MULTI-PERSPECTIVE ANALYSIS AND SYSTEMATIC BENCHMARKING FOR BINARY-CLASSIFICATION PERFORMANCE EVALUATION INSTRUMENTS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

GÜROL CANBEK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
THE INFORMATION SYSTEMS

SEPTEMBER 2019

MULTI-PERSPECTIVE ANALYSIS AND SYSTEMATIC BENCHMARKING FOR BINARY-CLASSIFICATION PERFORMANCE EVALUATION INSTRUMENTS

Submitted by **Gürol CANBEK** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Information Systems, Middle East Technical University** by,

Prof. Dr. Deniz ZEYREK BOZŞAHİN ____________________
Dean, **Informatics Institute**

Prof. Dr. Yasemin YARDIMCI ÇETİN ____________________
Head of Department, **Information Systems**

Assoc. Prof. Dr. Tuğba TAŞKAYA TEMİZEL ____________________
Supervisor, **Information Systems, METU**

Prof. Dr. Şeref SAĞIROĞLU ____________________
Co-supervisor, **Computer Engineering, Gazi University**

Examining Committee Members:

Assoc. Prof. Dr. Banu GÜNEL KILIÇ ____________________
Information Systems, METU

Assoc. Prof. Dr. Tuğba TAŞKAYA TEMİZEL ____________________
Information Systems, METU

Assist. Prof. Dr. Ercüment ÇİÇEK ____________________
Computer Engineering, Bilkent University

Assoc. Prof. Dr. Sevil ŞEN ____________________
Computer Engineering, Hacettepe University

Assoc. Prof. Dr. Altan KOÇYİĞİT ____________________
Information Systems, METU

Date: 02/09/2019

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this wok.

Name, Last name	**: Gürol CANBEK**
Signature	**: ___________________**

ABSTRACT

MULTI-PERSPECTIVE ANALYSIS AND SYSTEMATIC BENCHMARKING FOR BINARY-CLASSIFICATION PERFORMANCE EVALUATION INSTRUMENTS

Canbek, Gürol
Ph.D., Department of Information Systems
Supervisor: Assoc. Prof. Dr. Tuğba Taşkaya Temizel
Co-Supervisor: Prof. Dr. Şeref Sağıroğlu

September 2019, 136 pages

This thesis proposes novel methods to analyze and benchmark binary-classification performance evaluation instruments. It addresses critical problems found in the literature, clarifies terminology and distinguishes instruments as measure, metric, and as a new category indicator for the first time. The multi-perspective analysis introduces novel concepts such as canonical form, geometry, duality, complementation, dependency, and leveling with formal definitions as well as two new basic instruments. An indicator named Accuracy Barrier is also proposed and tested in re-evaluating performances of surveyed machine-learning classifications. An exploratory table is designed to represent all the concepts for over 50 instruments. The table's real use cases such as domain-specific metrics reporting are demonstrated. Furthermore, this thesis proposes a systematic benchmarking method comprising 3 stages to assess metrics' robustness over new concepts such as meta-metrics (metrics about metrics) and metric-space. Benchmarking 13 metrics reveals significant issues especially in accuracy, *F1*, and normalized mutual information conventional metrics and identifies Matthews Correlation Coefficient as the most robust metric. The benchmarking method is evaluated with the literature. Additionally, this thesis formally demonstrates publication and confirmation biases due to reporting non-robust metrics. Finally, this thesis gives recommendations on precise and concise performance evaluation, comparison, and reporting. The developed software library, analysis/benchmarking platform, visualization and calculator/dashboard tools, and datasets were also released online. This research is expected to re-establish and facilitate classification performance evaluation domain as well as contribute towards responsible open research in performance evaluation to use the most robust and objective instruments.

Keywords: Binary-classification, performance evaluation, performance metrics, machine learning, artificial intelligence

ÖZ

İKİLİ SINIFLANDIRMA BAŞARIM DEĞERLENDİRME ARAÇLARI İÇİN ÇOK PERSPEKTİFLİ ANALİZ VE SİSTEMATİK KIYASLAMA

Canbek, Gürol
Doktora, Bilişim Sistemleri Bölümü
Tez Yöneticisi: Doçent Dr. Tuğba Taşkaya Temizel
Ortak Tez Yöneticisi: Prof. Dr. Şeref Sağıroğlu

Eylül 2019, 136 sayfa

Bu tez, ikili sınıflandırma başarım değerlendirme araçlarının analizi ve kıyaslanması için yeni yöntemler önermektedir. Literatürden tespit edilen kritik sorunları ele alan çalışma, terminolojiyi açıklığa kavuşturmakta ve araçları ilk kez ölçü, ölçüt ve yeni bir kategori olarak gösterge şeklinde ayırt etmektedir. Çok perspektifli çözümleme; iki yeni araçla beraber kanonik biçim, geometri, ikilik, tümleme, bağımlılık ve seviyelendirme gibi yeni kavramları resmî tanımlarla tanıtmaktadır. Ayrıca, Doğruluk Engeli adında yeni bir gösterge önerilmekte ve etüt edilen makine öğrenmesi sınıflandırma çalışmaları üzerinden değerlendirilmektedir. Tüm önerilen kavramları 50 başarım aracı için gösteren bir keşif tablosu tasarlanmış ve tablonun sahaya özgü ölçütler gibi gerçek kullanım durumları gösterilmiştir. Tez, meta-ölçütler (ölçütler hakkında ölçütler) ve metrik uzayı gibi yeni kavramlarla ölçütlerin gürbüzlüğünü değerlendirmek ve karşılaştırmak için 3 aşamadan oluşan sistematik bir kıyaslama yöntemi önermektedir. 13 ölçütün kıyaslanması; doğruluk, *F1* ve normalleştirilmiş karşılıklı bilgi gibi yaygın kullanılan ölçütlerde kayda değer sorunları ortaya çıkarmakta ve Matthews Korelasyon Katsayısını en gürbüz ölçüt olarak belirlemektedir. Kıyaslama yöntemi, literatür ile karşılaştırılarak etraflı bir şekilde değerlendirilmiştir. Tez çalışmasında gürbüz olmayan ölçütlerin kullanımından kaynaklanan yayın önyargısı ve doğrulama sapması da resmî bir şekilde gösterilmektedir. Son olarak tez; kesin ve öz başarım değerlendirme, raporlama ve karşılaştırma konusunda önerilerde bulunmaktadır. Geliştirilen yazılım kütüphanesi, analiz/kıyaslama platformu, görselleştirme ve ölçüt hesaplama/gösterge araçları ve veri kümeleri çevrimiçi olarak yayımlanmıştır. Bu çalışmanın, ikili sınıflandırma başarım değerlendirme alanını temelden yeniden kurması ve kolaylaştırması yanında başarım değerlendirmesinde en gürbüz ve nesnel araç kullanımı ile sorumlu açık araştırmaya katkıda bulunması beklenmektedir.

Anahtar Kelimeler: İkili sınıflama, başarım değerlendirme, başarım ölçütleri, makine öğrenmesi, yapay zekâ

To my wife

ACKNOWLEDGMENTS

First of all, I would like to express my deepest gratitude to my supervisor Assoc. Prof. Dr. Tuğba TAŞKAYA TEMİZEL and co-supervisor Prof. Dr. Şeref SAĞIROĞLU who gave me their precious and continuous guidance, support, and motivation in all stages of this research study. I appreciate the invaluable contributions of my supervisor to the study and constant enlightening assistance of my co-supervisor. Working with both was a privileged experience.

I would like to thank Prof. Dr. Nazife BAYKAL for her support and guidance at the initial phase of the study. I am grateful to the thesis committee members, Assoc. Prof. Dr. Banu GÜNEL KILIÇ and Assoc. Prof. Dr. Sevil ŞEN for their valuable suggestions and positive feedback during the study and the examining committee members, Assoc. Prof. Dr. Altan KOÇYİĞİT and Assist. Prof. Dr. Ercüment ÇİÇEK for their helpful advice.

In particular, I would like to give very special thanks, love, and appreciation to my dear wife Elif. Words cannot express how grateful I am for all she provided during this enduring process.

And finally, big thanks to my lovely kids –the secret motivation of this study– for their understandings and pure love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF DEFINITIONS

LIST OF ABBREVIATIONS

ACC	Accuracy (efficiency, rand index)
ACCBAR	Accuracy Barrier
AML	Anti-Money Laundering
AUC	Area Under Curve
AUC-ROC	Area-Under-ROC-Curve
AUC-PR	Area-Under-Precision/Recall-Curve
BACC	Balanced Accuracy (strength)
BIAS	Bias
BM	Base Measures
CERN	the European Organization for Nuclear Research
CK	Cohen's Kappa (Heidke skill score, quality index)
CKc	Cohen's Kappa Chance
CPU	Central Processing Unit
CRISP-DM	CRoss-Industry Standard Process for Data Mining
CRR	(Correct) Rejection Rate
CSV	Comma Separated Values
DET	Determinant
DNA	Deoxyribo Nucleic Acid
dom	Domain
DP	Discriminant Power
DPR	D Prime
DR	Detection Rate
e.g.	“For example,” (abbreviation for *exempli gratia*)
ELISA	Enzyme-Linked Immunosorbent Assay
F1	F1 (F-score, F-measure, positive specific agreement)
FC	False Classification
FDR	False Discovery Rate
Fm	F-metrics
FN	False Negative
FNR	False Negative Rate
FOR	False Omission Rate (imprecision)
FP	False Positive
FPR	False Positive Rate
G	G-metric (G-mean, Fowlkes-Mallows index)
GB	Giga Byte
GHz	Giga Hertz
HC	Class Entropy
HIV	Human Immunodeficiency Virus
HO	Outcome Entropy

HOC	Joint Entropy
IBA	Index of Balanced Accuracy
i.e.	“That is,” (abbreviation for *id est*; used to give specific clarification)
INFORM	Informedness (Youden’s index, delta P', Peirce skill score)
LRN	Likelihood Ratio Negative
LRP	Likelihood Ratio Positive
MARK	Markedness (delta P, Clayton skill score, predictive summary index)
MCC	Matthews Correlation Coefficient (Phi correlation coefficient, Cohen’s index, Yule phi)
MCR	Misclassification Rate
MI	Mutual Information
ML	Machine Learning
N	Negative
N/A	Not Applicable or Not Available
NaN	Not-a-Number
NER	Null Error Rate
NIR	No Information Rate (non-information rate)
nMI	Normalized Mutual Information
NPV	Negative Predictive Value
OACC	Optimized Precision (Optimized Accuracy)
ON	Outcome Negative
OP	Outcome Positive
OR	Odds Ratio
p.	Page (in citations)
P	Positive
PPV	Positive Predictive Value (precision, confidence)
PREV	Prevalence
PToPI	Periodic Table of Performance Instruments
$\overline{\mathbb{R}}$	Affinely Extended Real Numbers ($\mathbb{R} \cup \{-\infty, +\infty\}$)
RAM	Random Access Memory
RGB	Red, Green, Blue
RMS	Root Mean Square
ROC	Receiver Operating Characteristic
RQ	Research Question
SAR	(combination of) Accuracy, Area Under ROC Curve, and Squared Error
SD	Standard Deviation
SKEW	(Class) Skew
SMS	Short Message Service
Sn	Sample Size
SRPM	Standardized Relative Performance Metric
TasKar	*Tasnif Karnesi* in Turkish (classification report)
TC	True Classification
TN	True Negative
TNR	True Negative Rate (inverse recall, specificity)
TP	True Positive
TPR	True Positive Rate (recall, sensitivity, hit rate, recognition rate)
UBMcorr	Universal Base Measure Correlations (Meta-metric-1)
UCons	Universal Consistency (Meta-metric-6)
UDisc	Universal Discriminancy (Meta-metric-7)
UDist	Universal Distinctness (Meta-metric-3)
UICons	Universal Inconsistency
UMono	Universal Monotonicity (Meta-metric-5)
UOsmo	Universal Output Smoothness (Meta-metric-4)

UPuncorr	Universal Prevalence Uncorrelation (Meta-metric-2)
vs.	“Versus”
WACC	Weighted Accuracy

CHAPTER 1

INTRODUCTION

This thesis is the result of a work spanning over five years. As a result of examining the relevant literature regarding the classification process of machine learning-based Android mobile malware detection, I saw important problems in applied methodologies and considerable gaps in the literature within each phase such as classification problem-domain taxonomies, sample collection and preprocessing, feature extraction and engineering, building datasets, modeling machine learning algorithms, and finally performance evaluation and tried to propose a systematic overall process. Afterward, looking further into the literature for each phase, independent from the classification problem domain, I found that there are significant problems in performance evaluation and hence focused on the performance evaluation instruments completely.

Thus, this thesis examines binary-classification performance evaluation instruments that are accepted as primary references which all researchers use to see what achieved for their evaluations as well as to refer them in reporting, comparing, and highlighting the performances. To begin with, briefly, no study in the literature makes a comprehensive evaluation of binary-classification performance evaluation instruments. Considering my literature review as well as observations, we (with my advisors) have seen that performance evaluation instruments should be revised again and so this has become the main motivation of the thesis.

Due to the new developments and increasing interest in machine learning algorithms such as deep learning, many researchers use or propose new machine learning algorithms for various problems. Since the general focus is on improving classification performance in a problem domain with their proposed methodology, researchers often refer to the metrics previously utilized in the papers they cited and do not question pertinence of these metrics for their problem. For example, some might claim 98% success with accuracy but another classification method in the same dataset could appear to achieve higher performance when a more appropriate metric is chosen.

With this respect, some practical questions such as “Are we sure of the instrument we used as a performance metric?”, “What are the drawbacks of specific instruments?”, “Is it an objective reference?”, “Does it match our specific requirements or goals?”, or “Is there any aspect that might affect the classification performance other than the metric used?” are not discussed or addressed in the previous studies.

Hence, researchers continue to use the legacy or *stock* metrics in existing domains or choose the metrics reported by previous works in new domains which are also inspired by other existing domains. For example, it is hardly possible to see a different metric other than *F1* reported in information retrieval domain. At the same time, it is difficult to know the reason behind using *F1* in other domains. Likewise, accuracy is still the most preferred metric in most of the domains. Note that the same problems are also valid in multi-label and/or multi-class classification performance evaluation.

Therefore, we can add theoretical or methodological questions about performance evaluation into the practical ones aforementioned above: “Do we agree on basic concepts, definitions and methods about performance instruments?”, “How to act while reporting the performance of a classification study or comparing among studies?”, “What other problems can be expected in performance evaluation?” or considering a large number of studies in the literature and ongoing researches and practices within each domain, “How can we be sure that scientific measurable progress is made?”. The literature does not elaborate on these questions specifically. Worse, as easily observed in any domain, a wide-spread confusion is seen even in fundamental terminology. Based on this main motivation, this thesis revisits performance evaluation more comprehensively and systematically and redefines the performance evaluation instruments from a broad perspective.

In all these respects, this thesis offers a new perspective to the literature by validating with scientific methods. The thesis is structured around this motivation and the research is guided by the following main research question:

RQ: How to establish and improve our knowledge on binary-classification performance instruments comprehensively and systematically in order to enable researchers to make informed decisions on choosing the right instrument(s) and follow objective approaches in performance evaluation, reporting, and comparison?

Binary-classification performance evaluation corresponds to the evaluation phase in CRISP-DM (CRoss-Industry Standard Process for Data Mining) comprising

- business understanding,
- data understanding,
- data preparation,
- modeling,
- evaluation, and
- deployment

phases (Huber, Wiemer, Schneider, & Ihlenfeldt, 2019).

As anyone who studies and conducts experiments in machine learning (ML) based binary-classification problems confirms that performance evaluation is an overlooked activity in the entire knowledge discovery process or ML workflow compared to others including data preparation/cleaning in preprocessing, feature engineering, and model building. Researchers usually focus, put efforts, and spends time on collecting samples, preprocessing, building and mining datasets, and refining the ML algorithms (CrowdFlower, 2016, 2017).

Considering performance evaluation, choice of a performance evaluation metric among a large number of alternatives is conventional or not explicit in many domains. Practitioners, as well as researchers, most likely think that performance evaluation is a well-studied and established topic without any uncertainties. For the practitioners, “accuracy” or “true positive rate" names might have a clear and convincing meaning for evaluating the performance of their classification applications[1].

From the research perspective, on the other hand, performance evaluation as a domain seems to have no area to improve or no gap to study further. The following headings highlight the seven main problems that are observed both in the literature and in practice and addressed in this thesis. Note that most of the problems are also clearly demonstrated over a case study domain in Section 2.3 and the preliminaries are summarized in Section 2.1. The main research question was formed based on the prominent arguments in these problems.

Note that italic terms are the performance instruments that have a limited range (*e.g.*, *ACC* in [0, 1]) whereas bold-italic terms refer to measures without a lower and/or upper limited (*e.g.*, ***FP*** = 44) and bold-only terms refer to corresponding metric-space that is proposed in this study (*e.g.*, **MCC**, *see* Section 5.1.1).

1. Confusing terminology: performance measure or performance metric?

In a general perspective, performance instruments are expressed by various terms such as "performance metrics", "performance measures", "evaluation measures", and "prediction scores", *etc*. Performance evaluation based on 2x2 contingency table is named as "diagnostic accuracy" or "test accuracy" in medicine (van Stralen et al., 2009) or "skill score" or "forecast skill" in meteorology (forecast *vs.* observation classes) (Wilks, 2006). Classification term itself is called as "categorization" in philosophy and statistics (Sammut & I.Webb, 2011). The lack of consensus in naming the instruments indicates a fundamental problem in performance evaluation.

Historically, evaluating the trends of different phrases expressed in the corpus of one million English books between 1930 and 2008 (Michel et al., 2011); "performance measure(s)", "performance indicator(s)", and "performance metric(s)" are the most frequent terminologies, which have been used since the 1950s, 1960s, and 1980s, respectively as shown in Figure 1.1[2]. Other evaluated phrases are "performance score", "evaluation measure", "skill score", "forecast score", and "prediction score". Thus, "performance metric" is rather a new phrase.

Concerning the literature in classification scope; it is observed that "measures", "indicators", "metrics", "scores", "criteria", "factors" or "indices" terms are used interchangeably. Even the studies related to classification performance use the related terms (especially "performance measures" and "performance metrics") interchangeably. There are review studies expressing performance instruments with different notations, abbreviations, and symbols (Powers, 2011).

[1] Nevertheless, confusion might also occur when it comes to different naming of the instruments such as "precision", "recall", "sensitivity", "specificity", "strength", "efficiency", *etc*.

[2] The books can be in any subject apart from classification performance.

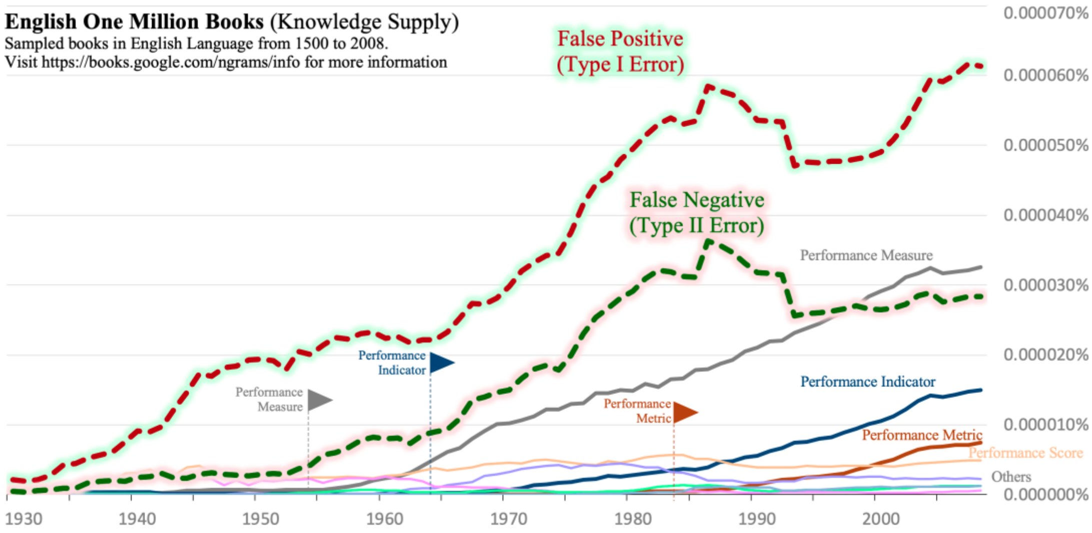


Figure 1.1 The trend of phrases mentioned in books: "False Positive" *vs.* "False Negative" and "Performance Measure" *vs.* "Performance Indicator" *vs.* "Performance Metric".

2. Disregarding negative-class performance, domain-specific tradeoffs, and end-user requirements

Performance of a classifier can be examined from the standpoint of failure instead of success. In this case, the number of false-classifications in positive and negative classes namely type I errors (***FP***) and type II errors (***FN***), respectively, become the foremost concern. However, it is common that type I errors are the main focus and type II errors are disregarded. Interestingly, for instance, people are more interested in type I errors than type II errors according to Google search engine trends since 2004 (78:29 on average) as shown in Figure 1.2.

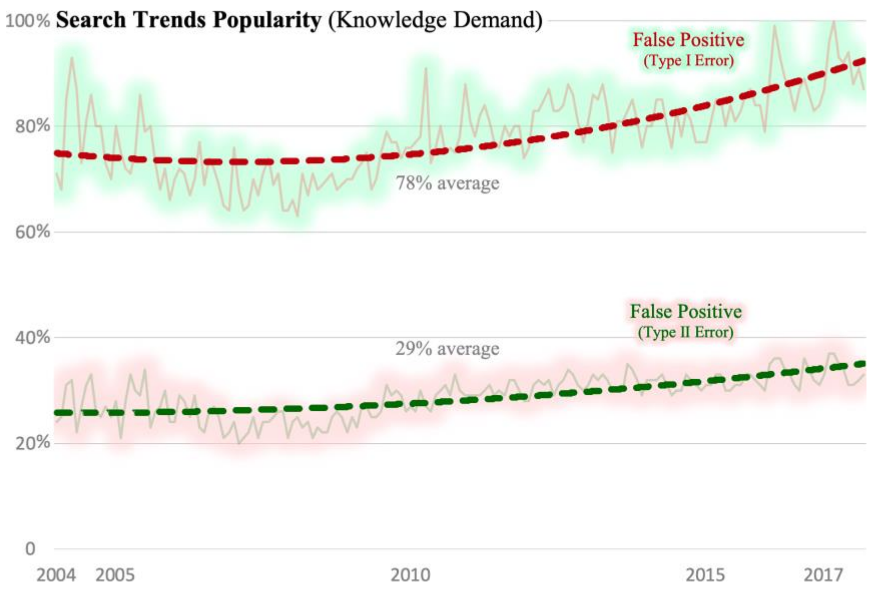


Figure 1.2 The search trends showing the interests to "false positive" and "false negative" according to Google search data worldwide between 2004 and September 2017. Y-axis shows the popularity of the search between 0% (none) to 100% (maximum)

The following are the top search suggestions associated with “False Positive in” and “False Negative in” phrases that I extracted in the Google search engine at the time of writing:

- *Type I error / “False Positive in”*: pregnancy test, network security, ELISA or HIV test, security, Anti-Money Laundering (AML), visual field test, psychology, indirect Coombs test, logistic regression, and software testing.

- *Type II error / “False Negative in”*: security, statics, a classification table, network security, psychology, software testing, object detection, and early pregnancy.

Each example areas actually shows the performance priorities of the end-users that should be considered for performance evaluation of the classifiers modeled in that scope. In some fields such as security, network security, and psychology, the top searches appearing in both error types suggest that those fields put equal emphasis on both types of errors. Thus, a successful classifier accepted by end-users in network security (*e.g.*, a network intrusion detection system), for instance, should minimize both error types, therefore the performance should be evaluated from both perspectives. Note that the trends can change afterward.

The dominance of type I errors over type II errors can also be seen in the corpus of English books from 1930 to 2008 as shown in Figure 1.1 above. Provided figures show that a sort of related knowledge supply (*i.e.* written books) and corresponding knowledge demand (*i.e.* search queries) attach more importance to false positives (type I errors) than false negatives (type II errors).

Besides, a tradeoff between type I and type II errors might be observed in each domain and its specific applications. Such as;

- In critical engineering and especially in medical research, type II errors can be more serious or worse than type I errors (*e.g.*, breast cancer diagnosis (N. Liu, Qi, Xu, Gao, & Liu, 2019)).

- In information retrieval applications such as document filtering, false positives might be critical (Kenter, Balog, & De Rijke, 2015).

- In malware analysis, it could be better to mistakenly label a “benign” software as “malign” (also known as malicious software or malware) than miss a malign software by incorrectly labeling it as benign (lower ***FP*** or type I error). Because labeled malware could be prioritized and an expert could go through further manual malware analysis to eliminate false positives (Yerima, Sezer, & McWilliams, 2014).

- An anti-malware product should be designed or configured to decrease ***FP*** to avoid annoying interruptions due to excessive malware warnings.

- In law or social perspective, the opposite (*i.e.* low type II errors against high type II errors) is likely to be valid to ensure the presumption of innocence in the same way as precautionary logic focuses on more underestimates (***FN***) than overestimates (***FP***) in criminal justice (Lomel, 2012).

Some classification applications might care for both error types equally. Thus, a performance instrument that is sensitive to both types should be used. Briefly, researchers might choose instruments without knowing the domain tradeoffs and user requirements and matching them to the chosen instruments. Hence, the expected performance of a classification application might not consider domain-specific conditions and end-users' perspectives.

3. Using instruments without being aware of the pros and cons

The performance instruments are selected according to the conventions per domains (*e.g.*, *F1* is frequently used in information retrieval domain) or the researchers unconsciously follow the practices of previous studies they would like to improve. The weak and strong characteristics of the instruments are not explicit in a broad perspective. Some instruments do not behave as expected in certain conditions or from specific aspects and mislead both authors and readers. For example, accuracy exhibits high performance in class imbalanced datasets (*i.e.* the number of positive samples are less than the negative ones).

Continuing using such *stock* instruments such as true positive rate or accuracy generates a saturation where the proposed classifiers' performances become closer to a maximum value (*e.g.*, 0.99 accuracy) that blurs the distinguished achievement of a remarkable study. Researchers need more granular instruments to identify the best classifier. Moreover, which performance instruments should be preferred is unknown if the standpoint of achievement (*i.e.* practical goal of a binary classification application) shifts into other aspects instead of in favor of true positives only such as

- False classifications or error types (false positive and false negative or type I and type II errors),
- Negative class performance at the same weight as for positive class, and/or
- Eliminating other external factors such as class imbalance.

The following four problems are especially observed in the literature.

4. Need for explaining performance instruments

In academic publications on binary-classification problems, it is frequently observed that researchers need explaining performance evaluation for the sake of completeness. Within a smaller or larger body of text, confusion matrix, performance instruments and their abbreviations, equations, and brief descriptions are expressed usually in a separate section of the articles. However, the terminology and notations vary unexpectedly. Moreover, this repeating section takes considerable space in the text, requires effort in every study, and takes the time of not only the authors but also the reviewers and readers.

5. Indeterministic performance reporting and comparison

With respect to performance publication, I have not seen any consensus on how many and what instruments should be used in reporting performances. The number of instruments reported and the instruments selected vary from study to study[3]. Because comparisons of performances of different classifiers in terms of different instruments (*e.g.*, accuracy or *F1*) yield different results, it is unclear which instrument should be used for the ultimate ranking. In a broader scope, the relations among such a large number of instruments are also not truly explored. Similar metrics might be used redundantly to report the performance.

[3] An interactive graphic is prepared and released online at http://bit.ly/performanceranks to show the ranks of mobile malware classification studies in terms of different performance metrics

6. The gap in responsible open research

As an up-to-date development in scientific studies, the initiatives such as OpenAIRE[4] by European Union and Zenodo[5] by CERN aim common, responsible, and reproducible open research approaches where research data become available to all researchers. While these initiatives encourage researcher community to share their studies along with the datasets to establish a widely common platform, we could not see the same efforts in developing a common standard for evaluating the performance of those studies. Yet, scientific progress cannot be achieved in the right direction unless the objective comparison methodologies are determined clearly and followed by all the researchers.

7. The complexity of the performance instruments

From the practical point of view, the practitioners who are not experienced in statistics need assistance to report the classification performance while researchers need also assistance to deep dive into the instruments' specifications in order to select the most appropriate performance instrument(s) according to their objectives and/or domain-specific requirements. For example, which metric or metrics should be used among *TPR*, *PPV*, and *ACC* and what their differences are not clear.

1.1 Research Questions

As already mentioned at the start of this chapter, this research addressing the problems above is guided by the following main research question:

- **RQ**: How to establish and improve our knowledge on binary-classification performance instruments comprehensively and systematically in order to enable researchers to make informed decisions on choosing the right instrument(s) and follow objective approaches in performance evaluation, reporting, and comparison?

The specific research questions for binary-classification performance evaluation as an enumerated list are as follows:

RQ1:

- What are the problems in performance evaluation reporting?

RQ2:

- Can classification performance evaluation terminology be clarified and improved?
- Are all the performance instruments the same semantically and formally?
- Are there any properties related to performance instruments that reveal and define their characteristics?
- Are there any similarities, relationships, and dependencies among the performance instruments?

RQ3:

- How to enhance comprehending, using, representing, reporting, learning, and teaching binary-classification performance instruments?

RQ4:

- Which instruments are robust to use in binary classification?
- What should be reported for expressing classification performance?

[4] https://www.openaire.eu
[5] https://zenodo.org

1.2 Research Contributions and Strategy

The thesis provides the following summarized contributions addressing the research questions:

- First, the problems in performance evaluation terminology and reporting are revealed specifically via a comprehensive survey in Android mobile malware detection as a typical and emerging example domain in binary-classification problems. Such a systematic survey is the first in the literature. The generic findings that are observed in other domains are clear evidence of the problems aforementioned above.

- Second, novel concepts are introduced via a multi-perspective analysis of performance evaluation instruments which is conducted on the largest set of instruments studied in the literature by far. Hence, the foundation of performance evaluation is completely defined for the first time both in a semantic and formal manner. The concepts introduce essential properties to comprehend and identify each of the instruments as well as to see the similarities and differences among them by categorizing the instruments from different perspectives. As a result of this breadth and depth analysis, the terminology is also clarified, existing instruments are categorized as "performance measures" and "performance metrics", and the representation of the instruments (*e.g.*, notation and visualizing) is standardized as a proposal.

- Third, two basic instruments, a new instrument category named "performance indicators", and a novel indicator called "Accuracy Barrier" as the first example of the new category are introduced to simplify and enhance the understanding of the instruments and avoiding common pitfalls that cause misleading performance evaluation. A case study is conducted to re-evaluate the surveyed classification studies via the proposed indicator. Moreover, aggregating all the concepts, an exploratory table for 50 binary-classification performance instruments called "PToPI" (Periodic Table of Performance Instruments), which is the pictorial specification or blueprint of instruments and their essential properties, is designed. The real use-cases of PToPI, which is a unique application of knowledge organization similar to the periodic table of elements, are also described. A handy calculator and dashboard tool called "TasKar" [6] to calculate and visualize classification results in terms of all the instruments is also designed. To enhance the interpretation of performance and subsequent performance variations, TasKar also visualizes the common performance metrics in new types of graphics proposed in this study. Note that both tools that can also be used for educational purposes are presented online to the research community.

- Forth, a novel systematic benchmarking method named "BenchMetric" for binary-classification performance metrics is proposed by introducing new concepts such as meta-metrics (*i.e.* metrics about performance metrics) and metric-space. BenchMetric method, tested on thirteen metrics (*TPR*, *TNR*, *PPV*, *NPV*, *ACC*, *INFORM*, *MARK*, *BACC*, *G*, *nMI*, *F1*, *CK*, and *MCC*) reveals interesting robust or non-robust behaviors even in common and/or suggested metrics. The method is comprehensively evaluated with the limited comparison approaches offered in the

[6] ***Tas**nif* ***Kar**nesi* in Turkish (classification report)

literature and it is also tested with recently proposed metrics in the literature. The results of both of the tests have shown that *MCC* is the most robust metric. The thesis is further notable to suggest what the optimal instruments are in classification performance reporting in academic or industrial studies. Moreover, it demonstrates via a case study that reporting biases such as confirmation and publication biases might occur in the literature where classification performances reported in terms of non-robust metrics. The last two contributions in performance reporting are expected to initiate discussions from responsible open research perspectives.

As a summary, this thesis study comprises one exploratory study via multi-perspective analysis, two complementing tools, three surveys, three case studies, and two experiments for the benchmarking all of which were performed in order to explore, generalize and validate the proposed concepts, instruments, tools, and methods.

1.3 Research Objectives

This thesis is intended to

- make the research community understand the criticality of performance evaluation instruments and aware of the fundamental but overlooked problems in theory and in practice which cause misleading results,
- provide novel concepts from multiple perspectives to increase our overall understanding of a large number of performance evaluation instruments and their characteristics,
- present convenient tools to facilitate performance evaluation activity including learning and teaching performance evaluation instruments,
- assist the researchers in making informed decisions on choosing the right metrics by ranking the metrics and showing the robustness issues, and
- introduce a comprehensive systematic method to assess the robustness of any number of metrics which can also be used to benchmark recently proposed metrics.

Note that this thesis attaches as much importance to organization and representation of the proposed concepts as the concepts themselves.

More specifically the aims are to describe binary-classification performance evaluation instruments in a clear and understandable manner and reestablish classification performance evaluation foundation by clarifying and standardizing the terminology, providing formal definitions, categorizing the instruments, and providing new instruments, organization/visualization/calculation tools, and benchmarking methods to facilitate the overall approach. Hence, researchers will be able to grasp each of 50 instruments without any doubts or mistakes via the proposed concepts, know their similarities, differences, and robust/weak behaviors, and select and use the proper instruments conveniently.

The high-level goal of this thesis is to allow researchers to be certain in their classification performance evaluations and concentrate on the other critical phases of their classification problems such as ensuring dataset quality or selecting the most optimum ML model and help to standardize performance evaluation process.

1.4 Research Scope

The scope of this thesis is the instruments summarizing the confusion matrix to evaluate the classification performance of binary-classifiers. In high level, ML workflow is the problem domain and performance evaluation is the problem topic of this thesis study.

Including the three additional instruments (***TC***, ***FC***, and *ACCBAR*) proposed in this study, over 50 instruments are covered. The fifteen metrics, namely *TPR*, *TNR*, *PPV*, *NPV*, *ACC*, *INFORM*, *MARK*, *BACC*, *G*, *nMI*, *F1*, *CK*, and *MCC*, which are the eventual set of non-redundant performance instruments, are included in the benchmark. Parametric instruments such as *WACC* or *Fβ* and instrument variants such as *nMI* (nMI_{ari} [default], nMI_{geo}, nMI_{joi}, nMI_{min}, nMI_{max}) are also referred for the sake of completeness. Besides, five recently proposed metrics are also reviewed and two of them included in the second benchmarking experiment. Note that these new metrics seem not to be accepted by the research community, therefore they were not included in PToPI. Moreover, the covered instruments can be extended in a straightforward manner with the new instruments that will be proposed in the future.

To the best of my knowledge, the literature does not cover such a great extent of instruments that are also examined and evaluated with a broad perspective. *TPR*, *PPV*, *ACC*, and *F1* are the most studied instruments as described in Section 2.2 (Literature Review).

Note that Area-Under-ROC-Curve (*AUC*) metric and instruments based on a probabilistic interpretation of classification error (*i.e.* the deviation from the true probability, *e.g.*, mean squared error and Log Loss) are out of the scope of this study because the former is not based on single instance of confusion matrix and calculated by varying a decision threshold for different *TPR* and False Positive Rate (*FPR*) pairs in a specific binary-classification application (Berrar & Flach, 2012) and the later ones are for multi-label classification (Ferri, Hernández-Orallo, & Modroiu, 2009).

1.5 Significance of the Study

In general perspective, this thesis is an epistemological study following exploratory research that focuses and clarifies "'how to know that we know' about classification performance evaluation, especially binary-classification performance instruments?" by laying the foundation of knowledge with the comprehensive formal definitions, organizing the knowledge, aligning the common approaches resulted from conventions with truths or objective facts, and avoiding error-prone or misleading conclusions about the performance. The thesis developed novel methods and concepts with respect to exploratory research.

This thesis is significant from several perspectives.

- First, it revisits and reestablishes the existing literature with comprehensible concepts along with new clear formal definitions. The proposed concepts will help to assess the individual performance instruments as well as see the similarities and subtle differences among instruments conveniently. Even, distinguishing between "performance measures" and "performance metrics" and naming all kind of items derived from a confusion matrix as "performance instruments" will clarify and lay the foundations of classification performance evaluation.

- Second, the thesis will extend the current literature by proposing a new instrument category named "performance indicators" and also proposing a novel indicator named "accuracy barrier" for detecting class imbalance problems. The indicators are expected to bring a completely new perspective in performance evaluations for whom wants to quick sense of classification performances or need evaluating the performances of the bulk of classifiers or presenting them for visualization or dashboard applications.

- Third, this thesis is notable to present a unique application of knowledge organization for representing the multi-dimensional concepts in a single picture called PToPI (periodic table of performance instruments). Similar to the unique *de facto* position of the periodic table of elements in chemistry, PToPI is a handful tool for not only researchers and practitioners but also anyone who wants to learn or teach performance instruments[7]. The thesis provides another tool called TasKar to calculate all the instruments as well as visualize the base metrics in new graphics to enhance the interpretation of classification performance.

- Fourth, this thesis is the only study that answers what the most robust performance metric is, comprehensively. Furthermore, the thesis points to the insufficiency in using even a robust metric alone and for the first time suggests additional measures to avoid misleading conclusions. The holistic performance reporting approach suggested in this thesis is expected to change the assessments of future applications in classification problem examples (*i.e.* switching to using a more robust metric) as well as make the performances achieved by the existing or previous applications susceptible to reconsider (*e.g.*, representing the performances in terms of the robust metric).

- The last but not the least, this thesis is expected to engage the attention of the whole research community to a possibility of a confirmation or publication bias in classification performance reporting where the performances are reported in terms of metrics demonstrating high performance.

The thesis attempts to overcome most of the obstacles in front of precise and concise objective performance evaluation for all the parties from researchers, practitioners to students, teachers and align the research community independent from the specific domains to conduct a common objective and responsible research.

