Egyptian Informatics Journal 31 (2025) 100711

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Full length article

Analysis of classification metric behaviour under class imbalance

Jean-Pierre van Zyl [a],*, Andries Petrus Engelbrecht [a,b,c]

[a] Division of Computer Science, Stellenbosch University, Stellenbosch, 7600, Western Cape, South Africa
[b] Department of Industrial Engineering, Stellenbosch University, Stellenbosch, 7600, Western Cape, South Africa
[c] GUST Engineering and Applied Innovation Research Center, Gulf University for Science and Technology, Mubarak Al-Abdullah, 7207, Kuwait

ARTICLE INFO

Keywords:
Class imbalance
Classification
Evaluation metric
Performance evaluation
Sensitivity analysis

ABSTRACT

Class imbalance is the phenomenon defined as skewed target variable distributions in a dataset. In other words class imbalance occurs when a dataset has an unequal proportion of target variables assigned to the instances in the dataset. Although the level of class imbalance is simply an inherent property of a dataset, highly skewed class imbalances cause misleading performance evaluations of a classification model to be reported by certain evaluation metrics. This paper reviews the history of existing performance evaluation metrics for classification, and uses a normalisation process to create new variations of these existing metrics which are more robust to class imbalance. Conclusions about the performance of the analysed metrics are drawn by performing the first extensive global sensitivity analysis of classification metrics. A statistical analysis technique, *i.e.* analysis of variance, is used to analyse the robustness to class imbalance of the existing metrics and the proposed metrics. This paper finds that most performance evaluation metrics for classification problems are highly sensitive to class imbalance, while the newly proposed alternative metrics tend to be more robust to class imbalance.

1. Introduction

One of the key aspects in the application of any classification model to a problem is the definition of a suitable evaluation function. The evaluation function of a classification model or rule set inducer creates the reference that the model uses to evaluate how well or how poorly it performs. The metrics used to define the evaluation function of a predictive model determines whether the end-user is able to make a correct assessment about the suitability of the model. In addition, it is incredibly important to provide an accurate representation of how well a model performs so that a satisfactory objective function can be defined for the implemented training algorithm. The metrics used to define the objective function of a training algorithm influences the quality of the model that the training process produces. Unfortunately, many commonly used classification metrics are sensitive to the level of CI and provide an inaccurate portrayal of a model when there is not a uniform distribution of the number of instances in each class [1]. The sensitivity of metrics to CI causes problems for both the evaluation of the performance of models, and for training models with meta-heuristics when these metrics are used in objective functions.

A considerable complication encountered when working with skewed (imbalanced) datasets is that very often the most important or interesting class is the minority class [2]. One of the first modern examples of studies on the issue of deceptive classifier evaluation caused by problematic CI was in the use of computer vision systems to analyse satellite imagery [3]. The system was trained to find oil spills on the ocean surface, but was sensitive to the user-requirement regarding the false positive and false negative trade-offs [3]. Machine learning (ML) applications in the medical field have proved useful for cancer diagnosis; however, a positive diagnosis is most often the exception in patients. This can be quite problematic considering the implications of an incorrect diagnosis of the (positive) minority class [4]. Since the advent of "big data", manual fraud investigations have become increasingly impractical which has required a more rapid switch to automated systems [5]. The increased use of automated systems has escalated human dependence on reliable classifier evaluation methods in even the most unbalanced datasets. Fraud detection systems have to investigate hundreds of thousands of transaction records, of which the vast majority are authentic. This creates an extremely skewed dataset which often results in poor performance using traditional classification methods [6]. Furthermore, the internet has increasingly become a breeding ground for malicious individuals to attack vulnerable systems on the web. This has created the need for intrusion detection systems which also represent a league of problems where the target class is in the extreme minority [7].

* Corresponding author.
E-mail address: 20706413@sun.ac.za (J. van Zyl).

https://doi.org/10.1016/j.eij.2025.100711
Received 5 August 2024; Received in revised form 2 April 2025; Accepted 29 May 2025
Available online 1 July 2025

2. Scope and contributions

To the knowledge of the authors, this is one of the most complete reviews of metrics for the evaluation of classification performance with respect to CI. Literature which analyses the behaviour of classification metrics under CI does exist and provides insight into the shortcomings of classification metrics [8,9]. However, current studies do not provide insight into how to improve classification metrics. The analysis performed in this paper is both model-agnostic and dataset-free, which makes the results generally applicable to all areas of artificial intelligence and fields of application. The three main contributions of this paper are as follows:

- This paper reviews the historical origins of binary classification metrics popular in literature today. There are publications, with contemporary use of performance metrics, which do not cite the proposers of the implemented metrics. This paper provides a comprehensive review of the history of these metrics, and gives the necessary credit to the original authors.
- This paper proposes a normalisation technique to make metrics more robust to CI. CI results in large differences in the ranges of the entries of the confusion matrix. Because of the range differences, a technique which ensures that components of metrics have equal contributions is proposed. The normalisation technique results in new metrics which are more robust to CI. There are two cases where the normalisation technique results in the rediscovery of existing metrics; for these cases, proper credit is given to the original authors.
- This paper performs the first global sensitivity analysis (SA) of existing and proposed metrics, which provides insight into how sensitive metrics are to CI. A method based on analysis of variance is used to perform the global SA.

The remainder of this paper is structured as follows: Section 3 provides the required background information drawn from relevant literature, after which Section 4 describes the normalisation process used to create new metrics. Section 5 reviews binary classification performance evaluation metrics popular in this field followed by Section 6 which proposes new variations of certain existing metrics. The empirical process used to evaluate the discussed metrics is outlined in Section 7, and the results of the evaluations are presented and discussed in Section 8.

3. Background

This section presents the concepts relevant to the rest of this paper. These concepts include the foundations relevant to performance evaluation metrics for classification systems which are discussed in Section 3.1. This section starts with a background on the problem wrought by CI, followed by a subsection on the definitions of the parts which constitute classification metrics. Finally, the statistical technique (*i.e.* analysis of variance with Sobol' sequences) used to analyse the popular performance metrics is described.

3.1. Imbalanced classification

Datasets can be described by a few salient features like the number of instances in the dataset, the number of input features, the data type of each feature, and the distribution of the target feature values of the dataset. This so-called distribution of the target feature values is a very important aspect to take into account when evaluating the performance of a classification algorithm on a dataset. To illustrate, imagine a case where there is a dataset D that contains 99 instances of class A for every 1 instance of class B. By using the popular baseline comparison approach of always predicting the majority class (class A), an accuracy of 99% is achieved.

This "untrustworthiness" of performance evaluation metrics as seen in the example above is a well-known phenomenon, as demonstrated in the following studies from literature. Swets reviewed the shortcomings of accuracy on a range of real-world problem domains which include weather forecasting, information retrieval, aptitude testing, medical imaging, materials testing, and polygraph lie detection [10]. Brzezinski et al. used barycentric visualisation to analyse ten different properties of metrics, and showed the importance of considering the effect of the CI when selecting a metric to use [11]. Amadzadeh and Angryk created a contingency space to determine how metric behaviour changes with CI, the study showed that four popular metrics are highly sensitive to CI [12]. Brzezinski et al. used a binning technique to create histogram visualisations of the probability mass functions of different metrics under CI and found that data-streams with variations in CI are particularly susceptible to result in misleading metric interpretations [13]. Luque et al. created and visualised a metric bias function to determine for which metrics CI introduces bias, and used clustering to quantify the similarities of the biases of different metrics [14].

Popular approaches in ML literature which deal with CIs typically involve either a form of artificial manipulation of the dataset, using ML algorithms which are more robust to imbalance datasets, or redefining the problem as an anomaly detection problem [2,15]. Artificial manipulation of the dataset materialises in the form of either reducing the number of instances in the majority class (undersampling), increasing the number of instances in the minority class (oversampling), or both [16,17]. An example of reducing the majority class is to sample a random subset of the instances of the majority class to reduce its number of instances to be similar to that of the minority class. This random undersampling technique is a straightforward way to reduce CI, but can cause the majority class to become misrepresented, because important instances may be discarded. Alternatively, more advanced methods can be used to reduce the majority class; for example, using a cluster-based stratified undersampling technique [18]. Further, minority oversampling is an approach used when training data is scarce and the user cannot afford to reduce the size of the majority class. This can easily be achieved by duplicating instances of the minority class and adding them to the training set. More advanced methods have also been proposed which create new artificial instances of the minority class. These techniques (*e.g.* the synthetic minority over-sampling technique (SMOTE) [19]) create additional variety among the minority class training instances, but may also unintentionally introduce false concepts that a classifier might learn. Zhang et al. investigated the effect of combining data resampling and feature selection in different orders, and the effect this has on imbalanced learning [20].

Many of the modern day approaches to dealing with class imbalance involve some form of dataset manipulation. For example, Dube and Verster studied the effects of nine different levels of class imbalance (between 1:9 and 1:1) on ten different ML models using five different performance metrics [21]. Dube and Verster found various ML models to be sensitive to class imbalance, with varying results between different evaluation metrics. For example the stochastic gradient descent classifier (SGDC) had an F_1 score of 0.0335 for 1:9 imbalance and a score of 0.5223 for a 1:1 imbalance; similarly SGDC had a Matthew's correlation coefficient (MCC) score of 0.1024 for 1:9 and 0.4548 for 1:1 imbalance.

De la Cruz Huayanay et al. compared the effectiveness of 12 different metrics in determining the best ML model for binary classification problems [22]. In order to evaluate the effectiveness of different metrics in selecting the optimal ML model, De la Cruz Huayanay et al. simulated imbalanced data using the Power-Cauchy distribution and applied the Kolmogorov-Simrnov test between the known Power-Cauchy curves and the metric curves. De la Cruz Huayanay et al. claimed that MCC, g-mean, and Cohen's kappa yielded metric curves closest to the expected Power-Cauchy curve.

Another recent study of classification metrics by Siblini et al. [23] proposed the use of the likelihood ratio [24] as the optimal scoring function, *i.e.*

$$s_t(x) := \frac{\mathbb{P}_t(x|y=1)}{\mathbb{P}_t(x|y=0)}. \tag{1}$$

With reference to the likelihood ratio, Siblini et al. proposed that a desirable performance metric satisfies the property that the likelihood of x, *i.e.* $\mathbb{P}_t(x|y)$, does not change as the prior, *i.e.* $\mathbb{P}_t(y)$, varies. Siblini et al. proposed metrics which satisfy this property, referred to as "calibrated metrics", by incorporating positive class ratio, $\pi = \frac{\sum_{i=1}^{N} y_i}{N}$ (*i.e.* the percentage of instances in the target class) into the metric calculation. The calibrated metrics from Siblini et al. showed that the prior class distribution of the dataset plays an important role in the use of performance evaluation metrics.

Gu et al. outlined the shortcomings of a few popular performance metrics on imbalanced datasets [25]. Specifically, Gu et al. discussed the issues which afflict accuracy, precision, recall, and the ROC curve. Accuracy does not discriminate between the type of error made; misclassifications of the target class and non-target class are both penalised equally, which is not always ideal. When using precision and recall to evaluate a ML model, the result does not take the correctly classified non-target instances into account which can lead to misleading interpretations of model performance. As a result, situations can arise where a classifier has vastly different behaviours on different datasets with varying numbers of non-target instances, even though the precision and recall remains the same. Gu et al. also critiqued the ROC curve for not taking precision into account, since it renders the metric blind to cases when there are a significant number of misclassified non-target instances.

3.2. Fundamentals of measures and metrics

What would be described as the qualities of a good metric varies depending on the application of the metric [26]. When a metric is used to evaluate the performance of a model after training has been completed, the metric should quantify the generalisation capabilities of the model. However, if a metric is to be used in the training process of a model, then using a metric suited to assess the generalisation performance of a completely trained model is short-sighted, because the metric may cause the training process to stagnate in a suboptimal local minimum. Instead, the metric should be able to capture the future classification potential of the model-in-training at a given time-step during the training process [26,27].

When constructing a metric to quantify how well a classification model performed, there are a few constituent measures which are important building blocks of all available classification performance metrics. In a given dataset of T instances, there are P instances from the target class (the feature value that the model tries to predict) and N instances that are not in the target class ($N = T - P$). Of the instances which a model has classified as belonging to the target class there are correctly classified and incorrectly classified instances. Instances correctly classified as belonging to the target class are referred to as true positives; the number of true positives is denoted as t_p. Instances incorrectly classified as the target class are called false positives; the number of false positives is denoted as f_p. Similarly, the number of instances correctly identified as not belonging to the target class (these instances are true negatives) are denoted by t_n, while the number of instances incorrectly associated as the non-target class (the false negatives) is denoted as f_n. The sum of the instances classified as belonging to the positive class (irrespective of correctness) is denoted by P' and the sum of the instances classified as the non-target class is N'.

These classification measures and the relationships between them are summarised in the confusion matrix for binary classification problems, as shown in Table 1.

Table 1
Example of a general confusion matrix.

		Ground truth		Total
		Positive	Negative	
Predicted value	Positive	t_p	f_p	P'
	Negative	f_n	t_n	N'
	Total	P	N	T

In order to be thorough, and to avoid any possible confusion that may be caused due to notation inconsistencies in existing literature, attention is brought to the following notation:

- The number of instances in the positive class is $P = t_p + f_n$.
- The number of instances in the negative class is $N = t_n + f_p$.
- The number of instances classified as in the positive class is $P' = t_p + f_p$.
- The number of instances classified as in the negative class is $N' = f_n + t_n$.
- The confusion matrix has four degrees of freedom. However, this can be reduced to two degrees of freedom as follows:
  - because P is constant, f_n can be omitted and $(P - t_p)$ used, and
  - because N is constant, t_n can be omitted and $(N - f_p)$ used.

Additionally, for the sake of brevity, it is often seen that metric functions are stylised as not having any input parameters. Hence, a metric which is a function of the number of true positives, true negatives, false positives, and false negatives, *i.e.* $f(t_p, t_n, f_p, f_n)$, is simply referred to as f.

Table 2 outlines simple metrics using the symbols defined above.

The nomenclature inconsistencies from existing literature are illustrated further by the entries in Table 2, where some entries have multiple names (*e.g.* TPR = sensitivity = recall). In order to clarify the confusion that can be caused by these naming inconsistencies, Canbek et al. defined a periodic table of performance instruments (PToPI) in [28]. The PToPI defines a hierarchy which outlines the relational structure of "performance instruments". Canbek et al. performed an exhaustive analysis and provided a full taxonomy, hence the term "performance instruments" is used as an umbrella-term for any formula which is used to evaluate how well a model performs. Categories of performance instruments defined by Canbek et al. include base measures, 1st level measures, 2nd level measures, 3rd level measures, base metrics, 1st level metrics, and 2nd level metrics.

This paper is concerned with binary classification metrics, and not the full spectrum of performance instruments. Hence, for brevity, a simplified naming schema is used which consists of the following:

- **Measures:** components of the confusion matrix. This category includes true positives, false positives, true negatives, false negatives, the number of positive instances, the number of negative instances, the total number of instances, the sum of instances classified as the target class and the sum of instances classified as part of the non-target class.
- **Rates:** normalised versions of entries in the confusion matrix (entries which have been divided by their maximum value). The category of rates includes the true positive rate, false positive rate, true negative rate, false negative rate, positive predictive value, negative predictive value, false discovery, rate and false omission rate.
- **Metrics:** more complex performance instruments constructed from measures and rates.

The above-mentioned classification schema is not exhaustive, and is simply provided to improve coherence. For a comprehensive system, the reader is referred to [28].

Table 2
Symbols used throughout this paper.

Symbol	Meaning	Equation
Φ_{t_p}	The true positive rate (TPR) is also referred to as sensitivity or recall. It is the number of instances correctly classified as the target class compared to the total number of instances in the target class.	$\frac{t_p}{P}$
Φ_{f_p}	The false positive rate (FRP), also known as the fall-out, is the ratio of the number of instances incorrectly identified as a part of the target class to the number of instances in the target class.	$\frac{f_p}{N}$
Φ_{t_n}	The true negative rate (TNR) contextualises the number of instances correctly classified as not belonging to the target class. This metric is also referred to as specificity or selectivity.	$\frac{t_n}{N}$
Φ_{f_n}	The false negative rate (FNR), or miss rate, weighs the number of instances incorrectly classified as not belonging to the target class against the number of instances in the target class.	$\frac{f_n}{P}$
ρ_{t_p}	The positive predictive value (PPV) is the proportion of true positives to the total number of instances classified as the target class. It is also referred to as precision.	$\frac{t_p}{P'}$
ρ_{t_n}	The negative predictive value (NPV) is the proportion of true negatives to the total number of instances attributed to not belonging to the target class.	$\frac{t_n}{N'}$
ρ_{f_p}	The false discovery rate (FDR) quantifies the number of type I errors made by a classifier because it proportions the number of false positives to the total number of instances assigned to the target class.	$\frac{f_p}{P'}$
ρ_{f_n}	The false omission rate (FOR) is the proportion of instances incorrectly assigned a non-target class value compared to the total number of instances assigned to the non-target class.	$\frac{f_n}{N'}$

3.3. Sobol' sensitivity analysis

SA is the process by which the effect of different input variables on a function output is quantified. Sobol pioneered sensitivity estimates for nonlinear mathematical models in 1993 and proved a theorem that an integrable function can be decomposed into the sum of different components for SA [29]. This approach, functional analysis of variance, is a technique used to quantify the effect that an input variable has on a mathematical function. Sobol investigated performing Monte Carlo simulations to determine sensitivity estimation with respect to a group of variables, as well as the effect of freezing unessential variables. Functional analysis of variance has since been expanded to include approaches based on sequence kernel association tests [30], functional linear models [31], and Bayesian non-parametric modelling [32]. This paper uses a popular open-source python implementation[1] for SA from [33,34]. The remainder of this section outlines the background information on the utilised SA method, referred to as S'SA.

