropy 22(9):1057

Gray D, Bowes D, Davey N, Sun Y, Christianson B (2011) The misuse of the NASA metrics data program data sets for automated software defect prediction. In: 15th annual conference on evaluation & assessment in software engineering (EASE 2011), pp 96–103

Hall T, Beecham S, Bowes D, Gray D, Counsell S (2011) A systematic literature review on fault prediction performance in software engineering. IEEE Trans Softw Eng 38(6):1276–1304

Hernández-Orallo J., Flach PA, Ferri C (2012) A unified view of performance metrics: translating threshold choice into expected classification loss. J Mach Learn Res 13:2813–2869. http://dl.acm.org/citation.cfm?id=2503332

Jureczko M, Madeyski L (2010) Towards identifying software project clusters with regard to defect prediction. In: Proceedings of the 6th international conference on predictive models in software engineering, pp 1–10

Lavazza L, Morasca S (2022) Considerations on the region of interest in the ROC space. Stat Methods Med Res 31(3):419–437
Li M, Zhang H, Wu R, Zhou ZH (2012) Sample-based software defect prediction with active and semi-supervised learning. Autom Softw Eng 19(2):201–230
Luque A, Carrasco A, Martín A, de Las Heras A (2019) The impact of class imbalance in classification performance metrics based on the binary confusion matrix. Pattern Recogn 91:216–231
Matthews BW (1975) Comparison of the predicted and observed secondary structure of t4 phage lysozyme. Biochimica et Biophysica Acta (BBA)-Protein Structure 405(2):442–451
Menzies T, Di Stefano JS (2004) How good is your blind spot sampling policy. In: Eighth IEEE international symposium on high assurance systems engineering, 2004. Proceedings. IEEE, pp 129–138
Morasca S, Lavazza L (2016) Slope-based fault-proneness thresholds for software engineering measures. In: Proceedings of the 20th international conference on evaluation and assessment in software engineering, pp 1–10
Morasca S, Lavazza L (2017) Risk-averse slope-based thresholds: Definition and empirical evaluation. Information & Software Technology 89:37–63. https://doi.org/10.1016/j.infsof.2017.03.005
Morasca S, Lavazza L (2020) On the assessment of software defect prediction models via ROC curves. Empir Softw Eng 25(5):3977–4019
Pierri F, Piccardi C, Ceri S (2020) A multi-layer approach to disinformation detection in us and italian news spreading on twitter. EPJ Data Science 9(1):35
Powers DM (2011) Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation
Scaranti GF, Carvalho LF, Barbon S, Proença ML (2020) Artificial immune systems and fuzzy logic to detect flooding attacks in software-defined networks. IEEE Access 8:100172–100184
Serafini P (1985) Mathematics of multi objective optimization. International Centre for Mechanical Sciences. Springer
Singh PK, Agarwal D, Gupta A (2015) A systematic review on software defect prediction. In: 2015 2nd international conference on computing for sustainable global development (INDIACom). IEEE, pp 1793–1797
Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. Information processing & management 45(4):427–437
Sonbol R, Rebdawi G, Ghneim N (2020) Towards a semantic representation for functional software requirements. In: 2020 IEEE seventh international workshop on artificial intelligence for requirements engineering (AIRE). IEEE, pp 1–8
Song Q, Guo Y, Shepperd M (2019) A comprehensive investigation of the role of imbalanced learning for software defect prediction. IEEE Trans. Software Eng. 45(12):1253–1269
van Rijsbergen CJ (1979) Information retrieval. Butterworth
Yao J, Shepperd M (2020) Assessing software defection prediction performance: Why using the Matthews correlation coefficient matters. In: Proceedings of the evaluation and assessment in software engineering, pp 120–129
Yao J, Shepperd M (2021) The impact of using biased performance metrics on software defect prediction research. Inf Softw Technol 139:106664
Zhang F, Keivanloo I, Zou Y (2017) Data transformation in cross-project defect prediction. Empir Softw Eng 22(6):3186–3218

Affiliations

Luigi Lavazza[1] · **Sandro Morasca**[1]

Sandro Morasca
sandro.morasca@uninsubria.it

[1] Dipartimento di Scienze Teoriche e Applicate, Università degli Studi dell'Insubria, Varese, Italy