1.6 Online Research Data and Materials

Table 1.1 lists the online data, software, and materials prepared to present extra information about thesis contributions.

[7] The periodic table was also formed by Mendeleyev in a textbook in 1870 to teach students the elements and facilitate their understanding (Brito, Rodríguez, & Niaz, 2005, p. 85).

Table 1.1 Online research data and materials

Data / Platform	Contribution	Description / Online Access Address
Data 1 (Mendeley Data)	Survey 1	Binary-Classification Performance Evaluation Reporting Survey Data with the Findings http://dx.doi.org/10.17632/5c442vbjzg.2
Tool 1 (GitHub)	PToPI	The proposed periodic table of (binary-classification) performance evaluation instruments (PToPI) in full-resolution in various views (full, plain, simplified, minimal, and minimum). https://github.com/gurol/PToPI
Tool 2 (GitHub)	TasKar	Binary-Classification Calculator/Dashboard and Metric Graphics https://github.com/gurol/TasKar
Code 1 (GitHub)	Method 1: *ACCBAR*	Open-source scripts for calculation Accuracy Barrier indicator. https://github.com/gurol/PToPI
Code 2 (GitHub)	Method 1: Dependency Graph	The full-resolution dependency graph for all the instruments and the DOT (graph description language) files to produce it via Graphviz. https://github.com/gurol/PToPI
API 1 (GitHub)	Method 3	Open-source performance metrics benchmarking software library. R scripts of the developed API for conducting the proposed benchmarking method. https://github.com/gurol/metametrics
Experimenter 1 (CodeOcean)	Method 3	An online interactive experimentation platform running the provided API for benchmarking of thirteen metrics. https://doi.org/10.24433/co.1564477.v2
Data 2 (Mendeley Data)	Method 3	Metric-spaces data: Base measure permutations and corresponding metric-spaces for 13 performance metrics per different sample size values. The data is used in the benchmark. http://dx.doi.org/10.17632/64r4jr8c88.1
Data 3 (Mendeley Data)	Method 3	The detailed benchmarking results data. http://dx.doi.org/10.17632/2g36672s5f.2
Visualization 1 (Tableau)	Method 2	Ranks of mobile malware classification studies in terms of different performance metrics http://bit.ly/performanceranks

1.7 Published Works during the Thesis Study

During the thesis study, five articles were published in peer-reviewed conferences and journal as listed in Table 1.2. Note that the article (Gürol Canbek, Sagiroglu, Taskaya Temizel, & Baykal, 2017) directly related to the thesis study has been cited by three works from medicine (Nnamoko, Hussain, & England, 2018), cyber security (Kaiafas et al., 2018), and software engineering (Ulysses, 2019)[8] disciplines.

[8] At the time of writing, the article is also appeared at the top or in the first page of Google search with the following queries: binary classification performance, classification performance metrics, etc.

Table 1.2 Published works in thesis study

	Publication title	Year	Thesis relation
1	New Comprehensive Taxonomies on Mobile Security and Malware Analysis (Gürol Canbek, Sagiroglu, & Baykal, 2016)	2016	Understanding the case study domain (*i.e.* ML-based Android mobile malware detection) (Section 2.3)
2	Clustering and visualization of mobile application permissions for end users and malware analysts (Gürol Canbek, Baykal, & Sagiroglu, 2017)	2017	
3	Binary classification performance measures/metrics: A comprehensive visualized roadmap to gain new insights (Gürol Canbek, Sagiroglu, et al., 2017)	2017	Preliminary work of multi-perspective analysis (Chapter 3)
4	New Techniques in Profiling Big Datasets for Machine Learning with a Concise Review of Android Mobile Malware Datasets (Gurol Canbek, Sagiroglu, & Taskaya Temizel, 2018)	2018	The criticality of sample size in precise and concise performance reporting (Section 5.8)
5	Cyber Security by a New Analogy: "The Allegory of the 'Mobile' Cave" (Gürol Canbek, 2018)	2018	Understanding the case study domain (Section 2.3) and analogical methods (Appendix D)
6	Canbek, G., Taskaya Temizel, T., & Sagiroglu, S. (2019). Multi-Perspective Analysis of Binary-Classification Performance Evaluation Instruments. Information Processing and Management (under review)	2019	Publication in multi-perspective analysis scope (Chapter 3)

1.8 Organization of the Thesis

The rest of the thesis is organized as follows. Chapter 2 gives the preliminaries for the thesis study and reviews the literature on binary-classification performance evaluation and instruments. It presents a comprehensive survey (Survey 1) on the performance evaluation approaches in ML-based Android mobile malware detection as a binary-classification case study domain. The domain is introduced and the significant findings based on systematically selected 78 studies are given in Chapter 2.

Chapter 3 clarifies the terminology, introduces categories for performance instruments, and proposes concepts related to instruments such as formal definitions of measures and metrics, canonical forms in instrument equations, instrument geometries, dualities, and complements. Two new measures are introduced in this chapter where dependencies and levels are also defined. Summary functions, other equations forms (base measures, direct/high-level dependency and equivalent form), class counterparts and redundancy are also introduced and described. Chapter 3 also proposes the first example of performance indicators called “Accuracy Barrier” (*ACCBAR*) to indicate so-called “accuracy paradox” or class imbalance effect. As a case study, the proposed indicator is used in re-evaluating performances in Android mobile malware detection domain.

Chapter 4 designs and proposes a knowledge organization tool called PToPI to represent over 50 instruments in a single compact picture by employing structural and visual techniques. Real-world use cases of this exploratory tool that is similar to the periodic table of elements are presented through the literature examples in various domains. Chapter 4 also proposes a tool called TasKar complementing PToPI to calculate all the instruments and visualize some metrics with three new graphics. The chapter gives a new specification of the coloring scheme and provides some example usages of TasKar.

Chapter 5 provides a systematic benchmarking method named BenchMetric with three stages to assess and compare the robustness of performance metrics. The novel concepts such as universal base measure permutations, metric-space, and meta-metrics (metrics about performance metrics) are introduced. BenchMetric is described in stage by stage and also being tested on 13 existing performance metrics. The intermediate results per stage and overall result are provided and interpreted. The chapter also makes a detailed assessment of very similar *MCC* and *CK* metrics. Further, BenchMetric is evaluated with the literature comprehensively and tested by including two recently proposed metrics. Finally, Chapter 5 discusses precise and concise performance evaluation and reporting and suggests a proper approach.

Finally, Chapter 6 summarizes the thesis contributions, discusses the limitations, provides ongoing and planned future studies, and gives conclusions.

Appendix A gives a complete list of performance instruments along with their names, abbreviations, alternative names, categories, and levels as well as details of the proposed color scheme. Appendix B gives a complete list of the equations of the instruments. Appendix C shows the full view of PToPI whereas Appendix D gives insights about the analogy between PToPI and the periodic table of elements by listing and depicting the similarities among the source and target domain. Appendix E describes the selection methodology for the survey of 78 ML-based Android mobile malware detection as a case study domain. Appendix F provides the references of those surveyed studies along what analyzes are conducted per each study. Appendix G summarizes BenchMetric findings as well as the overall robustness issues combined per metric sorted in alphabetic order.

Appendix H focuses on a critical aspect of classification performance reporting in the literature and searches for the potential signs of biases where the performances are reported in terms of non-robust metrics. In this regard, this thesis introduces some equations to reveal the confusion matrix of a given study that reports a few metrics. Having performances in terms of the most robust metric, a case study is conducted and the presence of publication and confirmation biases are discussed.

CHAPTER 2

LITERATURE REVIEW

This chapter summarizes the preliminaries for binary-classification performance evaluation and reviews the related literature in general perspective.

2.1 Preliminaries

Classification and supervised machine learning

Classification is a leading specific problem or task in machine learning (ML) at which a computer program (*i.e.* classifier) improves its performance through learning from experience and it requires a well-specified task, robust performance instruments, and representative source of training experience (Mitchell, 1997, p. 17). The experience is gained by providing labeled examples (*i.e.* training dataset) of one or more classes (*e.g.*, positive or negative) that share common properties or characteristics to the classifier mapping the properties into the class labels. The performance of the trained classifier (*i.e.* in what degree it predicts the labels of known examples) is optionally re-evaluated and then finalized on different labeled examples (*i.e.* validation and test datasets, respectively). After this supervised learning phase, the classifier is supposed to be ready to predict the class of additional unknown or unlabeled examples.

Binary-classification and classes

In binary-classification or two-class classification, a classifier separates a given example into two contrasting classes. In symmetric binary-classification, each class is equally important (*e.g.*, “female” *vs.* “male”) whereas in asymmetric binary classification, one class is more valuable than the other (*e.g.*, “positive” over “negative” for a medical test or a condition in a disease, “respond” over “no respond” for a treatment, “spam” over “non-spam” for an e-mail, “malicious” software (*i.e.* malware) over “benign” software, or “faulty” over “normal” in fault identification of electric power systems). Such binary classes having one state is actually called “monary”. Symmetric binary classes are collectively exhaustive (*i.e.* there is no possibility except the two classes).

A classification separating more than two mutually exclusive classes is called multi-class classification. If the classes are not mutually exclusive (*i.e.* an example could be one or more of the available classes simultaneously), it is a multi-label classification (or any-of or multivalue classification). Binary and multi-class classifications are single-label classifications.

Confusion matrix

The binary-classification performances in training, validation and test datasets are presented by a 2x2 contingency table or confusion matrix (*i.e.* the number of correct and incorrect classification per positive and negative classes)[9]. The four figures are the number of ***TP***, ***TN***, ***FP***, and ***FN*** of the classified examples with known labeled ***Sn*** samples[10].

Overall machine learning workflow

From a broader perspective, classifier modeling, machine learning, and performance evaluation are the critical activities of an overall ML workflow (*i.e.* ML-based classification). The workflow from start to end comprises staged activities each of which defines the repeating and/or incremental tasks. Briefly, several ML models are tried and tested to achieve the best performance.

Performance instruments and evaluation/reporting/comparison

Evaluation and comparison of a binary-classification performance stated in terms of four figures simultaneously are difficult. Therefore, several classification performance instruments have been proposed to summarize these four figures as a single figure (*i.e.* multi-objective optimization or in other words compressing these values into a single number)[11]. This thesis systematically covers and analyzes over 50 performance instruments for the first time in the literature. Refer to Table 3.1 and Table A.1 in Appendix A for the instruments, notation, formatting and other related information.

Highlighting that performance instruments are used in the following scopes:

- *Performance evaluation* in training, validating and testing a classifier
- *Performance reporting* in publishing the performance of a classifier
- *Performance comparison* in comparing a classifier with the other proposed ones

2.2 Literature Review

Japkowicz and Shah (2015, p. 45) give a basic taxonomy of performance evaluation instruments apart from binary-classification instruments based on the confusion matrix. Considering binary-classification performance evaluation instruments, most of the literature gives introductory information about common metrics such as their equations. Others interpret common metrics over a number of common ML algorithms tested on example or hypothetical datasets to demonstrate the behaviors of different metrics. For example, Sokolova et al. (2006) cover three measures and six metrics via two classifiers and Tharwat (2018) addresses four measures and twelve metrics described with a single simple classification result.

[9] Those four elements (***TP***, ***FP***, ***FN***, and ***TN***) will be named as "base measures" as described in Section 3.2.1.

[10] Two terms used especially in clinical research are also related to confusion matrix and classification performance: "gold standard" and "ground truth". The former refers to a diagnostic method with the best performance and the latter the reference values used as standard for comparison. Additionally, prevalence has also slightly different meaning than it has in general classification context (i.e. positive class ratio): the probability of an individual to have the disease (based on clinical characteristics and demographic data) in a population including both newly diagnosed and existing cases (Cardoso et al., 2014, p. 28).

[11] Apart from some cost-based approaches (*i.e.* reward for correct classifications and penalty for incorrect classifications)

Most of the related literature partly addresses the issues researchers encounter when they use binary-classification performance evaluation instruments. The effect of class imbalance (or "class skew" or "prevalence effect") on performance metrics is the most studied issue (Brzezinski, Stefanowski, Susmaga, & Szczęch, 2018; Luque, Carrasco, Martín, & de las Heras, 2019; Straube & Krell, 2014). Most of the performance metrics that are based on confusion matrix elements from both class are actually sensitive to class skew (Fawcett, 2004, p. 10). Without any change in the classifier, those metrics change as the distribution of the positive and negative class samples is changed.

The skew sensitivity in metrics is examined in a narrow perspective in the literature. For example, Straube and Krel (2014) conclude the skew sensitivity of *ACC*, *F* metrics, *MCC*, and *nMI* and the skew insensitivity of ***DPR***, *BACC*, *WACC*, and *Gm* based on a single example via a hypothetical classifier having *TPR*=0.9 and *TNR*=0.7. Note that *ROC* graphs based on *TPR* and *FPR* dimensions where each dimension strictly depends on one class exclusively (*TPR* within the positive class, *FPR* within the negative class) are not sensitive to class skew.

Valverde-Albacete and Peláez-Moreno (2014) analyze so-called "accuracy paradox" where a classifier with a lower value of accuracy might have a greater level of predictive power and vice versa. Other aspects reviewed in the literature are chance correction (Labatut & Cherifi, 2011), cost-based evaluation (Hu & Dong, 2014), constraints (Forbes, 1995), and the relationship between diversity (*i.e.* the degree of disagreement within an ensemble) and performance metrics (Wang & Yao, 2013).

Some studies examine the properties of instruments from specific perspectives such as invariance in confusion matrix (Sokolova & Lapalme, 2009), chronology of the instruments (Seung-Seok, Sung-Hyuk, & Tappert, 2010), patterns in the instruments' equations (Warrens, 2008), and decomposability into the sum or average of individual losses on each sample (Yan, Koyejo, Zhong, & Ravikumar, 2018). Multi-class/multi-labelled performance evaluation is also addressed (Kolo, 2011; Pereira, Plastino, Zadrozny, & Merschmann, 2018; Sokolova & Lapalme, 2009). Others propose approaches to compare metrics. In a qualitative approach, Straube and Krell (2014, p. 2) indicate the following criteria for choosing a proper metric: *i*) performance-oriented (not data-oriented), *ii*) intuitive (interpretable), and *iii*) comparable (accepted in the literature). In a quantitative approach, Huang and Ling (2005, p. 302) suggest consistency and discriminancy degrees for comparing performance metrics through *ACC* and *AUC-ROC* example metrics in balanced and imbalanced dataset examples.

Some of the binary-classification performance instruments are the same as binary similarity or distance measures (Kocher & Savoy, 2017) that are also based on a 2x2 contingency table. For example, *F1* and *ACC* are referred to as Dice and simple matching coefficient, respectively. Tulloss (1997) first suggests some requirements and recommendations for similarity measures. Theoretically, performance and similarity instruments can be formed in numerous ways by changing the coefficients or weights in the equations. Koyejo et al. (2014) and Paradowski (2015) provides parametric equations that also generalize most of the performance evaluation instruments.

The literature touches upon problematic performance metrics especially *TPR*, *PPV*, *ACC*, and *F1*. The recommended metrics are also varied because of evaluating them from a single perspective:

- Valverde-Albacete and Peláez-Moreno (2014) report that higher Accuracy values could be misleading.
- Powers (2015, p. 5) discusses some fallacies of *F1* that come from information retrieval such as focusing on one class only, assuming prediction and real class distributions are identical and biased by the majority class.
- Shepperd (2013, p. 16) also indicates that *F1* yields significantly high values (about 0.7) on highly skewed datasets and also exhibits a misleading high performance in low prevalence datasets.
- Labatut and Cherifi (2011, p. 13) recommend *ACC* as covering both classes otherwise *TPR* and *PPV*.
- Straube and Krell (2014) recommend *DPR*, *BACC*, *WACC*, and *G* instead of *ACC*, *F1*, *MCC*, and *nMI* considering class imbalance effect.
- Schröder, Thiele, and Lehner (2011, p. 6) suggest using *INFORM*, *MARK*, and *MCC* instead of *PPV*, *TPR*, and *F1*.
- Forbes recommends *nMI* as a nontraditional metric (Forbes, 1995).

Considering the literature in general, the studies on performance evaluation instruments examine a small number of issues most of which are related to class imbalance on a few common metrics especially *F1* and *ACC*. To the best of my knowledge, such a broad analysis of a large number of performance evaluation instruments as well as a systematic benchmarking on those instruments has not been conducted in the literature.

The problems in performance evaluation approaches in a case study domain is described via a survey in the next section. Note that the following comprehensive literature reviews on specific areas of performance evaluation instruments are expressed and evaluated separately in the related chapters:

- Survey 2 in Section 4.2.2: Confusion matrix visualization methods.
- Survey 3 in Chapter 5: Metric comparison methods.

The methods reviewed in Survey 2 and Survey 3 are also compared with the methods proposed in this thesis.

2.3 Survey 1: Problems in Classification Performance Evaluation Approaches

ML-based binary classification is used in numerous domains such as unusual event detection, medical diagnosis, customer target marketing, multimedia, biological, and social media analysis, and document categorization (Aggarwal, 2015). This thesis evaluates the performance evaluation approaches of some specific classification application examples such as term extraction in medical records, computer system intrusion detection, e-mail spam detection, and software design defects detection in Section 4.1.3.

This chapter addresses "What are the problems in performance evaluation reporting?" research question (**RQ1**) by systematically surveying ML-based Android mobile malware detection as a case study domain throughout this thesis. The domain described below was chosen because it is subject to a rapid change and is one where binary classification (malicious or benign software) is frequently used. The survey also presents clear evidences for the problems 1, 2, and 5 below, all of which are described in Chapter 1 above:

1. Confusing terminology: performance measure or performance metric?
2. Disregarding negative-class performance, domain-specific tradeoffs, and end-user requirements
3. Using instruments without being aware of the pros and cons
4. Need for explaining performance instruments
5. Indeterministic performance reporting and comparison
6. The gap in responsible open research
7. The complexity of the performance instruments

Note that some findings of Survey 1 that are not expressed here are given in the other sections where they are related.

2.3.1 Brief introduction to survey domain

Mobile applications are normally expected as benign software satisfying different user requirements without any implicit/explicit and/or direct/indirect harm. However, they could be malign software or malware seeming innocent but actually contain payloads besides the intended purpose to cause harm to end-users in different forms such as sending SMS (Short Message Service) messages to the premium numbers without users' consent (Gürol Canbek et al., 2016). The followings give some insights about the domain:

- 6,140 new mobile applications on average are released every day in Google Play Store ("Average Number of New Android App Releases Per Day," 2018),
- The official application market includes 3.8 million applications in total in 2018 ("Number of Apps in Leading App Stores," 2018),
- Many more Android applications are also released in over 300 third-party application markets worldwide in a rather uncontrolled way (Dogtiev, 2018).

Within this volume and speed, distinguishing whether an existing or new mobile application is malign or benign is highly challenging. Because, detecting them solely by malware analysis conducted by a small number of specialists is impossible, ML-based malware classification is a promising and effective solution. Both in academia and industry, researchers build and test classifiers trained on labeled mobile application samples to detect malware in new applications.

Within these conditions along with the increasing threat environments, diversifying risks, and technical challenges, ML-based mobile malware classification is a prominent research area. In Android malware classification studies, the number of available malware datasets, especially positive samples are small, which results in class imbalance. The features which can be extracted by static (*i.e.* file/code analysis) and/or dynamic (*i.e.* run-time) malware analysis is high dimensional (Gürol Canbek et al., 2016, fig. 11). A very-specific attack vector (*i.e.* technique to deliver the malicious payloads) could be embedded into a benign popular application and transformed into malware (*i.e.* repackaged apps). Malware writers (*i.e.* hackers) alter these vectors and/or combine others that lead to different instances of malware (*i.e.* malware variants in malware families).

With respect to these attributes summarized above, we saw that performance evaluation is the critical part of malware detection studies in the literature where researchers claim their improvements by comparing different classifiers with performance metrics. Nevertheless, the problems introduced here can be encountered in any other domain like in software defect prediction summarized at the beginning of Appendix H.

2.3.2 Survey scope

Total 78 studies from 2012 to 2018 reporting their binary-classification performances with different ML algorithms on Android static mobile-malware detection (*see* Appendix E and F for the selection methodology and the references of the studies) are surveyed.

2.3.3 Findings: Blurring terminology

The studies use different terms while reporting classification performance evaluation. Of the surveyed studies, 42% use "metrics" for performance metrics, which is correct as this thesis will formally define it (*see* Definition 3.2 in Section 3.2.3), whereas 15% use "measures" and even 25% use both interchangeably.

Various other phrases such as "accuracies", "measurements", "performance indexes", "quality measures", "summary measures", "assurance scores", "classification quality", "detection performance", "evaluation criteria or indicator", *etc*. are also expressed in 31% of the studies. The terms for individual metrics also vary as listed in Table 2.1.

Table 2.1 The distribution of alternative terms used in 78 studies for referring to individual metrics.

Metrics	**Terms**
ACC	*ACC* (80%), Detection Rate (or Ratio) (11%), Detection Accuracy (7%). Success Rate (or Ratio), Overall Accuracy (or Efficiency), Correctly Classified Instances Rate
F1	F-measure (43%), F-score or *F1* score (39%), *F1* (22%), *Fm*
TPR	*TPR* (39%), Recall (26%), *TPR* and Recall (at the same time) (15%), Detection Rate (9%), Sensitivity (5%), Accuracy Rate, Fraction of Malware Thread Identified Correctly, Hit Rate, Rate of Correctly Detection of Malware, Recall Malicious, Recall Malign
PPV	Precision (86%), *PPV* (8%), Precision Malicious, Precision Malign, Detection Rate
FPR	*FPR* (96%), False Alarm Rate (7%), Rate of Incorrectly Detection of Innocent Application as Malware
TNR	*TNR* (60%), Specificity (27%), Recall Benign (13%), Pass Rate, Benign Application Recognition Rate

Other metrics: *FNR*: *FNR*; *NPV*: *NPV*, Precision Benign; *MCR*: *ERR*; *CK*: *CK*; *MCC*: *MCC*

The blurring terminology in a fundamental level is so widespread that the literature even on performance evaluation sometimes intermingles "performance measures" and "performance metrics" terms (*e.g.*, (Baldi, Brunak, Chauvin, Andersen, & Nielsen, 2000; Ferri et al., 2009; Huang & Ling, 2005; Labatut & Cherifi, 2011; Sokolova & Lapalme, 2009)).

The terms commonly used in other domains such as pattern recognition and information retrieval are also borrowed for use in generic binary-classification context (*e.g.*, "recall" or "sensitivity" instead of "true positive rate", "precision" instead of "positive predictive value", and "specificity" instead of "true negative rate"). The class relation is also not explicit (for example, "inverse recall" is used for *TNR* and "inverse precision" is used for *NPV* rather in a compulsory manner (Tharwat, 2018)).

Different terms for the same metrics could be used in the same study. For example, 15% of the surveyed studies use both "true positive rate" and "recall", which are commonly used in information retrieval, at the same time in pure binary-classification context. More interestingly, even six of the surveyed studies (7.7%) published the same ***TP* / *P*** value two times one referring as *TPR* and one as recall redundantly[12]. In addition, the same terms could be used for different metrics (e.g., “detection rate” for *TPR* and *ACC*). The results suggest that in order to establish a common approach in scientific studies, the fundamental terminology should be clarified first and correct terminology should be followed by all the researchers.

2.3.4 Findings: Reporting inconsistencies and tendencies

Table 2.2 shows the key findings of our survey in reporting the performance of ML-based malware classification. As seen in Table 2.2 (a), the number of performance evaluation instruments reported in a single study is discrepant. The studies tend to report two or three instruments but they may choose from only one instrument (only *ACC* or *F1*) to seven instruments inclusive. Tough they are primitive; *TPR*, *FPR*, and *ACC* are the most reported metrics as shown in Table 2.2 (b). Note that the same variance in selected metrics was also observed in multi-labeled performance reporting (Pereira et al., 2018).

Note that 12% of the studies report the confusion matrix for their best classifier configurations. Reporting confusion matrix enables calculating all the performance evaluation instruments but comparisons via the four elements of the matrix are impractical unless the sample size and class ratios are the same.

Table 2.2 The statistics of performance metrics reported from 69 applicable studies of 78 surveyed studies: **a)** the distribution of the number of metrics reported in a study (minimum one metric and maximum seven metrics were reported in the same study) **b)** Distribution of the reported 11 metrics **c)** Distribution of 31 unique combinations of the reported metrics. For example, out of 69 studies, 14 studies reported only *TPR* and *FPR* metrics, 7 studies reported *TPR*, *PPV*, and *F1*. The top six combinations (53%) are shown (other 25 combinations: 47%) **d)** The distribution of the components of the reported metrics according to their distribution in (b) revealing positive-class focus tendency

(a)

one	**two**	**three**	**four**	**five**	**six**	**seven**	-metrics
9%	32%	13%	13%	13%	1%	3%	

(b)

TPR	*FPR*	*ACC*	*PPV*	*F1*	*FNR*	*TNR*	*NPV, MCR, CK, MCC*
75%	64%	55%	36%	30%	20%	17%	7%

(c)

TPR	*FPR*					20%	53%
TPR			*PPV*	*F1*		10%	
		ACC				7%	
	FPR	*ACC*			*FNR*	7%	
TPR		*ACC*	*PPV*			4%	
TPR	*FPR*	*ACC*				4%	

(d)

OP	***TP***	***FN***	***P***	***ON***	***TN***	***FP***	***N***	***TC***	***FC***
6%	36%	2%	15%	1%	4%	9%	13%	8%	5%
positive-class related 60%				negative-class related 27%					

[12] Studies: #17 (in Table 7); #32 (in Table 6, 7, and 8); #39 (in Table 5); #40 (in Table 1); #57 (in Table 5); and #18 (in Table 5, *TPR* and recall equations are given at the same time)

Table 2.2 (c) shows the most notable finding regarding an inconsistency found in performance evaluation reporting: in almost half of the surveyed studies, the researchers use varying combinations of metrics for performance reporting.

When we looked into the roots of the metrics reported (*i.e.* calculated according to the canonical measures introduced in Definition 3.1 in Section 3.2.3), 60% of the selected metrics are positive-class related (*i.e.* based on ***TP***, ***FN***, ***P***, and ***OP***) whereas 27% are negative. Interpreting the overall findings in Table 2.2, the performance evaluation reporting seems discretionary.

Though it is out-of-scope of this study, the performance of a classification workflow and/or a classifier should be evaluated in time-space. We saw that 35% of the studies publish some sort of time measures (e.g. classifier training time in seconds). Time performance should be published and standardized in all the classification studies considering the computational or time complexity of the machine learning algorithms used. ML algorithm complexity is also a related subject in time performance (Kearns, 1990).

2.4 Conclusion

Contrary to the common assumption that performance evaluation is a well-understood and studied area, this chapter reveals fundamental problems in performance evaluation approaches in ML-based classification studies in the literature. Besides, wide-spread confused terminology, there is no consensus in performance reporting and publication.

It should be highlighted that performance instruments are also the key to making decisions in other ML workflow activities besides final performance evaluation and reporting such as

- comparing different feature sets selected for the same ML model,
- comparing different ML models with the same feature sets, and
- finally comparing the best approach achieved overall with other ML studies in the same context.

The findings have shown that researchers use a different number of metrics selected from a limited number of conventional ones namely *TPR*, *FPR*, *ACC*, *PPV*, and *F1*. On the other hand, other alternatives covered in this thesis such as *BACC*, *G*, *nMI*, *CK*, and *MCC* have not been commonly used in the literature.

Note that the findings of the survey conducted on a specific domain covering seven years are actually generic that could be encountered in other domains. In the next sections, the proposed approach is presented, which aims to help to overcome such problems by clarifying the fundamental terminology and providing a formal multi-perspective analysis and tools for binary-classification performance evaluation instruments.

CHAPTER 3

MULTI-PERSPECTIVE ANALYSIS OF PERFORMANCE INSTRUMENTS

This chapter addresses a group of **RQ2** research questions by introducing novel concepts via a multi-perspective analysis method to

- clarify and improve the terminology,
- examine whether any difference exists in instruments semantically and formally,
- introduce new essential properties uncovering and defining their characteristics, and
- reveal their similarities, relationships, and dependencies.

3.1 Categorization of Instruments: Measure, Metric, and Indicator

As revealed in Survey 1 in Section 2.3.3, terminology confusion is widespread. The first conceptualization of this thesis is to propose a fundamental terminology in classification context. For the first time in the literature, this thesis

- refers the references derived from a binary-classification confusion matrix as “performance instruments”,
- categorizes instruments as “measures” and “metrics, and further
- introduces a new instrument category named “indicators”.

A measure is defined as “the dimensions, capacity, or amount of something ascertained by measuring”[13] and metric (often metrics) is “a standard of measurement”[14] according to Merriam-Webster. A measure is quantitatively derived from measurement while a metric is close to inferring qualitative subjects. A metric is a calculated or composite measure based on two or more measures and typically stated as percentages, ratios, or fractions.

Two related works are found in the literature that specifically covers the terminology confusion observed in software engineering where “measures”, “metrics”, and “indicators” are also used interchangeably along with other related terms such as “attributes” and “scales”. Olsina and de los Angeles Martín (2004) points at the lack of consensus in the terms evaluating related concepts such as quality and productivity. They present an ontology to suggest clarification based on software-related ISO standards and recognized research articles. Similar to the adopted approach described below, they also order the terms as measures, metrics, and indicators going through distinct activities namely measurement,

[13] https://www.merriam-webster.com/dictionary/measure
[14] https://www.merriam-webster.com/dictionary/metric

calculation, and decision. García et al. (2006) approach to the terms in a different way and completely avoid the use of “measures”. Because the term “software metric” seems to be imprecise to the contrary to any other engineering disciplines.

As specifically examined from the general perspective by Texel (2013), measures, metrics, and indicators refer to different but dependent concepts. In parallel with the semantic distinction among instruments proposed in this thesis, measures are numerical values with little or no context whereas metrics possess a collection of measures in context, and indicators are the comparison of measures and/or metrics to a baseline. Figure 3.1 illustrates performance measure–metric–indicator dependencies, their relative characteristics, and typical values or ranges. The levels per instrument type are described and formally defined in the next section.

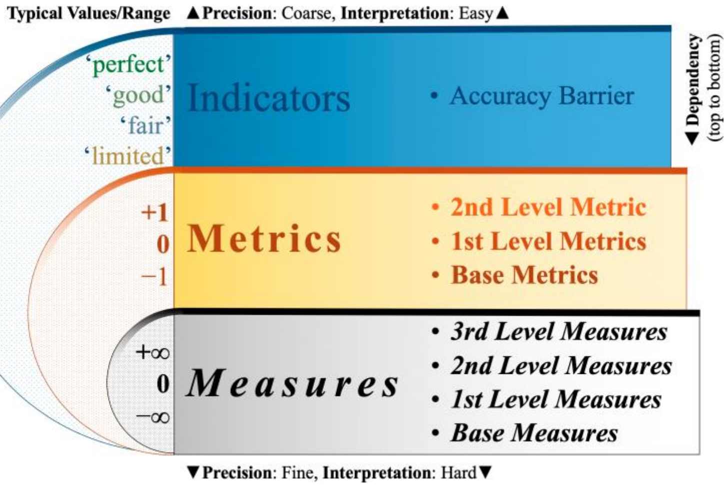


Figure 3.1 Dependency and relative characteristics of performance evaluation instrument types. The attached semicircles on the left show the typical values or ranges for each instrument type. For binary-classification performance measures and metrics, the ranges are usually [0, ∞) and [0, 1] respectively whereas indicators have nominal values.

3.2 Formal Definition and Organization of Instruments

The following formal definitions are proposed for organizing and describing binary-classification performance evaluation instruments. Table 3.1 shows the special notation proposed for differentiating measures and metrics as well as the instrument transformations described in this study.

Table 3.1 Performance instrument notations

Notation	Style	Meaning	Example
M	Italic	Any metric/indicator or measures in a limited range	*ACC* in [0, 1], *MCC* in [-1, 1], *PREV* in [0, 1]
$\boldsymbol{M}$	+ Bold	Measures with no lower and/or upper limit	***TP***, ***Sn***, ***OR***
M^* or $\boldsymbol{M}^*$	* superscript	Dual of	$PREV = BIAS^*$
$\overline{M}$ or $\overline{\boldsymbol{M}}$	Over bar	Complement of	$TPR = \overline{FNR}$
M	Regular bold	Metric-space of the metric (*see* Section 5.1.1)	**MCC**

Note that canonical measures are usually not written in bold in equations.

3.2.1 The base measures (*TP*, *FP*, *TN*, and *FN*)

In this thesis, the four conventional direct outputs of classification performance, that are presented in a 2x2 contingency table or confusion matrix are called base measures because all other instruments can be expressed by them. The different names of the base measures are provided in Table A.1 in Appendix A.

3.2.2 First level measures (*P*, *N*, *OP*, *ON*, *TC*, *FC*, and *Sn*)

The first level measures are the composition of the base measures by summation. ***P*** and ***N*** measures that are column totals (also known as marginal totals in probability theory) of a confusion matrix represent the real or actual sizes of the two classes (*i.e.* the real labels). For example, a classification test dataset with 3,000 malign and 2,000 benign application samples is expressed as ***P*** = 3000 and ***N*** = 2000. These measures correspond to the reality, observed or ground truth. ***OP*** and ***ON*** measures that are row totals (also known as marginal totals in probability theory) of a confusion matrix represent the prediction (test or classification result) of the two classes. For the same example, a decision tree classifier predicting the examples as 3,100 malign and 1,900 benign is expressed as ***OP*** = 3100 and ***ON*** = 1900. These measures correspond to the prediction, hypothesized or estimated (classification) output.

Moreover, True Classification (***TC***) and False Classification (***FC***) are defined as the totals of diagonal base measures (***TP*** and ***TN***) and/or off-diagonal ones (***FP*** and ***FN***), respectively. Substituting those totals have significantly simplified the metrics' equations and their interpretation. For instance, *ACC* that is defined as (***TP*** + ***TN***) / ***Sn*** (even as (***TP*** + ***TN***) / (***TP*** + ***FP*** + ***FN*** + ***TN***)) could be expressed simply as ***TC***/***Sn*** with ***TC***. Including ***TC*** and ***FC*** where appropriate makes the equation easy to interpret (*e.g.*, the ratio of the number of correct classifications to total sample size instead of the ratio of the number of positive and negative samples correctly classified to total sample size). Note that this notation also simplifies the multi-class performance instruments. For example, the accuracy of a ternary-classification is again ***TC***/***Sn***.

It may be argued that ***Sn*** could be classified as a base measure because sample size is always available at the very beginning before starting classification. However, our formal performance evaluation approach in this study is based on direct outputs of classification performance (***TP***, ***FP***, ***FN***, and ***TN***) and the leveling is determined by dependencies. Hence, sample size (***Sn*** = ***TP*** + ***FP*** + ***FN*** + ***TN***) is above base measures like ***P*** or ***N***.

3.2.3 Instrument equations: the canonical form

The terminology confusion described in Section 2.3.3 can be efficiently avoided by defining a formal logic that determines whether a given equation of a performance evaluation instrument is a metric or measure. The first step in the proposed formal definition is to standardize the equations. In canonical form, the equations are expressed with the base measures and the first level measures (***TP***, ***FP***, ***FN***, ***TN***, ***P***, ***N***, ***OP***, ***ON***, ***TC***, ***FC***, and ***Sn***). For example, *MCR* = ***FC***/***Sn*** and *F1* = 2***TP*** / (2***TP*** + ***FC***) are expressed in canonical form.

Definition 3.1 (*Canonical Form*).

M is a performance metric or a measure expressed in a canonical form $M\colon \boldsymbol{X} \rightarrow \overline{\mathbb{R}}$

$\boldsymbol{X} = \{(TP, FP, FN, TN, P, N, OP, ON, TC, FC, Sn) \in \mathbb{Z}^{*11}\colon [0, \infty)\}$ and $\mathbb{Z}^{*} = \{0\} \cup \mathbb{Z}^{+}$

where $P = TP + FN;\ N = FP + TN;\ OP = TP + FP;\ ON = TN + FN;\ TC = TP + TN;\ FC = FP + FN;\ Sn = P + N = OP + ON = TC + FC = TP + FP + TN + FN.$

Note that any equations listed in Definition 3.1 above must be reduced into total form (***P***, ***N***, ***OP***, ***ON***, ***TC***, ***FC***, ***Sn***) while converting an equation into canonical form (*e.g.*, a ***TP***+***FN*** should always be reduced into ***P***). High-level dependency form is described in “More Geometries” subheading below.

3.3 Performance Measure/Metric Definition

A binary-classification performance evaluation metric in canonical form is expected to have at least one of the base measures and its range is limited as dictated in semantic interpretation described in Section 3.1. Hence, the following definition is applicable to performance measures. Otherwise, the given equation in the canonical form is called as a performance metric.

Definition 3.2 (*Measure/Metric*).

M is a (binary-classification performance) “measure” expressed in canonical form where $M\colon \boldsymbol{X} \rightarrow \overline{\mathbb{R}}$ and ($\mathrm{dom}(M) \subseteq \{P, N, OP, ON, Sn\}$ or ($\min(M) = -\infty$ and/or $\max(M) = +\infty$)).

Otherwise, *M* is a “metric”.