In S'SA, points are sampled from the domain of valid variable inputs through a quasi-Monte Carlo technique, called Sobol' sequence (SS) sampling. The approach for sampling SSs is given in [35], with complexity improvements for the sampling process given in [36]. Enhancements of the quasi-randomness was proposed in [37], after which a stability study was performed on the sampling algorithm and is given in [38].

From the sampled SS, global sensitivity indices are calculated to perform analysis of variance (ANOVA); these sensitivity indices are used to estimate the influence of individual variables and subsets of variables on the model output. The calculation of Sobol' sensitivity indices are outlined in [35]. Sobol' sensitivity indices are categorised as first-order (S_1), which quantify the influence of individual input variables on the output, and second order (S_2), which quantify the influence of pairs of variables. Modifications to the first-order sensitivity calculations were presented in [39], with modifications for the second order sensitivity calculations given in [36]. Finally, error reducing improvements were incorporated in [40]. A brief overview of the calculations required to calculate the global sensitivity indices as presented by Sobol' [35] is provided below.

In order to perform ANOVA, Sobol' uses a unit interval ($I = [0, 1]$) to represent the input space as a d-dimensional unit hypercube (I^d). Consider an integrable function defined on I^d in the form

$$f(\boldsymbol{x}) = f_0 + \sum_{s=1}^{d} \sum_{i_1<\cdots<i_s}^{d} f_{i_1\ldots i_s}(x_{i_1}, \ldots, x_{i_s}), \tag{2}$$

where $1 \le i_1 < \cdots < i_s \le d$. Eq. (2) is called the ANOVA-representation of the function $f(x)$ if

$$\int_0^1 f_{i_1\ldots i_s}(x_{i_1}, \ldots, x_{i_s})\, dx_k \quad \text{for } k = i_1, \ldots, i_s. \tag{3}$$

Assuming that f is square integrable, and that each component of the decomposition of f is square integrable, squaring and integrating results in

$$\int_0^1 f^2 dx - f_0^2 = \sum_{s=1}^{d} \sum_{i_1<\cdots<i_s}^{d} \int_0^1 f_{i_1\ldots i_s}^2 dx_{i_1} \ldots dx_{i_s}. \tag{4}$$

From which the constants,

$$D = \int_0^1 f^2 dx - f_0^2, \tag{5}$$

and

$$D_{i_1\ldots i_s} = \sum_{s=1}^{d} \sum_{i_1<\cdots<i_s}^{d} \int_0^1 f_{i_1\ldots i_s}^2 dx_{i_1} \ldots dx_{i_s} \tag{6}$$

are defined. The constant D is the total variance of the output and $D_{i_1\ldots i_s}$ is the variance attributed to the subset of variables, $i_1 \ldots i_s$.

[1] https://github.com/salib/salib.

The global sensitivity index of a subset of variables, $i_1 \dots i_s$, is defined as

$$S_{i_1 \dots i_s} = \frac{D_{i_1 \dots i_s}}{D} \tag{7}$$

The sensitivity indices are used to calculate how much of the total variance of a function output is attributable to a specific variable or subset of variables. The main effect of an input variable x_j is given by the corresponding first-order sensitivity index S_{x_j}.

For this study on the effect of CI on metric performance, S'SA is used to determine how sensitive a given metric is to an input parameter (*i.e.*, t_p, f_p, t_n, f_n) under different levels of CI.

4. Metric normalisation process

The purpose of this paper is three-fold: to provide a comprehensive review on the origin of existing metrics, to determine the sensitivity of the metrics to CI, and to propose variations of the existing metrics which are more robust to CI.

This paper uses normalisation to create modified versions of existing metrics, in the hope that existing metrics can be made more robust to CI. The fact that components which represent the majority class come to dominate existing metrics, serves as motivation behind the normalisation process. The approach to create new metrics is to normalise the constituent parts of existing metrics. The normalisation of the components of a metric scales the output of each component of the metric to $[0, 1]$. Components with the same output range have the same influence on the final metric value, contrary to components which vastly different influences due to vastly different output ranges. The normalisation process followed is:

1. simplify the metric into elementary form such that it consists solely of measures,
2. normalise each measure in the metric, so that each measure becomes a rate,
3. scale the metric so that the output is in $[0, 1]$ and the optimum is at 1.

In order to normalise measures within a metric, a function $g : \mathcal{M} \to \mathcal{R}$ is defined, where $\mathcal{M}$ is the set of measures and $\mathcal{R}$ is the set of rates. The function, g, is defined as

$$g(m) = \begin{cases} \frac{m}{P} & \text{if } m \in \{t_p, f_n\}, \\ \frac{m}{N} & \text{if } m \in \{t_n, f_p\}. \end{cases} \tag{8}$$

Instances of other measures, *e.g.* P, N, P' and N', are implicitly handled by g since these measures can be reformulated in terms of t_p, f_p, t_n and f_n.

For example, to modify an existing metric, true positives (t_p) are divided by the total number of positive instances (P), which results in the true positive rate (Φ_{t_p}). Similarly, the false positives (f_p) are divided by the number of available negative instances (N), which results in the false positive rate (Φ_{f_p}). This normalisation process is applied to existing metrics from literature which are not already defined in terms of rates.

5. Existing performance metrics for classification

This section outlines the existing metrics currently popular in literature.

5.1. Accuracy

Accuracy is a ubiquitous metric in the modern ML community, and it is unfortunately not possible to pinpoint a single seminal paper which proposed the metric of classification accuracy. The idea of "percentage of correct instance" pre-dates ML and artificial intelligence in fields such as biology and meteorology. The use of "accuracy" in ML is present in the inaugural paper on *The Perceptron*, where Rosenblatt calculated the probability that the perceptron gives the correct response to a stimulus as P_r [41].

The accuracy metric as defined by Fürnkranz and Flach [27] accounts for both the correctly covered instances and incorrectly covered instances. Accuracy is an improvement over the basic strategy of simply trying to maximise the number of true positives or to minimise the number of false positives. By accounting for both objectives, the metric aims to find a solution that compromises well between classifying as many instances of the target class correctly, while not inadvertently also classifying instances from the non-target class as part of the target class.

The accuracy metric is defined as

$$A = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \tag{9}$$

5.2. Gilbert's ratio of verification

Gilbert's ratio of verification has its origins in the study of meteorology, with a full history of this metric summarised in [42]. The fundamental idea behind this metric has been in literature for over 100 years as first communicated in [43] as criticism of Finley's publication on the accuracy of tornado predictions [44]. Gilbert [43] outlined the fallacy behind the evaluation method that Finley used to quantify the success of tornado predictions and suggested an alternative approach called the "ratio of verification". Gilbert's ratio of verification was proposed with the symbol v and expressed as

$$v = \frac{c}{o + p - c} \tag{10}$$

where c is the number of verified predictions, o is the total number of occurrences, and p is the number of positive predictions.

Gilbert's ratio has been reproposed multiple times, with two notable proposals characterised by new names for the metric, *e.g.* threat score (TS) from [45] and critical success index (CSI) from [46]. Both the TS and the CSI are essentially v rewritten using t_p, f_p and f_n. However, both are more commonly used terms in the domain of ML evaluation metrics. The formula for these scores is

$$\text{TS} = \text{CSI} = \frac{t_p}{t_p + f_n + f_p} \tag{11}$$

The remainder of this paper uses the index as defined in Eq. (11), but uses the symbol v in homage to Gilbert as the original proposer.

5.3. Balanced accuracy

Five metrics, *i.e.* balanced accuracy, Pierce's I, Youden's J statistic, bookmaker's informedness, and area under the ROC curve, are inextricably linked; hence, these five metrics are presented together.

Pierce's I. Pierce, similarly to Gilbert, saw the error of Finley's method and proposed another alternative which is now referred to as Pierce's I [47]. Pierce originally proposed

$$I = \frac{(aa)}{(aa) + (ba)} - \frac{(ab)}{(ab) + (bb)}, \tag{12}$$

where $(aa) = t_p$, $(ab) = f_p$, $(ba) = f_n$, and $(bb) = t_n$.

Youden's J. Youden initially developed the J statistic to judge how well a diagnostic test performs [48]. Youden's proposed statistic is an early attempt at addressing the shortcomings of simpler metrics which do not account for both correctly classifying target instances and non-target instances. Reviews of the J statistic in medical applications have expressed concern that there is no mechanism to apply weights to either of the two components (Φ_{t_p}, Φ_{t_n}) to make one or the other more important [49]. Youden's J statistic is

$$J = \Phi_{t_p} + \Phi_{t_n} - 1. \tag{13}$$

Bookmaker's informedness. The bookmaker's informedness metric was proposed in [50] by Powers. The rationale of Powers was that a good way to estimate how suitable a classifier is, is to compare the classifier to how well a person betting on fair odds would perform against a random guesser. The formulation of bookmaker's informedness is the same as Eq. (13).

Area under the receiver operating characteristic curve. ROC curve analysis is a technique developed in the second world war to determine how successful military radars were at distinguishing between enemy aeroplanes and signal noise. The origin of ROC curves is slightly obscured due to the nature of the situation in which it was created. However, to the knowledge of the author, one of the earliest references to ROC curves was made by Peterson in a paper completed for the U.S. Army Signal Corps [51].

After its successful use in signal detection for military application, ROC curve analysis was applied to evaluate a new theory of visual detection in a psychological context by Tanner and Swets [52]. Since the application in [52], ROC analysis has been an integral tool for the medical diagnosis field [53–55]. However, it was not until Spackman's application of ROC curve analysis to ML problems in [56] that the technique started gaining popularity in the ML community.

A ROC curve is generated by plotting the Φ_{t_p} on the y-axis against the Φ_{f_p} on the x-axis for different threshold values of the classifier. These two-dimensional graphs depict the trade-off that a classifier has to make when assigning instances to the target class. The classification of all instances as the target class results in a perfect score for that class, but incorrectly classifies all non-target class instances. This gives the user the ability to measure the sensitivity of the classifier against the fall-out of the classifier.

A ROC graph has a range and domain of $[0, 1]$, with four salient (x, y) points on the graph. The following list outlines what points on the ROC curve in the vicinity of different x and y values (denoted as $\sim (x, y)$) indicate:

- $\sim (0, 0)$ represents a classifier which classifies all instances as part of the non-target class.
- $\sim (0, 1)$ represents a classifier a perfect classifier.
- $\sim (1, 0)$ represents a classifier which classifies all instances (of the target class and non-target class) incorrectly, *i.e.* all classifications are opposite to the ground truth.
- $\sim (1, 1)$ represents a classifier which classifies all instances as part of the target class.

Fig. 1 depicts a single point on the ROC curve.

The area under the curve (AUC) of the ROC curve can be estimated by decomposing the plot into one rectangle and two triangles. The formulas for the area of a square, $\dot{A}_{\square}$, and the area of a triangle, $\dot{A}_{\triangle}$, are defined as $\dot{A}_{\square} = l \cdot w$ (where l is the length of a side and w the width) and $\dot{A}_{\triangle} = \frac{1}{2} \cdot b \cdot h$ (where b is the base of the triangle and h is the hight). The total AUC is defined as the sum of the main rectangular body ($\square$) and the two triangles ($\triangle_1$ and $\triangle_2$), resulting in $\dot{A}_{ROC} = \dot{A}_{\square} + A_{\triangle_1} + A_{\triangle_2}$. The simplification from the area calculation to the arithmetic mean is shown below:

$$\begin{aligned} \dot{A}_{ROC} &= (\dot{A}_{\square} + \dot{A}_{\triangle 1} + \dot{A}_{\triangle 2}) \\ &= \left(l \cdot b + \frac{1}{2} \cdot b_1 \cdot h_1 + \frac{1}{2} \cdot b_2 \cdot h_2\right) \\ &= \left((1 - \Phi_{f_p}) \cdot \Phi_{t_p} + \frac{1}{2} \cdot \Phi_{f_p} \cdot \Phi_{t_p} + \frac{1}{2} \cdot (1 - \Phi_{f_p}) \cdot (1 - \Phi_{t_p})\right) \\ &= \frac{\Phi_{t_p} + \Phi_{t_n}}{2} \end{aligned} \tag{14}$$

Balanced accuracy. Brodersen et al. used a probabilistic view of performance evaluation to propose the balanced accuracy metric [57]. Balanced accuracy aims to provide generic safeguards against reporting an optimistic accuracy estimate, which can be caused by CI. A further motivation given for balanced accuracy is to fix the impossible statistical situation where confidence intervals of conventional accuracy can exceed 100% [57]. Balanced accuracy is given as

$$\text{BA} = \frac{\Phi_{t_p} + \Phi_{t_n}}{2}. \tag{15}$$

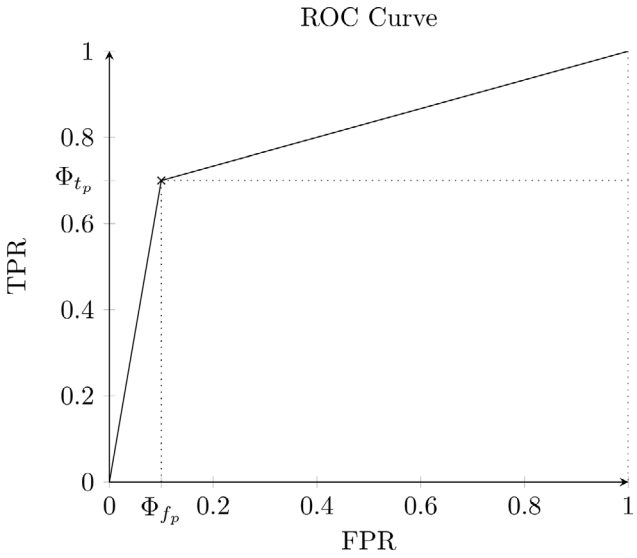


Fig. 1. Trigonometric properties of a single point ROC curve.

"Metric B". The five metrics discussed in this section are either equivalent to, or estimators of each other ($\Phi_{t_p} + \Phi_{t_n} - 1 \propto \frac{\Phi_{t_p} + \Phi_{t_n}}{2}$). It would be superfluous to analyse each individually, given that all metrics in this paper are modified to be in the range $[0, 1]$. Therefore, the following metric is used

$$B = \frac{\Phi_{t_p} + \Phi_{t_n}}{2}. \tag{16}$$

5.4. F-Measures

The F-measure is a family of metrics which weight the precision and recall of a classifier against each other. These metrics stem from work done by Van Rijsbergen [58] in the book *Information Retrieval*. Van Rijsbergen developed an effectiveness measure to balance the trade-off between the precision and recall of a search result from text. Initially, Van Rijsbergen's measure achieved this balance by using a parameter α as follows:

$$E = 1 - \frac{1}{\alpha(\frac{1}{\rho_{t_p}}) + (1 - \alpha)\frac{1}{\Phi_{t_p}}} \tag{17}$$

where ρ_{t_p} is the precision and Φ_{t_p} is the recall (as defined in Table 2). To facilitate interpretation of the function, Van Rijsbergen applied the transformation $\alpha = \frac{1}{\beta^2+1}$ to Eq. (17). This results in the general form of the F score as follows:

$$F_\beta = (1 + \beta^2)\frac{(\frac{t_p}{t_p+f_p}) \cdot (\frac{t_p}{t_p+f_n})}{\beta^2 \cdot (\frac{t_p}{t_p+f_p}) + (\frac{t_p}{t_p+f_n})} \tag{18}$$

Eq. (18) defines the popular metric F_1 which is used in this paper.

To define the F_1 metric, β is set to 1 in Eq. (18). This results in a function which calculates the harmonic mean between the precision and recall metrics of a classifier. Through simple algebraic manipulation, the F_1 score is transformed to consist of only measures as follows:

$$F_1 = \frac{2 \cdot t_p}{2 \cdot t_p + f_n + f_p} \tag{19}$$

5.5. Kappa

Cohen's Kappa (κ) is a similarity statistic developed by Cohen in 1960 for use in determining the overlap in identical values between variables (in the original article, these variables were the diagnoses given by various psychologists) [59]. Although Cohen's Kappa is the most popular version of this form of metric, it was not the first. An equivalent formulation of this metric was proposed 34 years prior by Heidke and was used to measure how well wind forecasts are made. Heidke's original metric was termed the Heidke skill score (HSS) [60]. However, Cohen's formulation and notation is currently used almost exclusively.

What makes Cohen's metric unique and significant, is that it takes into consideration the probability of a coincidental similarity between variables. Cohen proposed the Kappa metric to account for unrelated chance being interpreted as apparent causation.

Cohen's Kappa coefficient is defined as

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{20}$$

where p_o is the observed agreement between variables and p_e is the expected probability of agreement between variables due to chance. In the context of classification, the observed agreement is the accuracy of the classification ($p_o = \frac{t_p+t_n}{P+N}$). The expected probability of agreement is defined as the expected accuracy between two statistically independent observations, *i.e.* $p_e = \left(\frac{t_p+f_p}{P+N} \cdot \frac{t_p+f_n}{P+N}\right) + \left(\frac{t_n+f_p}{P+N} \cdot \frac{t_n+f_n}{P+N}\right)$. Substitution of the values for p_o and p_e and the simplification of the subsequent formula results in a more convenient form of the Kappa coefficient using base measures as follows:

$$\kappa = \frac{1}{2} \cdot \left(\frac{2 \cdot (t_p \cdot t_n - f_p \cdot f_n)}{(t_p + f_p) \cdot (f_p + t_n) + (t_p + f_n) \cdot (f_n + t_n)} + 1 \right). \tag{21}$$

Note that this form of Kappa is scaled to be in the range $[0, 1]$ for balanced data, but under extreme levels of CI the range of Kappa tends to $[0.5, 1.0]$.