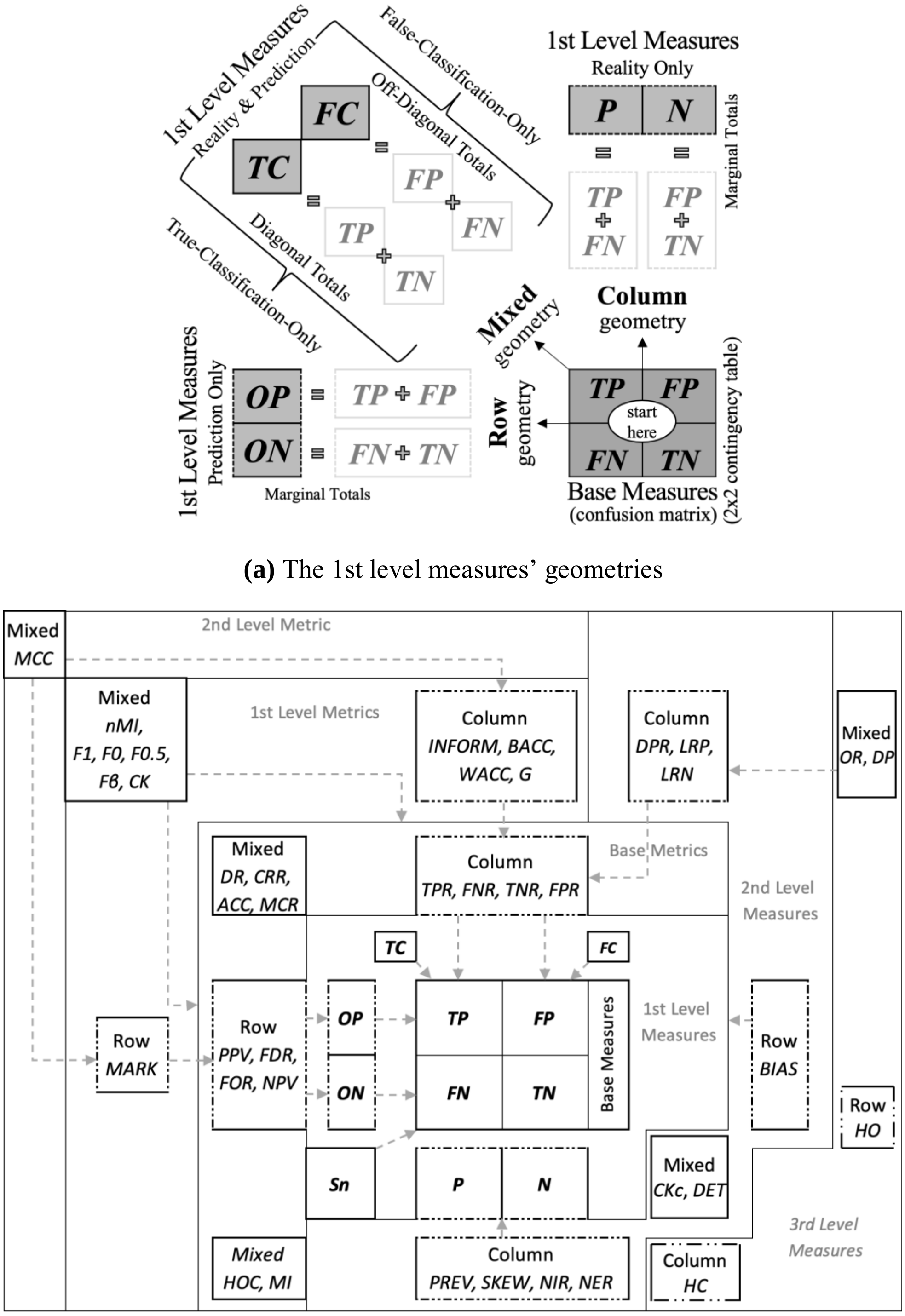
For example, *PREV* = ***P*** / ***Sn*** is a measure because dom(*PREV*) is equal to {***P***, ***Sn***} whereas $OR = {}^{TP \cdot TN}/_{FP \cdot FN}$ is still a measure even $\mathrm{dom}(OR) = \{TP, FP, FN, TN\} \not\subseteq \{P, N, OP, ON, Sn\}$ because range of ***OR*** is limitless, *i.e.* [0, ∞). $G = \sqrt{{}^{TP \cdot TN}/_{P \cdot N}}$ is a metric because neither dom(*G*) is not subset of {***P***, ***N***, ***OP***, ***ON***, ***Sn***} (because of ***TP*** and ***TN***) and nor its range is limitless (range(*G*) = [0, 1]).[15]

3.4 The Geometry for Measures/Metrics

Figure 3.2 (a) is drawn to depict the geometry of canonical measures defined above. ***P*** and ***N*** are column type (total of base measures in vertical cells in confusion matrix) that is related to reality only, ***OP*** and ***ON*** are row type (total of base measures in horizontal cells) related to prediction only, and ***TC*** and ***FC*** that are named are mixed type (total of base measures in diagonal or off-diagonal cells). Note that ***Sn*** is mixed geometry and has no effect on

[15] First measure is defined because metrics are derived from or above measures. Nevertheless, metrics could be defined explicitly by ($\mathrm{dom}(M) \supseteq \{TP, FP, FN, TN\}$ and ($\min(M) \neq -\infty$ and $\max(M) \neq +\infty$)).

geometry type when it is involved in other instruments' equations. Figure 3.2 (b) depicting the geometries of all the measures and metrics is used as a guide for the proposed exploratory table (PToPI) shown in Figure 4.1 in Chapter 4 to position the different measures and metrics in the table layout. Note that the geometry type is represented by dashed and solid edges described in Table 4.1 in Chapter 4.



(a) The 1st level measures' geometries

(b) Geometry types and layout of all measures and metrics

Figure 3.2 The origin of laying out of performance evaluation instruments in PToPI

This thesis extends this column/row geometry to any measure/metric as shown in Figure 3.2 (b) apart from ***P***, ***N***, ***OP***, ***ON***, ***TC***, ***FC***, and ***Sn*** with the following definitions:

Definition 3.3 (*Instrument Geometry*).

M is a metric/measure expressed in a canonical form where $M\colon \boldsymbol{X} \rightarrow \overline{\mathbb{R}}$

The geometry of *m* is 'Column' (depicted as M^c)

if $\mathrm{dom}(M) \supseteq \{P, N\}$ and $\mathrm{dom}(M) \not\supseteq \{0P, 0N, TC, FC\}$

The geometry of *m* is "Row" (depicted as M^r)

if $\mathrm{dom}(M) \supseteq \{OP, ON\}$ and $\mathrm{dom}(M) \not\supseteq \{P, N, TC, FC\}$

Otherwise, the geometry of *M* is 'Mixed' (depicted as M^x)

In our survey, 26% of the studies, published column geometry metrics (*e.g.*, *TPR, TNR, FPR,* and/or *FNR*). 19% published true-classification-only metrics (*e.g.*, *TPR*, *TNR*, *PPV*, *NPV*, and/or *ACC*). Interestingly, 3% published *FPR* with *FNR*, which is a subset of false-classification-only metrics.

3.5 Transforming Geometry: Metrics/Measures Duality

The extended geometry divides classification performance measures/metrics into two orthogonal dimensions besides the mixed ones: column (reality only) *vs.* row (prediction only). This approach brings about transformations in corresponding measures/metrics. Essentially, duality is to transform one concept into another concept in a bilateral manner. It could be perceived as interchanging antecedent and consequent (Powers, 2011).

Definition 3.4 (*M**, *Duality*).

M is a metric/measure expressed in a canonical form where $M\colon \boldsymbol{X} \rightarrow \overline{\mathbb{R}}$ and the geometry of *M* is "Column", "Row", or "Mixed". The dual of *M* is *M** where

if the geometry of *m* is "Column" (M^c)

$$\mathrm{dom}(M) \xrightarrow[]{\substack{P\rightarrow OP\\ N\rightarrow ON}} \mathrm{dom}(M^*)$$

if the geometry of *M* is "Row" (M^r)

$$\mathrm{dom}(M) \xrightarrow[]{\substack{OP\rightarrow P\\ ON\rightarrow N}} \mathrm{dom}(M^*)$$

A transformation via switching the column to row geometries and vice versa corresponds to reality versus prediction perspective change. The introduced transformation via duality makes researchers become aware of the special relations in corresponding metrics/measures. Basically, a dual of a column/row type measure/metric is formed by swapping between $\{\boldsymbol{P}\}$ and $\{\boldsymbol{OP}\}$ and between $\{\boldsymbol{N}\}$ and $\{\boldsymbol{ON}\}$ respectively. For instance, $TPR = PPV^*$ or $PPV = TPR^*$. As seen in the examples, the symmetry (involution) is always valid for the duality of performance measures/metrics ($M_1{}^* = M_2$ and $M_2{}^* = M_1$, *i.e.* if M_1 is the dual of M_2, then M_2 is the dual of M_1).

The duality is important to transform a mapping in one concept (dimension) to its dual concept. For example, a function (*f*) of two column-geometry metrics ($M^c{}_1$ and $M^c{}_2$) could be transformed or sought in their corresponding dual (*i.e.* row geometry metrics) metrics ($M^r{}_1$ and $M^r{}_2$) as described as below:

$$\begin{gathered}\forall M_{i,j} \in P(TP, FP, FN, TN, P, N, OP, ON, TP, TC, Sn), \exists f\ \exists M^r{}_1\ \exists M^r{}_2 \\ f(M^c{}_1, M^c{}_2)\ \Rightarrow f(M^c{}_1{}^*, M^c{}_2{}^*) = f(M^r{}_1, M^r{}_2)\end{gathered} \tag{3.1}$$

For example, ***LRP*** is a mapping between *TPR* and *TNR*. The dual of ***LRP*** = *TPR* / (1 – *TNR*) is TPR^* / (1 – TNR^*) = *PPV* / (1 – *NPV*), which is not common in existing classification performance evaluations. It is called “Relative Risk” that is especially used in statistics, epidemiology, clinical research, and diagnostic tests (Siegerink & Rohmann, 2018). The relation revealed by duality can connect classification performance evaluation domain with these domains.

The example given for ***LRP*** is related to the transformations of column or row geometry instruments. As for mixed geometry, duality transformation of high-level mixed-geometry instruments reveals different dependencies (note that dual of a mixed type metric/measure is equal to itself). For instance, the following transformation of *ACC* from Eq. (3.2) showing *PREV* dependency reveals *BIAS* dependency of *ACC*:

$$ACC = TNR + PREV \cdot (TPR - TNR) \tag{3.2}$$

$$ACC^* = TNR^* + PREV^* \cdot (TPR^* - TNR^*) \tag{3.3}$$

$$ACC = ACC^* = NPV + BIAS \cdot (PPV - NPV) \tag{3.4}$$

Increasing the class imbalance leads to a higher performance value via *ACC* as shown in Eq. (3.2), which causes a higher bias as shown in Eq. (3.4). Dual instruments should be interpreted correctly. For example, Powers’ statement (Powers, 2011, p. 3) that the goal of the classification model is achieving the equality of dual instruments such as *PREV* = *BIAS*, *TPR* = *PPV*, or *TNR* = *NPV* should be clarified by adding “in the highest possible metric values” constraint (*e.g.*, *TPR* = *PPV* = *TNR* = *NPV* ≈ 1.0). Because a random classifier with all the base measures equal (*e.g.*, ***TP*** = ***FP*** = ***FN*** = ***TN*** = 50) also satisfies all these three equalities.

3.6 Instrument Complements

Binary-classification performance metrics and some of the measures are normalized ratios having ranges in [0, 1] or [-1, 1]. The complement of a measure/metric is defined as follows:

Definition 3.5 ($\overline{M}$, *Complement*)

M is a metric/measure where $M: \boldsymbol{X} \rightarrow \overline{\mathbb{R}}$. The complement of the *i*th value of *M* is $\overline{M}$ where

$$\overline{M_i} = \begin{cases} \max(M) - M_i, & M \text{ in } [0, \max(M)] \\ \min(M) - M_i, & M \text{ in } [\min(M), 0] \\ -M_i, & \min(M) < 0 \text{ and } \max(M) > 0 \end{cases}$$

For instance, *TPR* is a metric *M*, which has a range [0, max(*M*) = 1], if TPR_i = 0.999, then the complement of TPR_i (*i.e.* FNR_i) is 1 − 0.999 = 0.001. Likewise, *INFORM* is a metric *M*, which has a range [min(*M*) = −1, max(*M*) = 1]. If $INFORM_i$ = 0.500, then the complement of $INFORM_i$ is −0.500. In contrast with duality, having both a measure/metric and its complement does not contribute any extra information. A complement could be used for simplification of equations or switching the primary point of view to another one such as switching from positive class-based view (*e.g.*, *TPR* or *PPV*) to a negative one (*e.g.*, *FNR* or *FDR*) or focusing on errors (*i.e. MCR*) instead of correctness (*i.e. ACC*). Redundancy in performance reporting is another issue related to complementation. Out of 51 studies surveyed in the performance reporting context, 16% have redundant metrics namely *TPR* with *FNR* (14%), *TNR* with *FPR* (12%), and *ACC* with *MCR* (2%).

3.7 Class Counterparts

Class-specific instruments have counterpart instruments. For example, *TPR* for positive class with *TNR* for negative class (with their complements: *FNR* with *FPR*), *PPV* with *NPV* (*FDR* with *FOR*), and ***LRP*** with ***LRN***. Counterpart relations can be uncommon unless otherwise is required. For example, the counterpart of *PREV* (=***P***/***Sn***) is *NER* (=***N***/***Sn***) that is not common. However, the counterpart of *BIAS* (=***OP***/***Sn***), (***ON***/***Sn)*** or the counterpart of *F1* (2***TN*** / (2***TN*** + ***FC***)) are not used at all. Counterparts are also applicable in multi-class performance evaluation above binary classification.

3.8 More Geometries: Dependencies, Levels, and High-Level Dependency Forms

A dependency graph is prepared to show the dependencies among 49 binary classification measures/metrics and reveals their similarities. Figure 3.3 and Figure 3.4 show a partial and full view of the dependency graph, respectively. The full-resolution graph and the DOT (graph description language) files to produce it via Graphviz are provided online at https://github.com/gurol/PToPI. High-level equation forms (*i.e.* substituting measures/metrics other than base level measures/metrics and 1st level measures) are used where possible to identify direct dependencies. Otherwise, the dependencies are calculated based on the equations in canonical form. For example,

- *TPR*, *TNR*, *PPV*, and *NPV* metrics and their complements depend on canonical measures. Therefore, they are considered as base metrics.

- *INFORM* depends on *TPR* and *TNR*; *MARK* depends on *PPV* and *NPV*. Therefore, they are 1st level metrics.

$MCC = \sqrt{INFORM \cdot MARK}$ shows that *MCC* has direct dependencies on *INFORM* and *MARK* metrics in high-level. Therefore, *MCC* is a 2nd level metric.

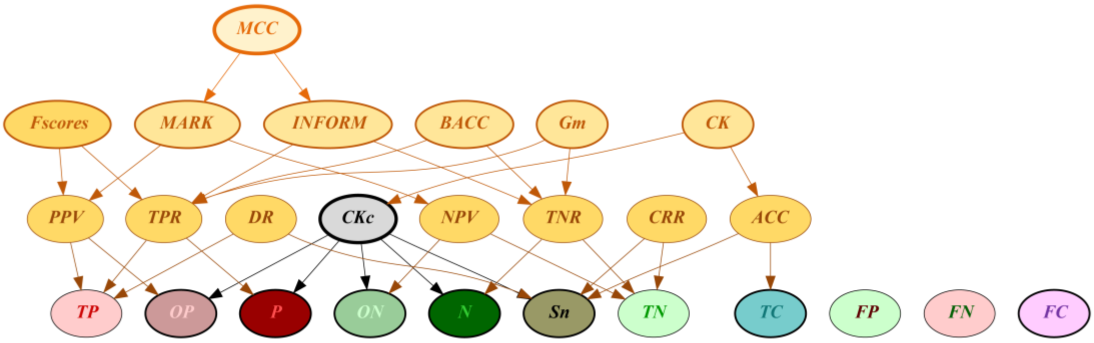


Figure 3.3 Partial view of dependency graph showing non-redundant metrics only (*i.e.* without *FPR*, *FNR*, and *MCR*). See https://github.com/gurol/PToPI for source files to generate dependency graphs

Beyond the well-known ones, the literature rarely examines the instrument equations with different expressions like in Eq. (3.2) and Eq. (3.4). Press (Press, 2008, p. 12), for example, finds the equivalent form of *PPV* and *NPV* by expressing them with *TPR* and *TNR*. The high-level dependency actually reveals another kind of redundancy observed in performance evaluation reporting (*i.e.* reporting a metric with its direct dependencies). For example, out of 51 studies surveyed in the performance reporting context, 27% published *F1* along with the two direct-dependencies (the harmonic mean of *TPR* and *PPV*).

3.9 Upper-Level Measures and Metrics Leveling

Applying the leveling approach described above, measures have four levels and metrics have three levels including base levels as shown in Figure 3.2 (b). The final levels are

- Measures: Base, 1st, 2nd, and 3rd level
- Metrics: Base, 1st, and 2nd level.

The complete list of levels and corresponding instruments in three-dimensional representation are depicted in Figure 3.5 and listed in alphabetic order in Table A.1 in Appendix A. Note that ***DP*** is not in a new level (*i.e.* 4th-level measures) because it only transforms ***OR*** measure without changing its dependents (***LRP*** and ***LRN*** or *TPR* and *TNR*).

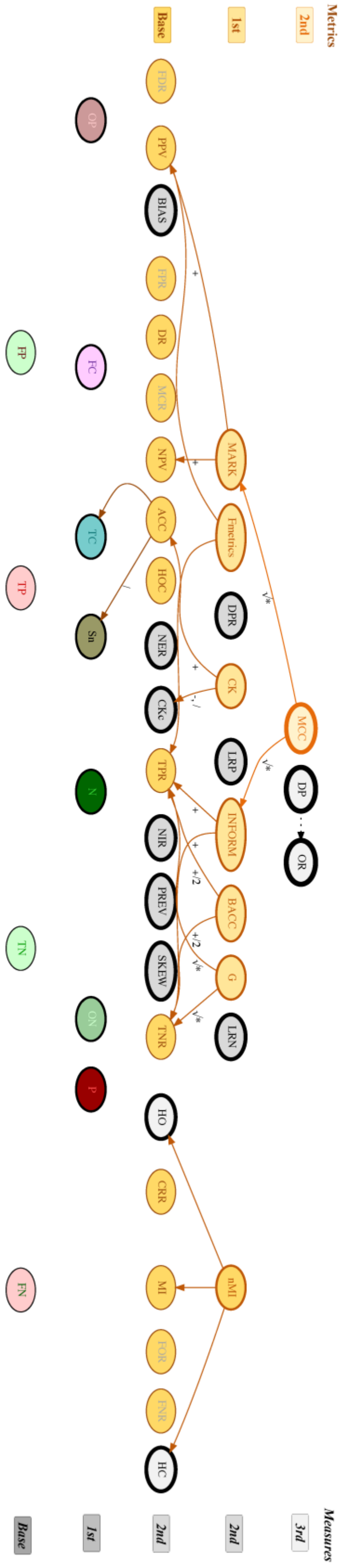


Figure 3.4 Full view of the dependency graph

3.10 Summary Functions

High-level metrics summarize the dependent metrics into a single figure on:

- Dual dependent metrics for a mixed geometry metric: *MCC* is the geometric mean of *INFORM* and *MARK* and *F1* is the harmonic mean of *TPR* and *PPV*. *nMI* has various summary functions (*e.g.*, arithmetic/geometric means, minimum and maximum) applied on *HC* and *HO*.

- Class-counterpart metrics for a column geometry metric: *INFORM* with addition, *BACC* with arithmetic mean, *WACC* with weighted mean, and *G* with geometric mean of *TPR* and *TNR*.

In parametric instruments such as *WACC* or *Fβ* (*see* equations (17') and (20') in Table B.2 in Appendix B), the summary function depending on two or more instruments can be adjusted according to the importance given each dependent (Kenter et al., 2015).

Leveling not only allows the researchers to distinguish similar instruments from a large number of instruments but also shows the dependencies among levels and their summarization degree. For example, *MCC* as a 2nd level metric depends on and summarizes the 1st level metrics that depend on and summarize the base metrics.

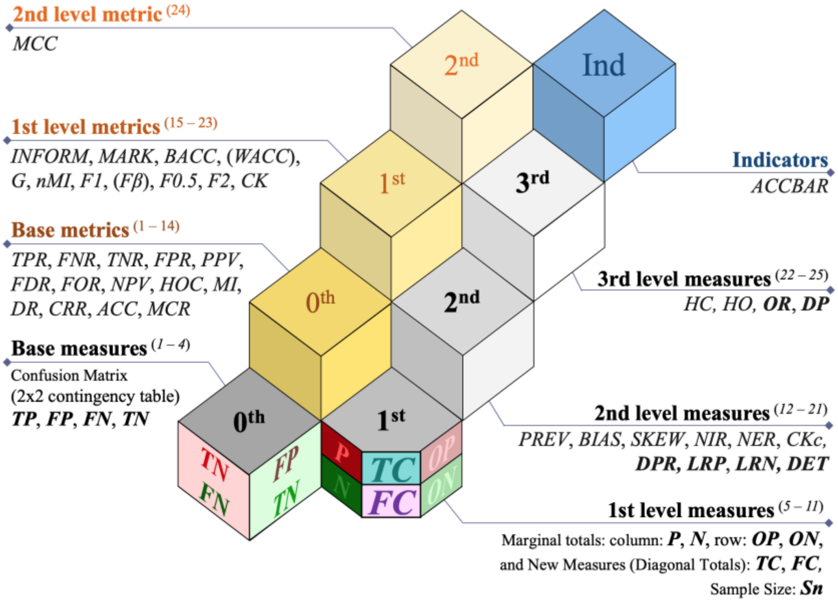


Figure 3.5 Three-dimensional representation of levels and dependency of performance instruments

3.11 “Accuracy Barrier” As the First Example of Performance Indicators

Metric or measure values are important particularly for comparison of the performance of different classifiers. However, they may be limited in terms of interpretability by end-users. In particular, nonlinear or limitless measures such as ***OR*** in [0, ∞) are hard to interpret (Schmidt & Kohlmann, 2008).

Indicator is the new category of performance instruments as proposed and described in Section 3.1 above. Addressing the research question "How to enhance comprehending, using, representing, reporting, learning, and teaching binary-classification performance instruments?" (**RQ3**), this chapter proposes a novel indicator that specifically enhances performance instrument using and reporting. Those enhancements are demonstrated via a case study where previously reported binary-classification performances in the literature are re-evaluated by the novel indicator. A negative result experienced in defining an indicator summarizing a limitless measure is also shared.

Indicators facilitate the comprehension and comparison of the metrics and measures; therefore, they are recommended for end-users or public applications. The outputs of an indicator are qualitative and they are obtained by dividing metric or measure values into coarse categories. Although categorizing a quantitative variable in a given range via cut points to facilitate understanding some phenomena and distinguish the specific intervals is applied in some domains, such as biology (Mayya, Monteiro, & Ganapathy, 2017), only one attempt of metric categorization is found where *CK* was divided into the six strength of agreement with the following half-open intervals:

- <0: "poor",
- [0, 0.2): "slight",
- [0.2, 0.4): "fair",
- [0.4, 0.6): "moderate",
- [0.6, 0.8): "substantial", and
- [0.8, 1]: "almost perfect"

by Landis and Koch (1977, p. 165) who stated that the divisions were arbitrary and provided for benchmarking.

ACC results can be high even for a random classifier (Valverde-Albacete & Peláez-Moreno, 2014). Therefore, it is essential to define a minimum performance that should be expected from a binary classifier. *NER* and *NIR*, which are not well-known or reported (Bond et al., 2018, p. S9; García-Magariño, Chittaro, & Plaza, 2018, p. 35), are two measures that can be used to define that limit as shown in Eq. (3.6), *NER* specifies the minimum successful classification rate of a classifier without a classification model that always labels a given instance with ***N***. As a class-independent version, *NIR* specifies the minimum performance by taking the larger class sample-size as either Positive or Negative into account.

A case of having a classifier with a close performance to *NER* and *NIR* measures is called as "accuracy paradox" from which this thesis introduces and formally defines the "Accuracy Barrier" indicator:

$$ACC \cong NIR \geq NER \tag{3.5}$$

$$\frac{TC}{Sn} \cong \frac{\max{(P, N)}}{Sn} \geq \frac{N}{Sn} \tag{3.6}$$

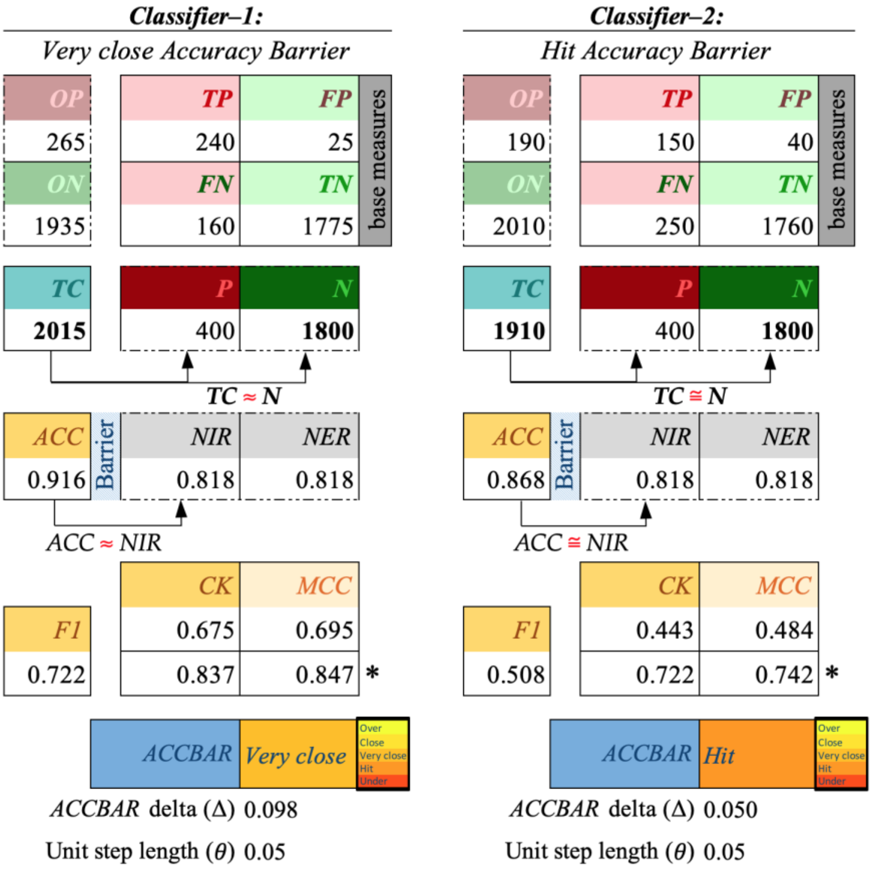
$$TC \cong \max{(P, N)} \geq N \tag{3.7}$$

A classifier with a reasonably high *ACC* where ***TC*** is close to the number of Positives (***TC*** ≈ ***P***) or Negatives (***TC*** ≈ ***N***) cannot overcome the Accuracy Barrier. Table 3.2 shows the performance measures and *ACC* metrics of two hypothetical classifiers tested on 2,200 samples (***Sn***) with 18% prevalence (as frequently observed in domains having rare positive samples such as known mobile malware or a specific disease).

When the performance is reported with only *ACC* metric, both classifiers achieve notable performances (*ACC* values are 0.916 and 0.868). Nevertheless, their *ACCs* are very close to the *ACC* of an ordinary classifier (0.818) whose outcome is always "Negative" (***N*** >> ***P***). Therefore, the Accuracy Barrier is recommended to be checked by either Eq. (3.5) or Eq. (3.7) (*see ACC*, ***TC***, *NIR*, and *NER* that are shown in PToPI in Figure C.2 in Appendix C). When the classification performance is reported in terms of other metrics such as *F1*, *CK*, and *MCC*, the results are lower than *ACC* as shown in Table 3.2.

You can test different classification results and see the accuracy barrier outputs in the online extra material provided at https://github.com/gurol/PToPI as well as using the developed tool TasKar described in Section 4.2, which is also provided online at https://github.com/gurol/TasKar.

Table 3.2 Accuracy barriers and other metrics on two example hypothetical classifiers



	Classifier–1: *Very close Accuracy Barrier*	***Classifier–2:*** *Hit Accuracy Barrier*
OP	265	190
ON	1935	2010
TP	240	150
FP	25	40
FN	160	250
TN	1775	1760
TC	**2015**	**1910**
P	400	400
N	**1800**	**1800**
ACC	0.916	0.868
NIR	0.818	0.818
NER	0.818	0.818
F1	0.722	0.508
CK	0.675 0.837 *	0.443 0.722 *
MCC	0.695 0.847 *	0.484 0.742 *
ACCBAR	*Very close*	*Hit*
ACCBAR delta (Δ)	0.098	0.050
Unit step length (θ)	0.05	0.05

base measures
Barrier
TC ≈ *N*
ACC ≈ *NIR*
TC ≅ *N*
ACC ≅ *NIR*
Over
Close
Very close
Hit
Under

* When *CK* and *MCC* ranges [-1,1] are normalized to [0, 1] like in *ACC* and *F1*

Five accuracy barrier categories are defined from the most proper to the least:

- "Over"
- "Close (to)"
- "Very close (to)"
- "Hit", and
- "Under" + the "Accuracy Barrier"

The following equations are proposed to calculate the proposed indicator called *ACCBAR*.

$$\Delta = ACC - \frac{\max(P, N)}{Sn} \tag{3.8}$$

$$ACCBAR = \begin{cases} \text{Over}, & \Delta > 3\theta \\ \text{Close}, & \Delta > 2\theta \\ \text{Very close}, & \Delta > \theta \\ \text{Hit}, & \Delta >= 0 \\ \text{Under}, & \text{otherwise} \end{cases} \tag{3.9}$$

The unit step length (θ) value is determined as 0.05 by considering the range of related metrics (*ACC*, *NIR*, *NER*) [0, 1] and the minimum difference in which the performances of different competing classifiers are compared (*i.e.* high-performance values between 0.95 and 1.0 that researchers would like to achieve). Note that Figure 4.7 also depicts accuracy barrier categories in example delta values in TasKar tool.

ACCBAR can give notable insight into the performance by evaluating one metric (*ACC*) and one measure (*NIR*). The indicator is straightforward to calculate and clarify the vague condition interpretation of Accuracy Paradox in the literature and provide an exact measurement. *ACCBAR* can be a significant instrument for classification studies when publishing their performances via *ACC*. For example, a classification performance stated as *ACC* = 0.916 alone cannot be disregarded in especially applications in emerging areas. Nevertheless, it is actually very close to the Accuracy Barrier as shown in Table 3.2.

3.11.1 Case Study 1: Classification performance re-evaluation via *ACCBAR*

The ideal approach in ranking different classification studies for the same classification problem (*e.g.*, ML-based Android mobile malware detection) is to test the classifiers on the same datasets (*i.e.* benchmarking datasets) and compare the test results in terms of a chosen metric. However, this approach could not be possible due to the various reasons. For example, a researcher could not

- access the datasets used in other compared classifiers to test her/his classifiers or
- build the compared classifiers' models to test them on her/his own datasets.

ACCBAR actually adds a pre-control for classification performances expressed in terms of *ACC*. In order to show the usage of *ACCBAR* indicator, 28 of the surveyed studies that report their classification performances in terms of *ACC* are analyzed via *ACCBAR*. As there were more than one alternative classifier models published in most of the studies, the configurations yielding the highest *ACC* are chosen. Table 3.3 shows the details of the analysis conducted on 28 studies but presents the top 15 of 28 studies having the highest *ACC* reported for the sake of space and simplicity (more detailed information for all the studies are provided in online data).

Unexpectedly, the results show that the top five of the classifications ranked by *ACC* are actually at the bottom when the studies are ranked by their *ACCBAR* category (from best condition: "Over", "Close" "Very Close", "Hit", and "Under") then delta (Δ) values per *ACCBAR* category, and then *ACC* in decreasing order. For example, the #51 study with the highest *ACC* (0.9982) is reduced by 23 ranks and to 5th from last. This is also seen in other studies (for example, #33 study is reduced from 2nd position to 7th from last and #57 study from 3rd to 3rd from last).

The exact delta (Δ) values can be used to evaluate and compare the performances of the classifiers within the same *ACCBAR* category. The conducted experiment shows that *ACCBAR* delta values help in interpreting the overall ranking. If they are not included (*i.e.* ranked by *ACCBAR* category from best then *ACC* in decreasing) the rankings become different.

The primary sort field *ACCBAR* and the secondary sort field *ACC* (*e.g.*, the sorting of #33, #1, #2, *etc*. studies in "Over" accuracy barrier) in Table 3.3 explain this condition. In the "Over" group, #33 and #1 studies having the highest two *ACC*s should be the first and second in the group. However, their delta values (0.22 and 0.19, respectively) are lower (*i.e.* closer to accuracy barrier) than the values of the preceding two studies (#2 and #47 with 0.49 and 0.48, respectively). Hence, the #2 and #47 studies are expected to be the first and second, respectively even their *ACC*s were lower (*i.e.* the achieved accuracy can be considered more credible).

Table 3.3 Performance rankings of different classifications in terms of *ACC* metric are completely different when *ACCBAR* indicator is taken into account.

		Sorted:	*ACC*↓	*ACCBAR*↓, Δ↑, *ACC*↓							
N	***P***	***ACC***	**Initial Rank**	**Δ Rank**		**Rank change**	**Change at bottom / top**	**Δ**	***ACCBAR***	**Reported metrics/measures**	**#Study reference**
8,000	400	0.9860	**7**	**28**	the last	-21	▼	0.03	Hit	*ACC*, ***BM***, *TPR*, *FPR*, *PPV*	30
99,037	10,581	0.9982	↓ **1**	**24**	5th last	-23	▼	0.09	Very close	*ACC*, *TPR*, *FPR*, *F1*	**51**
107,327	8,701	0.9949	**3**	**26**	3rd last	-23	▼	0.07		*ACC*, *TPR*, *FPR*, *PPV*, *F1*	**57**
122,176	9,756	0.9906	**4**	**27**	2nd last	-23	▼	0.06		*ACC*, *TPR*, *FPR*, *PPV*, *F1*, *CK*, *MCC*	27
1,853	6,909	0.8828	26	25	4th last	1		0.09		*ACC*, *BM*, *TNR*	52
9,804	2,794	0.9970	↓ **2**	**22**	7th last	-20	▼	**0.22**	Over	Only *ACC*	**33**
16,000	3,987	0.9900	**5**	**23**	6th last	-18	▼	**0.19**		*ACC*, *TPR*	**1**
7,494	7,494	0.9890	**6**	**1**	first	5	▲	**0.49**		*ACC*, *FPR*, *FNR*	**2**
1,260	1,260	0.9840	**8**	**2**	second	6	▲	**0.48**		*ACC*, *FPR*	**47**
480	743	0.9787	9	17		-8		0.37		*ACC*, *TPR*, *PPV*, *F1*	13
3,938	2,925	0.9750	10	13		-3		0.40		*ACC*, *TPR*, *TNR*, *FPR*, *FNR*, *PPV*, *AUC-ROC*	37
12,026	5,264	0.9740	11	20		-9		0.28		*ACC*, *TPR*, *FPR*	63
3,938	2,925	0.9720	12	14		-2		0.40		*ACC*, *TPR*, *TNR*, *FPR*, *FNR*, *AUC-ROC*	66
1,250	610	0.9688	14	19		-5		0.30		*ACC*, *TPR*, *PPV*, *AUC-ROC*	41
5,560	5,560	0.9688	13	**3**	third	10	▲	0.47		Only *ACC*	**6**

• Studies are sorted by *ACC* values from maximum to minimum per *ACCBAR* category to differentiate the effect of *ACCBAR*. • For simplicity, only the top 15 of 28 studies with "hit" and "very close" to *ACCBAR*. There is no classification with "under" "close (to)" *ACCBAR*. The names of the reported metrics are displayed instead of the values. • Delta (Δ) values for example misleading *ACC* ranks are shown in **underlined bold** against the proper Δ ranks shown in **bold**.

The possible reduction is not limited to the top-performing classifications. Classification with a relatively lower *ACC* can move to the higher ranks as observed for the #6 study that goes up from rank 13th to 3rd via *ACCBAR* indicator correction. Future work will evaluate the performance values of other metrics such as *BACC*, *F1*, *CK*, and *MCC* for under, hit and very close to Accuracy Barrier cases and compare the differences with *ACC* from a broad perspective as shown in Table 3.2. Note that an open-source R script developed for *ACCBAR* is provided at https://github.com/gurol/PToPI.

CHAPTER 4

KNOWLEDGE ORGANIZATION AND DASHBOARD/CALCULATOR TOOLS FOR PERFORMANCE INSTRUMENTS

As mentioned in Chapter 3, novel concepts are proposed to present the essential properties to distinguish the instruments. As a summary, the followings are some example interpretations of each concept:

- *ACC* is a performance metric whereas *PREV* is a measure (recall that a metric is directly related to classification performance and a measure in a fixed range is indirect, related to classification configuration).

- *TPR* is a base metric whereas *BACC* is a 1st level metric (*i.e.* directly depends on *TPR* and *TNR*).

- Of three base metrics; *TPR* has a column geometry related to reality only, *PPV* has a row geometry related to prediction only, and *ACC* has a mixed geometry covering both reality and prediction.

- *FNR* is the complement of *TNR* whereas *PREV* is the dual of *BIAS*. ***LRP*** and ***LRN*** are the class counterparts with the same summary function and direct dependents (for positive and negative class, respectively).

It is expected that all these concepts will establish a well-defined foundation for performance evaluation instruments from a theoretical point of view.

Beyond defining the concepts such as instrument type, leveling, geometry, complementation, and duality, this study also focuses on the representation of these concepts for all the instruments as a practical contribution that addresses the research question “How to enhance comprehending, using, representing, reporting, learning, and teaching classification instruments?” (**RQ3**).

4.1 PToPI: A Knowledge Organization Tool

As an original implementation proposal of knowledge organization in information science, a compact exploratory table is designed for 50 binary-classification performance evaluation instruments called PToPI, which is the pictorial specification or blueprint of instruments from multiple perspectives covering all the proposed concepts that described and formally defined in Chapter 3. PToPI depicts the patterns among performance evaluation instruments including 25 measures, 24 metrics, and one indicator by organizing them according to their types and relationships. The real-world use cases of PToPI is also demonstrated over the literature studies.

A simplified version of PToPI is shown in Figure 4.1 and the full view is included in Figure C.2 in Appendix C. PToPI is presented in an all-in-one style, thus it is a compact schema resembling the periodic table of elements. A total of 50 classification performance instruments all of which originated from four base measures are grouped into measures, metrics, and indicators, then the measures and metrics are divided into a leveled structure, and positioned according to geometries, similarities, and dependent metrics/measures.

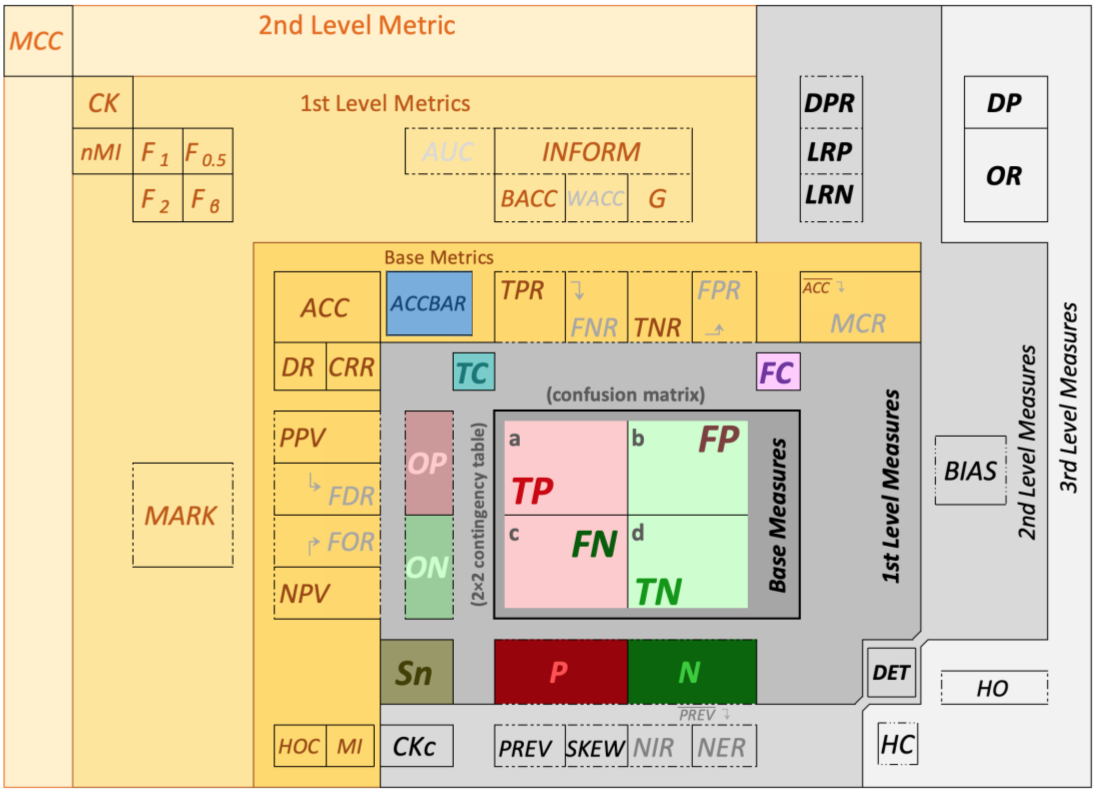


Figure 4.1 Plain view of PToPI for 50 instruments: 25 measures, 24 metrics, and one indicator for binary-classification performance. See Figure C.2 in Appendix C for the full view and visit https://github.com/gurol/PToPI for other views and future updates.

4.1.1 Design methodology

PToPI is designed with the following methodology:

1. Reviewing the literature to compile information such as alternative names and equations of the metrics, measures, and indicators,

2. Equations are converted into different forms where possible such as canonical form (Definition 3.1) and with high-level dependency form (*see* Section 3.8, "More Geometries")

3. Measure and metrics are identified by canonical equations (via Definition 3.2),

4. Geometry types are determined as “column”, “row”, or “mixed” (via Definition 3.3),
5. A dependency graph is prepared to formulate the levels and discover the similarities and dependencies (*see* Dependency Graph),
6. The determined levels and dependencies along with the geometry types are used to position and level the measures/metrics around base measures shown in a 2x2 contingency table,
7. After the layout is completed, the dual and complement of measures/metrics are determined (via Definition 3.4 and Definition 3.5, respectively),
8. The ranges and whether a measure/metric yield not-a-number (division by zero) are calculated,
9. Special colors are used on text and/or background for distinguishing measures, metrics, and indicators, their complements (*e.g.*, FDR is grayed out because it is $\overline{PPV}$), and individual base and first level measures (*see* colors in Table A.2 in Appendix A),
10. Measures and metrics are separately numbered according to levels and dependencies from innermost. Within each level, the numbers are assigned from column to row and mixed geometry and from positive to negative class dependencies. The duals are numbered in succession.
11. Geometry is depicted by solid and dashed lines (dashed bottom/top edges for column types, dashed left/right edges for row types, and all solid for mixed geometries, *see* Table 4.1), and
12. Canonical and simplified equations are shown around the measures, metrics, and indicators.

4.1.2 How to interpret PToPI?

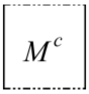
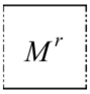
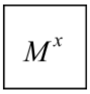
Table 4.1 lists the visual design elements employed in PToPI to represent the properties of individual instruments and/or instrument types. PToPI in full view also presents abbreviated names, full names, alternative names, assigned numbering, and some special attributes of measures and metrics (see also the legend in Figure C.2 in Appendix C) such as duality, complementation, whether having not-a-number value (*i.e.* no 0/0), ranges that are different from [0, 1].

Recall that the names of measures and metrics that have no upper limit are written in bold and numbering for measures are written in italic as shown in Table 3.1 above. The measures and metrics above or below the confusion matrix are column geometry type (depended solely upon base measures and ***Sn*** with ***P*** and/or ***N***) whereas the ones located on the left or right of the confusion matrix are row geometry type (the same as row type but with ***OP*** and/or ***ON***). In equations, bold font styles depict canonical forms and normal styles depict high-level forms.

4.1.3 Applications of PToPI Use

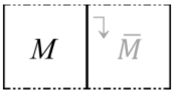
PToPI facilitates standardized specifications of a large number of performance evaluation instruments and avoids terminological confusion and uninformed choice of a metric.

Table 4.1 Descriptions of the visual design elements used in PToPI

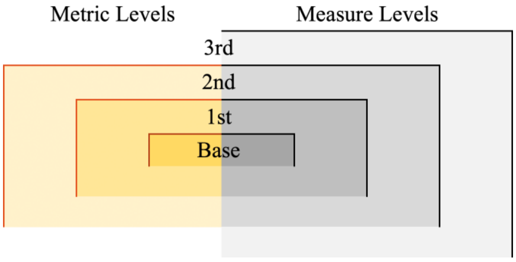
Geometries	Position (1)	Box edges	Arrows in Equations (2)	Example
Column	Below / above		Up (↑) / Down (↓)	*TPR* ↓ ***TP*** / ***P*** *TPR* has a column geometry and depends on ***TP*** and ***P***.
Row	Left / right		Right (⟶) / Left (⟵)	*PPV* ⟶ ***TP*** / ***OP*** *PPV* has a row geometry and depends on ***TP*** and ***OP***.
Mixed	Diagonal / Off-diagonal		Diagonal (↘↖↗↙)	*ACC* ↘ ***TC*** / ***Sn*** *ACC* has a mixed geometry and depends on ***TC*** and ***Sn***.

Complements

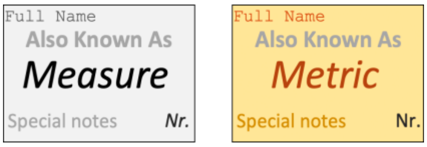
Complement relations (*e.g.*, *TPR vs. FNR*) are shown in rightwards arrows with corner downwards (↴) or upwards (⤴) in redundant pair (*e.g.*, *FNR*) having gray text color.



Leveling Background Colors



Instrument Boxes (3)



Special notes:
Instrument Ranges: ±1 or [0, ∞), otherwise: [0, 1] (not displayed)
Error Types: type I (***FP***) and/or type II (***FN***)
NaN (not have 'not-a-number', *i.e.* division by zero);
Dual: M_*; **Complement**: $\overline{M}$

(1) According to canonical measures frame **(2)** Also shows the dependencies (*e.g.*, $P = TP + TN$ ↑)
(3) Instruments are numbered (Nr.) per instrument type. Measure numbers are italic.

Seeing the true limitations of the instruments eliminates unnecessary performance reporting and allows the researchers to select the most appropriate instrument or instruments according to specific requirements. PToPI is intended to be a single comprehensive reference that will be updated upon new instrument proposals.

The practical use of PToPI can be described in two pillars:

- Overall instrument analysis: Seeing and comparing the relationships, differences, and similarities of all the instruments.

- The proper metric choice for performance reporting and comparison: Deciding which instruments are suitable for establishing classification models, comparing different classifiers, and reporting classification performances.

Overall instrument analysis: PToPI shows the similarity of the instruments. For instance, comparing *INFORM* and *MARK* dual metrics in the 1st level, three additional column-geometry metrics are shown near *INFORM*, namely *BACC*, *WACC*, and *G*. However, the duals of those additional metrics corresponding to row-geometry are not seen near *MARK*. For example, there is no metric taking the arithmetic mean of *PPV* and *NPV* like *BACC* (arithmetic mean of $TPR{=}PPV_*$ and $TNR{=}NPV_*$).

No metric is found that corresponds to *G* taking geometric mean of the same dependents. The reason for the lack of dual metrics in row geometry is attributed to the fact that performance metrics based on the prediction of a classifier (*i.e.* depending on ***OP*** and ***ON***) are not as significant as the ones based on the reality (*i.e.* depending on ***P*** and ***N***). The duals of ***LRP***, ***LRN***, and ***OR*** column-type measures are also missing due to the same reason. This thesis revealed such findings that were not addressed in the literature after seeing the big picture via PToPI.

The proper metric choice for performance reporting and comparison:

The following performance evaluation example approaches are compiled from different domains in the recent literature to show the practical assistance of PToPI in selecting an optimum number of metrics in performance comparison and performance reporting.

- *F1* is frequently used as a single metric in many domains especially in information retrieval conventionally (*e.g.*, in extracting medical terms in clinical texts (Matsuo & Ho, 2018)). Referring to PToPI, we can see that *F1* is the harmonic mean of *TPR* and *PPV*, which then depends on positive class only measures (***TP***, ***P***, and ***OP***). While using *F1* could be acceptable considering the domain requirements focusing on positive performance, it would be better to report a supportive metric with *F1* to distinguish the negative class performance. The best alternative is *TNR* or *NPV* that are shown near *TPR* and *PPV*. Briefly, the main metric (*i.e.* used as a single figure in a performance comparison of different classifiers) is *F1* and the supportive metric (*i.e.* additional metrics used in performance reporting to indicate other perspectives) is *TNR* in this case. A classifier with higher performance in terms of a main metric could have a lower performance in terms of supportive metrics.

- Another common approach in performance reporting as shown in Table 2.2 above is reporting *F1* along with its direct dependencies namely *TPR* and *PPV* (*e.g.*, in predicting hospital admissions from emergency department medical records (Lucini et al., 2017)). Following the same approach above and addressing the negative class performance, *F1* can be reported as the main metric. Furthermore, *TNR* and one of *TPR* and *PPV* direct dependent metrics can be reported as supportive metrics. In the given medical example, *PPV* can be selected as a supportive metric because *PPV* values are less than *TPR*. Thus, the lower *PPV* performances are disclosed to the readers.

- Some domains prioritize false classifications (either or both of *FPR* and *FNR*). For example, an intrusion detection system focuses on and reports *FPR* (type I error) and then *FNR* (type II error) along with *TPR* and *ACC* (Shah & Issac, 2018). Because, high false positives can be annoying for end-users, in the given example, reporting

TPR, which is the complement of *FNR* as shown in PToPI, is redundant. Note that reporting a metric (*INFORM*, *BACC*, and *G* groups in PToPI) above *FPR* and *FNR* is also redundant unless focusing on both error types. As an alternative to reporting *ACC*, a mixed geometry metric above *FPR* and *FNR* level such as *CK* or *MCC* can be used as a main metric besides supportive *FPR* and *FNR* metrics (*e.g.*, reporting three metrics: *MCC*, *FPR*, *FNR* instead of *ACC*, *TPR*, *FPR*, *FNR*).

- An *ad hoc* increase in the number of the reported metrics does not necessarily guarantee the revelation of the superiority of a classification method. It makes the comparison harder for the readers conversely. For example, an e-mail spam detection study highlights the performance via three base metrics, namely *ACC*, *TPR*, and *PPV* (Faris et al., 2019). Besides, *TNR*, *NPV*, and *G* metrics are also reported in detailed performance tables. As shown in PToPI, having four non-complement base metrics each of which depends on corresponding base measures is equivalent to reposting a confusion matrix. Going up in one level per reported column and row base metrics, *INFORM* is reported instead of *TPR* and *TNR* and *MARK* (as the dual of *INFORM*) is reported instead of *PPV* and *NPV*. There is no need to report *G* metric because it is similar to *INFORM* as shown in PToPI. Reporting *MCC* is also appropriate by not only summarizing *INFORM* and *MARK* dependents but also including ***FP*** and ***FN*** as shown in the canonical forms of *MCC*. Hence, three metrics are sufficient for this example of performance comparison and reporting instead of six metrics (*MCC* as the main metric and *INFORM* and *MARK* as the supportive metric).

- Another binary-classification performance reporting example that classifies code smells (issues in software codes potentially causing error or failure) reports ten instruments: *ACC*, *TPR*, *TNR*, *FPR*, *FNR*, *PPV*, *TPR*, *F1*, *PREV*, and *NER* (Ubayawardana & Karunaratna, 2019). As shown in PToPI, three instruments are redundant: *FPR*, *FNR*, and *NER*. From a class-balanced performance view, *CK* or *MCC* can be used instead of *ACC* and *F1* along with supportive *INFORM*. *PREV* should also be reported as a supportive instrument indicating class-imbalance in datasets. Hence, three instruments can be reported instead of ten. Supportive instruments can be further taken into account where *ACC* and *F1* yield the maximum performance (1.000).

4.1.4 Analogy between PToPI and Periodic Table of Elements

From information science perspective, the periodic table of elements can be considered as an unprecedented example application of information or knowledge organization where the classification of the elements (*i.e.* grouping, ordering, positioning the elements) is pragmatic (*e.g.*, producing the most helpful one), methodological and fruitful suggesting new hypothesis, explanations, and theories (Hjørland, 2013). Likewise, PToPI is also a schematic representation of available performance evaluation instruments conveying different forms of essential properties (*i.e.* concepts) (Hjørland, Scerri, & Dupré, 2011).

After designing PToPI, an analogy with the periodic table of elements was also explored. It is observed that there is a strong analogy among them. Analogy is defined as the inference that if two or more systems of things agree with one another in some respects, they will probably agree in others. Generally, there are two specific systems of things: source domain and target domain where a strong and large number of similar patterns exist from source to target (Gürol Canbek, 2018, pp. 65–66)[16]. The mapping of the analogy is prepared and summarized in Table D.1 in Chapter D to present the interesting revealed similarities.

Note that the *ACCBAR* performance indicator, which is shown next to *ACC* in PToPI, can also be used along with *ACC* so that class imbalance is addressed in performance evaluation.

It is expected that the proposed definitions and PToPI itself will assist researchers in comprehension, computation, interpretation, selection, and representation of classification performance evaluation instruments and their relationships. Considering a large number of available instruments, such a table is essential for not only experienced researchers but also young academicians and practitioners in machine learning.

4.2 TasKar: Dashboard and Calculator

Despite PToPI is a convenient tool from theoretical aspects, researchers in practice still need to calculate and see performances in terms of those large number of instruments. To the best of my knowledge, there is no convenient tool having this comprehensive capability besides some engineering packages providing commands to calculate metrics. Such packages obviously are not compatible with the proposed concepts presented in this thesis.

To address such a need, a compact dashboard and calculator called TasKar[17] is designed and shared with the research community online at https://github.com/gurol/TasKar. TasKar is a practical tool to calculate and visualize a large number of performance instruments not only the common and well-known ones but also the others that should be paid strong attention.

Recall that PToPI described above represents the proposed concepts for 50 instruments described in Chapter 3. As an implementation of knowledge organization, PToPI presents the instruments in an organized structure with visualization techniques in order to facilitate learning, comprehending, and teach performance instruments. The concepts and detailed information about the instruments including the equations are all represented in a single page.

TasKar complements PToPI by providing a tool to calculate the instruments and visualize their outputs along with new graphics to interpret the classification results dynamically. In this regard, this chapter addresses the research question (**RQ3**) again especially from the aspects such as using, reporting, learning, and teaching instruments.

Before introducing TasKar tool, the proposed coloring scheme for representing the instruments and concepts is described in detail. Note that this scheme is applied throughout this thesis where applicable such as in figures and tables. Likewise, PToPI was also designed according to this scheme. The colors in this scheme are selected based on the concepts and their meanings.

4.2.1 Proposed coloring scheme

The proposed scheme comprised color palettes designed for distinguishing:

- Instrument types (measure *vs.* metrics *vs.* indicators),
- Instrument levels per instrument type (*i.e.* base measures or 1st level metrics), and

[16] Note that this article was prepared during the initial phase of my thesis study.

[17] TasKar is the abbreviation of ***Tas****nif* ***Kar****nesi* in Turkish (Classification Report)

- Canonical measures (*i.e.* ***TP***, ***FP***, ***FN***, ***TN***, ***P***, ***N***, ***OP***, ***ON***, ***TC***, ***FC***, and ***Sn***).

The color palettes for these items (*i.e.* text and background colors) are listed and shown in Table A.2 in Appendix A. These palettes are designated to reflect the notion behind performance instruments as well as enhance the comprehensibility of the proposed concepts so that overall perception can be achieved by all the parties (*e.g.*, researchers, practitioners, students, and teachers) dealing with performance instruments. The coloring scheme also provides harmony among different tools such as PToPI and TasKar.

A three-dimensional representation shown in Figure 4.2 (a) is prepared as a guide to describing the scheme of the base and first level measures (*i.e.* canonical measures). The figure reflects the view of a researcher who evaluates classification performance by looking into the confusion matrix above.

4.2.1.1 Color palettes for 1st level measures

The following items describe the color palettes for first level measures (***P***, ***N***, ***OP***, ***ON***, ***TC***, ***FC***, and ***Sn***) as depicted in Figure 4.2 (b).

- *Red-like for positive*: Because the target class is usually more concerned in classification studies (*e.g.*, malware, spam, illness, or any other rather abnormal phenomena), it is used for positive-class related canonical measures (***P***, ***OP***, ***TP***, and ***FN***). Following the common practices, red-like colors are used to distinguish such measures.

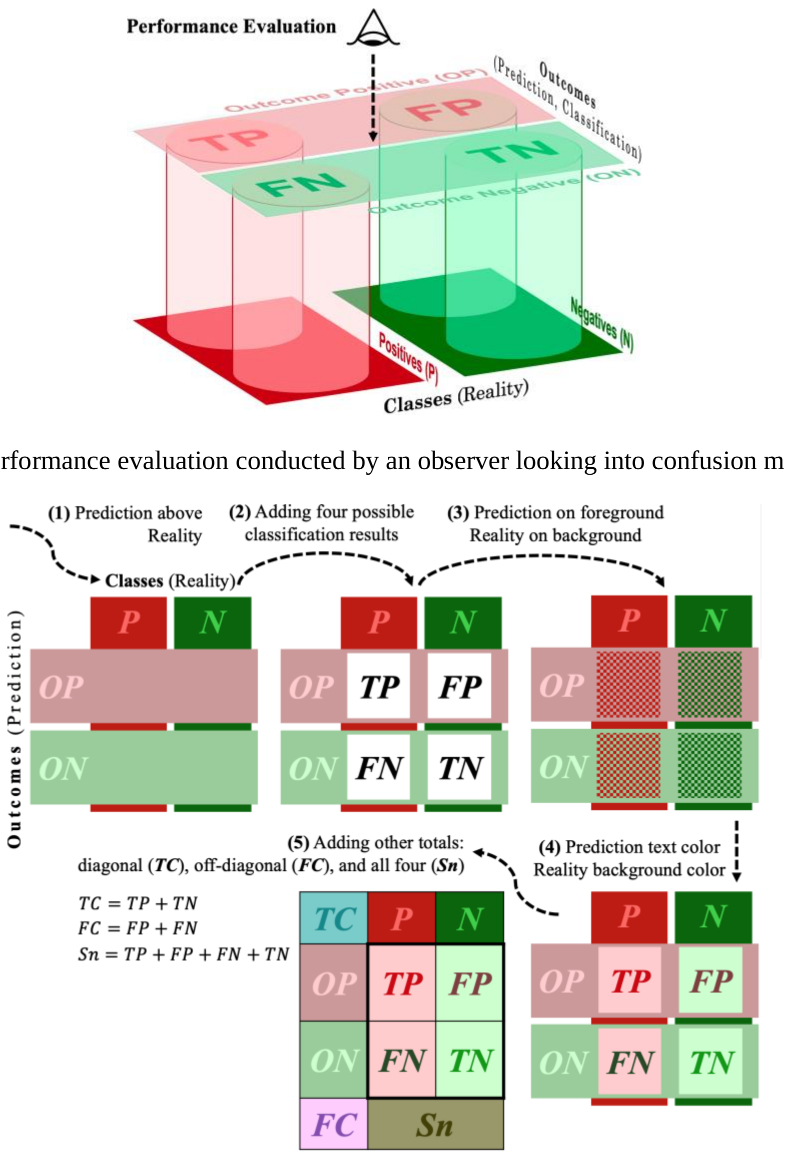
- *Green-like for negative*: Negative-class related canonical measures (***N***, ***ON***, ***TN***, and ***FP***) are green like colors implying secondary or normal concerns (*e.g.*, benign software, regular e-mail, or healthy).

In the proposed coloring scheme, foreground colors show the prediction or classification outcome colors whereas background colors show the reality or actual class colors as depicted in Figure 4.2 (b).

- *Clean background colors for real classes*: clean colors are used for depicting reality classes. Clean red (*red berry*)[18] for positive (***P***) and clean green (*camarone*) for negative class (***N***).

- *Dirty background colors for classification outcomes*: the background colors of ***OP*** and ***ON*** are dirty red and green, respectively, indicating that we do not know the reality (it can be either positive or negative). Therefore, dirty red (*eunry*) for outcome positive and dirty green (*de York*) for outcome negative.

[18] Color names are extracted from http://chir.ag/projects/name-that-color online color approximation tool


Performance Evaluation
Outcomes
(Prediction, Classification)
Outcome Positive (OP)
TP
FP
FN
TN
Outcome Negative (ON)
Negatives (N)
Positives (P)
Classes (Reality)
(1) Prediction above Reality
(2) Adding four possible classification results
(3) Prediction on foreground Reality on background
Classes (Reality)
Outcomes (Prediction)
P
N
OP
ON
TP
FP
FN
TN
(4) Prediction text color Reality background color
(5) Adding other totals: diagonal (TC), off-diagonal (FC), and all four (Sn)
TC = TP + TN
FC = FP + FN
Sn = TP + FP + FN + TN
TC
FC
Sn

(a) Performance evaluation conducted by an observer looking into confusion matrix

(b) Steps in defining color palettes for (1): ***P***, ***N***, ***OP***, and ***ON***; (2 – 4): ***TP***, ***FP***, ***FN***, and ***TN***; (5): ***TC***, ***FC***, and ***Sn***

Figure 4.2 Establishing the proposed coloring scheme via the geometries from the three-dimensional representation where the observer is above into the two-dimensional representations of base measures (confusion matrix)

Note that ***TC*** and ***FC*** are the canonical measures introduced in this thesis study enhancing the readability of the instrument equations (*e.g.*, *ACC* is defined as ***TC*** / ***Sn*** instead of (***TP*** + ***TN***) / ***Sn***. The colors dedicated to these measures should not be similar to red and green that are class-colors, because ***TC*** and ***FC*** are class-agnostic.

- *Turquoise-like for true classifications*: Because ***TC*** indicates favorable outputs of classification performance, turquoise-like colors are selected, namely *Monte Carlo* and *genoa* for the background and foreground colors of ***TC***.

- *Magenta-like for false classification*: To the contrary of ***TC***, ***FC*** indicates unfavorable outputs of classification performance. Therefore, magenta-like colors are selected, namely *pink lace* and *royal purple* for the background and foreground colors of ***FC***.

Brown-like color for sample size: The last 1st level measure is ***Sn*** that is the sum of all base measures (***TP*** + ***FP*** + ***FN*** + ***TN***), sum of column and row marginal totals of the base measures (***P*** + ***N*** and ***OP*** + ***ON***), and sum of diagonal and off-diagonal totals (***TC*** + ***FC***). Brown-like color is selected for ***Sn*** because it should not be similar to any of those measures (*avocado* for background color and *Verdun green* for foreground color).

4.2.1.2 Color palettes for base measures

Base measures give the classification result by checking the predictions against reality. The three-dimensional perspective depicted in Figure 4.2 helps to define the color palette depicting these conformances (i.e. ***TP*** and ***FP***) and non-conformances (i.e. ***FP*** and ***FN***). For example, the foreground color of ***FP*** is red-like indicating the outcome of the classification (positive) whereas background color is green-like indicating the reality (negative). Table 4.2 shows the color palette designed for representing the four base measures.

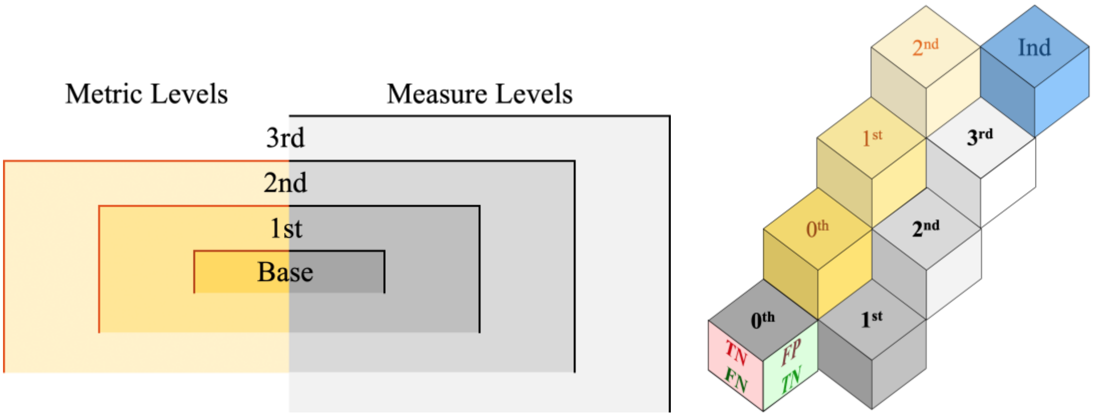
Table 4.2 Color palette for base measures (***TP***, ***FP***, ***FN***, and ***TN***)

Base Measures	Reality	Background	Prediction	Foreground	Fore/back-ground
True Positive	Positive	Red-like	Positive	Red-like	***TP***
False Positive	Negative	Green-like	Positive	Red-like	***FP***
True Negative	Negative	Green-like	Negative	Green-like	***FN***
False Negative	Positive	Red-like	Negative	Green-like	***TN***

4.2.1.3 Color palettes for instrument types and their leveling

The following colors are used to indicate instrument types and levels as shown in Figure 4.3.

- Gray-like colors for measures
- Orange-like colors for metrics
- Blue-like colors for indicators



(a) levels **(b)** levels and dependencies

Figure 4.3 Color palettes for instrument types and levels

Note that canonical measures in TasKar are also displayed in bold-italic as listed in Table 3.1. Next section reviews the literature on classification performance instrument visualization.

4.2.2 Survey 2: Visual representation of confusion matrix

The literature has not addressed the visualization of confusion matrix and performance metrics adequately. Researchers usually tend to report the success of their classification by some of the metrics at their choice instead of fully giving the four base measures. If they report, a 2x2 contingency tabular form is used without any visualization.

Alsallakh et al. (2014) designed a visualization tool called "confusion wheel" in order to show a multi-class classification confusion matrix. The visualization is based on a chord diagram having sectors representing the classes. The color palette chosen for representing base measures for given class against others are green (***TP***), orange (***FP***), red (***FN***), and gray (***TN***). To the contrary of the coloring scheme proposed and employed in this study, the colors do not suggest a semantic interpretation (*e.g.*, red for ***FN***).

Saito and Rehmsmeier (2015) use two semi-oval shapes to visualize the base measures as well as ***P***, ***N***, ***OP***, ***ON***, and ***Sn*** as shown in Figure 4.4 (b). The portion of the base measures shows the proportion of base measures. The size of ***P*** and ***N*** semi-ovals are changed for imbalanced samples.

Some engineering software packages also provide functions for plotting base measures. 3 (a) shows "plotconfusion" command in MATLAB ("Matlab: plotconfusion," 2018) whereas Figure 2 (d) shows "forfoldplot" in R (Friendly, 1995). The former displays the values of base measures, base measure rates, some base metrics in tabular form whereas the latter displays the values of base and first level measures along with scaled circular sections for base measures.

Figure 4.4 (c) and (e) are the examples displaying the base measures with Venn diagrams. Figure 4.4 (c) shows three cases of classification from top to bottom: regular case, no false positive, and no false negative (Nicolov, 2012). Figure 4.4 (e) shows an attempt to visualize performance metrics with Venn diagrams (Massich, 2015). It should be highlighted that the coloring scheme proposed in this study can also enhance the comprehension and interpretation of these visualization approaches. As seen in the review above, the representations of performance instruments are highly limited. The next section introduces a dashboard and calculator that is accompanied by PToPI for a wide range of performance instruments that can be used by the researchers, professionals, and students.

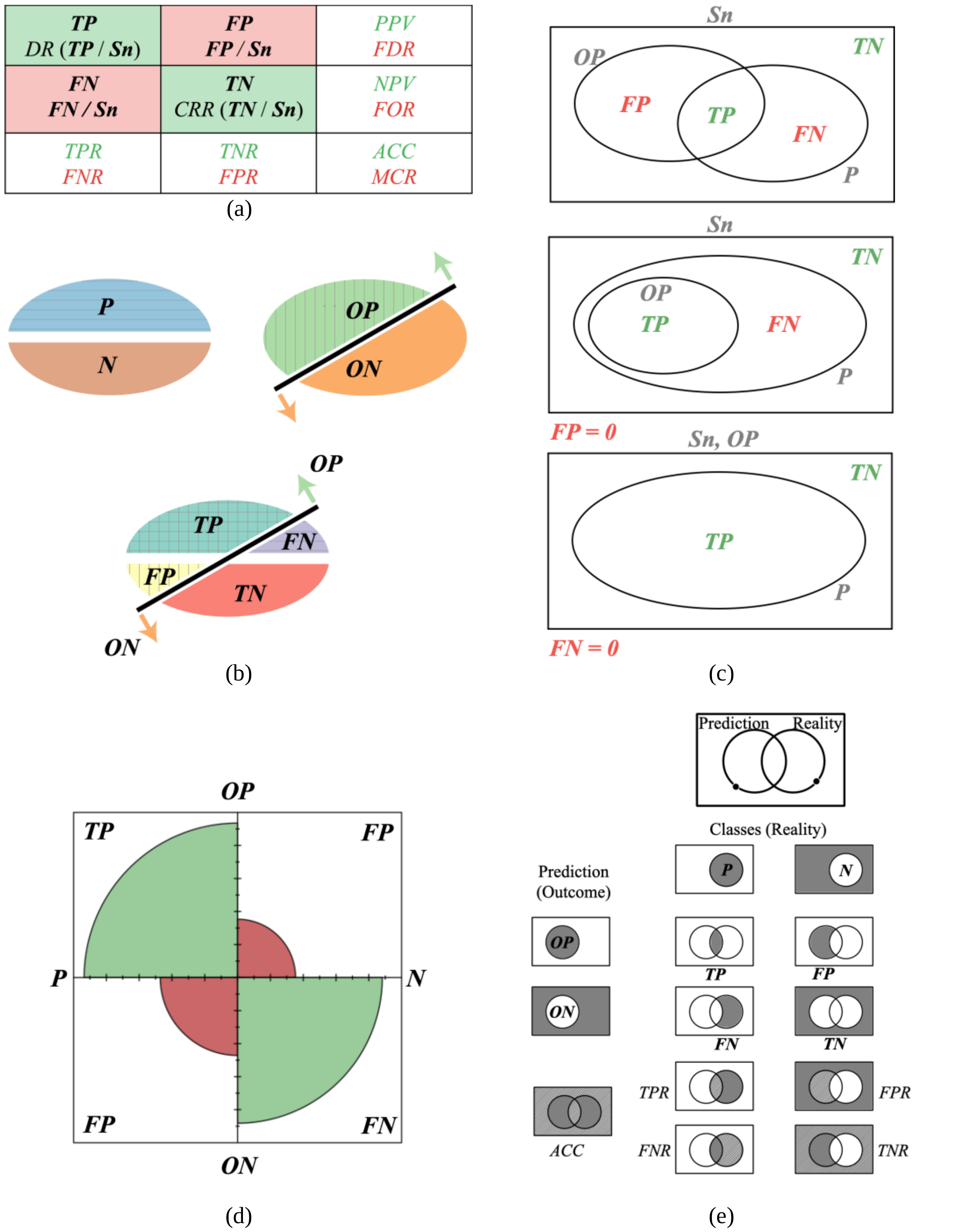


Figure 4.4 Related works on visualization of confusion matrix and performance metrics. **(a)** the tabular output of "plotconfusion" command in MATLAB. Only numbers are given ("Matlab: plotconfusion," 2018), **(b)** base measure visualization with semi-ovals (Saito & Rehmsmeier, 2015), **(c)** base measure visualization with Venn diagram (Nicolov, 2012), **(d)** the graphics output of "forfoldplot" command in R with circular sections (Friendly, 1995), **(e)** base measure and metrics visualization with Venn diagram (Massich, 2015)

4.2.3 TasKar overview

Based on the formatting scheme described above, a dashboard/calculator is designed named TasKar for binary classification performance instruments as shown in Figure 4.6. TasKar is implemented as an OpenDocument Spreadsheet document file (TasKar.ods) and provided online. Therefore, it does not require installing extra software besides an office package (the best viewed with LibreOffice version 6.2).

Figure 4.5 shows the parts and layout of the TasKar that consists of two parts vertically:

- Upper part: performance instruments
- Lower part: base metric graphics

The upper part comprises the instruments that are located as similar to PToPI as possible. The lower part provides three graphics to summarize base metrics.

The usage is straightforward. After opening the dashboard file, users can enter the classification results in the cells belonging to the confusion matrix (the cells under ***TP***, ***FP***, ***FN***, and ***TN*** base measure labels). The performance instruments are calculated and the graphics are updated automatically.

It is possible to compare two classification studies by opening two instances of the dashboard file and tiled horizontally on the desktop. The researchers can take the screenshot of the dashboard by adding the citation reference in the reserved cell in upper-middle and publish it.