5.6. Laplace estimate

Laplace developed the rule of succession as a probability estimator for seemingly certain events using the example of calculating the probability that the sun rises tomorrow [61]. The Laplace estimate is similar in form to the classical precision metric, but encourages more instances to be classified instead of encouraging a small number of perfectly classified instances. This is because with Laplace's estimate, a small number of classified instances equates to randomly guessing to which class an instance belongs [26]. When the number of instances classified tends to infinity, the Laplace estimator behaviour becomes identical to that of precision.

The Laplace estimator formula is given as

$$L = \frac{t_p + 1}{t_p + f_p + 2} \tag{22}$$

5.7. Matthew's correlation coefficient

MCC is a performance metric shown to be a good quantifier of the relationship between the predicted values of a classifier and the true values of the dataset [62]. Early versions of this metric were proposed by Yule and refined by Pearson as the ϕ (mean square contingency) coefficient [63,64].

However, the metric was later repopularised by Matthews [65], for whom this metric is named, to compare structural similarities in lysozymes in work in the biological sciences. This metric was then first used for ML applications by Baldi et al. [66] and has since been established in the field.

MCC uses all four possible input variables (*i.e.* t_p, f_p, t_n and f_n) and gives an indication of how well a pair of variables are correlated. When the performance of a classification or rule induction algorithm is evaluated, the two variables for which the correlation is measured is the true labels and the predicted labels.

The equation for the MCC metric is outlined as

$$\text{MCC} = \frac{1}{2} \cdot \left(\frac{t_p \cdot t_n - f_p \cdot f_n}{\sqrt{(t_p + f_p)(t_p + f_n)(t_n + f_p)(t_n + f_n)}} + 1 \right) \tag{23}$$

5.8. Markedness

Powers [67] proposed a metric called markedness (MK), named after the psychological and linguistic terms of condition and marker. A condition is an experimental outcome that is determined by indirect means and a predictor is the indicator that is used to determine the outcome.

The MK metric maps the classification performance of a model to the line $\rho_{t_p} + \rho_{t_n} = 1$, which represents the trade-off between the positive predictive value (ρ_{t_p}) and the negative predictive value (ρ_{t_n}). To keep with the general metric structure of this paper where all metrics are scaled between $[0, 1]$, the MK metric is modified slightly and takes the form of the arithmetic mean between PPV and NPV:

$$\begin{aligned} \text{MK} &= \rho_{t_p} + \rho_{t_n} - 1 \\ &\propto \frac{\rho_{t_p} + \rho_{t_n}}{2} \end{aligned} \tag{24}$$

5.9. Fowlkes-Mallows index

Fowlkes and Mallows [68] derived and outlined a measure of similarity for two different hierarchical clusterings, known as the Fowlkes-Mallows index (FMI).

The original definition (with non-overloaded symbols) for the FMI metric given by Fowlkes and Mallows is

$$M_k = \frac{T_k}{\sqrt{P_k \cdot Q_k}} \tag{25}$$

where

$$T_k = \sum_{i=1}^{k} \sum_{j=1}^{k} m_{ij}^2 - n$$

$$m_{i\cdot} = \sum_{j=1}^{k} m_{ij}$$

$$m_{\cdot j} = \sum_{i=1}^{k} m_{ij}$$

$$m_{\cdot\cdot} = n_c = \sum_{i=1}^{k} \sum_{j=1}^{k} m_{ij}$$

$$P_k = \sum_{i=1}^{k} m_{i\cdot}^2 - n_c$$

$$Q_k = \sum_{j=1}^{k} m_{\cdot j}^2 - n_c$$

From the original proposal in Eq. (25), an interpretation of FMI for classification performance evaluation was proposed in [69], which defines the FMI metric as the geometric mean between precision and recall as follows:

$$\text{FMI} = \sqrt{\Phi_{t_p} \cdot \rho_{t_p}} \tag{26}$$

5.10. Optimised precision

Optimised precision is a metric proposed by Ranawana and Palade as an improved heuristic used to train multi-classifier systems [70]. However, name "optimised precision" is a misnomer as the formulation given for precision by authors of the metric is in fact accuracy.

Ranawana and Palade outlined the potential issues of training on a dataset with a large proportion of non-target class instances, and proposed that a solution is to simultaneously minimise ($|\Phi_{t_n} - \Phi_{t_p}|$) and maximise ($\Phi_{t_n} + \Phi_{t_p}$). The solution given is to incorporate a new metric named relationship index, the relationship index is defined as

$$RI = \frac{|\Phi_{t_n} - \Phi_{t_p}|}{\Phi_{t_n} + \Phi_{t_p}}. \tag{27}$$

The final metric for optimised precision is

$$\begin{aligned} OP &= A - RI \\ &= \frac{t_p + t_n}{t_p + t_n + f_p + f_n} - \frac{|\Phi_{t_n} - \Phi_{t_p}|}{\Phi_{t_n} + \Phi_{t_p}} \\ &\propto \frac{1}{2} \cdot \left(\frac{t_p + t_n}{t_p + t_n + f_p + f_n} - \frac{|\Phi_{t_n} - \Phi_{t_p}|}{\Phi_{t_n} + \Phi_{t_p}} + 1 \right). \end{aligned} \tag{28}$$

5.11. Matthew's correlation coefficient-F1 metric

As part of an attempt to develop an improved alternative to the popular ROC curve and PR curve analysis techniques, Cao et al. proposed the MCC-F_1 curve and the corresponding MCC-F_1 metric [71]. Cao et al. claimed that the MCC-F_1 is more informative than the ROC and PR curves, because MCC-F_1 summarises the whole confusion matrix instead of only parts of the confusion matrix. The MCC-F_1 curve plots the values of a normalised MMC against F_1 for different threshold values and the MCC-F_1 metric then calculates the average distance between the points on the MCC-F_1 curve resulting from different thresholds to the point representing an idealistic classifier.

The metric is calculated as follows over different threshold values: n_T thresholds are used to test a classifier, for each point $i \in \{0, 1, \ldots, n_T - 1\}$ the prediction score $f(x_i)$ is calculated and the unit normalised MMC value, X_i, as well as the F_1 value, Y_i, is identified. The MMC values are divided into $W = 100$ subranges, each with a width of $w = (\max_i X_i - \min_i X_i)/W$. The Euclidean distance of a point i and the perfect classifier point $(1, 1)$ is calculated using

$$D_i = \sqrt{(X_i - 1)^2 + (Y_i - 1)^2} \tag{29}$$

The maximum distance possible between a point i and the ideal classifier occurs if i represents the worst possible classifier at $(0, 0)$. This worst-case distance is $\sqrt{2}$. The MCC-F_1 curve is then divided into two sides, left (L) and right (R), and the set of points in the subranges of each side is calculated as Z_j^s, with $j \in \{0, 1, \ldots, W - 1\}$ and $s \in \{L, R\}$. The number of points in these sets is defined by

$$n_j^s = |Z_j^s| \tag{30}$$

For the sets with a non-zero number of points, the mean distance $\bar{D}_j^s$ is defined by

$$\bar{D}_j^s = \frac{\sum_{i \in Z_j^s} D_i}{n_j^s} \tag{31}$$

All pairs of sides and subranges $\mathcal{D} = (s, j)$ for non-zero Z_j^s are identified; the generator function for this set is given as

$$\mathcal{D} = \{(s, j) | s \in \{L, R\}, j \in \{0, 1, \ldots, W - 1\}, n_j^s > 0\} \tag{32}$$

To get the grand average D^*, mean distances $\bar{D}_j^s$ are averaged over the $\mathcal{D}$ pairs as shown below:

$$D^* = \frac{\sum_{(s,j) \in \mathcal{D}} \bar{D}_j^s}{|\mathcal{D}|} \tag{33}$$

Division of the grand average by the hypothetical max distance results in the final metric of

$$\text{MCC-F}_1 = 1 - \frac{D^*}{\sqrt{2}} \tag{34}$$

Computation of the MCC-F_1 metric for only one threshold simplifies the process somewhat. One threshold creates the conditions $n_T = 1$ and $i = \{1\}$, meaning that $n_j^s = 0$ except for one value of j, *i.e.* $|\{n_j^s : n_j^s > 0\}| = 1$. The mean distance then becomes

$$\bar{D}_j^s = \frac{D_1}{1} \tag{35}$$

The set of (side, subrange) pairs then also contains only one point, giving the grand average the same formula as the mean distance, *i.e.*

$$D^* = \frac{D_1}{1} \tag{36}$$

Finally, the metric becomes

$$\text{MCC-F}_1 = \frac{D_1}{\sqrt{2}} = 1 - \frac{\sqrt{(\text{F}_1 - 1)^2 + (\text{MCC} - 1)^2}}{\sqrt{2}} \tag{37}$$

5.12. Sensitivity specificity geometric mean

Similar to how the FMI is the geometric mean between the precision and recall, the geometric mean between the Φ_{t_p} and Φ_{t_n} has also been used in [72]. Kubat et al. [72] proposed this metric to evaluate the performance of systems detecting oil spills. It aims to maximise the accuracy on both the positive class and negative class, and to minimise the discrepancy in the levels of accuracy between these two classes. The original authors simply referred to their metric as g (for geometric mean) and used a^+ to mean Φ_{t_p} and a^- to mean Φ_{t_n}. The original formula is

$$g = \sqrt{a^+ \cdot a^-} \tag{38}$$

In order to prevent confusion by using a general term like geometric mean (g) to mean a specific implementation thereof, this paper refers to the metric of this section specifically as the geometric mean between the sensitivity (Φ_{t_p}) and specificity (Φ_{t_p}). Therefore, the geometric mean between the sensitivity and specificity (G_{SS}) is

$$G_{SS} = \sqrt{\Phi_{t_p} \cdot \Phi_{t_n}} \tag{39}$$

5.13. Index of balanced accuracy

The index of balanced accuracy (IBA) was proposed by García et al. for evaluating two-class problems in imbalanced domains [73]. IBA combines the geometric mean of TPR and TNR with a dominance relation between TPR and TNR. García et al. also proposed that the dominance relation can be combined with any metric, in order to make the metric more robust to CI [74].

For IBA, the geometric mean component is as defined by Kubat et al. in Eq. (38), and the dominance relation is

$$d = \Phi_{t_p} - \Phi_{t_n}. \tag{40}$$

García et al. combined Eqs. (38) and (40) and defined IBA as

$$\begin{aligned} IBA &= (1 + d) \cdot g^2 \\ &= \Phi_{t_p} \Phi_{t_n} \cdot (1 + \Phi_{t_p} - \Phi_{t_n}) \end{aligned} \tag{41}$$

6. Modified performance metrics for classification

This section proposes the modified metrics, based on the metrics in Section 5 which are constructed from elements of the confusion matrix and not on rates. Sections 6.1 to 6.7 present modified versions of the metrics summarised in Section 5. Further, Sections 6.8 to 6.11 present new metrics inspired by the popularity of using averages of either sensitivity and specificity, or precision and recall. Sections 6.8 to 6.11 ensure that all variations of means (arithmetic, harmonic, geometric, and quadratic) are applied to these popular combinations.

6.1. Modified Gilbert's ratio of verification

This section proposes to modify Gilbert's ratio of verification. The starting point is ν from Eq. (10), and the final result of ν_ι is obtained by

$$\begin{aligned}&\nu\left(g(t_p), g(t_n), g(f_p), g(f_n)\right)\\&= \frac{\frac{t_p}{P}}{\frac{t_p}{P}+\frac{f_n}{P}+\frac{f_p}{N}}\\&= \frac{\boldsymbol{\Phi}_{t_p}}{\boldsymbol{\Phi}_{t_p}+\boldsymbol{\Phi}_{f_n}+\boldsymbol{\Phi}_{f_p}}\\&= \nu_\iota\end{aligned} \tag{42}$$

6.2. Modified F_1 measure

The algebraically simplified version of the F_1 metric posed in Eq. (19) is modified to create a new metric, F_ι, *i.e.*

$$\begin{aligned}&F_1\left(g(t_p), g(t_n), g(f_p), g(f_n)\right)\\&= \frac{2\cdot\frac{t_p}{P}}{2\cdot\frac{t_p}{P}+\frac{f_n}{P}+\frac{f_p}{N}}\\&= \frac{2\cdot\boldsymbol{\Phi}_{t_p}}{2\cdot\boldsymbol{\Phi}_{t_p}+\boldsymbol{\Phi}_{f_n}+\boldsymbol{\Phi}_{f_p}}\\&= F_\iota\end{aligned} \tag{43}$$

6.3. Modified Kappa

The Cohen's Kappa metric from Eq. (21) is normalised below:

$$\begin{aligned}&\kappa\left(g(t_p), g(t_n), g(f_p), g(f_n)\right)\\&= \frac{1}{2}\cdot\left(\frac{2\cdot(\frac{t_p}{P}\cdot\frac{t_n}{N}-\frac{f_p}{N}\cdot\frac{f_n}{P})}{(\frac{t_p}{P}+\frac{f_p}{N})\cdot(\frac{f_p}{N}+\frac{t_n}{N})+(\frac{t_p}{P}+\frac{f_n}{P})\cdot(\frac{f_n}{P}+\frac{t_n}{N})}+1\right)\\&= \frac{1}{2}\cdot\left(\frac{2\cdot(\boldsymbol{\Phi}_{t_p}\cdot\boldsymbol{\Phi}_{t_n}-\boldsymbol{\Phi}_{f_p}\cdot\boldsymbol{\Phi}_{f_n})}{(\boldsymbol{\Phi}_{t_p}+\boldsymbol{\Phi}_{f_p})\cdot(\boldsymbol{\Phi}_{f_p}+\boldsymbol{\Phi}_{t_n})+(\boldsymbol{\Phi}_{t_p}+\boldsymbol{\Phi}_{f_n})\cdot(\boldsymbol{\Phi}_{f_n}+\boldsymbol{\Phi}_{t_n})}+1\right)\\&= \kappa_\iota\end{aligned} \tag{44}$$

6.4. Modified Laplace/modified precision

The Laplace estimator is normalised using the following process:

$$\begin{aligned}&L\left(g(t_p), g(t_n), g(f_p), g(f_n)\right)\\&= \frac{\frac{t_p}{P}+1}{\frac{t_p}{P}+\frac{f_p}{N}+2}\\&= \frac{\boldsymbol{\Phi}_{t_p}+1}{\boldsymbol{\Phi}_{t_p}+\boldsymbol{\Phi}_{f_p}+2}\\&= L_\rho\end{aligned} \tag{45}$$

where both $\boldsymbol{\Phi}_{t_p}$ and $\boldsymbol{\Phi}_{f_p}$ have a range of $[0,1]$. This result seems analogous to the normalisation results of previous metrics, but the Laplace estimator has constants in both the numerator and denominator. The effect of the constants causes the range of output for the metric to be $[\frac{1}{3},\frac{2}{3}]$. However, the aim for all metrics, for the purposes of this paper, is that the range is to be $[0,1]$. Hence, the following additional steps are followed:

$$\begin{aligned}L_\rho &= \frac{\boldsymbol{\Phi}_{t_p}+1}{\boldsymbol{\Phi}_{t_p}+\boldsymbol{\Phi}_{f_p}+2}\\&\propto \left(\left(\frac{\boldsymbol{\Phi}_{t_p}+1}{\boldsymbol{\Phi}_{t_p}+\boldsymbol{\Phi}_{f_p}+2}\right)-\frac{1}{3}\right)/\left(\frac{2}{3}-\frac{1}{3}\right)\\&= \left(\frac{\boldsymbol{\Phi}_{t_p}+1}{\boldsymbol{\Phi}_{t_p}+\boldsymbol{\Phi}_{f_p}+2}\right)\cdot 3\\&= \frac{3\cdot\boldsymbol{\Phi}_{t_p}+3}{\boldsymbol{\Phi}_{t_p}+\boldsymbol{\Phi}_{f_p}+2}-1\\&= L_\iota\end{aligned} \tag{46}$$

6.5. Modified Matthew's correlation coefficient

This section proposes a modified version of MCC, which has been made more robust to CI through normalisation. This is accomplished through the normalisation process of dividing t_p and f_n by P and t_n and f_p by N as shown below:

$$\begin{aligned}&\text{MCC}\left(g(t_p), g(t_n), g(f_p), g(f_n)\right)\\&= \frac{1}{2}\cdot\left(\frac{\frac{t_p}{P}\cdot\frac{t_n}{N}-\frac{f_p}{N}\cdot\frac{f_n}{P}}{\sqrt{(\frac{t_p}{P}+\frac{f_p}{N})(\frac{t_p}{P}+\frac{f_n}{P})(\frac{t_n}{N}+\frac{f_p}{N})(\frac{t_n}{N}+\frac{f_n}{P})}}+1\right)\\&= \frac{1}{2}\cdot\left(\frac{\boldsymbol{\Phi}_{t_p}\cdot\boldsymbol{\Phi}_{t_n}-\boldsymbol{\Phi}_{f_p}\cdot\boldsymbol{\Phi}_{f_n}}{\sqrt{(\boldsymbol{\Phi}_{t_p}+\boldsymbol{\Phi}_{f_p})(\boldsymbol{\Phi}_{t_p}+\boldsymbol{\Phi}_{f_n})(\boldsymbol{\Phi}_{t_n}+\boldsymbol{\Phi}_{f_p})(\boldsymbol{\Phi}_{t_n}+\boldsymbol{\Phi}_{f_n})}}+1\right)\\&= \text{MCC}_\iota\end{aligned} \tag{47}$$