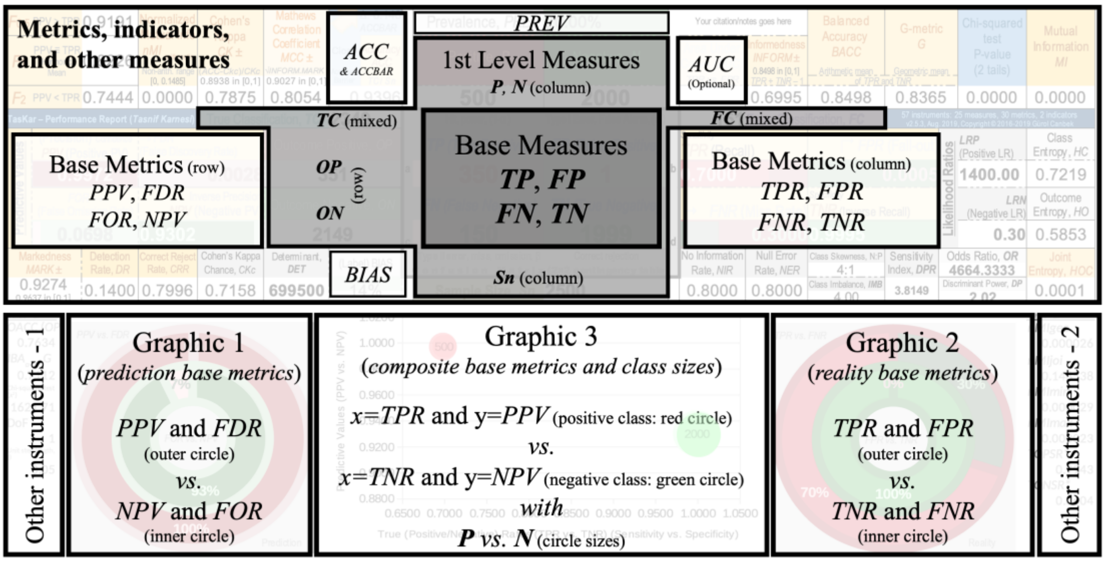
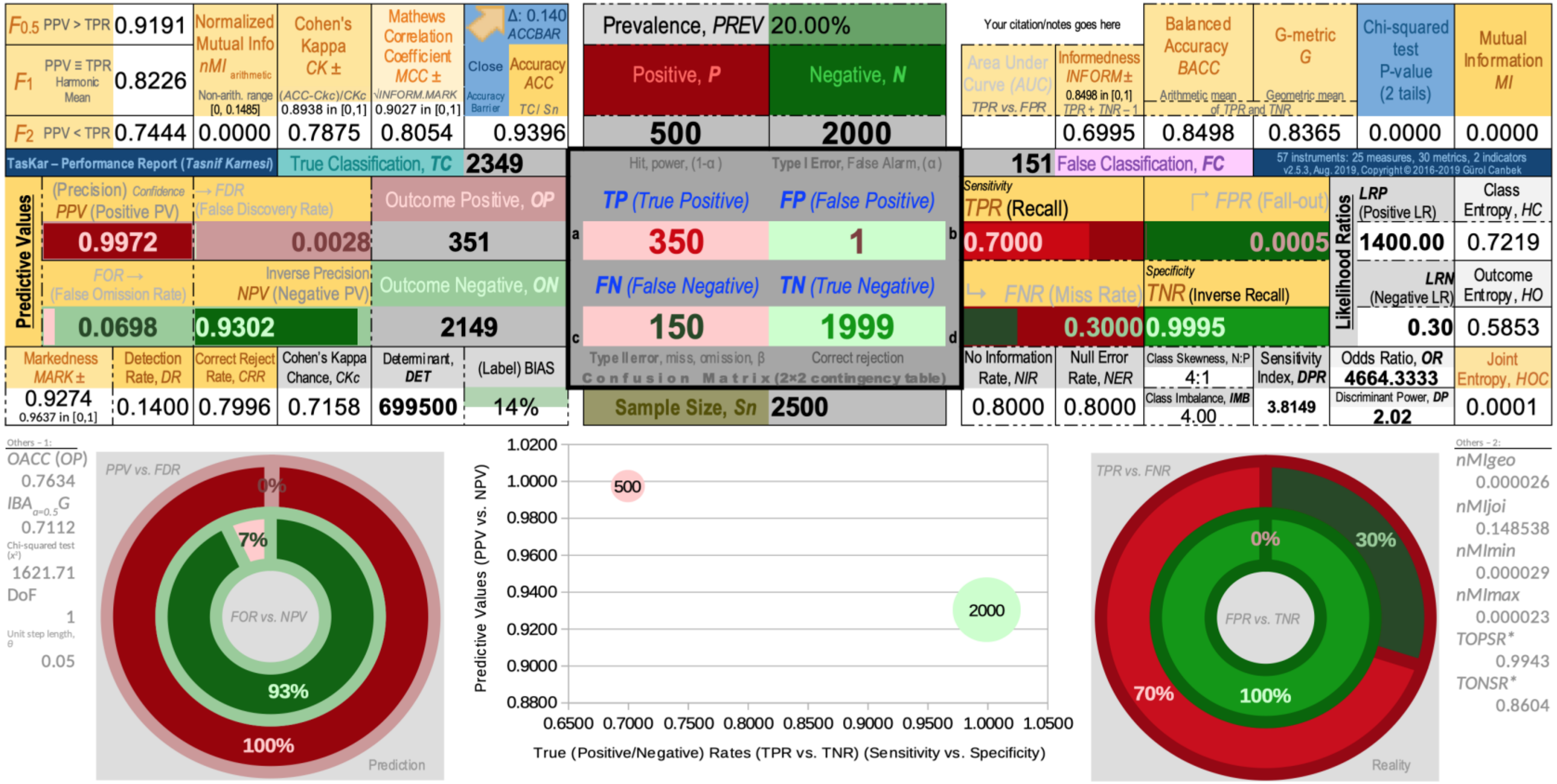


Figure 4.5 The layout of TasKar parts (performance instruments and graphics)



$F_{0.5}$ PPV > TPR 0.9191
F_1 PPV ≡ TPR Harmonic Mean 0.8226
F_2 PPV < TPR 0.7444

Normalized Mutual Info *nMI* arithmetic
Non-arith. range [0, 0.1485]
0.0000

Cohen's Kappa *CK*±
(ACC-Ckc)/CKc
0.8938 in [0,1]
0.7875

Mathews Correlation Coefficient *MCC*±
√INFORM.MARK
0.9027 in [0,1]
0.8054

Δ: 0.140 *ACCBAR*
Close Accuracy Barrier
Accuracy *ACC*
TC / Sn
0.9396

Prevalence, *PREV* 20.00%
Positive, *P* 500
Negative, *N* 2000

Your citation/notes goes here

Area Under Curve (AUC)
TPR vs. FPR

Informedness *INFORM*±
0.8498 in [0,1]
TPR + TNR – 1
0.6995

Balanced Accuracy *BACC*
Arithmetic mean of TPR and TNR
0.8498

G-metric *G*
Geometric mean
0.8365

Chi-squared test P-value (2 tails)
0.0000

Mutual Information *MI*
0.0000

TasKar – Performance Report (*Tasnif Karnesi*)
True Classification, *TC* 2349
151 False Classification, *FC*
57 instruments: 25 measures, 30 metrics, 2 indicators
v2.5.3, Aug. 2019, Copyright © 2016-2019 Gürol Canbek

Predictive Values

(Precision) Confidence
PPV (Positive PV)
0.9972

→ *FDR*
(False Discovery Rate)
0.0028

Outcome Positive, *OP*
351

FOR →
(False Omission Rate)
0.0698

Inverse Precision
NPV (Negative PV)
0.9302

Outcome Negative, *ON*
2149

Hit, power, (1-α)
Type I Error, False Alarm, (α)

a *TP* (True Positive) 350
b *FP* (False Positive) 1
c *FN* (False Negative) 150
d *TN* (True Negative) 1999

Type II error, miss, omission, β
Correct rejection
Confusion Matrix (2×2 contingency table)

Sensitivity
TPR (Recall)
0.7000

↱ *FPR* (Fall-out)
0.0005

↳ *FNR* (Miss Rate)
0.3000

Specificity
TNR (Inverse Recall)
0.9995

Likelihood Ratios

LRP (Positive LR)
1400.00

LRN (Negative LR)
0.30

Class Entropy, *HC*
0.7219

Outcome Entropy, *HO*
0.5853

Markedness *MARK*±
0.9274
0.9637 in [0,1]

Detection Rate, *DR*
0.1400

Correct Reject Rate, *CRR*
0.7996

Cohen's Kappa Chance, *CKc*
0.7158

Determinant, *DET*
699500

(Label) BIAS
14%

Sample Size, *Sn* 2500

No Information Rate, *NIR*
0.8000

Null Error Rate, *NER*
0.8000

Class Skewness, N:P
4:1

Class Imbalance, *IMB*
4.00

Sensitivity Index, *DPR*
3.8149

Odds Ratio, *OR*
4664.3333

Discriminant Power, *DP*
2.02

Joint Entropy, *HOC*
0.0001

Others - 1:
OACC (OP)
0.7634
$IBA_{\alpha=0.5}G$
0.7112
Chi-squared test (x^2)
1621.71
DoF
1
Unit step length, θ
0.05

Others - 2:
nMIgeo
0.000026
nMIjoi
0.148538
nMImin
0.000029
nMImax
0.000023
TOPSR*
0.9943
TONSR*
0.8604

Figure 4.6 A screenshot of TasKar

4.2.3.1 TasKar features

Some of the features of TasKar can be summarized as follows:

- Base measure cells are also captioned as “a” (***TP***), “b” (***FP***), “c” (***FN***), and “d” (***TN***) notation that is a convention in similarity/dissimilarity (distance) between two binary matrices, diagnostic tests, association measures, many 2x2 contingency table analysis such as meteorology forecasting skill scores (Wilks, 2006, p. 261), and even early classification performance evaluation studies.

- Like PToPI, the first level measures namely ***P***, ***N***, ***OP***, ***ON***, ***TC***, ***FC***, and ***Sn*** are located around the confusion matrix according to their dependencies (*e.g.*, ***P*** is above ***TP*** and ***FN***, because ***P*** = ***TP*** + ***FN***; also, ***OP*** is located at the left of ***TP*** and ***FP***).

- *PREV* and *BIAS* that are the important measures of classification studies are located near confusion matrix.

- The background color of the values of *PREV* and *BIAS* reflect the weight of the class: small values (less than 0.5) are getting green indicating negative class dominance, large values (more than 0.5) are getting red indicating the positive class, and values around middle (about 0.5) is white that is ideal for a classification study.

- Class skewness (*SKEW*) and class imbalance (*IMB*) are also displayed at the right-bottom of the upper part.

- Instrument geometries are depicted via the dashed edges similar to PToPI *(see* Table 4.1)

- Column geometry base metrics *TPR*, *FNR*, *TNR*, *FPR* and row geometry base metrics *PPV*, *FDR*, *NPV*, *FOR* are presented at the right and left of confusion matrix, respectively, as shown in Figure 4.5[19].

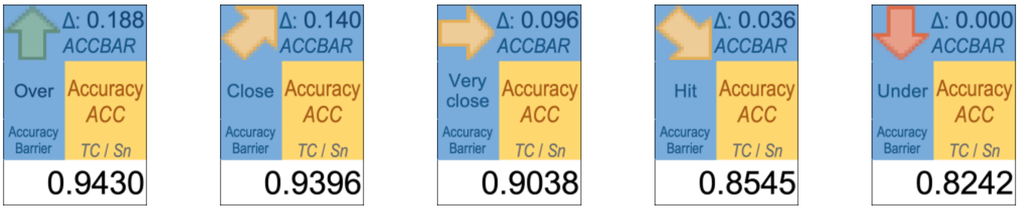
- Those eight metrics are also visualized via bar graphs besides their actual values using the coloring scheme.

- Metric complements are indicated with arrows and gray text color in their labels denotes redundancy (*FDR* is the complement of *PPV* and *FPR* is the complement of *TNR*).

- For the sake of completeness, although it is not based on confusion matrix, *AUC* can be entered into the cell at the middle-top for reporting purposes.

- *ACCBAR* indicator is also integrated into TasKar that shows how the classification is close to accuracy barrier as described in Section 3.11. Figure 4.7 shows the indicator categories.

[19] The column instruments are not positioned above confusion matrix except ***P***, ***N***, and *PREV* because of the design goal of the tool of making a compact tool in a minimum size.



(a) Over (b) Close (c) Very close (d) Hit (e) Under

Figure 4.7 TasKar showing the accuracy barrier indicator categories

The features provided in TasKar can facilitate the performance evaluation phase of binary classification studies accurately and objectively. Presenting all the instruments together avoid ignoring the prominent aspects of a specific classification application. For example, class imbalance and underperformance in terms of specific metrics. If we calculate and see only some of the metrics such as *ACC*, *F1*, *PPV*, *BACC*, *G*, or *TPR*, the performance evaluation misleads that the classifier achieves high performance.

Note that TasKar is implemented as a self-contained tool. Due to the lack of space and the nature of the tool with respect to end-user requirements, it cannot and does not need be as informative as PToPI.

4.2.3.2 TasKar graphics

The performance values in terms of various instruments are helpful for seeing the complete results and focusing on different metrics together. However, interpretation of the overall performance might be difficult by analyzing the numbers only. In order to help researchers in the interpretation of the metrics and give more insights, the following three kinds of graphics are developed further in this thesis study:

- Graphic 1 (prediction base metrics)
- Graphic 2 (reality base metrics)
- Graphic 3 (composite base metrics and class sizes)

Figure 4.8 shows these graphics, which are described below, in an example case with ***TP*** = 300, ***FP*** = 25, ***FN*** = 50, and ***TN*** = 475 base measures.

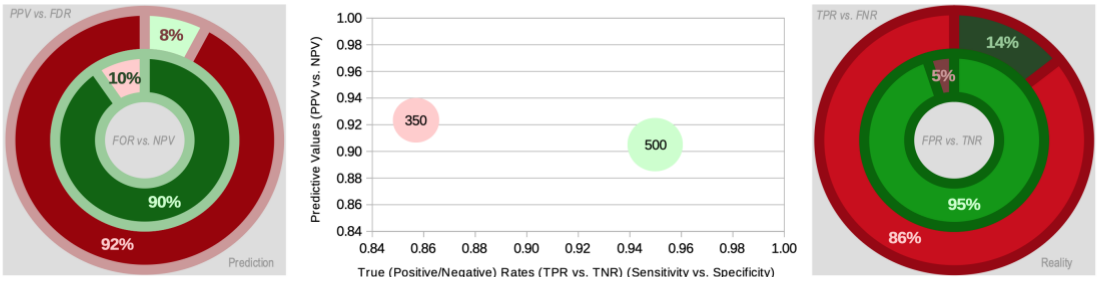


Figure 4.8 TasKar graphics for an example classifier with ***TP*** = 300, ***FP*** = 25, ***FN*** = 50, and ***TN*** = 475

Graphic 1 (prediction base metrics) shows the two complements of two prediction base metrics (*i.e.* in row geometry) per each class in two nested circles. The outer circle is for positive class and the inner circle is for negative class prediction base metrics.

Graphic 2 (reality base metrics) shows the same graphics for two reality base metrics (i.e. in column geometry).

Graphic 3 (composite base metrics and class sizes) provides an overall overview of the base metrics enhanced with the class sample sizes. It is alone a comprehensive graphic to summarize the overall performance, therefore it could be used in performance reporting in the literature (the same graphic for the most competing classification in a domain can also be presented side by side or in the same graphic)

Note that the precision of the base metric values are decreased to two digits to simplify performance evaluation (four digits are presented in the upper part).

4.2.3.3 Example real-word usage of TasKar graphics

Interpreting the graphics given in Figure 4.8, the followings could be inferred:

- Comparing Graphics 1 and 2 together; positive class performance is less than negative class in reality (as seen in Graphic 2) while it is better in prediction (as seen in Graphic 1). More specifically, *FNR* (14%, type II error) is higher than all the other false classifications (*FOR* = 10%, *FDR* = 8%, and *FPR* = 5%)[20].

- Via Graphic 3, the predictive power of the classifiers on both classes is close but this power does not reflect in reality (the circles are closer in vertical axis than the horizontal axis).

- Further, the class imbalance can be observed easily via the representation of the class sizes in Graphic 3.

TasKar graphics can provide different insights on evaluating a classifier’s performance in other real-world use cases. It is also helpful in comparing two different classifiers to help in noticing the differences.

The two graphics at the left and right are more detailed and comprehensible comparing a small number of attempts reviewed above. Graphic 3 especially is informative as it gives a clear insight by showing the performance in terms of both classes’ reality and prediction performances and reflecting the class imbalance in a single picture.

[20] Reporting *PPV* as 92% and *FPR* as 5% only makes this classifier as a promising one.

CHAPTER 5

BenchMetric: SYSTEMATIC BENCHMARKING OF PERFORMANCE METRICS

Analyzing performance instruments increase our overall understanding of performance evaluation and its instruments. The provided tools are also helpful in comprehending the instruments as well as conducting performance evaluation. Nevertheless, the critical question is “What is the best metric?” or put it in a more correct expression, “What is the most robust metric that should be used in performance evaluation, comparison, and reporting?” addressing (**RQ4**). This question must be answered in an incontrovertible proof on behalf of the researchers who even embrace the concepts and practically use the tools provided in this thesis. This chapter also addresses the second research question in (**RQ4**) “What should be reported for expressing classification performance?” by recommending a proper approach.

In order to answer these key questions, a benchmarking method named BenchMetric is proposed to evaluate all the performance metrics from a comprehensive perspective in a methodological manner. BenchMetric comprises the following three stages, which are described in the following sections:

- Stage-1: *Extreme cases*: Performance of thirteen extreme classification result cases are measured by each metric and the outputs are inspected.

- Stage-2: *Mathematical evaluation*: The equations of each metric and the metric-spaces are evaluated according to eleven different criteria.

- Stage-3: *Meta-metrics*: The robustness of each metric is evaluated by seven novel meta-metrics (*i.e.* metrics about (performance) metrics) defined formally in metric-space.

5.1 Benchmarking Data

This section introduces a new aspect of metrics named “metric-space” before describing the benchmarking method in stages. The benchmark stages are conducted on the metric-spaces.

5.1.1 Metric-space: metric distribution in pseudo-universal “base performance measure permutations”

A metric-space indicates all possible permutations of base (performance) measures (***TP***, ***FP***, ***FN***, and ***TN***) yielding the same ***Sn***. A metric-space (**M**) holds all possible results of a hypothetical classification conducted in a dataset with a given sample size in terms of a specific metric (*M*). Metric-space provides a pseudo-universal space to analyze and compare metrics in the complete coverage. Recall that metric-spaces are represented in bold (*e.g.*, **ACC** metric-space vector for *ACC* metric), single metric values in italic (*e.g.*, *ACC* = 0.900), and set or array of metric values and limitless measures in bold-italic (*e.g.*, ***BM*** = **{*TP*** = 7, ***FP*** = 1, ***FN*** = 0, ***TN*** = 2} and ***Sn*** = 25). Because metrics are the ratios and sample sizes are reduced in the numerator/denominator of the metrics' equations, we can calculate metric-spaces (*i.e.* all possible values of a given metric per base measure permutation in given sample size) as formally expressed in Definition 5.1.

Definition 5.1 (*Universal Base Measure Permutations*).

A vector $\mathbf{BM}^{\boldsymbol{Sn}}$ shows all possible base measure permutations with repetition where each *ith* element of $\mathbf{BM}^{\boldsymbol{Sn}}$ is $\mathbf{BM}_i^{\boldsymbol{Sn}}: \boldsymbol{BM} \rightarrow \mathbb{Z}^{*4}$ and $\boldsymbol{BM} = \{\boldsymbol{TP}, \boldsymbol{FP}, \boldsymbol{FN}, \boldsymbol{TN}\}$ and $\boldsymbol{TP}_i + \boldsymbol{FP}_i + \boldsymbol{FN}_i + \boldsymbol{TN}_i = \boldsymbol{Sn}$ and $\boldsymbol{BM} = \{\boldsymbol{bm} | 0 \leq \boldsymbol{bm} \leq \boldsymbol{Sn}\}$.

Definition 5.2 (*Metric-Space*).

A metric-space $\mathbf{M}$ or $\mathbf{M}^{\boldsymbol{Sn}}$ covers the outputs given by an *M* metric for all the elements of $\mathbf{BM}^{\boldsymbol{Sn}}$ universal base measure permutations.

For example, there is a total of 286 permutations of four base measures with repetition for 10 samples where the sum of the measures is equal to 10. An example permutation might be 10 true positives only (all others are zero) and another example might be 7 true positives, 1 false positive, and 2 true negatives. The metric-spaces of *F1*, *ACC*, and *MCC* are also calculated per each permutation.

Note that the size of base-measure permutations and metric-spaces increases exponentially with $\boldsymbol{Sn}$. For instance, it is 2,667,126 for 250 samples. When metric-spaces are used in the experiments, the related benchmarking criteria are tested with different $\boldsymbol{Sn}$ values. It is observed that the results are the same or converge as $\boldsymbol{Sn}$ increases but they are representative while comparing a group of metrics or at least consistent within a specific $\boldsymbol{Sn}$. As a result, the maximum sample size is limited to 250 in order to keep the permutation size and calculation time in a reasonable range. Calculation of the meta-metrics in metric-spaces in up to 250 sample size (except for consistency and discriminancy meta-metrics) takes maximum one minute on an R version 3.5.2 (2018-12-20) platform on a Darwin 15.6.0 operating system with 2.3 GHz CPU and 16 GB RAM. The calculation of the proposed meta-metrics for a single metric takes 21 hours and 45 minutes. Note that detailed time test results and metric-spaces for different sample sizes between 10 and 250 are provided in the online material described in Section 1.6.

5.2 Experiment 1: Benchmarking 13 Performance Metrics

The following sections from Section 5.3 and Section 5.6 define and describe the criteria and stages proposed in BenchMetric method as well as demonstrates them via the experimentation conducted on benchmarking 13 metrics namely *TPR*, *TNR*, *PPV*, *NPV*, *ACC*, *INFORM*, *MARK*, *BACC*, *G*, *nMI*, *F1*, *CK*, and *MCC*. Section 5.6 summarizes the overall benchmarking result.

5.3 BenchMetric Stage-1: Extreme Case Benchmarking

Stage-1 gives initial insights about the robustness of the metrics where thirteen extreme classification result cases are defined on 10.000 samples and the corresponding performances in terms of each metric are evaluated. Basically, a performance metric should be accurate in all these extremes. Table 5.1 shows the cases defined by some specific base measures and corresponding performances calculated in terms of thirteen performance metrics.

The base measures are calculated based on sample size (***Sn***) parameter according to the equations given in the footnote of the table. The performance values are in [0, 1] range where 0 and 1 denote lowest and highest performances, respectively. Note that the metrics with apostrophe (*e.g.*, *MCC'*) indicates that bi-directional metric (*i.e.* [-1, 1]) is normalized into [0, 1] range to simplify the assessment.

Three benchmark criteria are defined in Stage-1:

1) "Does a metric yield not-a-number (NaN, *i.e.* 0/0) in extreme cases?"
2) "Are the performance metric values of the cases from 5 to 9 decreasing?"
3) "Are the performance metric values symmetric for both classes?"

The problematic behaviors under those criteria are depicted in bold underlined texts. Note that the metrics are also sorted horizontally in Table 5.1 according to the total ranking of their non-conformance with the criteria. The followings are some highlights:

- The first criterion in Stage-1 is that a proper metric should not yield undefined results. For example, *PPV* and *MARK* are NaN for the case 12 on 1 positive 9999 negative samples.

- The second criterion examines the logical performance order of a metric in the same number of positive and negative samples. The performances for the extreme case 5 to case 9 expressed by a metric should satisfy *p*4 > *p*3 > *p*2 > *p*1 > *p*0, respectively. Notably, only *nMI* does not follow it accurately for case 8 and case 9, which corresponds to almost and exactly full false-classifications (0.9973 for *p*1 where ***TP*** = ***TN*** = 1 and 1 for *p*0 where ***TP*** = ***TN*** = 0).

- The third criterion is that a metric should not differentiate the performances in symmetric conditions of both classes. In extreme cases 1 and 13 having positive only and negative only samples and/or extreme cases 2 and 12 having almost positive and negative samples yield similar performances (*i.e. p*i ≈ *p*j, *p*ii ≈ *p*jj). *F1*, for example, yields 0.9999 for positive-only and almost-positive samples whereas it yields 0.0 for the symmetric cases. Hence, *F1* is not sensitive to negative-class performance.

Overall assessment of Stage-1 reveals that *ACC*, *CK*, and *MCC* are the most and *nMI* is the least robust performance metrics.

Table 5.1 Stage-1 benchmarking of 13 performance metrics according to assessment through 13 proposed extreme cases

Case	***P***	***N***	***TP****	***FP****	***FN****	***TN****	Performance	*ACC*	*CK'*	*MCC'*	*F1*	*TPR*	*TNR*	*PPV*	*NPV*	*BACC*	*G*	*INFORM'*	*MARK'*	*nMI*
1	10000	0	9999	0	1	0	*p*i	0.9999	0.5000	0.5000	**0.9999**	0.9999	**NaN**	**1.0000**	**0.0000**	**NaN**	**NaN**	**NaN**	0.5000	**NaN**
2	9999	1	9999	1	0	0	*p*ii	0.9999	0.5000	0.5000	**0.9999**	**1.0000**	**0.0000**	0.9999	**NaN**	0.5000	0.0000	0.5000	**NaN**	0.0000
3	9998	2	9997	1	1	1		0.9998	0.7499	0.7499	0.9999	0.9999	0.5000	0.9999	0.5000	0.7499	0.7071	0.7499	0.7499	0.3908
4	6666	3334	3333	3333	3333	1		0.3334	0.2501	0.2501	0.5000	0.5000	0.0003	0.5000	0.0003	0.2501	0.0122	0.2501	0.2501	0.2727
5	5000	5000	5000	0	0	5000	*p*4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
6			4999	1	1	4999	*p*3	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9973
7			2500	2500	2500	2500	*p*2	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.0000
8			1	4999	4999	1	*p*1	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	**0.9973**
9			0	5000	5000	0	*p*0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**1.0000**
10	3334	6666	1	3333	3333	3333		0.3334	0.2501	0.2501	0.0003	0.0003	0.5000	0.0003	0.5000	0.2501	0.0122	0.2501	0.2501	0.2727
11	2	9998	1	1	1	9997		0.9998	0.7499	0.7499	0.5000	0.5000	0.9999	0.5000	0.9999	0.7499	0.7071	0.7499	0.7499	0.3908
12	1	9999	0	0	1	9999	*p*jj	0.9999	0.5000	0.5000	**0.0000**	**0.0000**	**1.0000**	**NaN**	0.9999	0.5000	0.0000	0.5000	**NaN**	0.0000
13	0	10000	0	1	0	9999	*p*j	0.9999	0.5000	0.5000	**0.0000**	**NaN**	0.9999	**0.0000**	**1.0000**	**NaN**	**NaN**	**NaN**	0.5000	**NaN**
		Criteria	No not-a-number									-1	-1	-1	-1	-2	-2	-2	-2	-2
			*p*4 > *p*3 > *p*2 > *p*1 > *p*0				Contradictions													-1
			*p*i ≈ *p*j, *p*ii ≈ *p*jj								-1	-1	-1	-1	-1					
							Stage-1 Rank	1			4	5								13

* Base measures calculation for each extreme case according to given sample size (*Sn*). Case 1: *TP* = *Sn* – 1; Case 2: *TP* = *Sn* – 1; Case 3: *TP* = *Sn* – 3; Case 4: *TP* = *FP* = *FN* = (*Sn* – 1)/3; Case 5: *TP* = *TN* = *Sn*/2; Case 6: *TP* = *TN* = *Sn*/2 – 1; Case 7: *TP* = *FP* = *FN* = *TN* = *Sn*/4; Case 8: *FP* = *FN* = *Sn*/2 – 1; Case 9: *FP* = *FN* = *Sn*/2; Case 10: *FP* = *FN* = *TN* = (*Sn* – 1)/3; Case 11: *TN* = *Sn* – 3; Case 12: *TN* = *Sn* – 1; Case 13: *TN* = *Sn* – 1. Other base measures not given here are the same as in the *TP*, *FP*, *FN*, *TN* columns above. *P*, *N*, and metric values are calculated according to four base measures.

5.4 BenchMetric Stage-2: Mathematical Evaluation Benchmarking

In Stage-2, eleven criteria are proposed to evaluate different metrics from mathematical perspectives.

5.4.1 Criteria 2.1–2.3: All-purpose coverage

A robust metric –by definition– should not have a missing facade of fundamental performance elements (***TP***, ***FP***, ***FN***, ***TN***, ***P***, ***N***, ***OP***, ***ON***). Otherwise, they cannot be effective to summarize the confusion matrix and number of classes and classification outputs. The following three criteria are provided to distinguish the limitations of metrics by evaluating the metrics expressed in canonical form defined in Definition 3.1:

- *Criterion 2.1 (Outcome/class coverage)*: Metrics should not be sensitive to outcome base-measures-only (*i.e.* includes ***OP*** and/or ***ON*** without ***P*** and ***N***) or class base-measures-only (*i.e.* includes ***P*** and/or ***N*** without ***OP*** and ***ON***).

- *Criterion 2.2 (Class coverage)*: Metrics should fully cover the classes (***P***, ***N***) without excluding any class.

- *Criterion 2.3 (Base measure coverage)*: Metrics should cover base performance measures (***TP***, ***FP***, ***FN***, *and* ***TN***) without excluding any measure.

5.4.2 Criteria 2.4–2.6: Variance/invariance

Contrary to other measures/metrics such as association measures, invariance (*i.e.* not differentiating the swaps among base measures) might not be a desired characteristic of a robust performance metric, because any change making four base measures of the confusion matrix different overall should usually be distinguished. Figure 5.1 depicts the three types of swaps that are used to assess metrics' variance in BenchMetric.

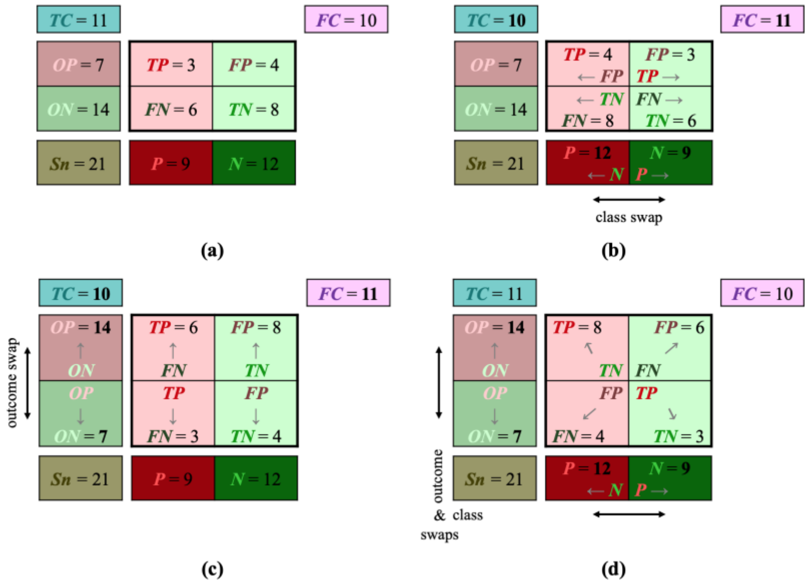


Figure 5.1 Three types of swaps of **(a)** an original confusion matrix (base measures): **(b)** class swap (horizontally: between ***TP*** and ***FP*** along with ***FN*** and ***TN***), **(c)** outcome swap (vertically: between ***TP***

and ***FN*** along with ***FP*** and ***TN***), and **(d)** class-and-outcome swaps (diagonally: between ***TP*** and ***TN*** along with ***FP*** and ***FN***)

A toy classification example is provided in Figure 5.1. A robust performance metric should be variant to class swap and variant to outcome swap because base measures become different as given in Figure 5.1 (b) and (c) with the original ones in Figure 5.1 (a). Otherwise, the metric does not differentiate such classification results.

In order to find the variance or invariance of a metric, the base measures in the equation of a metric should be changed according to the type of swaps as shown in Figure 5.1 (b – d) and the original and swapped version equations are compared. For example, swapping classes in $TPR = TP/P = TP/(TP + FN)$ makes the equation $FP/(FP + TN) = FP/N = FPR$, which is different from the original metric. Hence, *TPR* is variant to class swap. Whereas, class-and-outcome swaps in $MCC = (TP \cdot TN - FP.FN)/\sqrt{P \cdot N \cdot OP \cdot ON}$ make no variance: $(TN \cdot TP - FN.FP)/\sqrt{OP \cdot ON \cdot P \cdot N} = MCC$.

Table 5.2 also shows the known metrics corresponding to each swap. Only two metrics are identified that contradict these criteria: *nMI* and *F1*. *F1* is not invariant to class and outcome swaps because it has no ***TN*** coverage as addressed in base measure coverage in Table 5.2.

In the literature, Sokolova (2006) suggests four invariance properties, only one of which corresponds with the proposed criterion namely class-and-outcome swapping and examines six metrics only (*TPR*, *TNR*, *PPV*, *ACC*, *INFORM*, and *F1*). The other two actually indicate the variance by changing *TP*-only and *FP*-only that are easily evaluated by our base measure coverage criterion. Likewise, the fourth property is actually scaling ***OP*** components (***TP*** and ***FP***) and ***ON*** components (***FN*** and ***TN***) separately. This also corresponds to Criterion 2.1 (Outcome/class coverage).

5.4.3 Criteria 2.7–2.11: Descriptive statistics

The general analysis of all possible outcomes of a performance metric can increase the overall understanding of its behavior in a complete scope. The distribution and descriptive statistics such as range, mean, median and standard deviation calculated for the metric-space of a metric give fundamental insights about the dispersions and transitions of metric outputs.

Figure 5.2 illustrates density graphs along with some of the statistics per metric namely range, mean, median, and mode. Each density graph shows the metric-space in terms of relative frequencies per equally spaced breaks in the metric's range. A fitted normal distribution curve over the mean is also attached where possible (*i.e.* **ACC**, **INFORM**, **BACC**, **CK**, and **MCC**).

The most important findings shown in Figure 5.2 are that although all the metrics summarize the four or fewer base measures into a single figure in a specific range, the distributions are different from each other and not all the performance metrics show smooth and continuous transitions. The revealed difference could be another motivation to identify the most robust metric. The following defined criteria are important for metric evaluation:

- *Criterion 2.7 (Undefined (NaN) counts)*: The number of undefined values (not-a-numbers, NaN) is listed in Table 5.2. The NaN count of **MCC** is the highest with proportional to ***Sn***, whereas **ACC**, **F1**, and **CK** have 0, 1, and 2 NaNs, respectively regardless of ***Sn***. Robust metrics should calculate any base measure permutations, without any exception. Note that this criterion is different from the first criterion in Stage-1 that covers only a few extreme cases.

- *Criterion 2.8 (Central tendencies)*: The central tendency defined by mean, median, and mode should be examined for metric-spaces. Only **INFORM**, **MARK**, and **BACC** have exactly the same three central tendencies. However, a mean-median difference (*i.e.* arithmetic average *vs.* positional average in sorted metric-space), which could be the indication of an imbalance in mapping the uniform classification performance results (*i.e.* base measure permutations) to the corresponding uniform output ranges of a metric-space, was observed in **nMI** and **CK** (even though **CK** is symmetric).

- *Criterion 2.9 (Standard deviation)*: Informatively, the standard deviation of **nMI** and **CK** are the lowest indicating low dispersion around their mean values whereas others disperse over a higher range of values in metric-space as can be seen in Figure 5.2.

The shape of distributions: Criterion 2.10 (skewness) and Criterion 2.11 (kurtosis): Table 5.2 shows two values to recognize the shape of metric-space distribution and dispersion shown in the graphs in Figure 5.2. Most of the metric-spaces are symmetric and platykurtic (thin-tailed) except **CK**, **F1**, **G**, and **nMI**. Note that **G** and **F1** metric-spaces exhibit unexpected distortions by yielding zero dominantly, which indicates the unusual accumulation points in metric-space.

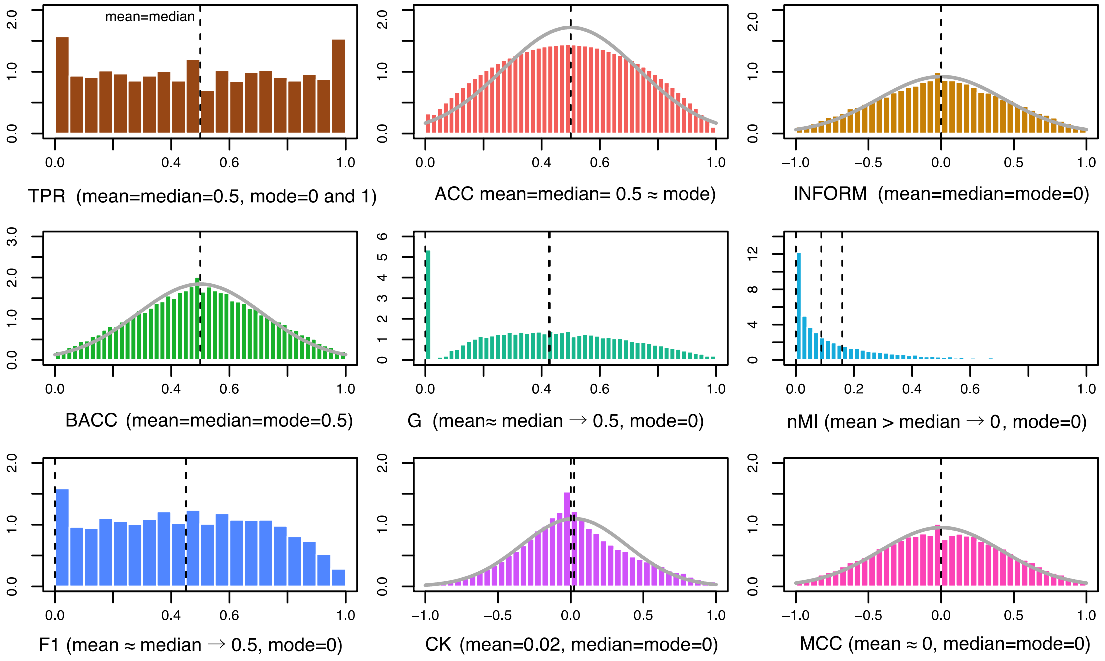


Figure 5.2 Density graphs summarizing each of the 9 metric-spaces (**TNR**, **PPV**, and **NPV** are the same as **TPR**; **MARK** is the same as **INFORM**). The area under curves are one.

Table 5.2 shows the results of the Stage-2 benchmarking along with the metrics' ranks. Note that underlined bold texts depict the deficiencies and each criterion is taken as equally important and the last three criteria (standard deviation, skewness, and kurtosis) are informative and not included in benchmarking evaluations.

Table 5.2 Stage-2 benchmarking of 13 performance metrics according to 8 proposed criteria along with three informative criteria

Stage-1 Criteria	**CK**	**MCC**	**F1**	**INFORM**	**MARK**	**BACC**	**G**	**ACC**	**TPR**	**PPV**	**TNR**	**NPV**	**nMI**
2.1 Outcome/class coverage	Yes	Yes	Yes	**Class-only**	**Outcome-only**	**Class-only**	**Class-only**	**None**	**Class-only**	**Outcome-only**	**Class-only**	**Outcome-only**	Yes
2.2 Class coverage (***P*** and ***N***)	Yes	Yes	Yes	Yes				**None**	***P*-only**		***N*-only**		Yes
2.3 Base measure coverage	Yes	*Yes*	**No *TN***	***TP, TN***				***TP, TN***	***TP***		***TN***		Yes
2.4 Variant to class swap	Yes	Yes	Yes	Yes				Yes (*MCR*)	Yes (*FPR*)	Yes (*FDR*)	Yes (*FNR*)	Yes (*FOR*)	***No (nMI)***
2.5 Variant to outcome swap	Yes	Yes	Yes	Yes				Yes (*MCR*)	Yes (*FNR*)	Yes (*FOR*)	Yes (*FPR*)	Yes (*FDR*)	***No (nMI)***
2.6 Invariant to class-and-outcome swaps	Yes	Yes	**No**	Yes				Yes	Yes				Yes
2.7 Undefined (NaN) count	2	***4Sn***	1	**2(*Sn*+1)**				0	***Sn*+1**				4
2.8 Central tendencies (mean-median difference)[1]	$\underline{\bar{\mathbf{M}} \neq \tilde{\mathbf{M}}} = \mathbf{Mo}$	$\bar{\mathbf{M}} \approx \tilde{\mathbf{M}} = \mathbf{Mo}$	$\bar{\mathbf{M}} \approx \tilde{\mathbf{M}} \neq \mathbf{Mo}$	$\bar{\mathbf{M}} = \tilde{\mathbf{M}} = \mathbf{Mo}$			$\bar{\mathbf{M}} \approx \tilde{\mathbf{M}} \neq \mathbf{Mo}$	$\bar{\mathbf{M}} = \tilde{\mathbf{M}} \approx \mathbf{Mo}$	$\bar{\mathbf{M}} = \tilde{\mathbf{M}} \neq \mathbf{Mo}$				$\underline{\bar{\mathbf{M}} \neq \tilde{\mathbf{M}}} \neq \mathbf{Mo}$
Stage-1 Rank	1		3	4				8	9				13
Other informative Criteria (i.e. not used in ranking)													
2.9 Standard Deviation	**0.18**[2]	0.20[2]	0.22	0.21[2]	0.21[2]	0.2	0.23	0.26	0.29	0.29	0.29	0.29	**0.17**
2.10 Skewness	Slightly positive[3] (0.16)	Symmetric (0)	Slightly positive[3] (0.05)	Symmetric (0)	Symmetric (0)	Symmetric (0)	Slightly positive[3] (0.18)	Symmetric (0)	Symmetric (0)	Symmetric (0)	Symmetric (0)	Symmetric (0)	**Highly positive**[3] **(1.69)**
2.11 Kurtosis[4]	Platykurtic (-0.2)	Platykurtic (-0.6)	Platykurtic (-1.07)	Platykurtic (-0.6)	Platykurtic (-0.6)	Platykurtic (-0.6)	Platykurtic (-0.85)	Platykurtic (-0.86)	Platykurtic (-1.2)	Platykurtic (-1.2)	Platykurtic (-1.2)	Platykurtic (-1.2)	**Leptokurtic (2.75)**

Notes: **(1)** $\bar{\mathbf{M}}$: Mean, $\tilde{\mathbf{M}}$: Median, and **Mo** Mode of a metric-space
(2) When normalized into [0, 1].
(3) Slightly or highly positive: right skewed
(4) Kurtosis types: Platykurtic: thin-tailed, Leptokurtic: fat-tailed, Mesokurtic (normal tail shape)

5.4.4 Detailed mathematical assessment of *MCC* and *CK*

This subsection is devoted to a further separate assessment of *CK* and *MCC* metrics that are the top-ranked metrics equally in both Stage-1 and Stage-2. The following arranged equations are introduced to reveal the subtle difference between them. As seen in the equations in Table B.2 in Appendix B, both *CK* and *MCC* have a determinant of base measures as a matrix in nominators. Rearranging the denominators, *CK* and *MCC* are inversely proportional to arithmetic mean ($\text{Arith}_{\text{mean}}$) and geometric mean ($\text{Geo}_{\text{mean}}$) of the same coefficients, respectively:

$$CK = \frac{DET}{\text{Arith}_{\text{mean}}(P \cdot ON, N \cdot OP)} \tag{5.1}$$

$$MCC = \frac{DET}{\text{Geo}_{\text{mean}}(P \cdot ON, N \cdot OP)} \tag{5.2}$$

As the nominators are the same, the only difference is the mean expressions in the denominators where $x = P \cdot ON$ and $y = N \cdot OP$ are multiplication of two performance dimensions (*i.e.* column and row geometries or reality and prediction) for opposite classes (i.e., cross-geometry margins in cross-class) as shown in the following equations.

$$x \ = P \cdot ON \text{ and } y = N \cdot OP \text{ and } \textit{class}=\{\text{positive, negative}\} \tag{5.3}$$

$$x \text{ and } y\text{: 1st\_level\_col\_measure}_{class} \cdot \text{1st\_level\_row\_measure}_{opposite\_class} \tag{5.4}$$

$$x \text{ and } y\text{: } reality_{class} \cdot prediction_{opposite\_class} \tag{5.5}$$

Hence, the mathematical assessment of *MCC* and *CK* comes down to a comparison of arithmetic mean with geometric mean.

$$CK \propto \text{Arith}_{\text{mean}}(x, y) \ \ ? \ \text{Geo}_{\text{mean}}(x, y) \propto MCC \tag{5.6}$$

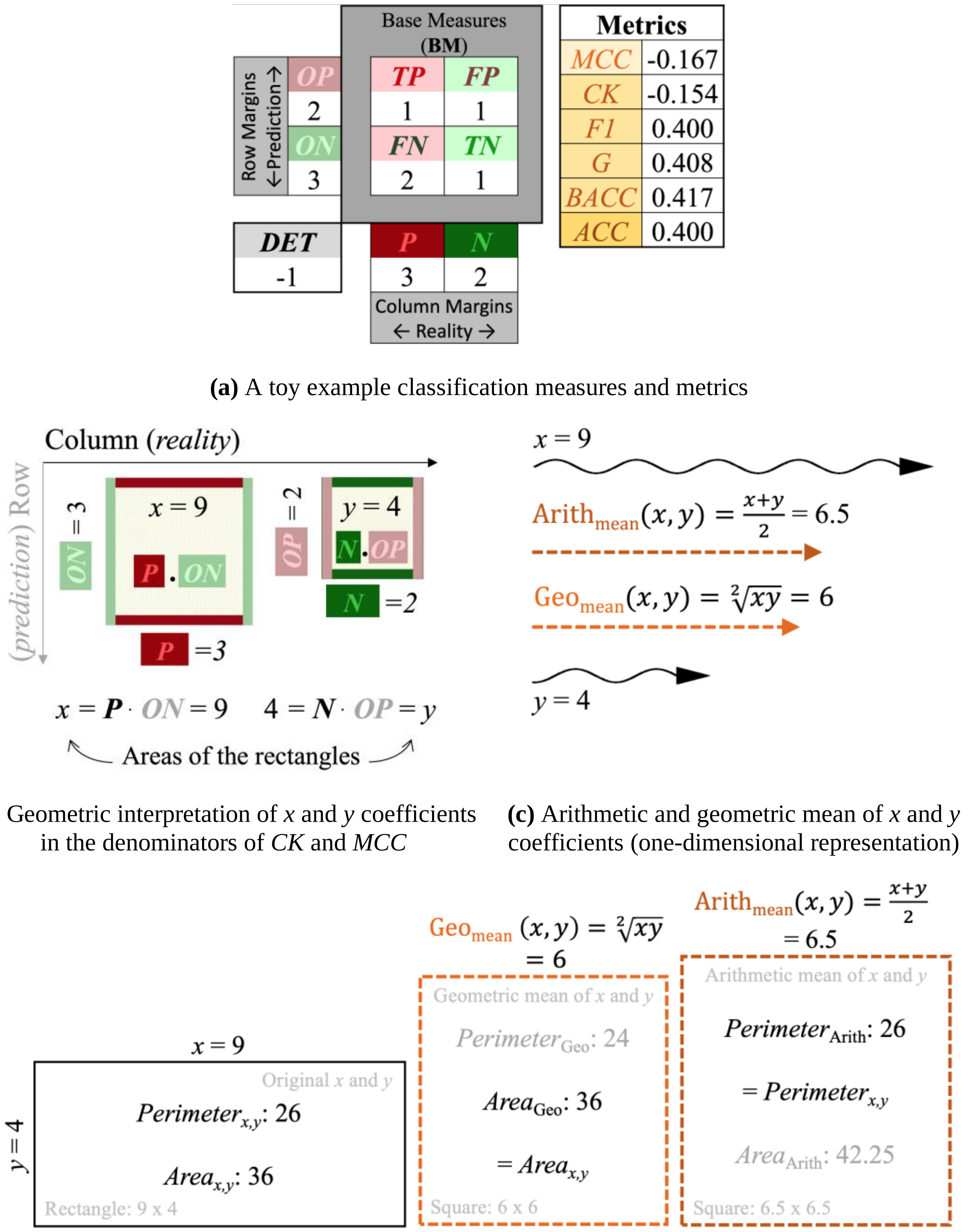
First of all, as two of the Pythagorean means, arithmetic means are always greater or equal to the geometric means for the same pair of values. Thus:

Remark. CK is always less than or equal to *MCC* though this does not imply any superiority.

$$\text{Arith}_{\text{mean}} \geq \text{Geo}_{\text{mean}} \geq \text{Harmonic}_{\text{mean}} \Rightarrow CK \leq MCC \tag{5.7}$$

Findings based on a toy example:

Figure 5.3 depicts the interpretation of the two types of means corresponding to *CK* and *MCC* metrics based on an example classification as shown in Figure 5.3 (a). The interpretation is conducted by using geometric modeling for the family of means (Maor, 1977). Note that the geometric elements are scaled to sense the given values, lengths and areas.



(a) A toy example classification measures and metrics

(b) Geometric interpretation of *x* and *y* coefficients in the denominators of *CK* and *MCC*

(c) Arithmetic and geometric mean of *x* and *y* coefficients (one-dimensional representation)

(d) Arithmetic and geometric mean of *x* and *y* coefficients (two-dimensional representation)

Figure 5.3 Geometric interpretation of arithmetic mean in *CK* and geometric mean in *MCC* via a toy example binary classification

The two factors (x and y) are multiplication of two geometric dimensions as given in Equations (5.3)–(5.5). The multiplication of two Cartesian dimensions refers to area in geometry as depicted in Figure 5.3 (b).

Representation of x and y in one-dimensional is depicted in Figure 5.3 (c) that shows original x and y values (represented as wavy lines indicating they are the multiplication of the two different dimensions) along with their arithmetic and geometric means. Those two means (6.5 and 6) are very close to each other and we could not tell which one has a better representation of original factors.

Figure 5.3 (d) shows the factors and means in a two-dimensional plane. Comparing the original x, y values forming a rectangle and their respected means forming a square:

- Area of the original x and y rectangle is the same as the area for geometric mean ($x \cdot y = 9 \cdot 4 = 36 = \text{Geo}_{mean}(x,y)^2 = 6^2$) and

- Perimeters are the same for arithmetic mean ($2(x+y) = 2(9+4) = 26 = 4 \cdot \text{Arith}_{mean}(x,y) = 4 \cdot 6.5$).

We could not judge based on these findings but when we look into unequal two measurements:

- the perimeters are closer to the original perimeter for geometric mean (the difference is $|24 - 26| = 2$

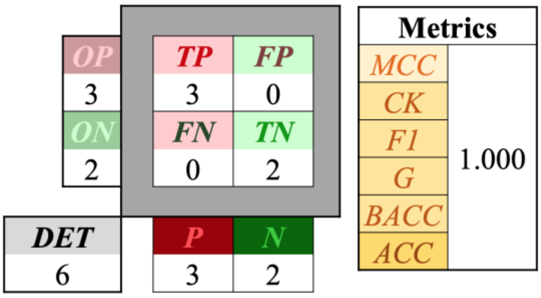
- than the areas to the original area for arithmetic mean (the difference is $\sqrt[2]{|42.25 - 36|} = \sqrt[2]{|6.25|} = 2.5$ by transforming from area back to perimeter in one dimensional).

Remark. Although it is based on a single example, this finding gives an idea that geometric mean is more representative.

Figure 5.4 shows another interesting finding based on the same example where the best classification is achieved (*i.e.* no false classifications) in the same dataset (***P*** = 3, ***N*** = 2, ***FP*** = ***FN*** = 0, ***OP*** = 3, and ***ON*** = 2). Let x' and y' denotes this second case, where $x' = P \cdot ON = 6$ and $y' = N \cdot OP = 6$. In this case, both arithmetic and geometric means of x' and y' are equal to 6 ($\text{Arith}_{mean}(x',y') = \text{Geo}_{mean}(x',y') = 6$), which is not equal to the arithmetic mean in the first case (6.5) but equal to the geometric mean in the first case.

Moreover, when we swap ***OP*** and ***ON*** in two factors**,**

- the geometric mean of $x'' = P \cdot OP = 3 \cdot 3 = 9$ and $y'' = N \cdot ON = 2 \cdot 2 = 4$ is also equal to 6 that is also less than (*i.e.* reducing outlier effect of) the corresponding arithmetic mean 6.5.



$$x' = P \cdot ON = 6 \quad y' = N \cdot OP = 6$$

$$\text{Geo}_{\text{mean}}(x', y') = \text{Arith}_{\text{mean}}(x', y') = 6$$
$$= \text{Geo}_{\text{mean}}(x, y) = 6 \neq \text{Arith}_{\text{mean}}(x, y) = 6.5$$

Figure 5.4 Comparison of the means in the best performance in the same dataset.

Findings based on the literature review:

From a statistical perspective, the literature has strong arguments in favor of geometric mean:

- Galton (1889, pp. 239–240), for example, has a decisive formulation stated as “the true mean is geometric rather than arithmetic” and “it (arithmetic mean) may lead to absurdity when applied to wide deviations”.

- Frank (2009, p. 31) agrees that “geometric mean often captures most of the information about a process or a set of data with respect to underlying distribution”.

- Compared to the arithmetic mean, a geometric mean is less sensitive to outliers’ disruptive effects and it is independent of different ranges of inputs (McAlister, 1879, p. 369).

- Though not justified, Colignatus (2007, p. 6) claims geometric means are more robust due to arbitrary influences among the values in contingency tables.

- Geometric mean is more appropriate for getting the most probable value where the data is inter-related (Matuszak, 2010). Here *x* and *y* factors are inter-related (*e.g.*, $P \cdot ON$ is related to $N \cdot OP$ that is $(Sn - P) \cdot (Sn - ON)$).

Conjecture. *Aggregating all the findings above, it is concluded that MCC is mathematically more robust than CK.*

The next stage becomes notable in whether it supports the concluded finding between *MCC* and *CK*.

5.5 BenchMetric Stage-3: Meta-metrics Benchmarking

Stage-3 measures the robustness of performance metrics via a proposed concept called meta-metrics (*i.e.* metrics about (performance) metrics as defined in Definition 5.2 above). The meta-metrics that are also in [0, 1] range are calculated in metric-spaces. In the experiments, each meta-metric is obtained for the reviewed performance metrics such as accuracy or *MCC* in the metric-spaces of different ***Sn*** sample sizes. It is observed that some meta-metric values are equal regardless of the sample size or they converge consistently as ***Sn*** increases. For the latter case, the intermediate meta-metric values for a number of ***Sn*** values are calculated and their averages are defined as the final meta-metric value. Figure 5.5 depicts the six of the seven proposed meta-metrics calculated for some example metrics in 10 sample size.

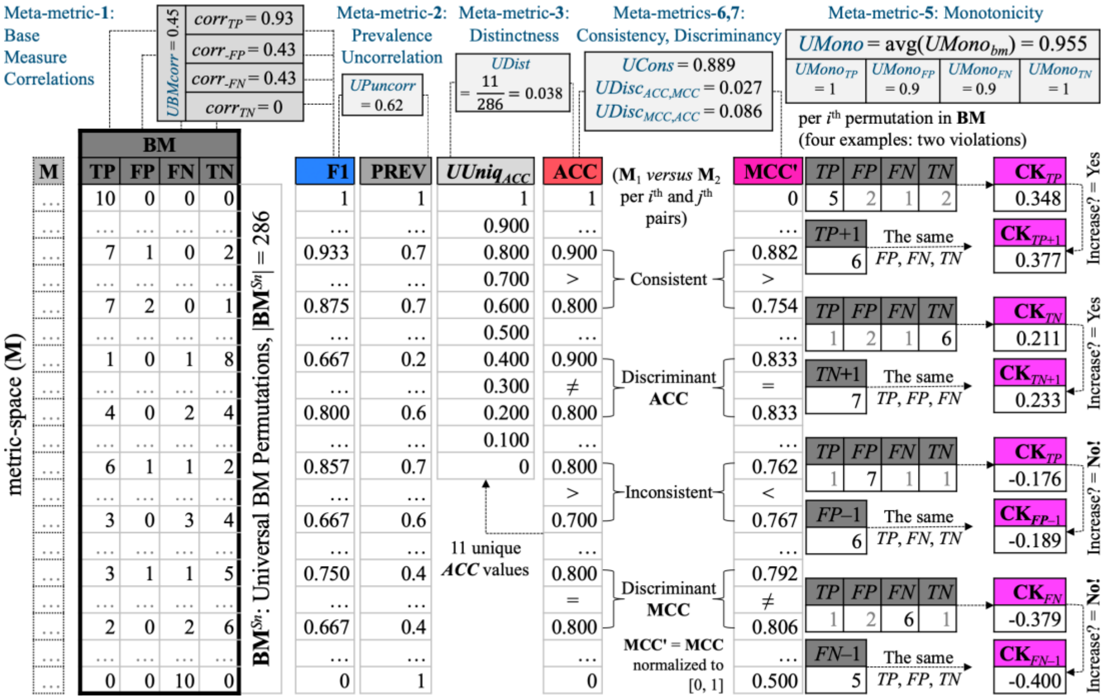


Figure 5.5 Depiction of six of seven meta-metrics for 286 base measure permutations (sample size 10): 1) *UBMcorr* for *F1* metric; 2) *UPuncorr* for *F1*; 3) *UDist* for *ACC*; 4) *UMono* for *CK*; and 5-6) *UCons* and *UDisc* for *ACC* versus *MCC'* (*MCC* normalized into [0, 1] range). Refer to Section 5.5.4 and Figure 4 for *UOsmo* meta-metric.

The following subsections describe and give formal definitions of each meta-metric.

5.5.1 Meta-metric-1: Base measure correlations (*UBMcorr*)

The correlation between a metric-space and each base measure gives their degree of relationship. Robust metrics should equally be correlated with all base performance measures from an objective perspective unless otherwise required. The correlations with **FP** and **FN** must be negative for a performance metric (*i.e.* false classifications should decrease the performance).

Figure 5.5 shows **F1** metric-space with corresponding **BM** permutations as an example. The correlations with **TP**, −**FP**, −**FN**, and **TN** along with the final *UBMcorr* meta-metric value are also displayed. Table 5.3 lists the Spearman's rho correlation values for all benchmarked metrics. Recall that underlined bold texts depict the deficiencies. Spearman correlation is used because it is less sensitive to outliers comparing with Pearson correlation that assumes linearity among the metric and base measures (or prevalence for *UPuncorr* meta-metric described below)[21].

UBMcorr meta-metric reveals that **F1** has zero correlation with **TN** values whereas it is highly correlated with **TP** but lower correlated with false positives/negatives than true positives. **CK** is lower correlated with true positives/negatives (*i.e.* more emphasis on performance errors than successes) compared to the others. ***G*** is class-balanced (*i.e.* correlations for **TP** *vs.* **TN** and −**FP** *vs.* −**FN** are the same) but it is lower correlated with negative false positives/negatives than true positives/negatives (0.49 < 0.54). **ACC**, **INFORM**, **MARK**, **BACC**, and **MCC** are ideally all balanced (*i.e.* absolute correlations for **TP** *vs.* −**FP** *vs.* **TN** *vs.* −**FN** are the same). **nMI** has the lowest correlations with base measures. Note that meta-metric *UBMcorr* for a metric-space is calculated as follows where corr$_{bm}$(**M**) depicts the spearman correlation between the metric-space and *bm* (base measures):

$$UBMcorr = \frac{\sum_{bm=\boldsymbol{TP},-\boldsymbol{FP},-\boldsymbol{FN},\boldsymbol{TN}} corr_{bm}(\mathbf{M})}{4} \tag{5.8}$$

Table 5.3 Meta-metric *UBMcorr* values [0, 1] and correlations with **TP**, −**FP**, −**FN**, **TN** (significance level, α = 0.05)

		ACC	**MCC**	**INFORM**	**MARK**	**BACC**	**CK**	**G**	**F1**	**TPR**	**PPV**	**TNR**	**NPV**	**nMI**
Correlations	**TP**	0.55	0.55	0.54	0.54	0.54	0.53	0.54	0.93	0.78	0.78	**0**	**0**	**-0.05**
	TN	0.55	0.55	0.54	0.54	0.54	0.53	0.54	**0**	**0**	**0**	0.78	0.78	**-0.05**
	−FP	0.55	0.55	0.54	0.54	0.54	0.55	0.49	0.43	**0**	0.78	0.78	**0**	0.05
	−FN	0.55	0.55	0.54	0.54	0.54	0.55	0.49	0.43	0.78	**0**	**0**	0.78	0.05
UBMcorr		0.55	0.55	0.54	0.54	0.54	0.54	0.52	0.45	0.39	0.39	0.39	0.39	0.00

5.5.2 Meta-metric-2: Prevalence uncorrelation (*UPuncorr*)

Robust metrics should not be influenced by class imbalance as addressed in the literature. The correlation between metric-space and **PREV** shows the degree of bias between classification performance and class imbalances. Figure 5.5 shows **F1** metric-space and corresponding **PREV** values with respect to **BM** permutations as an example. As can be seen in Table 5.4, only **PPV**, **NPV**, and **F1** are correlated with **PREV** regardless of the sample sizes. Note that meta-metric *UPuncorr* is calculated by $UPuncorr = 1\text{-}|corr_{\boldsymbol{PREV}}(\mathbf{M})|$ for a metric-space.

[21] The nonlinearity is confirmed by diagnosing the residuals of linear regression assumptions.

Table 5.4 Meta-metric *UPuncorr* values [0, 1] and correlations with **PREV** (significance level, α = 0.05)

	TPR	TNR	ACC	INFORM	MARK	BACC	G	nMI	CK	MCC	F1	PPV	NPV
PREV	0	0	0	0	0	0	0	0	0	0	**0.38**	**0.64**	**-0.64**
UPuncorr	1	1	1	1	1	1	1	1	1	1	0.62	0.36	0.36

Figure 5.6 is provided for presenting correlation values among metrics as well as *PREV* and *BIAS* measures.

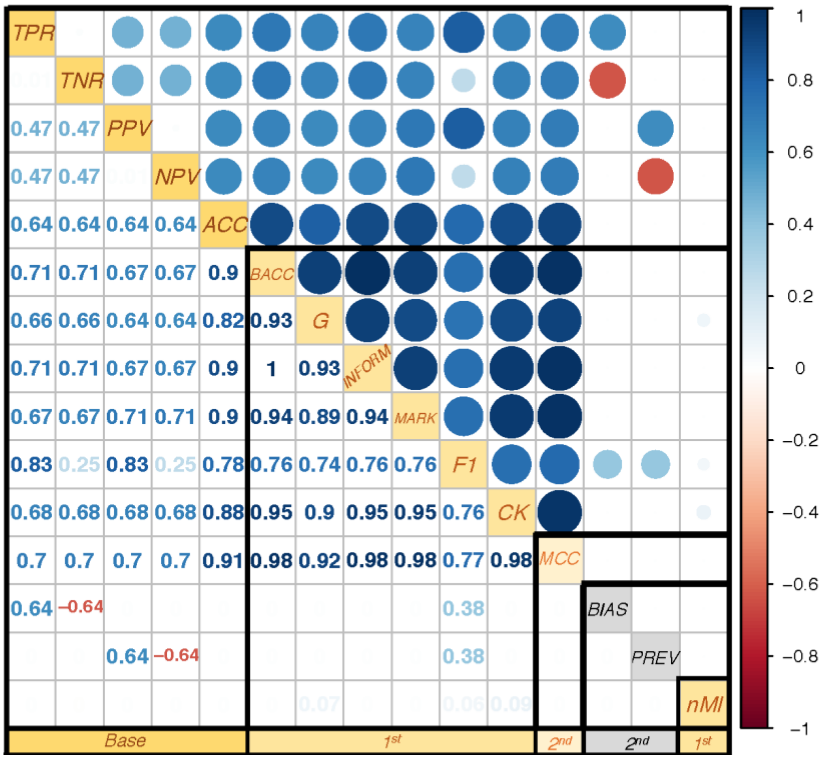


Figure 5.6 Correlations among metrics and *PREV*/*BIAS* measures

5.5.3 Meta-metric-3: Distinctness (*UDist*)

As each base measure permutation is different from each other, a robust metric should differentiate these different cases in metric-space. Figure 5.5 depicts how *UDist* is calculated for *ACC* metric as an example. The number of unique values of the metric-space (*e.g.*, 11 unique values for **ACC**) is compared against the size of the metric-space (*e.g.*, 286 for ***Sn*** = 10), which is the number of unique values in **BM** permutations. The distinctness meta-metric defined formally below gives the granularity of the metrics in metric-space as listed in Table 5.5.

Definition 5.3 (*Universal Distinctness*).

UDist measures the ratio of unique values in the metric-space of a metric M where $\mathbf{M}: \mathbf{BM}^{Sn} \to \overline{\mathbb{R}}$ and $\mathbf{UUniq}$ is a finite set where $\mathbf{M}: \mathbf{UUniq} \to \overline{\mathbb{R}}_{\geq 1}$ and $UDist = |\mathbf{UUniq}|/|\mathbf{BM}^{Sn}|$.

Table 5.5 Meta-metric *UDist* minimum, average, and maximum values [0, 1]. The metrics are sorted according to average *UDist* values

UDist	**nMI**	**BACC**	**INFORM**	**MARK**	**MCC**	**CK**	**G**	**TPR**	**TNR**	**PPV**	**NPV**	**F1**	**ACC**
Min	0.32	0.30	0.30	0.30	0.23	0.17	0.18	0.007	0.007	0.007	0.007	0.007	0.0001
Average	0.38	0.35	0.35	0.35	0.24	0.20	0.20	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.001**
Max	0.40	0.40	0.40	0.40	0.24	0.24	0.20	0.06	0.06	0.06	0.06	0.06	0.008

Sample Size (Permutations):
Sn = 25 (3,276); ***Sn*** = 50 (23,426); ***Sn*** = 75 (76,076); ***Sn*** = 100 (176,851); ***Sn*** = 125 (341,376);
Sn = 150 (585,276); ***Sn*** = 175 (924,176); ***Sn*** = 200 (1,373,701); ***Sn*** = 250 (2,667,126)

To the contrary of the first two meta-metrics, *UDist* values might differ per ***Sn***. *UDist* values are calculated for nine sample sizes (given in the footnotes of Table 5.5) and benchmarked the metrics according to their average values. While *nMI* has the most distinct metric-space, *ACC* has the least. Unexpectedly, *F1* has exactly the same level of distinctness as *TPR*, *TNR*, *PPV*, and *NPV* metrics.

5.5.4 Meta-metric-4: Output smoothness (*UOsmo*)

Output smoothness evaluates how a metric uniformly uses its output range. As each variation in corresponding base measures is a unit change, a metric-space should exhibit a smooth transition. Figure 5.7 shows the transition of metric-spaces sorted in ascending order.

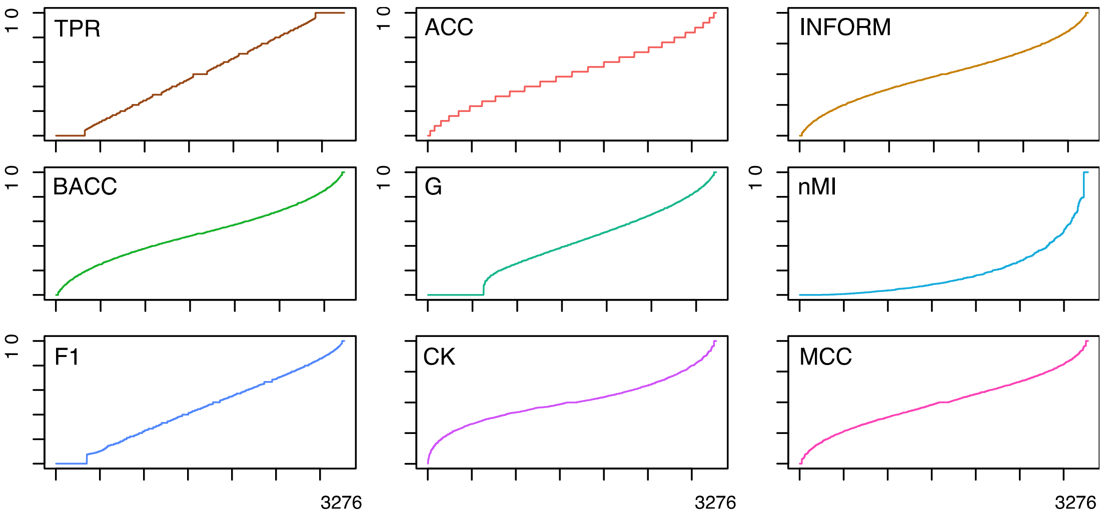


Figure 5.7 Transitions of the metric-spaces sorted. The transitions **MARK** with **INFORM** and **TNR**, **PPV**, **NPV**, and **TPR** are the same. Y-axis shows the metric's outputs and X-axis shows the sequence number of the elements in the metric-space (total 3,276 for ***Sn*** = 25).

Unexpectedly, a repeating stepped transition occurs in **ACC**. As mentioned in the shape of distributions criteria in Stage-3, **G** and **F1** dominantly yield zero. Stepped transitions indicate a robustness defect where a metric yields in coarse resolution in steps or accumulates in some values. These behaviors degrade a metric's ability to differentiate different classification results (*e.g.*, the performance of two classifiers are more likely to fall into the same value than if a smoother metric is used).

The following equation is used to measure the smoothness without visual inspection:

$$osmo = \frac{\mathrm{SD}(\mathbf{Ms}_k - \mathbf{Ms}_{k-1})}{\mathrm{Arith}_{\mathrm{mean}}(|\mathbf{Ms}_k - \mathbf{Ms}_{k-1}|)} \tag{5.9}$$

Ms denotes the sorted metric-space in increasing order, **Ms**$_k$ denotes the k_{th} value of the sorted metric-space and SD is the standard deviation function. The equation calculates the coefficient of variation for one lagged self-difference. The minimum the result, the maximum the smoothness is.

The smoothness values calculated for the sample sizes between 25 and 250 as listed in the footnote of Table 5.6 are averaged and Eq. (5.10) is used to get the *UOsmo* meta-metric for i_{th} metric by normalizing the smoothness values (*osmo*) among *n* compared metric-spaces (*e.g.*, 13 metrics).

$$UOsmo_i = \frac{\mathrm{Arith}_{\mathrm{mean}}\left(osmo_i^{Sn=25..250}\right) - \min\left[\left(\mathrm{Arith}_{\mathrm{mean}}\left(osmo_j^{Sn=25..250}\right)\right)_{j=1..n}\right]}{\max\left[\left(\mathrm{Arith}_{\mathrm{mean}}\left(osmo_j^{Sn=25..250}\right)\right)_{j=1..n}\right] - \min\left[\left(\mathrm{Arith}_{\mathrm{mean}}\left(osmo_j^{Sn=25..250}\right)\right)_{j=1..n}\right]} \tag{5.10}$$

Table 5.6 shows the smoothness and *UOsmo* meta-metric values for the compared metrics. In accordance with Figure 5.2, *ACC* and *nMI* have the least smooth metric-spaces whereas *CK* and *MCC* have slightly unsmooth metric-spaces compared to *INFORM*, *MARK*, and *BACC*.

Table 5.6 Meta-metric *UOsmo* values [0, 1] along with the minimum, average, and maximum smoothness values per base measure

		INFORM	**MARK**	**BACC**	**CK**	**MCC**	**G**	**TPR**	**TNR**	**PPV**	**NPV**	**F1**	**nMI**	**ACC**
Min		2.07	2.07	2.07	2.92	3.02	3.79	3.39	3.39	3.39	3.39	4.02	6.94	5.25
Avg.	*osmo**	4.73	4.73	4.73	8.08	8.46	11.67	15.61	15.61	15.61	15.61	18.03	45.44	91.71
Max		9.79	9.79	9.79	16.74	18.94	27.07	41.73	41.73	41.73	41.73	47.70	135.93	409.47
	UOsmo	1	1	1	0.96	0.96	0.92	0.87	0.87	0.87	0.87	0.85	0.53	0

* Smoothness. Minimum, average, and maximum smoothness are calculated for ***Sn*** = 25, 50, 75, 100, 125, 150, 175, 200, and 250

5.5.5 Meta-metric-5: Monotonicity (*UMono*)

A robust metric should also be sensitive to small changes in classification performance. *UMono* meta-metric is calculated per four base measures by increasing ***TP*** and ***TN*** by one and decreasing ***FP*** and ***FN*** by one separately for all **BM** permutations and checking whether the new metric value does not decrease. Otherwise, this is a bare violation in a metric-space. The formal definition is given in Definition 5.4. The analysis reveals that all the reviewed metrics have 100% monotonicity except for **INFORM**, **MARK**, **BACC**, **nMI**, and **CK** as listed in Table 5.7.

Definition 5.4 (*Universal Monotonicity*).

$UMono_{\boldsymbol{bm}}$ gives the ratio of cases where a metric-space $\mathbf{M}$ adjusts its performance value congruous with the unit changes (±1) by $bm \in \{\boldsymbol{TP}, \boldsymbol{TN}, \boldsymbol{FP}, \boldsymbol{FN}\}$ in metric-space. For all $\mathbf{M}_i$: $\mathbf{BM}^{Sn} \rightarrow \overline{\mathbb{R}}$ and $\mathbf{M}_{i\pm}$: $\mathbf{BM}^{Sn\pm1} \rightarrow \overline{\mathbb{R}}$:

$$\mathbf{M}_{i+}\colon \mathbf{BM}^{\boldsymbol{Sn+1}} = \begin{cases} \{TP_i + 1,\ FP_i,\ FN_i,\ TN_i\}, & bm = TP \\ \{TP_i,\ FP_i,\ FN_i,\ TN_i + 1\}, & bm = TN \end{cases}$$

$$\mathbf{M}_{i-}\colon \mathbf{BM}^{\boldsymbol{Sn-1}} = \begin{cases} \{TP_i,\ FP_i - 1,\ FN_i,\ TN_i\}, & bm = FP \\ \{TP_i,\ FP_i,\ FN_i - 1,\ TN_i\}, & bm = FN \end{cases}$$

$$\mathbf{Mono}_{bm} = \{(\mathbf{M}_i, \mathbf{M}_{i\pm})\colon \mathbf{M}_{i\pm} \geq \mathbf{M}_i\}$$

$$UMono_{bm} = |\mathbf{Mono}_{bm}| / |\mathbf{BM}^{Sn}|$$

Table 5.7 Meta-metric *UMono* values [0, 1] per base measure. The metrics are sorted according to *UMono* values (the average of the four meta-metric sub-values: *UMonoTP*, *UMonoTN*, *UMonoFP*, *UMonoFN*)

UMono	TPR	TNR	PPV	NPV	ACC	G	F1	MCC	INFORM	MARK	BACC	CK	nMI
UMonoTP	1	1	1	1	1	1	1	1	0.9990	0.9990	0.9990	1	0.5029
UMonoTN	1	1	1	1	1	1	1	1	1	1	1	1	0.5029
UMonoFP	1	1	1	1	1	1	1	1	1	1	1	0.9005	0.5032
UMonoFN	1	1	1	1	1	1	1	1	1	1	1	0.9005	0.5032
UMono	1	1	1	1	1	1	1	1	0.9995	0.9995	0.9995	0.9502	0.5031

CK −as parallel to *UBMcorr* meta-metric shown in Table 5.3− has 90% monotonicity for ***FP*** and ***FN*** decrements (10% violations) and **BACC** has 99% monotonicity (1% violation) for ***TP*** and ***TN*** increments. For example, *CK* is −0.176 for ***TP*** = 1, ***FP*** = 7, ***FN*** = 1, ***TN*** = 1 as shown in Figure 5.5. Decreasing ***FP*** only by one (***FP*** = 6) should increase the performance, but *CK* yields -0.189 violating monotonicity (*i.e.* −0.189 < −0.176). Increasing ***TP*** only by one (***TP*** = 1+1, ***FP*** = 7, ***FN*** = 1, ***TN*** = 1) yields −0.128 preserving monotonicity (−0.128 > −0.176). **nMI** monotonicity violations are almost exactly half-and-half.

5.5.6 Meta-metric-6 and 7: Inconsistency/Consistency (*UICons*/*UCons*) and Discriminancy (*UDisc*)

These meta-metrics formally defined in Definition 5.5 and Definition 5.6 below are proposed for comparing the robustness of two metrics. Figure 5.5 above depicts the example cases on real metric values of **ACC** and **MCC'** (**MCC** normalized to [0, 1]) where ***Sn*** = 10. Among all possible *ith* and *jth* pairs, the first given example pairs are consistent because *ith* values (*ACC* = 0.900 and *MCC'* = 0.882) are greater than *jth* values (*ACC* = 0.800 and *MCC'* = 0.754) for both metrics.

However, in the third example, the pairs are inconsistent because the *ith* value is greater than *jth* value for *ACC* (0.800 > 0.700) but the *ith* value is less than the *jth* value for *MCC'* (0.762 < 0.767). For discriminancy, *ACC* is discriminant against *MCC'* in the second example, because *ACC* yields different values (0.900 ≠ 0.800) where *MCC'* yields the same value (0.833 = 0.833) for corresponding pairs. Likewise, *MCC'* is discriminant against *ACC* in the fourth example.

Definition 5.5 (*Universal Consistency and Inconsistency*).

$UCons_{M_1,M_2}$ and $UICons_{M_1,M_2}$ give the agreement and disagreement in increments/decrements in metric-space of two metrics M_1 and M_2, respectively, where $\mathbf{M}_1, \mathbf{M}_2 \colon \mathbf{BM}^{Sn} \rightarrow \overline{\mathbb{R}}$. For all different pairs of i^{th} and j^{th} values of $\mathbf{M}_1$ and $\mathbf{M}_2$:

$$\mathbf{ICons}_{M_1,M_2} = \left\{ \begin{array}{c} (\mathbf{M}_{1_i}, \mathbf{M}_{1_j}), (\mathbf{M}_{2_i}, \mathbf{M}_{2_j}) \colon \\ \left(\left(\left(\mathbf{M}_{1_i} > \mathbf{M}_{1_j} \right) \wedge \left(\mathbf{M}_{2_i} < \mathbf{M}_{2_j} \right) \right) \vee \left(\left(\mathbf{M}_{1_i} < \mathbf{M}_{1_j} \right) \wedge \left(\mathbf{M}_{2_i} > \mathbf{M}_{2_j} \right) \right) \right) \end{array} \right\}$$

$$UICons_{M_1,M_2} = \left| \mathbf{ICons}_{\mathrm{M}_1,\mathrm{M}_2} \right| / \binom{|\mathbf{BM}^{Sn}|}{2}$$

$$UCons_{M_1,M_2} = 1 - \mathbf{UICons}_{M_1,M_2}$$

Definition 5.6 (*Universal Discriminancy*)

$UDisc_{M_1,M_2}$ gives the ratio of cases where the metric M_1 yields different values while the metric M_2 could not differentiate in metric-spaces where $\mathbf{M}_1, \mathbf{M}_2 \colon \mathbf{BM}^{Sn} \rightarrow \overline{\mathbb{R}}$. For all different pairs of i^{th} and j^{th} values of $\mathbf{M}_1$ and $\mathbf{M}_2$:

$$\mathbf{Disc}_{M_1,M_2} = \left\{ \begin{array}{c} \left(\mathbf{M}_{1_i}, \mathbf{M}_{1_j} \right), \left(\mathbf{M}_{2_i}, \mathbf{M}_{2_j} \right) \colon \\ (\mathbf{M}_{1_i} \neq \mathbf{M}_{1_j}) \ \wedge (\mathbf{M}_{2_i} = \mathbf{M}_{2_j}) \end{array} \right\}$$

$$UDisc_{M_1,M_2} = \left| \mathbf{Disc}_{M_1,M_2} \right| / \binom{|\mathbf{BM}^{Sn}|}{2}$$

Note that *UICons*/*UCons* and *UDisc* meta-metrics are based on the two formal criteria proposed by Huang and Ling for comparing two performance metrics (Huang & Ling, 2005). The application of these criteria ("degree of consistency" and "degree of discriminancy") has become one of the most used comparative methods in the literature. The improvement here is transforming the degrees that are ranged differently per compared metrics into a fixed ratio in [0, 1] representing the cases with respect to the universal **BM** permutations. Hence, the proposed meta-metrics can be used for comparing more than two performance metrics as can be seen in Table 5.8 and Table 5.9. Table 5.8 shows the *UCons* values calculated for ***Sn*** = 25 per pairs of the reviewed metrics as well as final *UCons* values. **MCC**, **INFORM** and **BACC** are the most consistent ones with other metrics on average (83%) whereas **nMI** is the least consistent metric (51%). For individual pairs, **INFORM** and **BACC** are only 100% consistent (*i.e.* $\mathit{UCons}_{\mathit{INFORM-BACC}}$ = 1.00).