6.6. Modified optimised precision

This section proposes a modified version of optimised "precision". The normalisation results in the accuracy in the first term becoming balanced accuracy, with the final metric given as

$$\begin{aligned}OP_\iota &= B - RI\\&= \frac{\boldsymbol{\Phi}_{t_p}+\boldsymbol{\Phi}_{t_n}}{2}-\frac{|\boldsymbol{\Phi}_{t_n}-\boldsymbol{\Phi}_{t_p}|}{\boldsymbol{\Phi}_{t_n}+\boldsymbol{\Phi}_{t_p}}\\&\propto \frac{1}{2}\cdot\left(\frac{\boldsymbol{\Phi}_{t_p}+\boldsymbol{\Phi}_{t_n}}{2}-\frac{|\boldsymbol{\Phi}_{t_n}-\boldsymbol{\Phi}_{t_p}|}{\boldsymbol{\Phi}_{t_n}+\boldsymbol{\Phi}_{t_p}}+1\right)\end{aligned} \tag{48}$$

6.7. Modified Matthew's correlation coefficient-F1 metric

The next proposed metric is a normalised version of the MCC-F_1 metric. However, for this metric the process is slightly simpler; since the constituent parts of the MCC-F_1 metric are the MCC metric and the F_1 metric, the existing normalised versions from Eqs. (47) and (43) are used to normalise the MCC-F_1 metric. The modification of the derived single-threshold version of the metric in Eq. (37) results in

$$\begin{aligned}&\text{MCC-F}_1\left(g(t_p), g(t_n), g(f_p), g(f_n)\right)\\&= 1-\frac{\sqrt{(\text{F}_\iota-1)^2+(\text{MCC}_\iota-1)^2}}{\sqrt{2}}\\&= \text{MCC-F}_\iota\end{aligned} \tag{49}$$

6.8. Area under precision recall curve

A PR curve can be generated and analysed to evaluate the performance of a classifier or rule induction system. One of the first in-depth formalisations of PR curve analysis was published by Buckland and Gey [75]. A PR curve demonstrates the trade-offs made by a classifier to optimise its performance on a binary classification problem by plotting

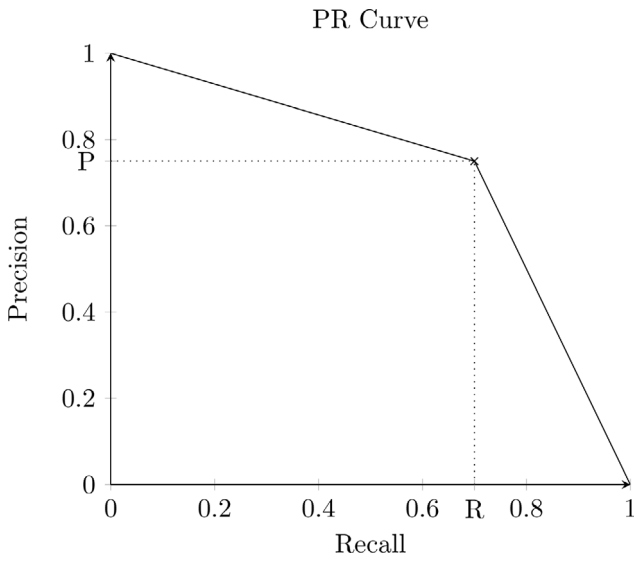


Fig. 2. Trigonometric properties of a single point PR curve.

the precision (PPV) against the recall (TPR) for different threshold values.

A PR graph has a range and domain $[0, 1]$, with four salient (x, y) points on the graph. The following list outlines what points on the PR curve in the vicinity of different x and y values (denoted as $\sim (x, y)$) indicate

- $\sim$ $(0, 0)$ represents a classifier which was unable to correctly classify any instances of the target class.
- $\sim (0, 1)$ represents a classifier which classifies very few instances of the target class, but those which are classified are correct.
- $\sim (1, 0)$ represents a classifier which classifies all instances of the target class correctly, but at the same time none of them. This nonsensical point is more of a parallel to a system which classifies all instances as the target class.
- $\sim (1, 1)$ is a perfect classifier.

Fig. 2 represents a generic PR plot of the performance of one classifier. This hypothetical classifier has a precision of 0.75 and a recall of 0.7. As with the ROC curve, the AUC of the PR curve can be estimated by decomposing the plot into one rectangle and two triangles. The total AUC is defined as the sum of the main rectangular body ($\square$) and the two triangles ($\triangle_1$ and $\triangle_2$), *i.e.* $\dot{A}_{PR} = \dot{A}_{\square} + A_{\triangle_1} + A_{\triangle_2}$.

In the same way that the FMI metric from Section 5.9 is the geometric mean between the precision and recall metrics, the area under a single-threshold PR curve is calculated as the arithmetic mean between the precision and recall metrics. The simplification from the area calculation to the arithmetic mean is shown below:

$$\begin{aligned}\dot{A}_{PR} &= \left(\dot{A}_{\square} + \dot{A}_{\triangle 1} + \dot{A}_{\triangle 2}\right) \\ &= \left(l \cdot b + \frac{1}{2} \cdot b_1 \cdot h_1 + \frac{1}{2} \cdot b_2 \cdot h_2\right) \\ &= \left(\rho_{t_p} \cdot \Phi_{t_p} + \frac{1}{2} \cdot \Phi_{t_p} \cdot (1 - \rho_{t_p}) + \frac{1}{2} \cdot (1 - \Phi_{t_p}) \cdot \rho_{t_p}\right) \\ &= \frac{\rho_{t_p} + \Phi_{t_p}}{2}\end{aligned} \tag{50}$$

6.9. Precision recall quadratic mean

The quadratic mean is a case of the more general power mean with specifically two components. This section proposes that the effect of CI on the final of the four popular means be evaluated. As with Section 6.8, this proposed metric is not a normalised version of an existing metric, but is an averaging technique applied to a popular metric combination (precision and recall).

The rates are combined into the quadratic mean as follows:

$$Q_{PR} = \sqrt{\frac{\Phi_{t_p} + \rho_{t_p}}{2}} \tag{51}$$

6.10. Sensitivity specificity harmonic mean

Similar to how the F_1 measure is the harmonic mean between precision and recall, this paper proposes testing the harmonic mean of another popular combination rates, namely sensitivity and specificity. The harmonic mean between sensitivity (TPR) and specificity (TNR) is not a normalised version of an existing metric, but is a newly proposed metric. There are no examples of this combination in the reviewed literature. The sensitivity and specificity (TPR and TNR) are already combined using the arithmetic mean in the balanced accuracy metric presented in Section 5.3. The harmonic counterpart is shown below:

$$H_{SS} = \frac{2}{\frac{1}{\Phi_{t_p}} + \frac{1}{\Phi_{t_n}}} \tag{52}$$

6.11. Sensitivity specificity quadratic mean

As in Section 6.9, this section defines the fourth average using the popular combination of Φ_{t_p} and Φ_{t_n} as a potential metric. The quadratic mean between sensitivity and specificity is calculated as follows:

$$Q_{SS} = \sqrt{\frac{\Phi_{t_p} + \Phi_{t_n}}{2}} \tag{53}$$

7. Metric behavioural analysis empirical process

This section outlines how the existing and proposed metrics presented in Sections 5 and 6 are analysed under different conditions of CI. The section also proposes a performance metric classification schema to simplify the description and explanation of the behaviour of a metric under CI.

7.1. Empirical process

In order to compare how robust each metric is to CI, the following procedure was applied. Firstly, levels of CI was defined, *i.e.* $(P : N)$, where P represents the number of instances in the target class and N the number of instances in the non-target class. For this empirical analysis, the levels of CI tested were $(1 : 1)$, $(1 : 2)$, $(1 : 100)$ and $(1 : 1000)$. Then input axes were constructed for different possible values of $t_p \in \{0, \ldots, P\}$ and $f_p \in \{0, \ldots, N\}$. The two axes were used to create a meshgrid of 10000 points, on which each metric was evaluated to simulate all possible configurations of the confusion matrix.

S'SA was used to calculate the importance of each input (t_p and f_p) by the process described in Section 3.3. S'SA was used to calculate first-order sensitivity indices on 2^{16} SSs, to ensure a sufficient number of samples for the Monte Carlo process. The first-order sensitivity indices were summarised as importance score means ($\mu_{I_{t_p,(P:N)}}$ and $\mu_{I_{f_p,(P:N)}}$) and standard deviations ($\sigma_{I_{t_p,(P:N)}}$ and $\sigma_{I_{f_p,(P:N)}}$). The average importance scores for each input for each CI was compared to the case of $(1 : 1)$ with the Welch's t-test [76]. The p-values from the t-tests ($p_{t_p(P:N)}$ and $p_{f_p(P:N)}$) were combined with the method from Stouffer et al. (S_S) [77,78], as outlined by Heard [79]. The p-values are used to compare the null hypothesis (H_0), which states that there is no statistically significant difference in the average importance scores, against the alternative hypothesis (H_a), which states that there is a statistically significant difference in the average importance scores.

The procedure for statistical comparison is outlined in Algorithms 1 and 2.

Algorithm 1 Statistical comparison of importance scores

Input: metric (f)
Output: p-values

1: Let comparison base case be importance scores of $(1:1)$ imbalance
2: **for** imbalance levels $(P:N) \in \{(1:2), (1:100), (1:1000)\}$ **do**
3: Calculate importance score values according to Algorithm 2
4: Let $p_{I_{t_p},(P:N)} = \text{t-test}(\mu_{I_{t_p},(1:1)} \pm \sigma_{I_{t_p},(1:1)}; \mu_{I_{t_p},(P:N)} \pm \sigma_{I_{t_p},(P:N)})$
5: Let $p_{I_{f_p},(P:N)} = \text{t-test}(\mu_{I_{f_p},(1:1)} \pm \sigma_{I_{f_p},(1:1)}; \mu_{I_{f_p},(P:N)} \pm \sigma_{I_{f_p},(P:N)})$
6: Combine p-values as $p_{(P:N)} = S_S(p_{t_p}, p_{f_p})$
7: Add $p_{(P:N)}$ to list of p-values, $\boldsymbol{p}$
8: **end for**
9: Return $\boldsymbol{p}$

Algorithm 2 S'SA for a metric under CI

Input: metric (f), level of CI $(P:N)$
Output: Importance scores (S_1 indices from S'SA)

1: Define the domain for t_p as $\mathcal{P} \subseteq [0, P]$
2: Define the domain for f_p as $\mathcal{N} \subseteq [0, N]$
3: Sample SS from $\mathcal{X} = \mathcal{P} \times \mathcal{N}$
4: Evaluate $\mathcal{F} = f(x_i),\ \forall x_i \in \mathcal{X}$
5: Apply S'SA to $(\mathcal{X}, \mathcal{F})$
6: Return means and standard deviations for importance scores, *i.e.* $\mu_{I_{t_p}} \pm \sigma_{I_{t_p}}$ and $\mu_{I_{f_p}} \pm \sigma_{I_{f_p}}$

Additionally, metric behaviour is quantified through the use of contour plots. For each level of CI, the equally spaced points of $t_p \in \{0, \dots, P\}$ and $f_p \in \{0, \dots N\}$ were used to calculate the metric output, $f(t_p, f_p)$. Let $y_{i,(P:N)}$ represent the metric output for input point $(t_p, f_p)_i \in \{0, \dots, P\} \times \{0, \dots, N\}$ from $(P:N)$ imbalanced data. The total deviation of the metric under $(P:N)$ imbalance in comparison to $(1:1)$ imbalance is quantified by the sum of the absolute value of the difference between each point, as

$$\Sigma_{(P:N)} = \sum_{(t_p,f_p)_i} |y_{i,(1:1)} - y_{i,(P:N)}|.$$

The greater the value of $\Sigma_{(P:N)}$, the less robust the metric is to $(P:N)$ imbalance.

7.2. Metric behaviour classification

After completion of the procedure in Section 7.1, the results from the S'SA importance scores statistical comparison and the contour plot deviations are used to classify metrics.

S'SA importance score classification. The classification based on S'SA importance scores are made by comparing the importance scores (I_{t_p} and I_{f_p}) for $(1:1)$ CI against the other levels of CI. Since metric behaviour on a $(1:1)$ balanced dataset represents the most trustworthy behaviour of a metric, metric behaviour of other levels of CI are compared against this base case. The average importance scores of each level of CI are used to calculate a p-value, $p_{(1:i)}$, which is used to accept or reject H_0. When $(p_{(1:i)} < 0.05)$ it indicates that H_0 is rejected, *i.e.* there is a statistically significant difference in the average importance scores. Rejection of H_0 indicates that a metric is not robust to that given level of CI, since the behaviour of the metric deviates from behaviour of the balanced case. Using this information, the S'SA importance scores are used to classify metrics into one of five types as outlined in Table 3.

Table 3
Types of metrics based on S'SA scores.

Type	Description
Type 1	$p_{(1:i)} < 0.05\ \forall\ i \in \{2, 10, 100, 1000\}$
Type 2	$p_{(1:i)} < 0.05\ \forall\ i \in \{10, 100, 1000\}$
Type 3	$p_{(1:i)} < 0.05\ \forall\ i \in \{100, 1000\}$
Type 4	$p_{(1:i)} < 0.05\ \forall\ i \in \{1000\}$
Type 5	$p_{(1:i)} < 0.05\ \forall\ i \in \{\}$

Table 4
Types of metrics based on contour plots.

Type	Description
Type 1	$\Sigma_{(1:i)} > 0\ \forall\ i \in \{2, 10, 100, 1000\}$
Type 2	$\Sigma_{(1:i)} > 0\ \forall\ i \in \{10, 100, 1000\}$
Type 3	$\Sigma_{(1:i)} > 0\ \forall\ i \in \{100, 1000\}$
Type 4	$\Sigma_{(1:i)} > 0\ \forall\ i \in \{1000\}$
Type 5	$\Sigma_{(1:i)} > 0\ \forall\ i \in \{\}$

Table 5
p-values for importance score comparisons.

Metric	$p_{(1:2)}$	$p_{(1:10)}$	$p_{(1:100)}$	$p_{(1:1000)}$
A	0.0000	0.0000	0.0000	0.0000
ν	0.0000	0.0000	0.0000	0.0000
B	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F_1	0.0000	0.0000	0.0000	0.0000
κ	0.0000	0.0000	0.0000	0.0000
L	0.0000	0.0000	0.0000	0.0000
MCC	0.0000	0.0000	0.0000	0.0000
MK	0.0000	0.0000	0.0000	0.0000
FMI	0.0000	0.0000	0.0000	0.0000
OP	0.0000	0.0000	0.0000	0.0000
MCC-F_1	0.0000	0.0000	0.0000	0.0000
G_{SS}	**0.9913**	**0.9550**	**0.8720**	**0.7989**
IBA	**0.9901**	**0.5113**	**0.9969**	**0.9540**
ν_i	**0.9988**	**1.0000**	**0.9973**	**0.9980**
F_i	**0.9839**	**0.9867**	**0.9838**	**0.9700**
κ_i	**1.0000**	**1.0000**	**1.0000**	**1.0000**
L_i	**1.0000**	**1.0000**	**1.0000**	**1.0000**
MCC_i	**0.9948**	**0.9262**	**0.9994**	**0.9902**
OP_i	0.0000	0.0000	0.0000	0.0000
MCC-F_i	**0.9962**	**0.9875**	**0.9999**	**0.9952**
A_{PR}	0.0000	0.0000	0.0000	0.0000
Q_{PR}	0.0000	0.0000	0.0000	0.0000
H_{SS}	**0.9359**	**0.9942**	**0.6421**	**0.9995**
Q_{SS}	**0.9858**	**0.9828**	**0.9779**	**0.9973**

Contour plot deviation. The contour plots for the different levels of CI of a metric are also used to classify the behaviour of the metric. Contour plots which maintain uniform shapes and contours even under different distributions of target feature values are seen as more robust to CI. The five types of contour plot-based classes are summarised in Table 4.

8. Results and discussion

This section presents the results obtained from the described analysis method in Section 7. Detailed results are given in Appendix A (Tables A.12, A.13, A.14, A.15 and A.16). These tables contain the S'SA importance scores for t_p and f_p for imbalances of (1:1), (1:2), (1:10), (1:100) and (1:1000) respectively. The p-values which compare the importance scores are given in Table 5. Table 6 classifies metrics according to the schema defined in Table 3.

The deviations between contour plots are presented in Table 7. Table 8 classifies each metric according to the categories of contour plots presented in Table 4. The contour plots used to make the classifications are given in Appendix B as Figs. B.3 to B.26. The visual representations of the contour plots provide additional insight into how the behaviour of metrics changes with increasingly extreme CI.

Table 6 shows that all evaluated metrics are classified as either Type 1 or Type 5. Type 1 metrics are greatly affected by CI as the ratios of

Table 6
S'SA-based classification of metric behaviour.

Type	Metrics
1	A, ν, F_1, κ, L, MCC, MK, FMI, OP, MCC-F_1, A_{PR}, Q_{PR}
2	
3	
4	
5	B, G_{SS}, IBA, ν_ι, F_ι, κ_ι, L_ι, MCC$_\iota$, OP_ι, MCC-F_ι, H_{SS}, Q_{SS}

Table 7
Contour plot cumulative deviation scores.