Table 5.8 *UCons* values per pairs of metrics and final *UCons* meta-metric values (the average of the meta-metric values per performance metric). For example, the cell marked with (1) (the consistency between **ACC** and **MCC**) is 88% ($UCons_{ACC-MCC}$ = 0.88), *UCons* for **MCC** (the average meta-metric values for **MCC**) and **ACC** are the cell marked with (2) (0.83) and the cell marked with (3) (0.80), respectively.

	MCC												
	0.96	**INFORM**											
	0.96	1.00	**BACC**										
	0.96	0.94	0.94	**CK**									
	0.96	0.91	0.91	0.94	**MARK**								
	0.90	0.91	0.91	0.89	0.89	**G**							
	(1)**0.88**	0.88	0.88	0.87	0.88	0.86	**ACC**						
	0.79	0.79	0.79	0.78	0.79	0.81	0.83	**F1**					
	0.76	0.77	0.77	0.75	0.76	0.77	0.76	0.85	**TPR**				
	0.76	0.76	0.76	0.75	0.77	0.76	0.76	0.85	0.69	**PPV**			
	0.76	0.77	0.77	0.75	0.76	0.77	0.76	0.60	0.53	0.69	**TNR**		
	0.76	0.76	0.76	0.75	0.77	0.76	0.76	0.60	0.69	0.53	0.69	**NPV**	
	0.50	0.50	0.50	0.51	0.50	0.54	0.52	0.53	0.52	0.52	0.52	0.52	**nMI**
UCons:	(2)**0.83**	0.83	0.83	0.82	0.82	0.81	(3)**0.80**	0.75	0.72	0.72	0.70	0.70	0.51
Rank:	1	1	1	4	4	6	7	8	9	9	11	12	13

Table 5.9 shows the *UDisc* values per ordered pairs of metrics analyzed in 25 samples. **nMI**, the least consistent metric, is the most discriminant metric (about 1%). Interestingly, **MCC** is both the most consistent and the third discriminant metric at the same time. The table also illustrates another important finding that all the metrics are highly discriminant (about 4%) with **ACC**.

Table 5.9 Meta-metric *UDisc* values [0, 1] per ordered pairs of metrics. The metrics are sorted according to the average of the meta-metric values per metric. The cell marked with (1) shows that the discriminancy of **G** against **F1** is 0.6%, and the cell marked with (2) shows that the discriminancy of **F1** against **G** is 2.8%, shown in bold.

	↳ **nMI**	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
	0.001	**CK**	0.000	0.001	0.001	0.001	0.002	0.001	0.002	0.002	0.002	0.002	0.001
	0.001	0.000	**MCC**	0.001	0.001	0.001	0.002	0.001	0.002	0.002	0.002	0.002	0.001
	0.001	0.001	0.001	**BACC**	0.000	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
	0.001	0.001	0.001	0.000	**INFORM**	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
	0.001	0.001	0.001	0.001	0.001	**MARK**	0.001	0.001	0.001	0.001	0.001	0.001	0.001
	0.018	0.018	0.018	0.017	0.017	0.017	**F1**	0.014	0.018	0.018	0.007	0.007	(1)**0.006**
	0.044	0.043	0.043	0.044	0.044	0.044	0.040	**ACC**	0.043	0.043	0.043	0.043	0.042
	0.029	0.029	0.029	0.029	0.029	0.027	0.028	0.028	**TNR**	0.019	0.029	0.019	0.019
	0.029	0.029	0.029	0.027	0.027	0.029	0.028	0.028	0.019	**NPV**	0.019	0.029	0.018
	0.029	0.029	0.029	0.029	0.029	0.027	0.019	0.028	0.029	0.019	**TPR**	0.019	0.019
	0.029	0.029	0.029	0.027	0.027	0.029	0.019	0.028	0.019	0.029	0.019	**PPV**	0.018
	0.039	0.038	0.038	0.038	0.038	0.034	(2)**0.028**	0.037	0.029	0.028	0.029	0.028	**G** ↰
UDisc:	0.019	0.018	0.018	0.018	0.018	0.018	0.014	0.014	0.014	0.014	0.013	0.013	0.011
Rank:	1	2	2	2	2	2	7	7	7	7	11	11	13

Table 5.10 shows the overall results of the Stage-3 benchmarking along with the metrics' ranks. Stage-3 differentiates the ranks of the benchmarked metrics some of which are equal in the previous stages (*e.g.*, *MCC* and *CK* have the same ranks).

Table 5.10 Stage-3 benchmarking of 13 performance metrics according to seven proposed meta-metrics

Meta-Metrics	**MCC**	**BACC**	**INFORM**	**MARK**	**CK**	**G**	**ACC**	**TNR**	**TPR**	**F1**	**nMI**	**NPV**	**PPV**
UBMcorr	1	3	3	3	3	7	1	9	9	8	13	9	9
UPuncorr	1	1	1	1	1	1	1	1	1	11	1	12	12
UDist	5	2	3	3	6	7	13	8	8	8	1	8	8
UOsmo	4	1	1	1	4	6	13	7	7	11	12	7	7
UMono	1	9	9	9	12	1	1	1	1	1	13	1	1
UCons	1	1	1	4	4	6	7	11	9	8	13	12	9
UDisc	2	2	2	2	2	13	7	7	11	7	1	7	11
Stage-3 Rank	1	2	3	4	5	6	7	8	9	10	10	12	13

According to overall meta-metrics benchmarking, *MCC* is ranked first whereas *PPV* is ranked last.

5.6 Overall BenchMetric Results and Summary of Findings

Table 5.11 summarizes and aggregates the benchmark results from the three stages and gives a finalized ranking of the 13 performance metrics reviewed. The stages defined by extreme cases, criteria, and metric-space were ordered according to complexity, coverage, and measurability. Taking the ranks of each stage equal, the final rankings would be misleading. Therefore, the weights are set as shown in Table 5.11 putting increasing weights through the stages.

Table 5.11 The ranking of three benchmark stages and final ranking results of BenchMetric

Stages	Weight	*MCC*	*CK*	*BACC*	*INFORM*	*MARK*	*G*	*ACC*	*F1*	*TNR*	*TPR*	*NPV*	*PPV*	*nMI*
Stage-1	1	1	1	5	5	5	5	1	4	5	5	5	5	13
Stage-2	2	1	1	4	4	4	4	8	3	9	9	9	9	13
Stage-3	3	1	5	2	3	4	6	7	10	8	9	12	13	10
BenchMetric		1	2	3	4	5	6	7	8	9	10	11	12	13

The followings are the main findings:

- *MCC* is the most robust performance metric.
- *CK* and *BACC* are the second and third most robust metrics.
- *MCC* is also better than *CK* in other aspects, which were not included in benchmarking such as according to the detailed mathematical comparison described in Section 5.4.4.
- Highly recommended and/or conventionally used metrics such as *TPR*, *PPV*, *ACC*, *G*, *F1*, and *nMI* have robustness issues and therefore should be used cautiously if they are used alone.

Some of the notable observations were obtained from the benchmarking:

In Stage-1:

i) The metrics yield not-a-number in some extreme cases except *ACC*, *F1*, *CK*, and *MCC*.

ii) *nMI* yields high values when ***FP*** and ***FN*** are higher than ***TP*** and ***TN***.

In Stage-2:

i) Only **INFORM**, **MARK**, and **BACC** have the same mean, median, and mode values.

ii) The metrics have symmetric metric-space except for *G*, *nMI*, *F1*, and *CK*.

iii) **G** and **F1** metric-spaces exhibit an accumulation at zero.

iv) Only *MCC*, *CK*, *F1*, and *nMI* cover both outcome measures (***OP*** and ***ON***) and class measures (***P*** and ***N***).

v) *TPR*, *PPV*, *TNR*, and *NPV* are single-class-only metrics (*i.e.* *P*-only and *N*-only).

vi) All metrics are insensitive to one or more base measures except *nMI*, *CK*, and *MCC*.

vii) *nMI* and *F1* exhibit some inconsistencies in swapping of base measures.

viii) *nMI* has a highly right-skewed metric-space.

ix) *MCC* with geometric means is mathematically better than *CK* with arithmetic means.

In Stage-3:

i) **ACC**, **INFORM**, **MARK**, **BACC**, and **MCC** have a high correlation with individual base measures whereas the others have either some imbalances or no correlations in some of the measures.

ii) **nMI** does not exhibit any relationship with base measures.

iii) Only *PPV*, *NPV*, and especially *F1* have metric-spaces correlated with prevalence.

iv) *TPR*, *TNR*, *PPV*, *NPV*, *ACC*, and *F1* do not exhibit granular output coverage in metric-spaces.

v) *nMI* and *ACC* do not output smoothly in metric-spaces.

vi) All metrics are monotonic except *INFORM*, *MARK*, and *BACC*. *CK* has minor and *nMI* has considerable monotonicity violations.

vii) *BACC*, *INFORM*, and *MCC* are the most consistent metrics among all the metrics.

viii) *INFORM* and *BACC* are the only metrics that are completely consistent with each other.

ix) *nMI* is the least consistent and most discriminating metric.

x) *G* is the least discriminant metric.

Note that Table G.1 in Appendix G shows the summary of the BenchMetric results per metric per criterion per BenchMetric stage. Table G.2 lists the robustness issues per metric in alphabetic order that could be helpful to be aware of the issues when a metric is used.

5.7 Survey 3: Evaluation of BenchMetric Method with the Literature

The proposed benchmarking method, BenchMetric, is compared with the other methods in the literature in threefold. First, the methodology is compared with the existing metrics evaluation methods. In the second step, the evaluation strategies of the studies, which proposed new metrics, are compared. Independent of the two use-cases, two areas are specifically focused while evaluating the related literature: "Are the approaches mapped onto BenchMetric?" and otherwise, "Did it cover, address or extend these approaches?"

Finally, the recently proposed metrics are directly evaluated with the proposed benchmarking criteria and the benchmarking results are compared with their findings. Hence, we can see whether *MCC* is still the most robust metric when those new metrics are included in the benchmark.

5.7.1 Comparison of BenchMetric with the existing metric evaluation methods

Table 5.12 gives details about the methods designed for metric comparisons in the literature, summarizes their limitations, and compares them with BenchMetric. The compared studies examined a few metrics. Some of them focus on basic behaviors of performance metrics that cannot be seen in practice (*e.g.*, extreme cases such as comparing two classifiers' results with swapped confusion matrix). Others cover only a very limited part of metric-spaces and show similarities from a simple perspective without using an explicit ranking.

Nevertheless, all the proposed comparison techniques are addressed in formal and easy to understand manner with measurable and comparable outputs. In addition, the existing approaches are improved either by extending them or defining them in a classification performance context. Furthermore, additional criteria are proposed and numerous unknown robustness issues are revealed in the metrics.

5.7.2 Comparison of BenchMetric with the methods evaluating recent metrics

Table 5.13 describes the recently proposed performance metrics and how they were compared with respect to the other metrics in the literature. The first three of the proposed metrics are intended to minimize the class imbalance effect of *ACC*. The validation of the new metrics is limited to comparing the new metrics by examining the relations of the input metrics comprising the new metric (*e.g., ACC*, *TPR*, and *TNR* for *OACC*) or by inspecting the input metrics' graphs for different class skews. As can be seen from the table, the validation of the new metrics has always been performed in a limited scope.

Table 5.12 Comparison of BenchMetric with existing metrics evaluation methods

The study / metrics	Conclusion	Comparison Method	Comparison Results and Corresponding Criteria
(Seliya, Khoshgoftaar, & Van Hulse, 2009a). *ACC*, *G*, *F1*, *FPR*, *FNR*, *NPV*, *PPV*, *AUC-ROC*, and *AUC-PR*	The study groups the compared metrics into two to four similar groups rather than comparing and rankings of the metrics.	The performances of two decision tree classifiers applied on 35 real-world datasets with 200 <= ***Sn*** <= 20.000 and 65% < *PREV* < 99% based on different decision thresholds (0 < *t* < 1, default: 0.5) are calculated in terms of the compared 9 metrics. The relations of the metric values are compared for 350 classifier-dataset runs in total: Comparison-1: Via correlations; Comparison-2: Via factor analysis (analyzing correlated metric values (observed variables) in terms of a small number of factors (unobserved variables).	1) Both reviewed studies cover limited cases of prevalence and metric-space. For example, in BenchMetric, there are 2,667,126 base measure permutations for ***Sn*** = 250. Whereas, for example, the 14,400 cases given by Yangguang et al. corresponds to only 0.5% of all possible cases. Thus, correlations and/or factors may not be representative. 2) The comparisons simply show similar metrics that are redundant when they are used together. They do not sufficiently dictate a proper metric and do not reveal any robustness issues. For example, *G* and *F1* are found similar in factor analysis, whereas, in BenchMetric, *G* is slightly more robust in general than *F1*. 3) The comparisons are limited as they are reliant on the performance of two decision tree classifiers. **BenchMetric:** *UBMcorr*, *UPuncorr*, *UMono*, *UCons*/*UDisc*
(Y. Liu, Zhou, Wen, & Tang, 2016) *CK*, *ACC*, and *F1*	The study shows the correlated metrics based on the example datasets.	Performances are calculated for eight ML algorithms on 18 real-world datasets with 80 <= ***Sn*** <= 8.124 and 50% < *PREV* < 94%. The relations of metrics are compared for 14,400 classifier-dataset runs via the Spearman and Pearson correlations of the metrics.	
(Huang & Ling, 2005) *ACC* and *AUC-ROC*	*AUC-ROC* is recommended instead of *ACC*.	The performances of simulated classifiers applied on balanced and imbalanced synthetic datasets and three classifiers applied on 18 real-world datasets (with 61 <= ***Sn*** <= 8.124) are calculated in terms of *ACC* and *AUC-ROC* and each paired metric value are compared for consistency and discriminancy.	4) Assessing the consistency and discriminancy among the metrics that are compared do not impose a superiority especially in paired comparisons. For example, consistency between *CK* and *ACC* is meaningful only if both of the metrics are robust. Likewise, if both or one of the metrics are not robust then the discriminancies could not be interpreted. 5) BenchMetric includes a large number of metrics, thus the conclusions are more meaningful. 6) BenchMetric also indicates that *CK* is better than *ACC*. **BenchMetric:** *UCons*/*UDisc*
(Fatourechi et al., 2008) *CK* and *ACC*	*CK* is recommended instead of *ACC*.	The consistency and discriminancy are compared only within "the desired region of operation" only (*i.e.* where *TPR* >= 0.5 and *FPR* <= 0.02). This is because the calculation of consistency and discriminancy degree as defined in the above study has time and calculation costs.	
(Joshi, 2002) *INFORM*, *ACC*, *G*, and *F1*	*F1* is the recommended metric.	They constructed performance trend graphics for different *TPR*, *PPV*, and *PREV* variations and observed whether the performances increase according to *PREV*.	7) Both techniques require visual inspection and manual interpretation and are not measurable as in BenchMetric. 8) For the former study, BenchMetric shows that *INFORM* is better among the compared four metrics. 9) For the latter study, *MCC* is more robust and in line with BenchMetric whereas *F1* has robustness issues in corresponding criteria. **BenchMetric:** *UPuncorr*, *UCons* (with *TPR* and *PPV*) and *UCons* (with *TPR* and *TNR*)
(Brown, 2018) *MCC*, *BACC*, *ACC*, *F1*, *TNR*, and *PPV*	*MCC* and *F1* exhibit more "realistic" estimation of classification performance.	They constructed performance trends graphics for different *TPR* and *TNR* variations as well as inverse cumulative distribution function plots per balanced and imbalanced datasets.	
(Sokolova, 2006) *BACC*, *ACC*, *F1*, *TNR*, *TPR*, and *PPV*	*TNR* and *BACC* are more appropriate metrics with respect to the variance or invariance of changes in confusion matrix elements.	Checking whether the performance output is varied upon the following changes in confusion matrix: 1) exchange ***TP*** with ***TN*** and ***FN*** with ***FP*** 2) change only in ***TN*** 3) change only in ***FP***, and 4) scale ***TP*** and ***FP*** along with ***TN*** and ***FN***.	10) I reformulate those four changes in order to fit with the classification performance evaluation context and make the assessments more comprehensible. 11) BenchMetric shows that *MCC* and *CK* are the most robust metrics from the corresponding three criteria. But, *TNR* and *BACC* have the same inconsistencies with *TPR*, *PPV*, and *ACC*. **BenchMetric:** 1) Criterion 2.6, 2 & 3) Criterion 2.3, 4) Criterion 2.1

Table 5.13 Comparison of BenchMetric with the methods, which were used to evaluate new metrics

Study, Proposed New Metric, and its Description	Notes and Validation of the New Metric	Corresponding Criteria
(Caruana & Niculescu-Mizil, 2004) (an abbreviation of Squared error, Accuracy, and *ROC* area) $$SAR = \frac{ACC + AUC\text{-}ROC + (1 - RMS)}{3}$$ *SAR* combines Accuracy, Area Under ROC Curve, and Squared Error into one measure.	*AUC-ROC* and *RMS* (root mean square) are different from all the metrics summarizing base measures like *ACC*. *RMS* is for regression problems instead of classification. The proposed metric is validated via correlation analysis as criticized in note 2) in Table 5.12.	**BenchMetric:** *UCons*/*UDisc*
(Ranawana & Palade, 2006) Optimized Precision (*OACC*): $$OACC = ACC - \frac{\lvert TPR - TNR\rvert}{TPR + TNR}$$ *OACC* reduces the sub-optimal performance measurement of *ACC* due to the skewed data sets by adding a heuristic correcting factor that minimizes *TPR* and *TNR* difference while maximizing their totals.	The proposed metric is validated by comparing *ACC* and *OACC* outputs with class balanced and highly-imbalanced theoretical datasets (*SKEW*s are 1:1 and 1:9) along with a single real dataset (human *DNA* sequences). They inspected graphics showing the variance of the metrics with respect to theoretical *TPR* and *TNR* ranges using $ACC = TPR \cdot N + TNR \cdot P$ equations. See note 7) in Table 5.12 for comparison.	**BenchMetric:** *UPuncorr*
(Huang & Ling, 2007) AUC-ROC:*ACC* $$AUC\text{-}ROC{:}ACC = \begin{cases} AUC\text{-}ROC, & AUC\text{-}ROC \text{ pairs are different} \\ ACC, & \text{pairs are the same} \end{cases}$$ *AUC-ROC*:*ACC* is a two-staged measure to enhance metric output differentiation.	The proposed metric is validated by examining the correlations of the new metric with *AUC-ROC* and *ACC* separately then comparing it with best *RMS* values (*AUC-ROC*:*ACC* is highly correlated with *RMS*). See note 4) in Table 5.12 for comparison.	**BenchMetric:** *UCons*/*UDisc*
(Seliya, Khoshgoftaar, & Van Hulse, 2009b) Standardized Relative Performance Metric (*SRPM*) Performances are calculated in terms of different metrics (*ACC*, *G*, *F1*, *NPV*, *PPV*, *AUC-ROC*, and *AUC-PR*) for 12 ML models on 35 real datasets. Factor analysis is applied to the metric values. For the given number of factors, a relative metric value that is calculated with factor scores and normalized proportions of the eigenvalues is standardized into [0, 100] range.	No validation.	**BenchMetric:** *UCons*/*UDisc*
(Garcia, Mollineda, & Sanchez, 2010) Index of Balanced Accuracy: $$IBA_\alpha(G) = \big(1 + \alpha(TPR - TNR)\big)G$$ *IBA* is a parametric metric like *OACC* that adjusts a known metric (here *G*) taking the difference between *TPR* and *TNR* into account.	1. The correlations of new metric with *TPR*, *TNR*, *ACC*, and *G* are evaluated with respect to class imbalance (*ACC* and *G*) and class focuses (*TPR* and *TNR*) 2. Checking the four invariance properties See notes 2) and 10) in Table 5.12 for comparison.	**BenchMetric:** *UCons*/*UDisc* Criterion 2.6, Criterion 2.3, and Criterion 2.1

5.7.3 Experiment 2: Testing recently proposed metrics via BenchMetric

As a limitation, the proposed benchmarking method is not intended for other types of performance metrics (*i.e.* not summarizing confusion matrix) such as *AUC-ROC* and *RMS*. Nevertheless, BenchMetric is re-conducted by including the two recently proposed binary-classification performance metrics, namely *OACC* and *IBA*$_a$(*G*) (shown in the second and fifth metric in Table 5.13 above) to answer the following questions:

i. Does *OACC* improve the robustness of *ACC* as intended?

ii. Does *IBA*$_a$(*G*) improve the robustness of *G* as intended?

iii. Which one is the most robust *OACC* or *IBA*$_a$(*G*)?

iv. Are any of the new metrics more robust than *MCC* as determined by the proposed benchmarking results?

The followings are the results of three benchmark criteria defined in Stage-1:

1) "Does a metric yield not-a-number (NaN, *i.e.* 0/0) in extreme cases?" *ACC* : No, *OACC* : **2 times**, *G* = **2 times**, *IBA*$_a$(*G*) = **2 times**, *MCC* = No
2) "Are the performance metric values of the cases from 5 to 9 decreasing?" Yes for all.
3) "Are the performance metric values symmetric for both classes?" *ACC* : Yes, *OACC* : **Asymmetric**, *G* = Yes, *IBA*$_a$(*G*) = **Asymmetric**, *MCC* = Yes

The followings are the summary of the findings according to the aforementioned questions for Stage-1:

i. *OACC* has no improvement on *ACC*.

ii. *IBA*$_a$(*G*) has no improvement on *G*.

iii. The robustness of *OACC* and *IBA*$_a$(*G*) is identical.

iv. *MCC* is more robust than these two recently proposed metrics.

Table 5.14 lists the details of the Stage-2 benchmarking results like in Table 5.2 for the benchmarking of 13 performance metrics. The various positive or negative robustness issues (underlined bold texts depict negative ones) are revealed. Note that *a* coefficient is taken as 0.05 as suggested by (Garcia et al., 2010).

The followings are the summary of the findings for Stage-2:

i. *OACC* improved *ACC* on outcome/class and class coverages, but robustness issues appeared in undefined metric outputs and mean-median difference. It also distorts symmetry observed in *ACC*.

ii. *IBA*$_a$(*G*) has no improvement on *G*, in fact, it is not invariant in class-and-outcome swaps, which is only seen in *F1* in the benchmarked metrics as seen in Table 5.2.

iii. Evaluating the eight criteria in Stage-2, the robustness of *OACC* and *IBA*$_a$(*G*) is almost identical. Only Criteria 2.6 and 2.8 are different mutually.

iv. *MCC* is more robust than the new metrics.

Table 5.14 Benchmarking Stage-2 results (***Sn*** = 50) for the two new proposed metrics in the literature

Stage-2 Criteria	**ACC**	**OACC**	**G**	**IBAα(G)**	**MCC**
2.1 Outcome/class coverage	**None**	**Class-only**(1)	**Class-only**	**Class-only**(2)	Yes
2.2 Class coverage (***P*** and ***N***)	**None**	Yes(1)	Yes	Yes(2)	Yes
2.3 Base Measure Coverage	***TP, TN***	***TP, TN***	***TP, TN***	***TP, TN***	Yes
2.4 Variant to class swap	Yes	Yes	Yes	Yes	Yes
2.5 Variant to outcome swap	Yes	Yes	Yes	Yes	Yes
2.6 Invariant to class-and-outcome swaps	Yes	Yes	Yes	**No**	Yes
2.7 Undefined (NaN) count	0	**3*Sn*+1**	**2(*Sn*+1)**	**2(*Sn*+1)**	**4*Sn***
2.8 Central tendencies (mean-median difference)	$\mathbf{\bar{M} = \tilde{M} \approx Mo}$	$\mathbf{\bar{M} \neq \tilde{M} \neq Mo}$	$\mathbf{\bar{M} \approx \tilde{M} \neq Mo}$	$\mathbf{\bar{M} \approx \tilde{M} \neq Mo}$	$\mathbf{\bar{M} \approx \tilde{M} = Mo}$
Other Informative Criteria					
2.9 Standard Deviation	0.23	0.23	0.26	0.26	0.21
2.10 Skewness	Symmetric	Slightly negative(3,4)	Slightly positive(5)	Slightly positive(5)	Symmetric
2.11 Kurtosis	Platykurtic(6)	Platykurtic(6)	Platykurtic(6)	Platykurtic(6)	Platykurtic(6)

(1) *OACC* = *f*(***TP***, ***TN***, ***P***, ***N***, ***TC***, ***Sn***), (2) *IBAα*(*G*) = *f*(***TP***, ***TN***, ***P***, ***N***), (3) Left-skewed, (4) Distorting symmetry, (5) Right-skewed, (6) Thin-tailed

Table 5.15 shows the results of Stage-3 benchmark according to the first five meta-metrics. Up arrows depict that a new metric improves the dependent metric (*i.e. IBAα*(*G*) improves *G* or *OACC* improves *ACC*). Down arrows depict a degradation.

Table 5.15 Benchmarking Stage-3 results (***Sn*** = 50) for the two new proposed metrics in the literature (excluding the *UCons* and *UDisc* meta-metrics). Metrics are sorted in descending order per meta-metrics from the most robust one to the least. *Osmo* is the smoothness value.

UBMcorr		*UPuncorr*		*UDist*		*UMono*		*Osmo*	
MCC	**0.78**	**MCC**	**1**	**IBAα(G) ▲**	**0.8**	**MCC**	**1**	**INFORM**	3.22
ACC	**0.78**	**ACC**	**1**	**OACC ▲**	**0.412**	**ACC**	**1**	**MARK**	3.22
INFORM	0.77	**OACC**	1	**nMI**	0.382	**G**	**1**	**BACC**	3.22
MARK	0.77	**G**	1	**BACC**	0.333	**IBAα(G)**	1	**OACC ▲**	**4.91**
BACC	0.77	**IBAα(G)**	1	**INFORM**	0.332	**F1**	1	**MCC**	**5.26**
CK	0.77	**INFORM**	1	**MARK**	0.332	**TPR**	1	**CK**	5.28
G	**0.75**	**MARK**	**1**	**MCC**	**0.232**	**TNR**	1	**IBAα(G) ▲**	**6.44**
IBAα(G)	**0.75**	**BACC**	**1**	**CK**	0.202	**PPV**	1	**G**	**6.98**
OACC ▼	**0.73**	**CK**	**1**	**G**	**0.196**	**NPV**	1	**TPR**	7.82
F1	0.72	**nMI**	1	**TPR**	0.033	**INFORM**	0.998	**TNR**	7.82
TPR	0.69	**TPR**	1	**TNR**	0.033	**MARK**	0.998	**PPV**	7.82
PPV	0.69	**TNR**	1	**PPV**	0.033	**BACC**	0.998	**NPV**	7.82
TNR	0.69	**F1**	0.61	**NPV**	0.033	**CK**	0.948	**F1**	9.15
NPV	0.69	**PPV**	0.37	**F1**	0.033	**OACC ▼**	0.76	**nMI**	19.7
nMI	0.5	**NPV**	0.37	**ACC**	**0.002**	**nMI**	0.517	**ACC**	**21.62**

The following is a summary of the findings for Stage-3:

i. *OACC* improves *ACC* on distinctness and output smoothness but decreases the robustness for base measure correlations and monotonicity in a contradictory manner.

ii. $IBA_a(G)$ has improvement on G by increasing distinctness and output smoothness.

iii. $IBA_a(G)$ is more robust than *OACC* considering the base measure correlations, distinctness, and monotonicity.

iv. *MCC* is more robust than the new metrics as in Stage-2.

Table 5.16 lists the remaining meta-metrics in Stage-3, namely *UCons* and *UDisc*. Instead of giving each pairwise meta-metric values among the metrics as in Table 5.8 and Table 5.9, they are summarized per each recently proposed metric. Bold values depict higher meta-metric summary values. For example, the mean consistency of $IBA_a(G)$ with the 13 benchmarked metrics (0.834) is higher than the mean consistency of *ACC* (0.773).

Table 5.16 Summary of the pairwise *UCons* (consistency) and *UDisc* (discriminancy) meta-metrics per *OACC* and $IBA_a(G)$ with the 13 benchmarked metrics (minimum, mean, standard deviation (SD), and maximum values) for Stage-3 with ***Sn*** = 20

New Metric(s)	Meta-Metrics	Min	Mean (SD)	Max
OACC	$UCons_{OACC,\ M1\text{-}M13}$	0.511	0.773 (0.090)	0.899
	$UDisc_{OACC,\ M1\text{-}M13}$	0.002	**0.022 (0.020)**	**0.052**
	$UDisc_{M1\text{-}M13,\ OACC}$	0	**0.003 (0.001)**	**0.004**
$IBA_a(G)$	$UCons_{IBAaG,\ M1\text{-}M13}$	**0.551**	**0.834 (0.110)**	**0.992**
	$UDisc_{IBAaG,\ M1\text{-}M13}$	0.002	0.014 (0.020)	0.051
	$UDisc_{M1\text{-}M13,\ IBAaG}$	0	0.042 (0.014)	0.053

OACC vs. $IBA_a(G)$	Meta-Metrics	Value
	$UCons_{IBAaG,\ OACC}$	0.898
	$UDisc_{IBAaG,\ OACC}$	0.001
	$UDisc_{OACC,\ IBAaG}$	**0.046**

Notes: Range of all *UCons* is [0.503, 1] and all *UDisc is* [0, 0.055]

Among the paired metric values in metric-space, *OACC* and $IBA_a(G)$ are 89.8% consistent. However, $IBA_a(G)$ is more consistent with the 13 benchmarked metrics on average whereas *OACC* is more discriminant than both benchmarked metrics (2.2%) and $IBA_a(G)$ (4.6%). Briefly, $IBA_a(G)$ is more consistent and *OACC* is more discriminant.

Combining all three stages, $IBA_a(G)$ is more robust than *OACC*. However, neither of them is as robust as *MCC*. This experiment shows that the proposed benchmarking method, BenchMetric, can be used to analyze and compare the robustness of any proposed metrics.

5.8 Precise and Concise Performance Evaluation and Reporting

In this study, thirteen performance metrics along with two recently proposed metrics have been benchmarked via BenchMetric method and the use of *MCC* is recommended for robust performance evaluation for the first time. Using a robust metric is significant to summarize the classification results with fewer errors. Nevertheless, "What should be reported for expressing classification performance?" research question (*see* **RQ4** in Section 1.1) is worth to discuss for the sake of completeness. Specifically, whether the use of a robust metric alone is sufficient to assess a classification approach?

Comparing different or same classifiers on different datasets using solely a metric (even with *MCC*) can be misleading. As revealed in BenchMetric stages, metrics can indicate contradicting, unexpected or undefined performance values in different conditions. Moreover, the literature uses various metrics together to report the classification performances as described in Section 2.3.4.

This section goes beyond the metrics and recommends what should be reported and considered minimally for precise and concise performance evaluation, comparison, and reporting avoiding possible drawbacks. One of the properties of performance metrics is that they are not sensitive to sample size that is reduced in the numerator/denominator of the metrics' equations (*i.e.* it is lost in summary functions of the metrics, *see* Section 3.10). Prevalence might have an implicit effect due to the nature of the functions.

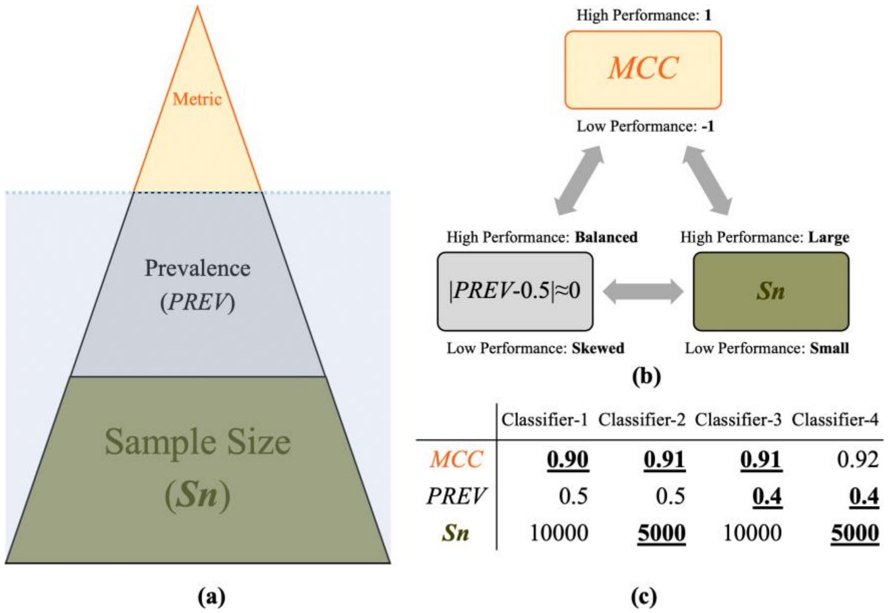
- With respect to sample size, for example, *ACC* = 0.9 for both
  - ***TC*** = 90 in ***Sn*** = 100 and
  - ***TC*** = 900 in ***Sn*** = 1000.
- With respect to prevalence, for example, *ACC* = 0.9 for both
  - *PREV* = 0.50 in ***Sn*** = 100 where ***TP*** = 45, ***FN*** = 5, ***TN*** = 45, and ***FP*** = 5 and
  - *PREV* = 0.75 in ***Sn*** = 100 where ***TP*** = 70, ***FN*** = 5, ***TN*** = 20, and ***FP*** = 5.

Provided two cases within each example above cannot be differentiated via the performance metric because it is 0.9 for all of the cases. From an intuitive perspective, sample size and size of the binary classes (or prevalence as a ratio) are also significant for classification studies. Generally speaking, some statistics are shown to be influenced by sample size and may not reflect the nature of the data (Calude & Longo, 2017, p. 6). As described in Section 2.2 above (Literature Review), the literature addressed the prevalence (or class imbalance, class skew) effect in some of the performance metrics and BenchMetric also reveals prevalence correlations in some metric-spaces for the first time.

With this holistic respect, this thesis proposes to define three dependent components of classification performance evaluation from top to bottom explicitly:

- A robust performance metric (*MCC*),
- Prevalence (*PREV*), and
- Sample size (***Sn***).

Researchers practically focus on performance metrics, which are at the tip of the iceberg, and usually ignore the other two components as shown in Figure 5.8 (a). This thesis engages the attention of the research community that evaluating performance solely based on a metric misleads. As depicted in Figure 5.8 (c), four classifiers on different and/or the same datasets can be compared according to three components of performance.



	Classifier-1	Classifier-2	Classifier-3	Classifier-4
MCC	**0.90**	**0.91**	**0.91**	0.92
PREV	0.5	0.5	**0.4**	**0.4**
Sn	10000	**5000**	10000	**5000**

Figure 5.8 **(a)** The three components of binary-classification performance evaluation. Performance metrics are at the tip of an iceberg. **(b)** Categorical values of each component for high and low performances **(c)** Hypothetical four classifiers with different component values

For example,

- The performance of classifier Classifier-4 is the best according to *MCC* metric only,

- Nevertheless, Classifier-2 and Classifier-3 perform better than Classifier-4 even *MCC* is slightly less than 0.92 because *PREV* or *Sn* values reflect high performance, respectively.

- Finally, Classifier-1 could actually be considered as the most promising of all, even *MCC* is slightly less. Because both *PREV* and *Sn* values reflect more ideal classification configuration (*i.e.* balanced class ratios and the highest sample size among the alternatives, respectively).

In a similar hypothetical case, assume that

- a study reports of a classifier's performance tested on 10.000 samples with fifty-fifty class ratio and

- another study reports the same classifier tested on 5.000 samples with the same class ratio.

It is reasonable to give more credit to the first study because the test is based on more samples or at least, you could expect the researchers of the second study to repeat their tests on 10.000 samples and report the performance again, on the same datasets if possible.

Considering the given arguments above, publishing sample size and prevalence complements the performance metrics. Especially, when comparing a group of studies, performance improvement expressed in terms of a metric should be justified by taking sample sizes and prevalence values into account. The better approach is to equalize them (*i.e.* testing the classifiers in the same *PREV* and ***Sn*** or at least in the same *PREV* value) and compare the performance metrics.

If the classification studies claiming an improvement in a specific classification problem domain (*e.g.*, mobile malware detection) can equalize the two base components of performance evaluation namely sample size and prevalence, then it is possible to compare those studies in terms of a robust performance metric. In this manner, the classifications with similar performance metric values could also be compared from other aspects (*e.g.*, the quality of the datasets, subsampling strategies, and/or time performances of the classification implementations).

A qualitative dataset assessment could be applied to support the quantitative approach that requires reporting two performance measures and one robust metric. A preliminary work that is out of scope of this thesis was already published to systematically profile datasets based on proposed four techniques with fourteen criteria including the sample and feature space sizes (Gurol Canbek et al., 2018).

Hence, it is seriously affirmed that classification studies should report and take sample size (***Sn***) and prevalence (*PREV*) performance measures into account along with *MCC* metric value in minimal to satisfy objective and responsible research. This should be a formal approach to performance reporting in the literature (e.g., listing the performances of compared classification studies with ***Sn***, *PREV*, and *MCC* values together in a table).