Metric	$\Sigma_{(1:2)}$	$\Sigma_{(1:10)}$	$\Sigma_{(1:100)}$	$\Sigma_{(1:1000)}$
A	561.11	1377.27	1650.0	1679.97
ν	716.14	2253.96	3211.02	3393.73
B	**0.0**	**0.0**	**0.0**	**0.0**
F_1	777.16	2791.69	4320.15	4652.12
κ	214.03	971.47	1516.24	1644.29
L	1281.47	3515.86	4680.3	4878.56
MCC	100.73	573.63	1271.53	1594.01
MK	223.25	1023.3	1751.6	1903.28
FMI	713.56	2391.24	3917.76	4516.34
OP	280.56	688.64	825.0	839.99
MCC-F$_1$	407.98	1599.66	2603.22	2851.58
G_{SS}	**0.0**	**0.0**	**0.0**	**0.0**
IBA	**0.0**	**0.0**	**0.0**	**0.0**
ν_ι	**0.0**	**0.0**	**0.0**	**0.0**
F_ι	**0.0**	**0.0**	**0.0**	**0.0**
κ_ι	**0.0**	**0.0**	**0.0**	**0.0**
L_ι	**0.0**	**0.0**	**0.0**	**0.0**
MCC$_\iota$	**0.0**	**0.0**	**0.0**	**0.0**
OP_ι	**0.0**	**0.0**	**0.0**	**0.0**
MCC-F$_\iota$	**0.0**	**0.0**	**0.0**	**0.0**
A_{PR}	647.22	1763.79	2340.18	2437.69
Q_{PR}	505.56	1422.57	1924.43	2013.31
H_{SS}	**0.0**	**0.0**	**0.0**	**0.0**
Q_{SS}	**0.0**	**0.0**	**0.0**	**0.0**

Table 8
Contour plot classification of metric behaviour.

Type	Metrics
1	A, ν, F_1, κ, L, MCC, MK, FMI, OP, MCC-F_1, A_{PR}, Q_{PR}
2	
3	
4	
5	B, G_{SS}, IBA, ν_ι, F_ι, κ_ι, L_ι, MCC$_\iota$, OP_ι, MCC-F_ι, H_{SS}, Q_{SS}

importance scores are dissimilar for all levels of CI. On the contrary, Type 5 metrics maintain equivalent ratios for all levels of CI, which indicates robustness to CI. There is a clear trend that the modified metrics proposed in Section 6, as well as metrics which average the sensitivity and specificity, are Type 5. The metrics popular in existing literature from Section 5 are Type 1.

An analysis of Table 8 reveals similar trends to the trends in Table 6. All the analysed metrics are either Type 1 or Type 5. The newly proposed metrics, as well as the averages between sensitivity and specificity, are all Type 5.

It is clear that there are two dominant cases of robustness to CI: metrics which are robust, and metrics which are not. Robust metrics include the metrics which were normalised using the method proposed in Section 4, and metrics which average the TPR and TNR using one of the four types of means. The majority of metrics which are popularly used in literature are not robust to CI.

9. Metric behaviour on real-world datasets

This section bolsters the results seen in the importance score and contour plot evaluation of Section 8. All existing and proposed metrics presented in Sections 5 and 6 were evaluated on six real-world anomaly detection datasets. The evaluation on these real-world datasets was performed to show the discrepancy in the expected value of the results

Table 9
Real-world dataset characteristics.

Name	Instances	Attributes	Imbalance	Reference
D_1	284 807	30	(1:577.88)	[80]
D_2	7200	21	(1:12.48)	[81]
D_3	202 599	39	(1:43.56)	[82]
D_4	41 188	62	(1:7.87)	[83]
D_5	299 285	500	(1:15.12)	[84]
D_6	37 136	500	(1:1)	[84]

Table 10
LR Hyperparameters.

Hyperparameter	Value
Optimiser	L-BFGS
Iterations	100
Regularisation	L_2
Stopping tolerance	10^{-4}
Regularisation strength	1.0

(shown by the robust metrics denoted with the subscript ι), and the non-robust metrics (*i.e.* many of the metrics popular in literature).

The datasets included credit card fraud detection [80], thyroid disease detection [81], celebrity attribute classification [82], bank marketing success prediction [83], and income levels from census data [84]. All six datasets were preprocessed as by Pang et al. in [85], where missing values were imputed by the mean and categorical variables were one-hot encoded. The dataset files are available from Pang et al. on GitHub.[2] Note that the sixth dataset, D_6, is simply D_5 after randomly undersampling the majority class to achieve (1:1) class balance. Dataset D_6 was created to compare metrics under balanced conditions.

Table 9 outlines the dataset characteristics, *i.e.* the total number of instances, the dimensionality of the data, the level of class imbalance, and the source of the data.

A simple ML model was trained on each of the six datasets. The purpose of this experiment was not to find the optimal ML model for each problem, but to see how the interpretation of the aptness of each model changes as different metrics are used. The ML approach selected for this task was LR due to its simplicity, ease of use, and relevance to binary classification. LR can be made more complex through the use of basis functions, but this was not explored. For a history and background of LR, the reader is referred to [86]. The implementation of LR used was from the popular Python package `scikit-learn`,[3] optimised with the limited memory Broyden-Fletcher-Goldfarb-Shanno algorithm (L-BFGS) [87], and used with the hyperparameters specified in Table 10. In order to split the dataset into train and test sets, stratified k-fold cross-validation was used. Due to the class imbalance, stratified sampling was used to ensure that the proportions of each class remained consistent across all train and test sets. Each dataset was split into $k = 5$ folds, which resulted in a 80%/20% train-test split. For each metric, the average values over each of the six tests sets were calculated and are reported in Table 11

The real-world results echo the results seen in Section 8, in which it is clear that many existing metrics deviate from the expected behaviour when applied to imbalanced learning. For example, accuracy (A) clearly overestimates the aptness of the models on all datasets in comparison to balanced accuracy (B). Similarly, even metrics which are heralded as suitable for imbalanced learning (*e.g.* F_1) do not show consistent results. Table 11 shows that, other than for the thyroid dataset, the values for F_1 do not match those of F_ι. Since F_ι was shown to be insensitive to class imbalance, F_ι represents the expected behaviour (*i.e.* F_ι behaves the same for all levels of class imbalance). If

[2] https://github.com/GuansongPang/ADRepository-Anomaly-detection-datasets/.

[3] https://scikit-learn.org/stable/.

Table 11
Real world dataset.

Metric	D_1	D_2	D_3	D_4	D_5	D_6
A	0.9991	0.9383	0.9781	0.9096	0.9529	0.8713
ν	0.5512	0.1778	0.1111	0.3274	0.3410	0.7766
B	0.8058	0.5895	0.5598	0.6831	0.6914	0.8713
F_1	0.7096	0.3002	0.1999	0.4933	0.5086	0.8742
κ	0.8546	0.6417	0.5967	0.7237	0.7431	0.8713
L	0.8366	0.8994	0.5584	0.6683	0.7210	0.8543
MCC	0.8594	0.6972	0.6269	0.7335	0.7555	0.8717
MK	0.9227	0.9392	0.7696	0.7977	0.8411	0.8721
FMI	0.7192	0.4089	0.2608	0.5112	0.5323	0.8745
OP	0.8788	0.6208	0.5978	0.7408	0.7605	0.9220
MCC-F_1	0.7719	0.4608	0.3758	0.5951	0.6119	0.8730
G_{SS}	0.7817	0.4211	0.3484	0.6173	0.6236	0.8709
IBA	0.3761	0.0339	0.0152	0.1584	0.1567	0.7944
ν_t	0.6118	0.1796	0.1216	0.3814	0.3889	0.7766
F_t	0.7583	0.3027	0.2167	0.5521	0.5600	0.8742
κ_t	0.8058	0.5895	0.5598	0.6831	0.6914	0.8713
L_t	0.8510	0.6226	0.5845	0.7274	0.7389	0.8655
MCC_t	0.8319	0.6552	0.6239	0.7257	0.7386	0.8717
OP_t	0.7822	0.4463	0.3887	0.6275	0.6298	0.9220
MCC-F_t	0.7918	0.4499	0.3856	0.6286	0.6381	0.8730
A_{PR}	0.7290	0.5599	0.3404	0.5298	0.5570	0.8747
Q_{PR}	0.8536	0.7478	0.5832	0.7278	0.7463	0.9353
H_{SS}	0.7584	0.3029	0.2170	0.5579	0.5625	0.8706
Q_{SS}	0.8976	0.7677	0.7482	0.8265	0.8315	0.9334

F_1 were also insensitive to class imbalance it is expected that F_1 would show similar results to F_t, but this is not the case.

The shortcomings of existing metrics is further demonstrated by the results on D_6. The results for the balanced census dataset (D_6) show that the behaviour of existing metrics do match the behaviour of the modified metrics. For example, the accuracy (A) and balanced accuracy (B) are both 0.8713 on D_6. Similarly, both F_1 and F_t are the same at 0.8742.

10. Conclusion and future work

This paper provided an extensive review of the behaviour of performance evaluation metrics for classification problems under the influence of different levels of CI. Multiple metrics from existing literature were reviewed, with in-depth histories and justifications for the proposals of these metrics presented. A normalisation-based technique for the creation of robust metrics from existing metrics was proposed, and used to modify the reviewed metrics. Additionally, all reviewed and proposed metrics were analysed under different levels of CI using Sobol' sensitivity analysis (S'SA).

To the knowledge of the authors, this paper performed the first extensive variance-based global SA of classification metrics with regards to CI. This paper found that the majority of existing metrics were sensitive to CI. This paper also found that the proposed normalisation technique resulted in metrics which are highly robust to CI.

Overall, this paper shed light on the issue of CI in classification problems, which has plagued researchers in fields like ML and diagnostics for decades. This paper also provided a potential solution to many problems posed by CI through the creation of new, more robust metrics.

This paper answered many questions about how binary classification metrics behave under different levels of CI. Future studies on this topic can include the sensitivity of multi-class classification metrics, the sensitivity of metrics when used in conjunction with a specific learning algorithm (*e.g.* neural networks), and the sensitivity of metrics when used on real-world datasets.

CRediT authorship contribution statement

Jean-Pierre van Zyl: Writing – review & editing, Writing – original draft, Methodology, Investigation, Formal analysis, Conceptualization. **Andries Petrus Engelbrecht:** Writing – review & editing, Supervision, Methodology, Investigation, Conceptualization.

Table A.12
Table of S'SA results 1:1 class imbalance.

Metric	1:1 imbalance		
	t_p	f_p	p
A	0.50 ± 0.01	0.50 ± 0.01	(1.00 : 1.00)
ν	0.10 ± 0.00	0.86 ± 0.01	(1.00 : 8.19)
B	0.50 ± 0.01	0.50 ± 0.01	(1.00 : 1.00)
F_1	0.08 ± 0.00	0.91 ± 0.01	(1.00 : 11.78)
κ	0.50 ± 0.01	0.50 ± 0.01	(1.00 : 1.00)
L	0.49 ± 0.01	0.49 ± 0.01	(1.00 : 1.00)
MCC	0.50 ± 0.01	0.50 ± 0.01	(1.00 : 1.00)
MK	0.49 ± 0.01	0.49 ± 0.01	(1.00 : 1.00)
FMI	0.09 ± 0.00	0.90 ± 0.01	(1.00 : 9.67)
OP	0.39 ± 0.01	0.39 ± 0.01	(1.00 : 1.00)
MCC-F_1	0.24 ± 0.01	0.76 ± 0.01	(1.00 : 3.20)
G_{SS}	0.47 ± 0.01	0.47 ± 0.01	(1.00 : 1.00)
IBA	0.19 ± 0.01	0.68 ± 0.01	(1.00 : 3.61)
ν_t	0.10 ± 0.00	0.86 ± 0.01	(1.00 : 8.19)
F_t	0.08 ± 0.00	0.91 ± 0.01	(1.00 : 11.78)
κ_t	0.50 ± 0.01	0.50 ± 0.01	(1.00 : 1.00)
L_t	0.50 ± 0.01	0.50 ± 0.01	(1.00 : 1.00)
MCC_t	0.50 ± 0.01	0.50 ± 0.01	(1.00 : 1.00)
OP_t	0.39 ± 0.01	0.39 ± 0.01	(1.00 : 1.00)
MCC-F_t	0.24 ± 0.01	0.76 ± 0.01	(1.00 : 3.20)
A_{PR}	0.12 ± 0.00	0.87 ± 0.01	(1.00 : 7.28)
Q_{PR}	0.10 ± 0.00	0.89 ± 0.01	(1.00 : 9.02)
H_{SS}	0.44 ± 0.01	0.44 ± 0.01	(1.00 : 1.00)
Q_{SS}	0.49 ± 0.01	0.49 ± 0.01	(1.00 : 1.00)

Table A.13
Table of S'SA results 1:2 class imbalance.

Metric	1:2 imbalance		
	t_p	f_p	p
A	0.80 ± 0.01	0.20 ± 0.00	(1.00 : 0.25)
ν	0.22 ± 0.01	0.70 ± 0.01	(1.00 : 3.18)
B	0.50 ± 0.01	0.50 ± 0.01	(1.00 : 1.00)
F_1	0.18 ± 0.01	0.79 ± 0.01	(1.00 : 4.42)
κ	0.51 ± 0.01	0.49 ± 0.01	(1.00 : 0.97)
L	0.60 ± 0.01	0.37 ± 0.01	(1.00 : 0.62)
MCC	0.54 ± 0.01	0.46 ± 0.01	(1.00 : 0.86)
MK	0.58 ± 0.01	0.40 ± 0.01	(1.00 : 0.69)
FMI	0.16 ± 0.01	0.82 ± 0.01	(1.00 : 5.16)
OP	0.52 ± 0.01	0.27 ± 0.01	(1.00 : 0.51)
MCC-F_1	0.32 ± 0.01	0.67 ± 0.01	(1.00 : 2.09)
G_{SS}	0.47 ± 0.01	0.47 ± 0.01	(1.00 : 1.00)
IBA	0.19 ± 0.01	0.68 ± 0.01	(1.00 : 3.61)
ν_t	0.10 ± 0.00	0.86 ± 0.01	(1.00 : 8.19)
F_t	0.08 ± 0.00	0.91 ± 0.01	(1.00 : 11.78)
κ_t	0.50 ± 0.01	0.50 ± 0.01	(1.00 : 1.00)
L_t	0.50 ± 0.01	0.50 ± 0.01	(1.00 : 1.00)
MCC_t	0.50 ± 0.01	0.50 ± 0.01	(1.00 : 1.00)
OP_t	0.39 ± 0.01	0.39 ± 0.01	(1.00 : 1.00)
MCC-F_t	0.24 ± 0.01	0.76 ± 0.01	(1.00 : 3.20)
A_{PR}	0.15 ± 0.00	0.84 ± 0.01	(1.00 : 5.74)
Q_{PR}	0.11 ± 0.00	0.88 ± 0.01	(1.00 : 7.75)
H_{SS}	0.44 ± 0.01	0.44 ± 0.01	(1.00 : 1.00)
Q_{SS}	0.49 ± 0.01	0.49 ± 0.01	(1.00 : 1.00)

Declaration of competing interest

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

Acknowledgements

The financial assistance of the National Research Foundation (NRF) of South Africa towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the authors and are not necessarily to be attributed to the NRF.

Table A.14
Table of S'SA results 1:10 class imbalance.

Metric	1:10 imbalance		
	t_p	f_p	p
A	0.99 ± 0.01	0.01 ± 0.00	(1.00 : 0.01)
ν	0.52 ± 0.01	0.30 ± 0.01	(1.00 : 0.57)
B	0.50 ± 0.01	0.50 ± 0.01	(1.00 : 1.00)
F_1	0.51 ± 0.01	0.38 ± 0.01	(1.00 : 0.75)
κ	0.61 ± 0.01	0.32 ± 0.01	(1.00 : 0.53)
L	0.77 ± 0.02	0.16 ± 0.01	(1.00 : 0.21)
MCC	0.61 ± 0.01	0.38 ± 0.01	(1.00 : 0.62)
MK	0.75 ± 0.01	0.22 ± 0.01	(1.00 : 0.30)
FMI	0.34 ± 0.01	0.59 ± 0.01	(1.00 : 1.73)
OP	0.68 ± 0.01	0.13 ± 0.01	(1.00 : 0.19)
MCC-F_1	0.55 ± 0.01	0.40 ± 0.01	(1.00 : 0.73)
G_{SS}	0.47 ± 0.01	0.47 ± 0.01	(1.00 : 1.00)
IBA	0.19 ± 0.01	0.68 ± 0.01	(1.00 : 3.61)
ν_t	0.10 ± 0.00	0.86 ± 0.01	(1.00 : 8.19)
F_t	0.08 ± 0.00	0.91 ± 0.01	(1.00 : 11.78)
κ_t	0.50 ± 0.01	0.50 ± 0.01	(1.00 : 1.00)
L_t	0.50 ± 0.01	0.50 ± 0.01	(1.00 : 1.00)
MCC_t	0.50 ± 0.01	0.50 ± 0.01	(1.00 : 1.00)
OP_t	0.39 ± 0.01	0.39 ± 0.01	(1.00 : 1.00)
MCC-F_t	0.24 ± 0.01	0.76 ± 0.01	(1.00 : 3.20)
A_{PR}	0.14 ± 0.01	0.85 ± 0.01	(1.00 : 6.20)
Q_{PR}	0.09 ± 0.00	0.91 ± 0.01	(1.00 : 9.96)
H_{SS}	0.44 ± 0.01	0.44 ± 0.01	(1.00 : 1.00)
Q_{SS}	0.49 ± 0.01	0.49 ± 0.01	(1.00 : 1.00)

Table A.15
Table of S'SA results 1:100 class imbalance.

Metric	1:100 imbalance		
	t_p	f_p	p
A	1.00 ± 0.01	0.00 ± 0.00	(1.00 : 0.00)
ν	0.70 ± 0.04	0.06 ± 0.01	(1.00 : 0.09)
B	0.50 ± 0.01	0.50 ± 0.01	(1.00 : 1.00)
F_1	0.73 ± 0.03	0.09 ± 0.01	(1.00 : 0.12)
κ	0.74 ± 0.03	0.09 ± 0.01	(1.00 : 0.12)
L	0.84 ± 0.05	0.04 ± 0.00	(1.00 : 0.04)
MCC	0.68 ± 0.01	0.27 ± 0.01	(1.00 : 0.40)
MK	0.84 ± 0.03	0.07 ± 0.01	(1.00 : 0.08)
FMI	0.53 ± 0.02	0.34 ± 0.01	(1.00 : 0.65)
OP	0.73 ± 0.01	0.10 ± 0.01	(1.00 : 0.14)
MCC-F_1	0.72 ± 0.03	0.16 ± 0.01	(1.00 : 0.22)
G_{SS}	0.47 ± 0.01	0.47 ± 0.01	(1.00 : 1.00)
IBA	0.19 ± 0.01	0.68 ± 0.01	(1.00 : 3.61)
ν_t	0.10 ± 0.00	0.86 ± 0.01	(1.00 : 8.19)
F_t	0.08 ± 0.00	0.91 ± 0.01	(1.00 : 11.78)
κ_t	0.50 ± 0.01	0.50 ± 0.01	(1.00 : 1.00)
L_t	0.50 ± 0.01	0.50 ± 0.01	(1.00 : 1.00)
MCC_t	0.50 ± 0.01	0.50 ± 0.01	(1.00 : 1.00)
OP_t	0.39 ± 0.01	0.39 ± 0.01	(1.00 : 1.00)
MCC-F_t	0.24 ± 0.01	0.76 ± 0.01	(1.00 : 3.20)
A_{PR}	0.04 ± 0.00	0.96 ± 0.01	(1.00 : 25.12)
Q_{PR}	0.02 ± 0.00	0.98 ± 0.01	(1.00 : 45.16)
H_{SS}	0.44 ± 0.01	0.44 ± 0.01	(1.00 : 1.00)
Q_{SS}	0.49 ± 0.01	0.49 ± 0.01	(1.00 : 1.00)

Table A.16
Table of S'SA results 1:1000 class imbalance.

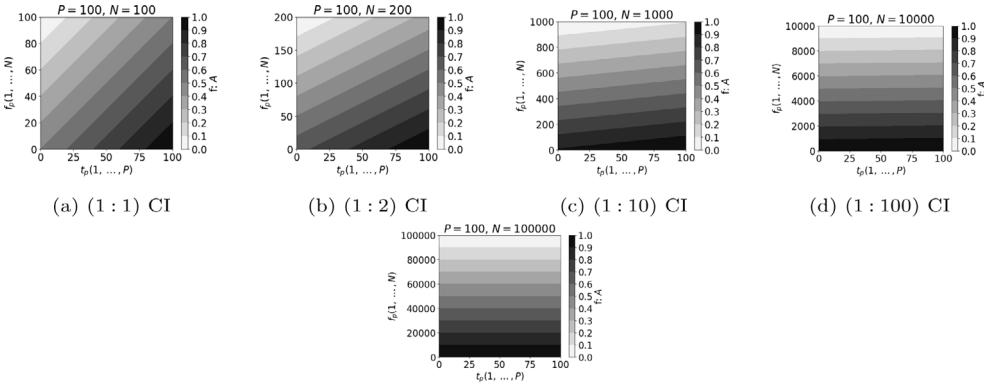
Metric	1:1000 imbalance		
	t_p	f_p	p
A	1.00 ± 0.01	0.00 ± 0.00	(1.00 : 0.00)
ν	0.74 ± 0.11	0.01 ± 0.00	(1.00 : 0.02)
B	0.50 ± 0.01	0.50 ± 0.01	(1.00 : 1.00)
F_1	0.78 ± 0.09	0.02 ± 0.01	(1.00 : 0.03)
κ	0.78 ± 0.09	0.02 ± 0.00	(1.00 : 0.02)
L	0.86 ± 0.15	0.01 ± 0.00	(1.00 : 0.01)
MCC	0.70 ± 0.03	0.20 ± 0.01	(1.00 : 0.28)
MK	0.86 ± 0.09	0.01 ± 0.00	(1.00 : 0.02)
FMI	0.61 ± 0.05	0.21 ± 0.01	(1.00 : 0.35)
OP	0.73 ± 0.01	0.10 ± 0.01	(1.00 : 0.13)
MCC-F_1	0.76 ± 0.07	0.08 ± 0.01	(1.00 : 0.10)
G_{SS}	0.47 ± 0.01	0.47 ± 0.01	(1.00 : 1.00)
IBA	0.19 ± 0.01	0.68 ± 0.01	(1.00 : 3.61)
ν_t	0.10 ± 0.00	0.86 ± 0.01	(1.00 : 8.19)
F_t	0.08 ± 0.00	0.91 ± 0.01	(1.00 : 11.78)
κ_t	0.50 ± 0.01	0.50 ± 0.01	(1.00 : 1.00)
L_t	0.50 ± 0.01	0.50 ± 0.01	(1.00 : 1.00)
MCC_t	0.50 ± 0.01	0.50 ± 0.01	(1.00 : 1.00)
OP_t	0.39 ± 0.01	0.39 ± 0.01	(1.00 : 1.00)
MCC-F_t	0.24 ± 0.01	0.76 ± 0.01	(1.00 : 3.20)
A_{PR}	0.01 ± 0.00	0.99 ± 0.01	(1.00 : 203.99)
Q_{PR}	0.00 ± 0.00	1.00 ± 0.01	(1.00 : 378.54)
H_{SS}	0.44 ± 0.01	0.44 ± 0.01	(1.00 : 1.00)
Q_{SS}	0.49 ± 0.01	0.49 ± 0.01	(1.00 : 1.00)

Appendix A. S'SA importance scores

This appendix provides the importance scores from the S'SA evaluations in tabular form. The importance scores for the true positive and false positive (I_{t_p} and I_{t_p}) input variables are provided, as well as the ratio between these scores. Importance scores are calculated as outlined in Section 7.

Appendix B. Contour plots

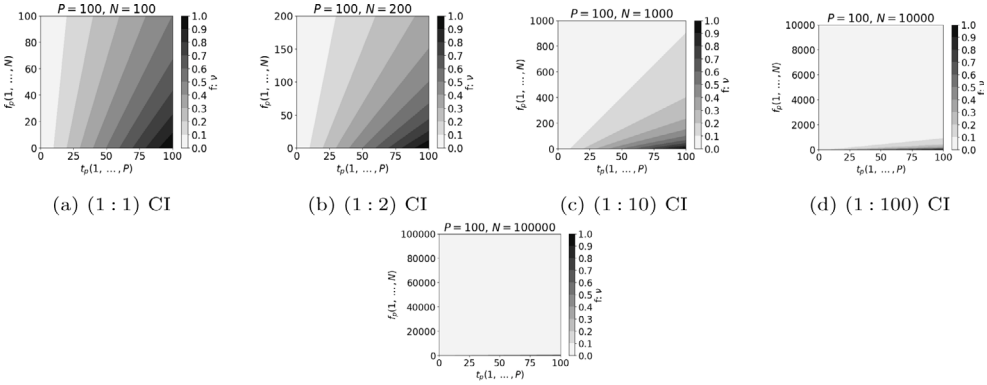
This appendix provides the contour plots for each of the evaluated metrics under the different levels of CI. The contour plots are generated by plotting the ranges of function values at different coordinates.



(a) (1 : 1) CI (b) (1 : 2) CI (c) (1 : 10) CI (d) (1 : 100) CI

(e) (1 : 1000) CI

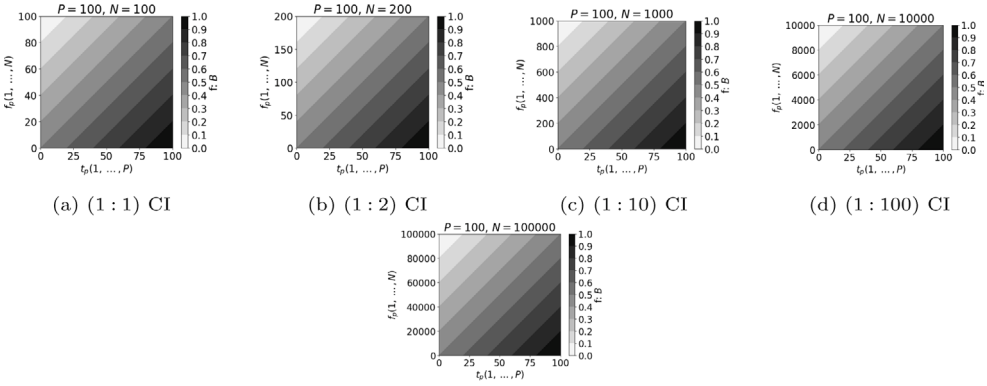
Fig. B.3. Contour plots of A behaviour under CI.



(a) (1 : 1) CI (b) (1 : 2) CI (c) (1 : 10) CI (d) (1 : 100) CI

(e) (1 : 1000) CI

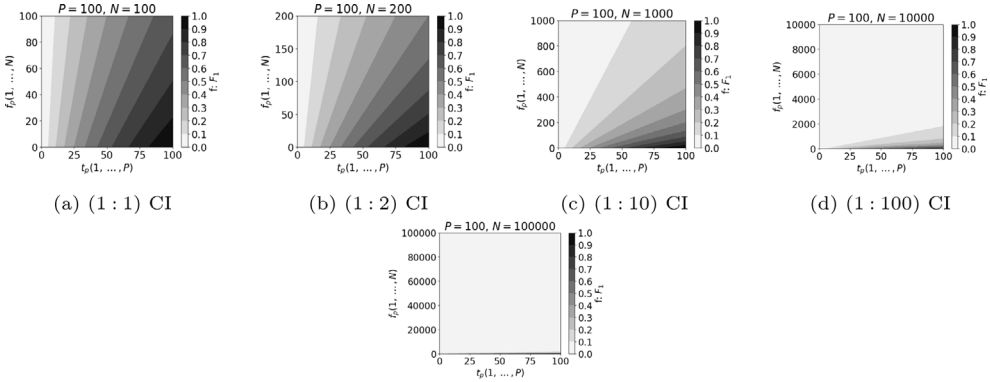
Fig. B.4. Contour plots of ν behaviour under CI.



(a) (1 : 1) CI (b) (1 : 2) CI (c) (1 : 10) CI (d) (1 : 100) CI

(e) (1 : 1000) CI

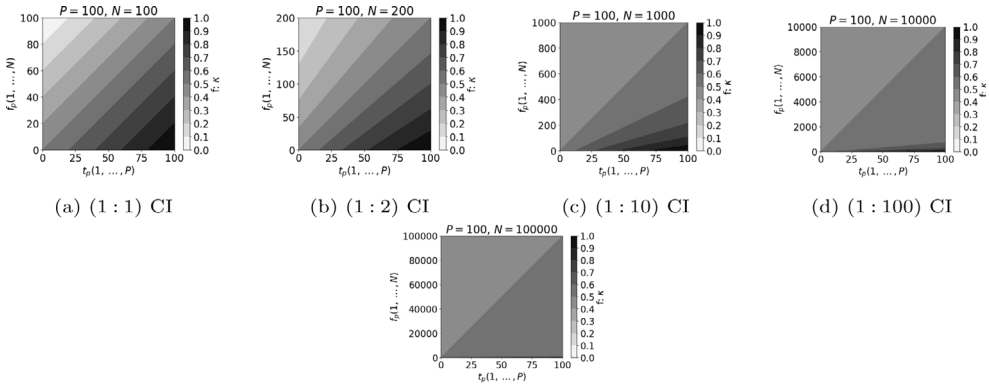
Fig. B.5. Contour plots of B behaviour under CI.



(a) (1 : 1) CI (b) (1 : 2) CI (c) (1 : 10) CI (d) (1 : 100) CI

(e) (1 : 1000) CI

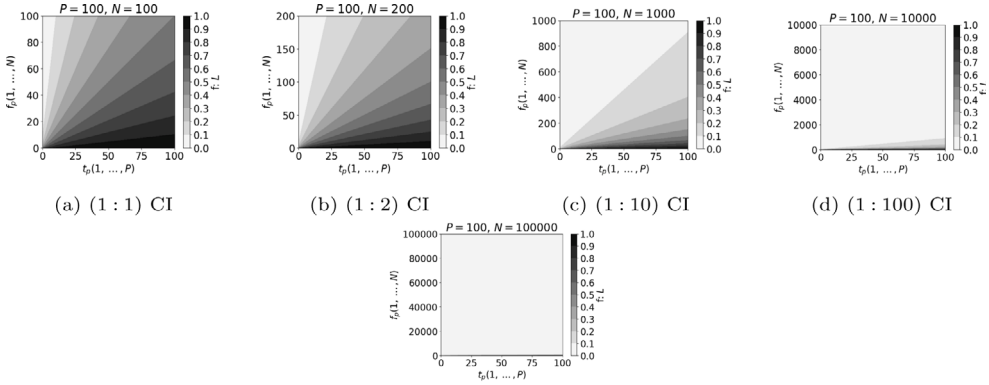
Fig. B.6. Contour plots of F_1 behaviour under CI.



(a) (1 : 1) CI (b) (1 : 2) CI (c) (1 : 10) CI (d) (1 : 100) CI

(e) (1 : 1000) CI

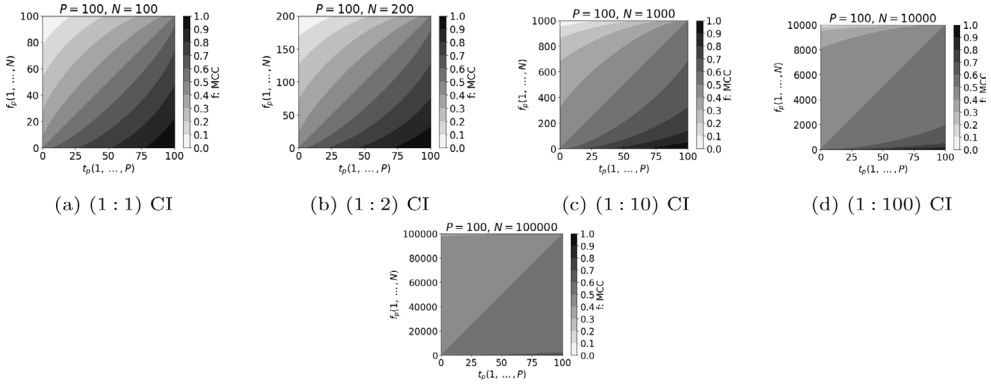
Fig. B.7. Contour plots of κ behaviour under CI.



(a) (1 : 1) CI (b) (1 : 2) CI (c) (1 : 10) CI (d) (1 : 100) CI

(e) (1 : 1000) CI

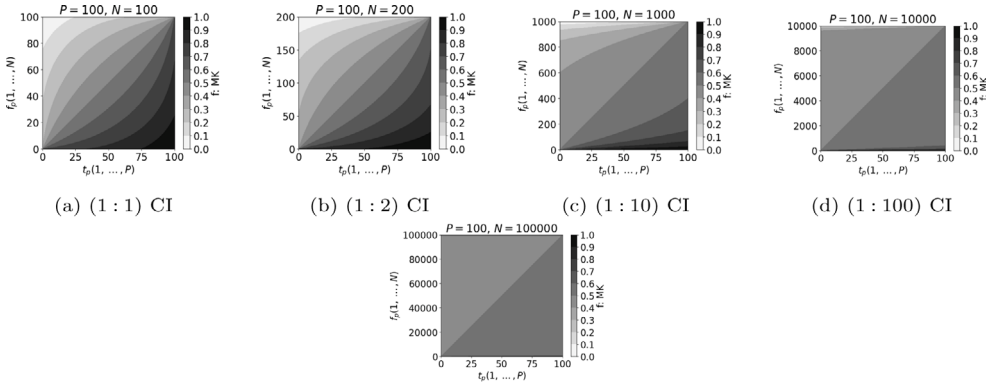
Fig. B.8. Contour plots of L behaviour under CI.



(a) (1 : 1) CI (b) (1 : 2) CI (c) (1 : 10) CI (d) (1 : 100) CI

(e) (1 : 1000) CI

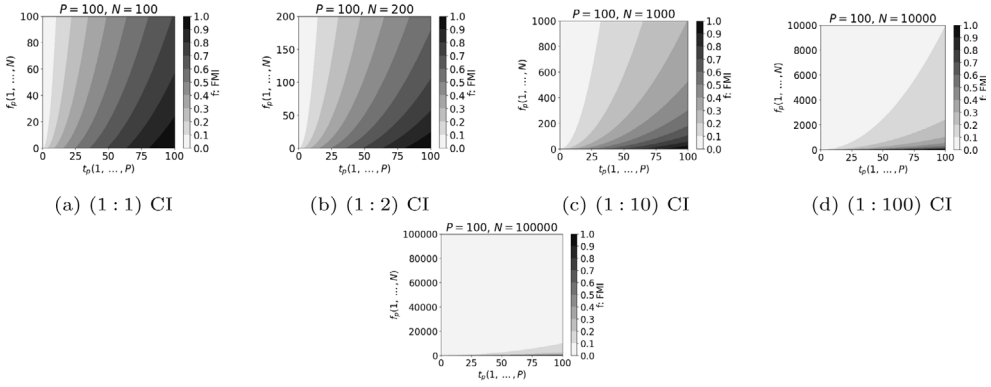
Fig. B.9. Contour plots of MCC behaviour under CI.



(a) (1 : 1) CI (b) (1 : 2) CI (c) (1 : 10) CI (d) (1 : 100) CI

(e) (1 : 1000) CI

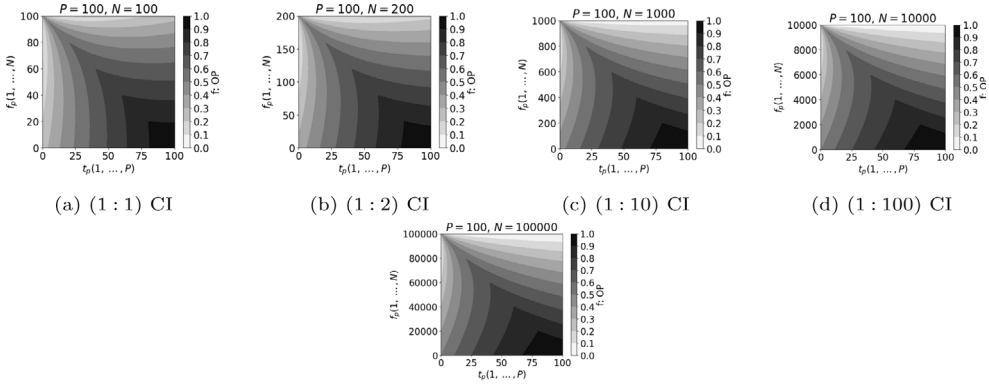
Fig. B.10. Contour plots of MK behaviour under CI.



(a) (1 : 1) CI (b) (1 : 2) CI (c) (1 : 10) CI (d) (1 : 100) CI

(e) (1 : 1000) CI

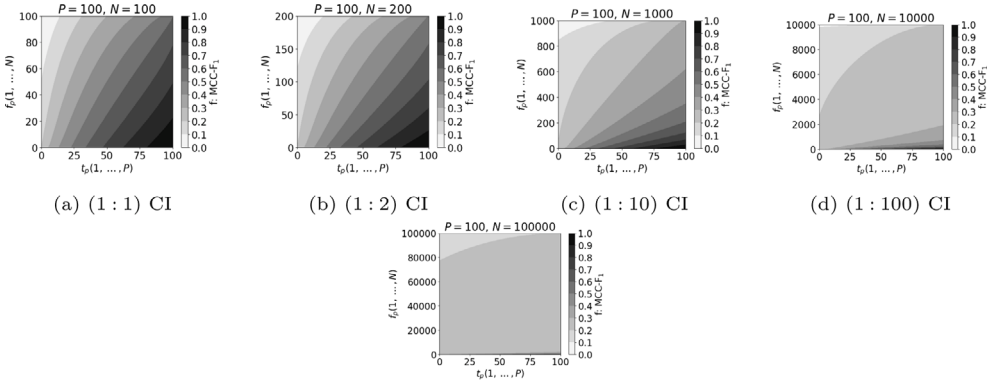
Fig. B.11. Contour plots of FMI behaviour under CI.



(a) (1 : 1) CI (b) (1 : 2) CI (c) (1 : 10) CI (d) (1 : 100) CI

(e) (1 : 1000) CI

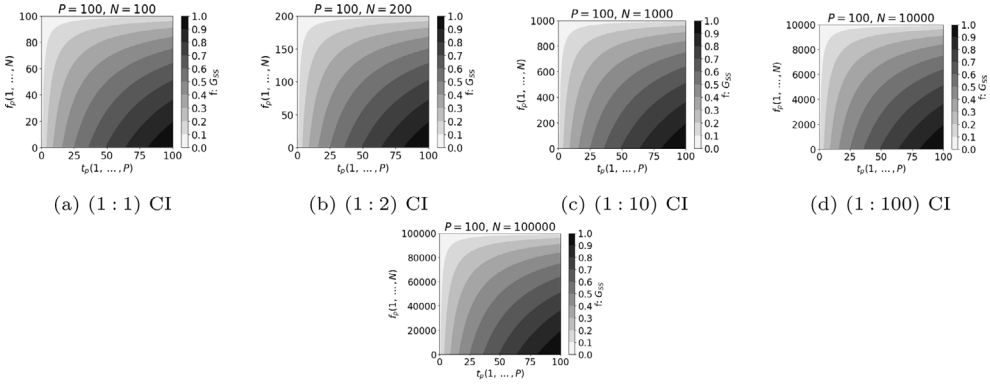
Fig. B.12. Contour plots of OP behaviour under CI.



(a) (1 : 1) CI (b) (1 : 2) CI (c) (1 : 10) CI (d) (1 : 100) CI

(e) (1 : 1000) CI

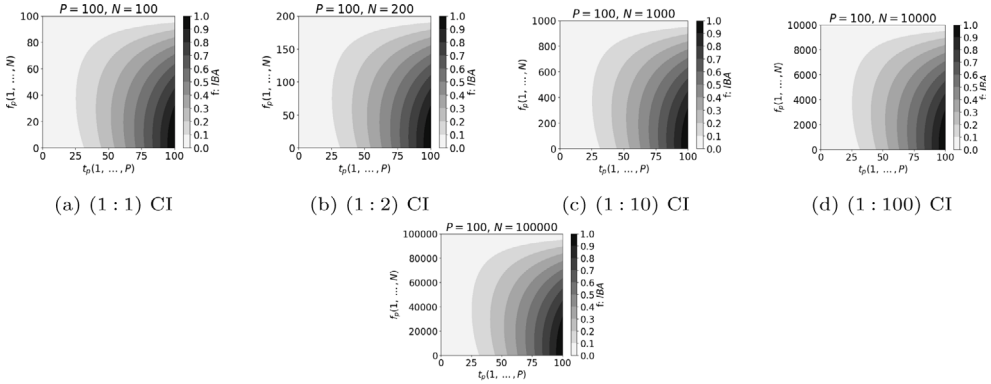
Fig. B.13. Contour plots of MCC-F_1 behaviour under CI.



(a) (1 : 1) CI (b) (1 : 2) CI (c) (1 : 10) CI (d) (1 : 100) CI

(e) (1 : 1000) CI

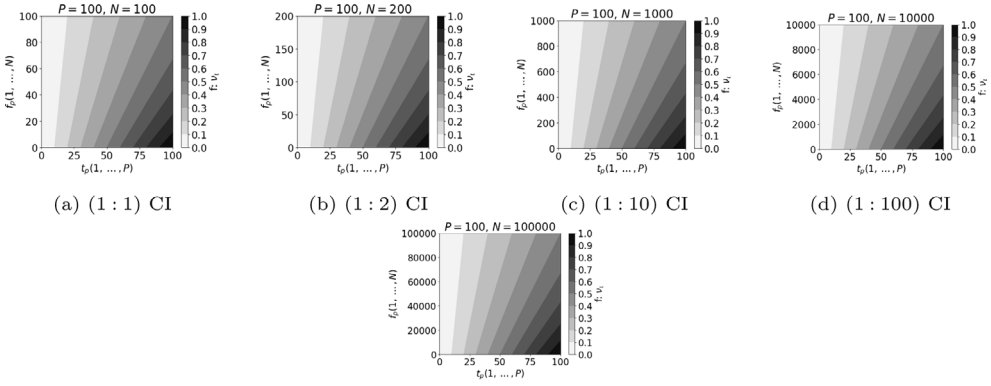
Fig. B.14. Contour plots of G_{SS} behaviour under CI.



(a) (1 : 1) CI (b) (1 : 2) CI (c) (1 : 10) CI (d) (1 : 100) CI

(e) (1 : 1000) CI

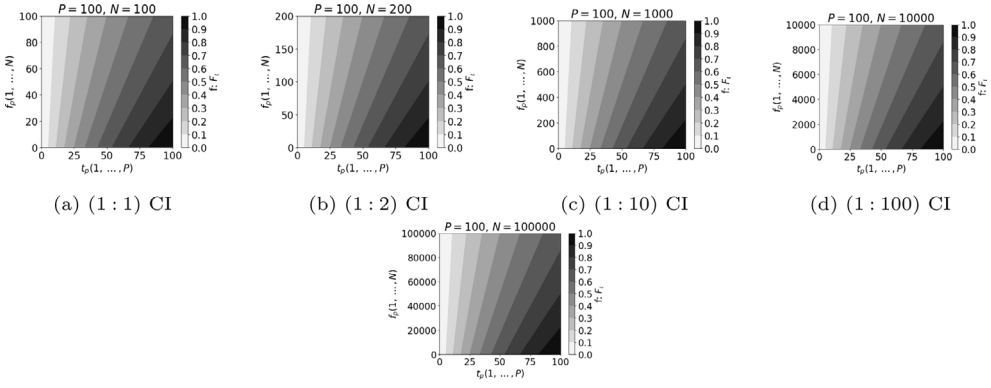
Fig. B.15. Contour plots of IBA behaviour under CI.



(a) (1 : 1) CI (b) (1 : 2) CI (c) (1 : 10) CI (d) (1 : 100) CI

(e) (1 : 1000) CI

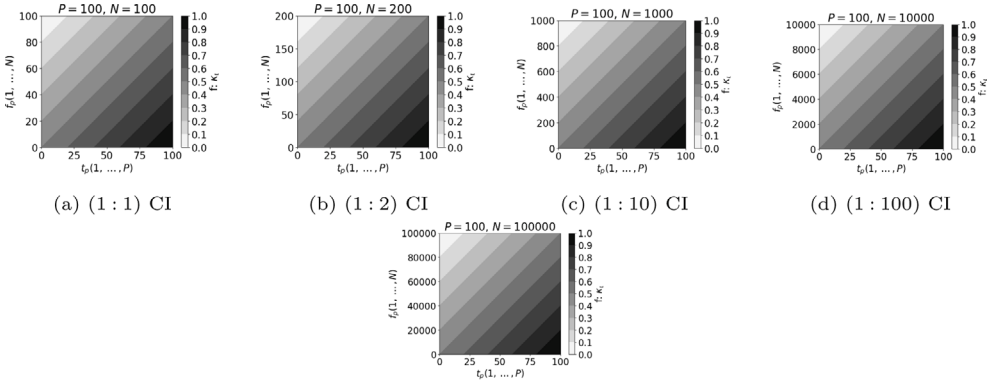
Fig. B.16. Contour plots of v_i behaviour under CI.



(a) (1 : 1) CI (b) (1 : 2) CI (c) (1 : 10) CI (d) (1 : 100) CI

(e) (1 : 1000) CI

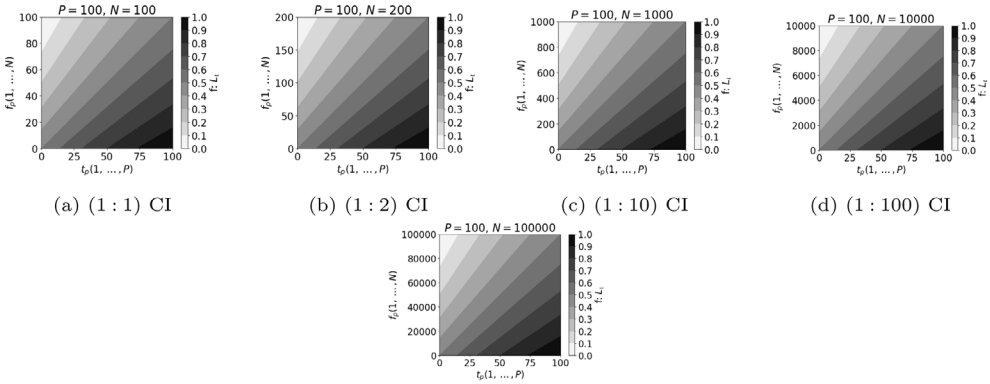
Fig. B.17. Contour plots of F_i behaviour under CI.



(a) (1 : 1) CI (b) (1 : 2) CI (c) (1 : 10) CI (d) (1 : 100) CI

(e) (1 : 1000) CI

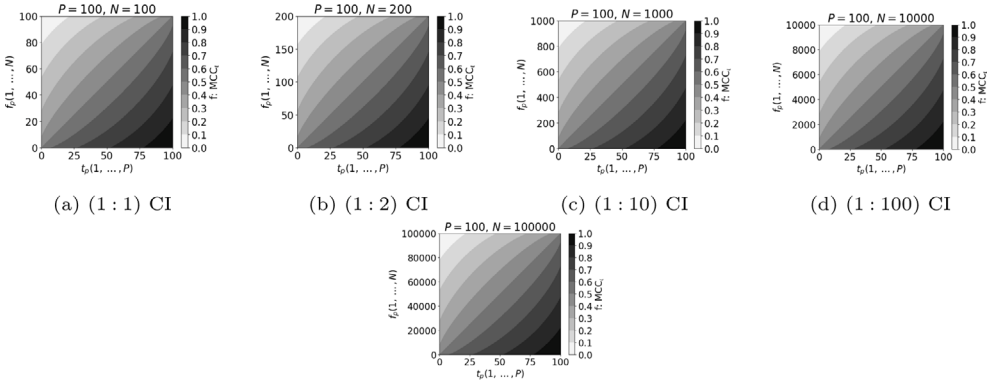
Fig. B.18. Contour plots of κ_t behaviour under CI.



(a) (1 : 1) CI (b) (1 : 2) CI (c) (1 : 10) CI (d) (1 : 100) CI

(e) (1 : 1000) CI

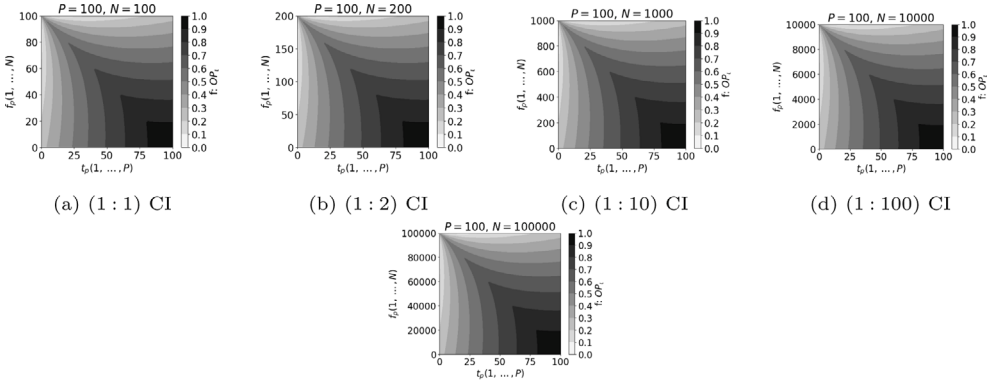
Fig. B.19. Contour plots of L_t behaviour under CI.



(a) (1 : 1) CI (b) (1 : 2) CI (c) (1 : 10) CI (d) (1 : 100) CI

(e) (1 : 1000) CI

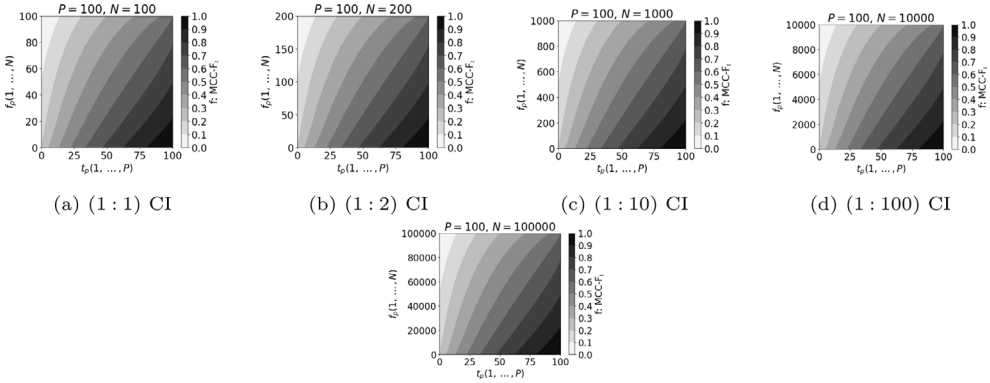
Fig. B.20. Contour plots of MCC_t behaviour under CI.



(a) (1 : 1) CI (b) (1 : 2) CI (c) (1 : 10) CI (d) (1 : 100) CI

(e) (1 : 1000) CI

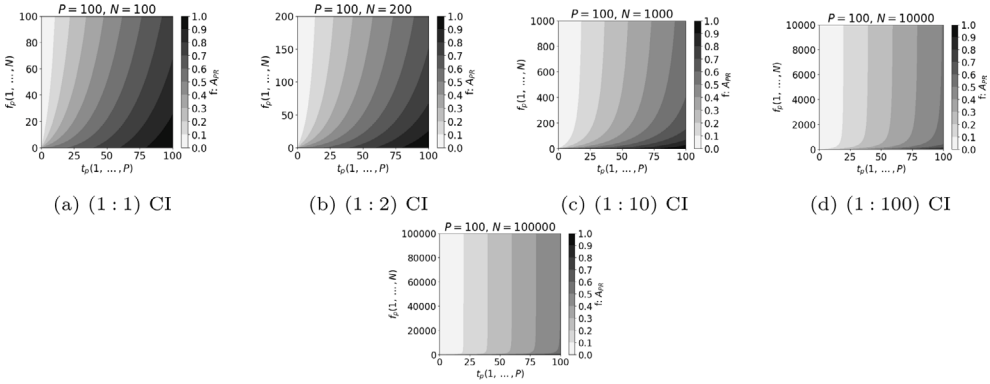
Fig. B.21. Contour plots of OP_t behaviour under CI.



(a) (1 : 1) CI (b) (1 : 2) CI (c) (1 : 10) CI (d) (1 : 100) CI

(e) (1 : 1000) CI

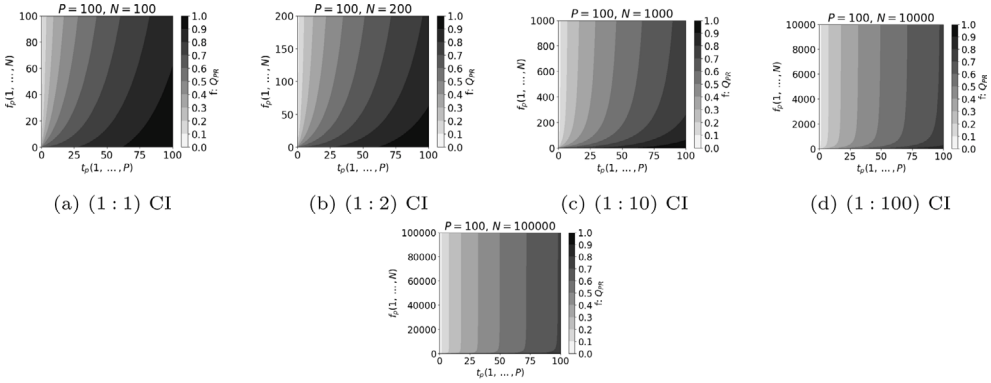
Fig. B.22. Contour plots of MCC-F_t behaviour under CI.



(a) (1 : 1) CI (b) (1 : 2) CI (c) (1 : 10) CI (d) (1 : 100) CI

(e) (1 : 1000) CI

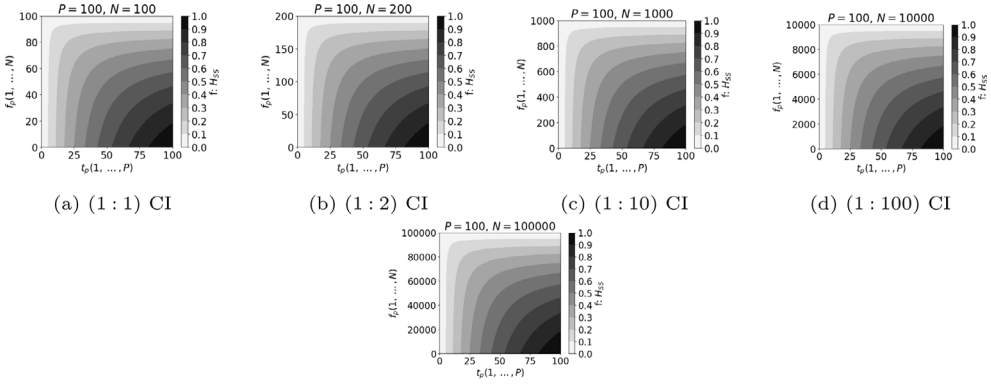
Fig. B.23. Contour plots of A_{PR} behaviour under CI.



(a) (1 : 1) CI (b) (1 : 2) CI (c) (1 : 10) CI (d) (1 : 100) CI

(e) (1 : 1000) CI

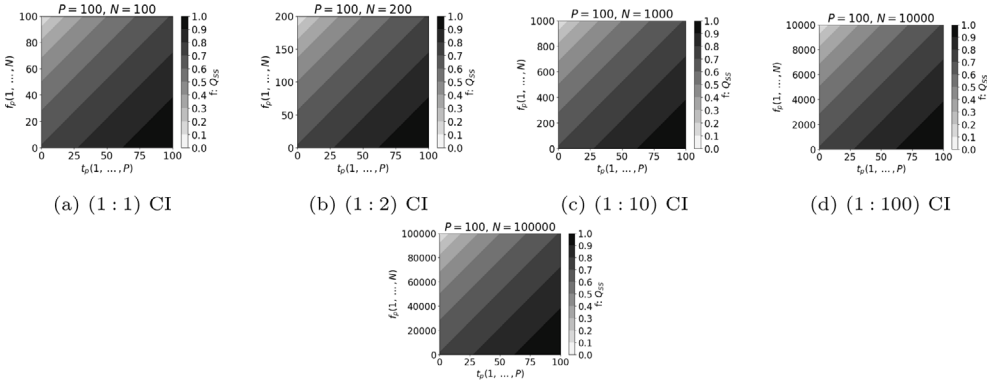
Fig. B.24. Contour plots of Q_{PR} behaviour under CI.



(a) (1 : 1) CI (b) (1 : 2) CI (c) (1 : 10) CI (d) (1 : 100) CI

(e) (1 : 1000) CI

Fig. B.25. Contour plots of H_{SS} behaviour under CI.



(a) (1 : 1) CI (b) (1 : 2) CI (c) (1 : 10) CI (d) (1 : 100) CI

(e) (1 : 1000) CI

Fig. B.26. Contour plots of Q_{SS} behaviour under CI.

References

[1] Leevy JL, Khoshgoftaar TM, Bauder RA, Seliya N. A survey on addressing high-class imbalance in big data. J Big Data 2018;5(1):1–30.

[2] Kotsiantis S, Kanellopoulos D, Pintelas P, et al. Handling imbalanced datasets: A review. GESTS Int Trans Comput Sci Eng 2006;30(1):25–36.

[3] Kubat M, Holte RC, Matwin S. Machine learning for the detection of oil spills in satellite radar images. Mach Learn 1998;30(2):195–215.

[4] Fotouhi S, Asadi S, Kattan MW. A comprehensive data level analysis for cancer diagnosis on imbalanced data. J Biomed Informatics 2019;90:103089.

[5] Herland M, Khoshgoftaar TM, Bauder RA. Big data fraud detection using multiple medicare data sources. J Big Data 2018;5(1):1–21.

[6] Wei W, Li J, Cao L, Ou Y, Chen J. Effective detection of sophisticated online banking fraud on extremely imbalanced data. World Wide Web 2013;16(4):449–75.

[7] Cieslak D, Chawla N, Striegel A. Combating imbalance in network intrusion datasets. In: Proceedings of the IEEE international conference on granular computing. 2006, p. 732–7.

[8] Bekkar M, Djemaa HK, Alitouche TA. Evaluation measures for models assessment over imbalanced data sets. J Inf Eng Appl 2013;3(10).

[9] Japkowicz N. Assessment metrics for imbalanced learning. Imbalanced Learn: Found Algorithms, Appl 2013;187–206.

[10] Swets JA. Measuring the accuracy of diagnostic systems. Sci. 1988;240(4857):1285–93.

[11] Brzezinski D, Stefanowski J, Susmaga R, Szczech I. Visual-based analysis of classification measures and their properties for class imbalanced problems. Inform Sci 2018;462:242–61.

[12] Ahmadzadeh A, Angryk RA. Measuring class-imbalance sensitivity of deterministic performance evaluation metrics. In: 2022 IEEE international conference on image processing. IEEE; 2022, p. 51–5.

[13] Brzezinski D, Stefanowski J, Susmaga R, Szczech I. On the dynamics of classification measures for imbalanced and streaming data. IEEE Trans Neural Networks Learn Syst 2019;31(8):2868–78.

[14] Luque A, Carrasco A, Martín A, de Las Heras A. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. Pattern Recognit 2019;91:216–31.

[15] Chandola V, Banerjee A, Kumar V. Anomaly detection: A survey. ACM Comput Surv 2009;41(3):1–58.

[16] Kubat M, Matwin S, et al. Addressing the curse of imbalanced training sets: one-sided selection. In: Proceedings of the international conference on machine learning. vol. 97, 1997, p. 179.

[17] Weiss GM. Mining with rarity: a unifying framework. ACM SigKDD Explor Newsl 2004;6(1):7–19.

[18] Li S, Zhou G, Wang Z, Lee SYM, Wang R. Imbalanced sentiment classification. In: Proceedings of the 20th ACM international conference on information and knowledge management. 2011, p. 2469–72.

[19] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. J Artificial Intelligence Res 2002;16:321–57.

[20] Zhang C, Soda P, Bi J, Fan G, Almpanidis G, García S, Ding W. An empirical study on the joint impact of feature selection and data resampling on imbalance classification. Appl Intell 2023;53(5):5449–61.

[21] Dube L, Verster T. Enhancing classification performance in imbalanced datasets: A comparative analysis of machine learning models. Data Sci Financ Econ 2023;3(4):354–79.

[22] de la Cruz Huayanay A, Bazán JL, Russo CM. Performance of evaluation metrics for classification in imbalanced data. Comput Statist 2024;1–27.

[23] Siblini W, Fréry J, He-Guelton L, Oblé F, Wang Y-Q. Master your metrics with calibration. In: International symposium on intelligent data analysis. Springer; 2020, p. 457–69.

[24] Neyman J, Pearson ES. IX. On the problem of the most efficient tests of statistical hypotheses. Philos Trans R Soc Lond Ser A, Contain Pap A Math Or Phys Character 1933;231(694–706):289–337.

[25] Gu Q, Zhu L, Cai Z. Evaluation measures of the classification performance of imbalanced data sets. In: Computational intelligence and intelligent systems: 4th international symposium, ISICA 2009, huangshi, China, October 23-25, 2009. proceedings 4. Springer; 2009, p. 461–71.

[26] Fürnkranz J. Separate-and-conquer rule learning. Artif Intell Rev 1999;13:3–54.

[27] Fürnkranz J, Flach PA. An analysis of rule evaluation metrics. In: Proceedings of the 20th international conference on machine learning. 2003, p. 202–9.

[28] Canbek G, Taskaya Temizel T, Sagiroglu S. PToPI: A comprehensive review, analysis, and knowledge representation of binary classification performance measures/metrics. SN Comput Sci 2022;4(1):13.

[29] Sobol IM. On sensitivity estimation for nonlinear mathematical models. Mat Model 1990;2(1):112–8.

[30] Wu MC, Lee S, Cai T, Li Y, Boehnke M, Lin X. Rare-variant association testing for sequencing data with the sequence kernel association test. Am J Hum Genet 2011;89(1):82–93.

[31] Luo L, Zhu Y, Xiong M. Quantitative trait locus analysis for next-generation sequencing with the functional linear models. J Med Genet 2012;49(8):513–24.

[32] Nguyen X, Gelfand AE. Bayesian nonparametric modeling for functional analysis of variance. Ann Inst Statist Math 2014;66(3):495–526.

[33] Herman J, Usher W. SALib: An open-source python library for sensitivity analysis. J Open Source Softw 2017;2(9).

[34] Iwanaga T, Usher W, Herman J. Toward SALib 2.0: Advancing the accessibility and interpretability of global sensitivity analyses. Socio- Environ Syst Model 2022;4:1–15.

[35] Sobol IM. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. Math Comput Simulation 2001;55(1–3):271–80.

[36] Saltelli A. Making best use of model evaluations to compute sensitivity indices. Comput Phys Comm 2002;145(2):280–97.

[37] Campolongo F, Saltelli A, Cariboni J. From screening to quantitative sensitivity analysis. a unified approach. Comput Phys Comm 2011;182(4):978–88.

[38] Owen AB. On dropping the first sobol' point. In: International conference on Monte Carlo and quasi-Monte Carlo methods in scientific computing. Springer; 2020, p. 71–86.

[39] Saltelli A, Annoni P, Azzini I, Campolongo F, Ratto M, Tarantola S. Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. Comput Phys Comm 2010;181(2):259–70.

[40] Sobol' I, Tarantola S, Gatelli D, Kucherenko S, Mauntz W. Estimating the approximation error when fixing unessential factors in global sensitivity analysis. Reliab Eng Syst Saf 2007;92(7):957–60.

[41] Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain.. Psychol Rev 1958;65(6):386.

[42] Murphy AH. The finley affair: A signal event in the history of forecast verification. Weather Forecast 1996;11(1):3–20.

[43] Gilbert GK. Finley's tornado predictions.. Am Meteorol J 1884;1(5):166.

[44] Finley JP. Tornado predictions.. Am Meteorol J 1884;1(3):85.

[45] Palmer W, Allen R. Note on the accuracy of forecasts concerning the rain problem. US Weather Bur 1949;4.

[46] Donaldson R, Dyer RM, Kraus MJ. Objective evaluator of techniques for predicting severe weather events. In: Proceedings of the bulletin of the American meteorological society. vol. 56, Amer Meteorological Soc 45 Beacon St, Boston, MA 02108-3693; 1975, 755–755.

[47] Peirce CS. The numerical measure of the success of predictions. Sci. 1884;4(93):453–4.

[48] Youden WJ. Index for rating diagnostic tests. Cancer 1950;3(1):32–5.

[49] Šimundić A. Measures of diagnostic accuracy: basic definitions. Electron J Int Fed Clin Chem Lab Med 2009;19(4):203.

[50] Powers D. Recall and precision versus the bookmaker. In: Proceedings of the cognitive science. 2003, p. 529–34.

[51] Peterson W, Birdsall T, Fox W. The theory of signal detectability. Trans the IRE Prof Group Inf Theory 1954;4(4):171–212.

[52] Tanner Jr. WP, Swets JA. A decision-making theory of visual detection.. Psychol Rev 1954;61(6):401.

[53] Hajian-Tilaki K. Receiver operating characteristic (ROC) curve analysis for medical diagnostic test evaluation. Casp J Intern Med 2013;4(2):627.

[54] Moses LE, Shapiro D, Littenberg B. Combining independent studies of a diagnostic test into a summary roc curve: Data-analytic approaches and some additional considerations. Stat Med 1993;12(14):1293–316.

[55] Hoo ZH, Candlish J, Teare D. What is an ROC curve? Emerg Med J 2017;34(6):357–9.

[56] Spackman KA. Signal detection theory: Valuable tools for evaluating inductive learning. In: Segre AM, editor. Proceedings of the sixth international workshop on machine learning. San Francisco (CA): Morgan Kaufmann; 1989, p. 160–3.

[57] Brodersen KH, Ong CS, Stephan KE, Buhmann JM. The balanced accuracy and its posterior distribution. In: 2010 20th international conference on pattern recognition. IEEE; 2010, p. 3121–4.

[58] Van Rijsbergen CJ. Information retrieval. Information retrieval. first ed.. London: Butterworths; 1975, p. 124.

[59] Cohen J. A coefficient of agreement for nominal scales. Educ Psychol Meas 1960;20(1):37–46.

[60] Heidke P. Berechnung des erfolges und der güte der windstärkevorhersagen im Sturmwarnungsdienst. Geogr Ann 1926;8(4):301–49.

[61] LaPlace PS. Essai philosophique sur les probabilites. Courcier; 1814.

[62] Chicco D, Warrens MJ, Jurman G. The matthews correlation coefficient (MCC) is more informative than cohen's kappa and brier score in binary classification assessment. IEEE Access 2021;9:78368–81.

[63] Pearson K. On the probable error of a coefficient of mean square contingency. Biometrika 1915;10(4):570–3.

[64] Yule GU. On the methods of measuring association between two attributes. J R Stat Soc 1912;75(6):579–652.

[65] Matthews B. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochim et Biophys Acta 1975;405(2):442–51.

[66] Baldi P, Brunak S, Chauvin Y, Andersen CA, Nielsen H. Assessing the accuracy of prediction algorithms for classification: an overview. Bioinform. 2000;16(5):412–24.

[67] Powers DM. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. Int J Mach Learn Technol 2011;2:37–63.

[68] Fowlkes EB, Mallows CL. A method for comparing two hierarchical clusterings. J Amer Statist Assoc 1983;78(383):553–69.

[69] Milligan GW, Soon SC, Sokol LM. The effect of cluster size, dimensionality, and the number of clusters on recovery of true cluster structure. IEEE Trans Pattern Anal Mach Intell 1983;PAMI-5(1):40–7.

[70] Ranawana R, Palade V. Optimized precision-a new measure for classifier performance evaluation. In: 2006 IEEE international conference on evolutionary computation. IEEE; 2006, p. 2254–61.

[71] Cao C, Chicco D, Hoffman MM. The MCC-F1 curve: a performance evaluation technique for binary classification. 2020, arXiv preprint arXiv:2006.11278.

[72] Kubat M, Holte R, Matwin S. Learning when negative examples abound. In: Proceedings of the European conference on machine learning. Springer; 1997, p. 146–53.

[73] García V, Mollineda RA, Sánchez JS. Index of balanced accuracy: A performance measure for skewed class distributions. In: Iberian conference on pattern recognition and image analysis. Springer; 2009, p. 441–8.

[74] García V, Mollineda RA, Sánchez JS. A bias correction function for classification performance assessment in two-class imbalanced problems. Knowl-Based Syst 2014;59:66–74.

[75] Buckland M, Gey F. The relationship between recall and precision. J Am Soc Inf Sci 1994;45(1):12–9.

[76] Welch BL. The generalization of 'student's' problem when several different population varlances are involved. Biometrika 1947;34(1–2):28–35.

[77] Stouffer SA, Suchman EA, DeVinney LC, Star SA, Williams Jr. RM. The American soldier: Adjustment during army life. vol. 1, Princeton University Press; 1949.
[78] Whitlock MC. Combining probability from independent tests: the weighted Z-method is superior to Fisher's approach. J Evol Biol 2005;18(5):1368–73.
[79] Heard NA, Rubin-Delanchy P. Choosing between methods of combining-values. Biometrika 2018;105(1):239–46.
[80] Dal Pozzolo A, Caelen O, Le Borgne Y-A, Waterschoot S, Bontempi G. Learned lessons in credit card fraud detection from a practitioner perspective. Expert Syst Appl 2014;41(10):4915–28.
[81] Quinlan R. Thyroid Disease. 1986, http://dx.doi.org/10.24432/C5D01, UCI Machine Learning Repository.
[82] Liu Z, Luo P, Wang X, Tang X. Deep learning face attributes in the wild. In: Proceedings of international conference on computer vision. ICCV, 2015, p. 3730–8.
[83] Moro S, Rita P, Cortez P. Bank Marketing. 2014, http://dx.doi.org/10.24432/C5K306, UCI Machine Learning Repository.
[84] Census-Income (KDD). 2000, http://dx.doi.org/10.24432/C5N30T, UCI Machine Learning Repository.
[85] Pang G, Shen C, Van Den Hengel A. Deep anomaly detection with deviation networks. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019, p. 353–62.
[86] Cramer JS. The origins of logistic regression. 2002.
[87] Liu DC, Nocedal J. On the limited memory BFGS method for large scale optimization. Math Program 1989;45(1):503–28.