5.9 Conclusion

This chapter was carried out to meet two objectives addressing (**RQ4**):

- First, to examine the behavior of all possible binary-classification performance metrics from a wider perspective in order to clarify what the most robust metric is by revealing the problematic issues.
- The second objective was to recommend a proper performance evaluation, comparison, and reporting approach for classification researches.

To meet the objectives, a new comprehensive benchmarking method called BenchMetric is introduced that can be used for any number of existing or newly proposed performance metrics. Contrary to existing approaches, the proposed method develops new concepts such as metric-space, meta-metrics, base measure permutations, and variance/invariance swapping to analyze the metrics and examines metrics from a wider perspective and reveals the weak and strong issues of individual metrics, metric pairs and/or group of metrics in an objective measurable manner.

BenchMetric was tested on thirteen performance metrics that are commonly used and/or recommended in the literature. To the best of my knowledge, this is the first time that such a larger number of metrics have been reviewed in this scope and one metric is suggested with solid justification. BenchMetric spotted specific cases where a metric can behave unexpectedly (*e.g.*, yielding high-performance values in a higher number of false classifications). Especially frequently used metrics such as *TPR*, *ACC*, *nMI*, *F1*, and *CK* exhibit significant robustness issues. The overall result of the proposed three-staged benchmark recommends that *MCC* (otherwise *CK*) as the best choice for performance evaluation.

Besides, two recently proposed metrics are also tested along with the 13 previously tested metrics by BenchMetric. Although the authors of those metrics claim improvement over the existing metrics, this second BenchMetric experiment showed limited improvements but also introduced many unaddressed robustness issues in the new metrics for the first time in the literature. Incorporating those new metrics, *MCC* is still the most robust one.

Monotonicity (*UMono*) calculated for ***Sn*** = 250 measures a small improvement per each base measure can be reflected by the metric, specifically whether there is any contradiction that causes misleading evaluations. There is the same degree of violations for the same metrics for other sample sizes. It might seem controversial that the violations are examined for two paired cases where the original sample size is increased or decreased by one (increase for ***TP*** and ***TN***) which cannot be observed while comparing the performances within the same sample size (*e.g.*, while trying different ML-models in the same dataset). However, such a condition could happen, at least hypothetically, when comparing two different classification studies with sample sizes by one difference.

Contradictory, it could be argued that the benchmarking highlights subtle issues in some metrics that cannot be seen in practice or in a well-prepared classification study. In my opinion, the issues re-summarized in Section 5.6 cannot be ignored as they may arise in several areas such as online machine learning classifications, decision-making applications including "what if" scenarios, and artificial general intelligence in the future where the classification performance possibilities are diverse.

Considering performance evaluation from a wider perspective, it is also suggested that classification studies shall report sample-space size and prevalence ratio explicitly along with metric value (*i.e. MCC*) together for objective and responsible open research. These three indispensable values should be evaluated together to get a better and entire perception of classification performance.

CHAPTER 6

DISCUSSION AND CONCLUSION

This thesis takes a breadth and depth look at binary-classification performance evaluation and covers the largest number of binary-classification performance evaluation instruments available in the literature including the recently proposed ones. The study that is guided by the following main research question revisits performance evaluation by focusing the important problems and essentially proposes two methods to make a multi-perspective analysis and systematic benchmarking for performance instruments.

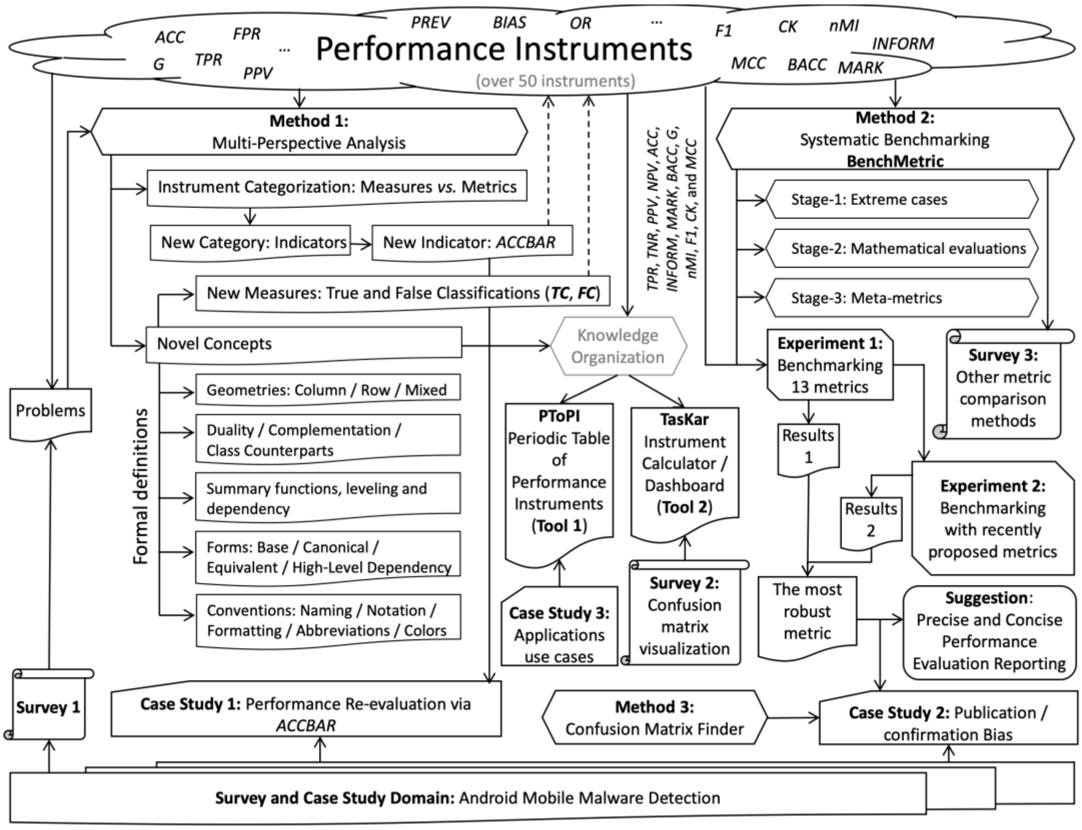
> **RQ**: How to establish and improve our knowledge on binary-classification performance instruments comprehensively and systematically in order to enable researchers to make informed decisions on choosing the right instrument(s) and follow objective approaches in performance evaluation, reporting, and comparison?

As depicted in Figure 6.1, this thesis addressed four research questions expressed in Section 1.1 and, in addition to the two main methods, presented three surveys, three case studies, two complementing tools, and two experiments besides other contributions.

Although performance instruments are widely-accepted global references and contrary to the common assumption that performance evaluation is a well-understood and studied area, this thesis pointed at the fundamental problems such as confusing terminology and lack of consensus in performance evaluation and reporting. Other problems such as misleading results via accuracy metric and publication/confirmation biases were also revealed by conducted case studies. The problems highlighted in Survey 1 and case studies addressing (**RQ1**) reveals previously unknown issues suggesting a root cause that the fundamentals of classification performance evaluation are neither established nor are they internalized by the research community.

Hence, this thesis first provided novel concepts derived from a multi-perspective analysis of performance evaluation instruments addressing (**RQ2**). This conceptualization brings a new perspective for performance evaluation instruments by the following contributions:

- Referring all confusion-matrix derived references as “performance instruments”,
- Terminology clarification with new “measure”–“metric”–“indicator” categorization,
- Naming convention in classification context with standardized abbreviations,
- Grouping and leveling instruments (*e.g.*, base measures: ***TP***, ***FP***, ***FN***, ***TN***, and 1st level measures: ***P***, ***N***, ***OP***, ***ON***), and
- Introducing new measures named ***TC*** and ***FC*** to enhance the comprehensibility of instrument equations.



Performance Instruments
(over 50 instruments)
ACC
FPR
G
TPR
PPV
...
PREV
BIAS
OR
...
F1
CK
nMI
INFORM
MCC
BACC
MARK
Method 1: Multi-Perspective Analysis
Instrument Categorization: Measures vs. Metrics
New Category: Indicators
New Indicator: ACCBAR
New Measures: True and False Classifications (TC, FC)
Novel Concepts
Knowledge Organization
Formal definitions
Geometries: Column / Row / Mixed
Duality / Complementation / Class Counterparts
Summary functions, leveling and dependency
Forms: Base / Canonical / Equivalent / High-Level Dependency
Conventions: Naming / Notation / Formatting / Abbreviations / Colors
Problems
Survey 1
Case Study 1: Performance Re-evaluation via ACCBAR
TPR, TNR, PPV, NPV, ACC, INFORM, MARK, BACC, G, nMI, F1, CK, and MCC
PToPI Periodic Table of Performance Instruments (Tool 1)
TasKar Instrument Calculator / Dashboard (Tool 2)
Case Study 3: Applications use cases
Survey 2: Confusion matrix visualization
Method 2: Systematic Benchmarking BenchMetric
Stage-1: Extreme cases
Stage-2: Mathematical evaluations
Stage-3: Meta-metrics
Experiment 1: Benchmarking 13 metrics
Survey 3: Other metric comparison methods
Results 1
Results 2
Experiment 2: Benchmarking with recently proposed metrics
The most robust metric
Suggestion: Precise and Concise Performance Evaluation Reporting
Method 3: Confusion Matrix Finder
Case Study 2: Publication / confirmation Bias
Survey and Case Study Domain: Android Mobile Malware Detection

Figure 6.1 Thesis contributions summary

This thesis has also introduced and formally defined the following concepts:

- Canonical, base, equivalent, direct and high-level forms in instruments' equations,
- Determination of measure and metric,
- The column, row, and mixed geometries,
- Duality and complementation via transformation in geometry, and
- Levels and dependencies among instruments.

The canonical form is especially helpful to reveal the essential properties of the instruments. Establishing a common language will avoid misunderstanding and facilitate communication among the research community. The concepts help to understand the significant properties of the instruments as well as recognizing the similarities and differences among a large number of instruments.

Concerning (**RQ3**), this thesis made novel contributions to enhance our understanding, facilitate our activities, and provide new approaches in performance evaluation by

- Proposing performance indicators as a new performance evaluation instrument type for the first time and highlighting their potential benefits as well as

- Proposing a novel indicator named “Accuracy Barrier” (*ACCBAR*) to assess whether the performance of a classifier is close to random classification.

ACCBAR indicator was applied in a case study that revealed a significant problem with performance evaluation whereby some of the studies with a high performance reported by *ACC* is misleading whereas the studies with lower *ACC*s had actually appeared to achieve a more reliable performance.

Furthermore, as an aggregation of all the proposed concepts, a new compact binary-classification performance evaluation instruments exploratory table named PToPI, which is like the periodic table of elements, is designed and provided online. PToPI covers over 50 instruments with the following characteristics:

- Clear measure–metric–indicator distinction via grouping and coloring,
- Leveling perception via nested groups,
- Showing the equations for all the instruments in canonical and/or high-level-dependency forms in one place,
- The comprehension of equations is enhanced via positioning according to instrument geometry and graphical decorations (*e.g.*, arrows and font styles),
- Presenting additional information per instruments via a uniform information box,
- A quick sensation of dependent measures/metrics via arrows, and
- Prediction and reality relations via positioning in a column, row, or mixed geometry.

Considering the presence of a large number of binary-classification performance evaluation instruments, it could be difficult to grasp those instruments, their intrinsic characteristics and the differences among them. Addressing these difficulties, the proposed table PToPI provides a big picture for presenting instruments within a single page only, which is also an efficient material for learning or teaching binary-classification performance measures, metrics, and indicators. PToPI can be used to select adequate instruments for performance reporting as demonstrated in this study.

Complementing PToPI, a calculator and dashboard tool called TasKar was also provided to assist the searchers to see the performance in terms of all the instruments as well as interpret the results via the graphical visualization of base metrics. It is expected that PToPI and TasKar will be an efficient material and tool for learning, teaching, and interpreting binary-classification performance measures, metrics, and indicators.

The last part of the thesis, addressing (**RQ4**), after revisiting and reestablishing the classification performance evaluation domain, is to focus on revealing the robustness of binary-classification performance instruments and answering “Which instruments are robust to use” and “What should be reported for classification performance”.

In this perspective, this thesis proposed a new comprehensive benchmarking method called “BenchMetric” to analyze the robustness of performance metrics. Comparing a few methods proposed in the literature, BenchMetric provides a systematic benchmarking comprising three stages and many measurable criteria. The concepts introduced in BenchMetric such as metric-space and meta-metrics (metrics about performance metrics) will enhance the overall understanding of metrics and their behaviors.

The results of the two conducted experiments of BenchMetric (first, on 13 performance metrics and second, on 15 metrics including two recently proposed ones) have shown that

- there are several robustness issues in even commonly used metrics and
- *MCC* is the most robust metric.

This thesis is the first to declare that researchers who want to be on the safe side, can use *MCC* as the most robust metric for general objective purposes. Otherwise, they can select a metric among others that are required or enforced by their domain of interest considering the ranks and specific robustness issues revealed by BenchMetric.

This thesis also demonstrated that publication and confirmation biases might exist because of non-robust metric usages. Some equations were introduced to reveal base measures (confusion matrix) of a classification study that reports a few metrics (e.g., ***P***, ***N***, ***TPR***, and ***ACC***). Hence, it is possible to calculate the performances in terms of other metrics such as *MCC*. It is expected that this method will be used to examine the classification studies in other domains.

Beyond choosing a metric, this thesis suggested that the proper approach is that classification studies should take sample-space size (***Sn***), prevalence ratio (*PREV*), and *MCC* values together into account for a precise and concise binary-classification performance evaluation, comparison, and especially reporting. It is expected that the rankings of the metrics, their robustness issues revealed, and the recommended evaluation approach will guide researchers to evaluate classification performances straightforwardly.

The followings are some remarks of this thesis study to highlight:

- There are severe robustness issues in widely used performance metrics such as *TPR*, *PPV*, *ACC*, and *F1* (see the summary of the findings in Section 5.6).
- Researchers who prefer to use *ACC* should use and consider *ACCBAR* indicator.
- Although *nMI* is recommended by the literature, it is not proper to handle different cases encountered in a classification problem[22].
- As mathematically demonstrated in Section 5.4.4, *CK* and *MCC* are very similar metrics. However, *CK* exhibits non-robust behaviors from certain aspects.
- The recently proposed metrics are not only behind the robust metrics but also they exhibit non-robust behaviors where the metric they try to optimize do not. Therefore, “finding a more robust metric than *MCC*”, as clearly declared in this thesis, might be a challenging research topic where a comprehensive benchmark is available with provided API.
- The visualization of various concepts in a consistent and comprehensible manner was a difficult activity. It is expected that the proposed formatting and coloring scheme will be an industry standard and/or academic visualization convention or utilized for other purposes (*e.g.*, visualizing diagnostic tests in medicine).

[22] Changing the default calculation method in some ML software packages could be a reason behind optimizing *nMI*

- Researchers can use two tools, PToPI and TasKar, complementing each other from theoretical and practical perspectives. It is expected that they will also be used in other domains/scopes, for example, in similarity and association measures conventionally represent base measures as “a”, “b”, “c”, and “d” as shown in TasKar shown in Figure 4.6.

- The analogy between PToPI in machine learning and the periodic table of elements in chemistry is also notable to highlight the significance of the conceptualization proposed in this study. Like the periodic table covering 118 elements, PToPI enhances the usability and comprehensibility of 50 instruments.

- Starting to report performance via a bi-directional robust metric (*MCC* in [-1, +1] range) will provide a wider range for the classification studies in a specific domain where the previous performance reports are saturated at near the maximum value (*i.e.* 1.000) of the non-robust conventional metrics in [0, 1] range especially *ACC* and *F1*.

- It is interesting that *MCC*, which was the top-level metric (the only metric in 2nd level) was also found as the most robust one in BenchMetric. This could be interpreted as an indicator of the consistency in the proposed methods and also a validation with respect to the robustness of *MCC*.

- Notably, including a new metric and repeating the benchmark of the new group of metrics is quite straightforward with the help of systematic methodology and developed ready-to-run API. I experienced this convenience when the benchmarked metrics were extended by adding normalized mutual information (*nMI*) metric that is rather recent and not used much in common.

6.1 Limitations

Although not within direct scope, this thesis also presents a baseline for performance evaluation apart from binary classification such as performance evaluation where binary-classification evaluation metrics are micro- or macro-averaged over time (Kenter et al., 2015) and multi-label or multi-class classification metrics most of which can be

- directly used by applying the canonical form proposed in this thesis. Some binary-classification performance instruments are expressed with the same notation as for multi-class performance evaluation instruments for the first time. For example, *ACC* = ***TC*** / ***Sn*** for binary and multi-class classification performances.

- adapted by using one versus all approaches, all binary-classification instruments can be used directly. For example, the performance of a classifier detecting “apples”, “pears”, and “apricots” in images can be expressed by converting the three classes to binary and calculating the confusion matrix accordingly (*i.e.* using binary-class performance instruments for “apricots” vs. “apples and pears” classes with respect to “apricots” class) (Hossin & Sulaiman, 2015; Kolo, 2011; Pereira et al., 2018).

Because of exponentially increasing number of permutations in metric-spaces (*e.g.*, 3,276 for ***Sn*** = 25 whereas 2,667,126 cases for ***Sn*** = 250) and corresponding limited computational resources, some meta-metrics such as *UDist* and *UOsmo* could be approximated by averaging the intermediate values for a number of sample sizes between 25 and maximum

250. Some optimization methods or high computational resources could be tried to improve the calculation time.

One might argue that examining problematic issues through a single domain cannot be generalized. First of all, Android mobile malware detection is a typical binary classification problem that the literature studies from a broad perspective. The problems could not be limited to this rather recently developed domain. Second, the issues revealed in this domain are also observed in other domains. Some representative examples are also given for term extraction in medical records, computer system intrusion detection in network security, e-mail spam detection in cyber security, and software design defects detection in Section 4.1.3. All these findings and observations suggest that the problems are independent of domains.

Limited feedbacks have been received with respect to PToPI and TasKar. Some of the critics (*e.g.*, from the reviewers of the journal we submit our related works) such as "being a complex tool" were taken into account in some degree. However, the usability of these tools could not be studied during the thesis study.

6.2 Future Work

As mentioned in Section 3.11.1, another future work will evaluate the performance values of other metrics such as *BACC*, *F1*, *CK*, and *MCC* for "under", "hit" and "very close" to Accuracy Barrier cases and compare the differences with *ACC* from a broad perspective. It is expected that this evaluation will give extra insight into metrics and could be integrated into BenchMetric.

For BenchMetric, some significant issues were also observed in metrics that were tested under controlled conditions such as synthetic classifiers and/or datasets. These observations of preliminary work need to be validated to identify whether the assessments could be integrated into BenchMetric as a fourth stage.

We are in the process of defining a single metric to follow the recommendation about precise and concise performance reporting described in Section 5.8. We obtained promising results in categorizing different datasets according to sample size with respect to both sample-space and feature-space size. We are also planning to improve our dataset profiling techniques (Gurol Canbek et al., 2018) to support performance evaluation activities to include assessing the dataset quality.

It is expected that this research will serve a base for future studies on exploring

- Accuracy barrier effect (as demonstrated in case study 1 in Section 3.11.1)
- Presence of publication and confirmation biases (as demonstrated in case study 2 in Appendix H)

in other classification domains in the literature.

An important matter to resolve for future studies is defining an indicator for limitless measures such as Discriminant Power (***DP***).

The following topics remain to be further explored and studied:

- the validity and/or extendibility of proposed concepts and tools in multi-class performance evaluation instruments.
- the effect of using *MCC* in micro- and macro-averaged metrics instead of conventional *TPR*, *PPV*, and *F1*.

It is expected that PToPI and TasKar will be helpful tools to facilitate performance evaluation from different perspectives. Therefore, another area of future work will be enhancing their capabilities and/or making some improvements.

The following capabilities will be developed to improve TasKar in practice:

- Copying all instruments results to the clipboard in a CSV format to paste into a spreadsheet for further analysis and reporting.
- Confusion matrix finder based on the equations given in Appendix H.
- Integrating other binary measures and metrics such as similarity/distance measures and association measures.
- Metric finder to identify a metric with a given value and confusion matrix.
- NaN (i.e. division-by-zero) correction option.

The second version of the proposed coloring scheme described in Section 4.2.1 that is also used in PToPI and TasKar could be optimized for color blindness.

Future work should give priority to develop a technology acceptance model for both PToPI and TasKar that help to assess perceived-usefulness and perceived-ease-of-use (Lai, 2017).

Finally, an interactive visualized performance instrument analysis platform will be released online.

REFERENCES

Aggarwal, C. C. (Ed.). (2015). *Data Classification Algorithms and Applications*. Boca Raton London New York: CRC Press.

Alsallakh, B., Hanbury, A., Hauser, H., Miksch, S., & Rauber, A. (2014). Visual methods for analyzing probabilistic classification data. *IEEE Transactions on Visualization and Computer Graphics*, *20*(12), 1703–1712. https://doi.org/10.1109/TVCG.2014.2346660

Average Number of New Android App Releases Per Day. (2018, April). Retrieved Aug 15th, 2019 from https://www.statista.com/statistics/276703/android-app-releases-worldwide

Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A. F., & Nielsen, H. (2000). Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, *16*(5), 412–424. https://doi.org/10.1093/bioinformatics/16.5.412

Berrar, D., & Flach, P. (2012). Caveats and pitfalls of ROC analysis in clinical microarray research (and how to avoid them). *Briefings in Bioinformatics*, *13*(1), 83–97. https://doi.org/10.1093/bib/bbr008

Bond, R. R., Novotny, T., Andrsova, I., Koc, L., Sisakova, M., Finlay, D., … Malik, M. (2018). Automation bias in medicine: The influence of automated diagnoses on interpreter accuracy and uncertainty when reading electrocardiograms. *Journal of Electrocardiology*, *51*(6), S6–S11. https://doi.org/10.1016/j.jelectrocard.2018.08.007

Brito, A., Rodríguez, M. A., & Niaz, M. (2005). A reconstruction of development of the periodic table based on history and philosophy of science and its implications for general chemistry textbooks. *Journal of Research in Science Teaching*, *42*(1), 84–111. https://doi.org/10.1002/tea.20044

Brown, J. B. (2018). Classifiers and their Metrics Quantified. *Molecular Informatics*, *37*(1), 1–11. https://doi.org/10.1002/minf.201700127

Brzezinski, D., Stefanowski, J., Susmaga, R., & Szczęch, I. (2018). Visual-based analysis of classification measures and their properties for class imbalanced problems. *Information Sciences*, *462*, 242–261. https://doi.org/10.1016/j.ins.2018.06.020

Calude, C. S., & Longo, G. (2017). The Deluge of Spurious Correlations in Big Data. *Foundations of Science*, *22*(3), 595–612. https://doi.org/10.1007/s10699-016-9489-4

Canbek, Gürol. (2018). Cyber Security by a New Analogy: "The Allegory of the 'Mobile' Cave." *Journal of Applied Security Research*, *13*(1), 63–88. https://doi.org/10.1080/19361610.2018.1387838

Canbek, Gürol, Baykal, N., & Sagiroglu, S. (2017). Clustering and visualization of mobile application permissions for end users and malware analysts. In *The 5th International Symposium on Digital Forensic and Security (ISDFS)* (pp. 1–10). Tirgu Mures: IEEE. https://doi.org/10.1109/ISDFS.2017.7916512

Canbek, Gürol, Sagiroglu, S., & Baykal, N. (2016). New Comprehensive Taxonomies on Mobile Security and Malware Analysis. *International Journal of Information Security Science (IJISS), 5*(4), 106–138. Retrieved Aug 15th, 2019 from http://www.ijiss.org/ijiss/index.php/ijiss/article/view/227

Canbek, Gurol, Sagiroglu, S., & Taskaya Temizel, T. (2018). New Techniques in Profiling Big Datasets for Machine Learning with a Concise Review of Android Mobile Malware Datasets. *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT),* 117–121. https://doi.org/10.1109/ibigdelft.2018.8625275

Canbek, Gürol, Sagiroglu, S., Taskaya Temizel, T., & Baykal, N. (2017). Binary classification performance measures/metrics: A comprehensive visualized roadmap to gain new insights. In *2017 International Conference on Computer Science and Engineering (UBMK)* (pp. 821–826). Antalya, Turkey: IEEE. https://doi.org/10.1109/UBMK.2017.8093539

Cardoso, J. R., Pereira, L. M., Iversen, M. D., Ramos, A. L., Cardoso, J. R., Pereira, L. M., … Ramos, A. L. (2014). What is gold standard and what is ground truth? *Dental Press Journal of Orthodontics, 19*(5), 27–30. https://doi.org/10.1590/2176-9451.19.5.027-030.ebo

Caruana, R., & Niculescu-Mizil, A. (2004). Data mining in metric space: an empirical analysis of supervised learning performance criteria. *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 69–78. https://doi.org/1-58113-888-1/04/0008

Colignatus, T. (2007). *Correlation and regression in contingency tables. A measure of association or correlation in nominal data (contingency tables), using determinants* (No. 2662). Retrieved Aug 15th, 2019 from https://mpra.ub.uni-muenchen.de/2662

CrowdFlower. (2016). *2016 Data Science Report*. Retrieved Aug 15th, 2019 from http://visit.crowdflower.com/rs/416-ZBE-142/images/CrowdFlower_DataScienceReport_2016.pdf

CrowdFlower. (2017). *2017 Data Science Report*. Retrieved Aug 15th, 2019 from https://visit.crowdflower.com/rs/416-ZBE-142/images/CrowdFlower_DataScienceReport.pdf

Dogtiev, A. (2018, September). App Stores List 2018. *BusinessOfApps*. Retrieved Aug 15th, 2019 from http://www.businessofapps.com/guide/app-stores-list/#1

Faris, H., Al-Zoubi, A. M., Heidari, A. A., Aljarah, I., Mafarja, M., Hassonah, M. A., & Fujita, H. (2019). An intelligent system for spam detection and identification of the most relevant features based on evolutionary Random Weight Networks. *Information Fusion*, *48*, 67–83. https://doi.org/10.1016/j.inffus.2018.08.002

Fatourechi, M., Ward, R. K., Mason, S. G., Huggins, J., Schlögl, A., & Birch, G. E. (2008). Comparison of evaluation metrics in classification applications with imbalanced datasets. In *7th International Conference on Machine Learning and Applications (ICMLA)* (pp. 777–782). https://doi.org/10.1109/ICMLA.2008.34

Fawcett, T. (2004). *ROC Graphs: Notes and Practical Considerations for Researchers* (No.

MS 1143). the Netherlands.

Ferri, C., Hernández-Orallo, J., & Modroiu, R. (2009). An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, *30*(1), 27–38. https://doi.org/10.1016/j.patrec.2008.08.010

Forbes, A. (1995). Classification-algorithm evaluation: five performance measures based on confusion matrices. *Journal of Clinical Monitoring and Computing*, *11*(3), 189–206. https://doi.org/10.1007/BF01617722

Frank, S. A. (2009). The common patterns of nature. *Journal of Evolutionary Biology*, *22*(8), 1563–1585. https://doi.org/10.1111/j.1420-9101.2009.01775.x

Friendly, M. (1995). *A Fourfold Display for 2 by 2 by k Tables* (No. 217). Toronto, ON. Retrieved Aug 15th, 2019 from http://datavis.ca/papers/4fold/4fold.pdf

Galton, F. (1889). *Natural Inheritance*. London and Newyork: Macmillan.

García-Magariño, I., Chittaro, L., & Plaza, I. (2018). Bodily sensation maps: Exploring a new direction for detecting emotions from user self-reported data. *International Journal of Human Computer Studies*, *113*(January), 32–47. https://doi.org/10.1016/j.ijhcs.2018.01.010

García, F., Bertoa, M. F., Calero, C., Vallecillo, A., Ruíz, F., Piattini, M., & Genero, M. (2006). Towards a consistent terminology for software measurement. *Information and Software Technology*, *48*(8), 631–644. https://doi.org/10.1016/j.infsof.2005.07.001

Garcia, V., Mollineda, R. a., & Sanchez, J. S. (2010). Theoretical Analysis of a Performance Measure for Imbalanced Data. *2006 IEEE International Conference on Pattern Recognition*, 617–620. https://doi.org/10.1109/ICPR.2010.156

Hjørland, B. (2013). Facet analysis: The logical approach to knowledge organization. *Information Processing and Management*, *49*(2), 545–557. https://doi.org/10.1016/j.ipm.2012.10.001

Hjørland, B., Scerri, E., & Dupré, J. (2011). Forum: The Philosophy of Classification. *Knowledge Organization*, *38*(1), 9–24.

Hossin, M., & Sulaiman, M. N. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*, *5*(2), 01–11. https://doi.org/10.5121/ijdkp.2015.5201

Hu, B.-G., & Dong, W.-M. (2014). A study on cost behaviors of binary classification measures in class-imbalanced problems. *Computing Research Repository (CoRR), abs/1403.7*. Retrieved Aug 15th, 2019 from http://arxiv.org/abs/1403.7100

Huang, J., & Ling, C. X. (2005). Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, *17*(3), 299–310. https://doi.org/10.1109/TKDE.2005.50

Huang, J., & Ling, C. X. (2007). Constructing new and better evaluation measures for machine learning. *IJCAI International Joint Conference on Artificial Intelligence*, 859–864.

Huber, S., Wiemer, H., Schneider, D., & Ihlenfeldt, S. (2019). DMME: Data mining methodology for engineering applications – a holistic extension to the CRISP-DM model. *Procedia CIRP*, *79*, 403–408. https://doi.org/10.1016/J.PROCIR.2019.02.106

Japkowicz, N., & Shah, M. (2015). Performance Evaluation in Machine Learning. In I. El Naqa, R. Li, & M. J. Murphy (Eds.), *Machine Learning in Radiation Oncology* (pp. 41–56).

Joshi, M. V. (2002). On evaluating performance of classifiers for rare classes. In *Proceedings IEEE International Conference on Data Mining* (pp. 641–644). IEEE. https://doi.org/10.1109/ICDM.2002.1184018

Kaiafas, G., Varisteas, G., Lagraa, S., State, R., Nguyen, C. D., Ries, T., & Ourdane, M. (2018). Detecting malicious authentication events trustfully. In *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium* (pp. 1–6). Taipei, Taiwan: IEEE. https://doi.org/10.1109/NOMS.2018.8406295

Kearns, M. J. (1990). *The Computational Complexity of Machine Learning*. Cambridge, MA: The MIT Press.

Kenter, T., Balog, K., & De Rijke, M. (2015). Evaluating document filtering systems over time. *Information Processing and Management*, *51*(6), 791–808. https://doi.org/10.1016/j.ipm.2015.03.005

Kocher, M., & Savoy, J. (2017). Distance measures in author profiling. *Information Processing and Management*, *53*(5), 1103–1119. https://doi.org/10.1016/j.ipm.2017.04.004

Kolo, B. (2011). *Binary and Multiclass Classification*. Weatherford Press.

Koyejo, O. O., Natarajan, N., Ravikumar, P. K., & Dhillon, I. S. (2014). Consistent Binary Classification with Generalized Performance Metrics. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada* (pp. 2744–2752). Montreal, Canada: ACM.

Labatut, V., & Cherifi, H. (2011). Evaluation of Performance Measures for Classifiers Comparison. *Ubiquitous Computing and Communication Journal*, *6*, 21–34.

Lai, P. (2017). The Literature Review of Technology Adoption Models and Theories for the Novelty Technology. *Journal of Information Systems and Technology Management*, *14*(1), 21–38. https://doi.org/10.4301/s1807-17752017000100002

Landis, J. R., & Koch, G. G. (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics*, *33*(1), 159–174. https://doi.org/10.2307/2529310

Liu, N., Qi, E. S., Xu, M., Gao, B., & Liu, G. Q. (2019). A novel intelligent classification model for breast cancer diagnosis. *Information Processing and Management*, *56*(3), 609–623. https://doi.org/10.1016/j.ipm.2018.10.014

Liu, Y., Zhou, Y., Wen, S., & Tang, C. (2016). A Strategy on Selecting Performance Metrics for Classifier Evaluation. *International Journal of Mobile Computing and Multimedia Communications*, *6*(4), 20–35. https://doi.org/10.4018/ijmcmc.2014100102

Lomel, H. M. (2012). Punishing the uncommitted crime: Prevention, pre-emption, precaution and the transformation of criminal law. In B. Hudson & S. Ugelvik (Eds.), *Justice and Security in the 21st Century: Risks, Rights and the Rule of Law* (1st ed.). Abingdon, Oxon, United Kingdom: Routledge. https://doi.org/10.4324/9780203125588

Lucini, F. R., S. Fogliatto, F., Giovani, G. J., L. Neyeloff, J., Anzanello, M. J., de S. Kuchenbecker, R., & D. Schaan, B. (2017). Text mining approach to predict hospital admissions using early medical records from the emergency department. *International Journal of Medical Informatics*, *100*, 1–8. https://doi.org/10.1016/j.ijmedinf.2017.01.001

Luque, A., Carrasco, A., Martín, A., & de las Heras, A. (2019). The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, *91*, 216–231. https://doi.org/10.1016/j.patcog.2019.02.023

Maor, E. (1977). A Mathematician’s Repertoire of Means. *The Mathematics Teacher*, *70*(1), 20–25. Retrieved Aug 15th, 2019 from http://www.jstor.org/stable/27960697

Massich, J. (2015). Open discussion in ROC curve, Confusion Matrix and Venn Diagram. Retrieved Aug 15th, 2019 from https://stats.stackexchange.com/questions/167511/open-discussion-in-roc-curve-confusion-matrix-and-venn-diagram

Matlab: plotconfusion. (2018). Retrieved Aug 15th, 2019 from https://mathworks.com/help/deeplearning/ref/plotconfusion.html

Matsuo, R., & Ho, T. B. (2018). Semantic term weighting for clinical texts. *Expert Systems with Applications*, *114*, 543–551. https://doi.org/10.1016/j.eswa.2018.08.028

Matuszak, A. (2010). Differences between Arithmetic, Geometric, and Harmonic Means. Retrieved Aug 15th, 2019 from http://economistatlarge.com/finance/applied-finance/differences-arithmetic-geometric-harmonic-means

Mayya, S. S., Monteiro, A. D., & Ganapathy, S. (2017). Types of biological variables. *Journal of Thoracic Disease*, *9*(6), 1730–1733. https://doi.org/10.21037/jtd.2017.05.75

McAlister, D. (1879). The Law of the Geometric Mean. *Proceedings of the Royal Society of London (1854-1905)*, *29*, 367–376. Retrieved Aug 15th, 2019 from http://www.jstor.org/stable/113784

Michel, J.-B., Shen, Y. K., Aiden, A. P., Veres, A., Gray, M. K., Pickett, J. P., … Aiden, E. L. (2011). Quantitative analysis of culture using millions of digitized books. *Science (New York, N.Y.)*, *331*(6014), 176–182. https://doi.org/10.1126/science.1199644

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill Science/Engineering/Math.

Nicolov, N. (2012). Machine Learning with Applications in Categorization, Popularity and Sequence Labeling. Retrieved Aug 15th, 2019 from https://www.slideshare.net/Nicolas_Nicolov/machine-learning-14528792

Nnamoko, N., Hussain, A., & England, D. (2018). Predicting Diabetes Onset: An Ensemble Supervised Learning Approach. In *2018 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1–7). Rio de Janeiro: IEEE. https://doi.org/10.1109/CEC.2018.8477663

Number of Apps in Leading App Stores. (2018, May). Retrieved Aug 15th, 2019 from https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores

Olsina, L., & de los Angeles Martín, M. (2004). Ontology for Software Metrics and Indicators: Building Process and Decisions Taken. *Journal of Web Engineering*, *2*(4), 262–281.

Paradowski, M. (2015). On the Order Equivalence Relation of Binary Association Measures. *International Journal of Applied Mathematics and Computer Science (AMCS)*, *25*(3), 645–657. https://doi.org/10.1515/amcs-2015-0047

Pereira, R. B., Plastino, A., Zadrozny, B., & Merschmann, L. H. C. (2018). Correlation analysis of performance measures for multi-label classification. *Information Processing and Management*, *54*(3), 359–369. https://doi.org/10.1016/j.ipm.2018.01.002

Porta, M. (Ed.). (2014). *A Dictionary of Epidemiology* (6th ed.). New York: Oxford University Press.

Powers, D. M. W. (2011). Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies*, *2*(1), 37–63.

Powers, D. M. W. (2015). *What the F-measure doesn't measure: Features, Flaws, Fallacies and Fixes* (No. KIT-14-001). https://doi.org/KIT-14-001

Press, W. H. (2008). Classifier performance: ROC, precision-recall, and all that. In *Computational Statistics with Application to Bioinformatics*. The University of Texas at Austin.

Ranawana, R., & Palade, V. (2006). Optimized Precision - A New Measure for Classifier Performance Evaluation. *2006 IEEE International Conference on Evolutionary Computation*, 2254–2261. https://doi.org/10.1109/cec.2006.1688586

Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE*, *10*(3), 1–21. https://doi.org/10.1371/journal.pone.0118432

Sammut, C., & I.Webb, G. (Eds.). (2011). *Encyclopedia of Machine Learning*. New York: Springer.

Schmidt, C. O., & Kohlmann, T. (2008). When to use the odds ratio or the relative risk? *International Journal of Public Health*, *53*(3), 165–167. https://doi.org/10.1007/s00038-008-7068-3

Schröder, G., Thiele, M., & Lehner, W. (2011). Setting Goals and Choosing Metrics for Recommender System Evaluations. In *UCERSTI 2 Workshop at the 5th ACM Conference on Recommender Systems*. Chicago, Illinois. Retrieved Aug 15th, 2019 from http://ceur-ws.org/Vol-811/paper12.pdf

Seliya, N., Khoshgoftaar, T. M., & Van Hulse, J. (2009a). A study on the relationships of classifier performance metrics. In *21st IEEE International Conference on Tools with Artificial Intelligence, ICTAI* (pp. 59–66). https://doi.org/10.1109/ICTAI.2009.25

Seliya, N., Khoshgoftaar, T. M., & Van Hulse, J. (2009b). Aggregating performance metrics for classifier evaluation. In *IEEE International Conference on Information Reuse and Integration, IRI* (pp. 35–40). https://doi.org/10.1109/IRI.2009.5211611

Seung-Seok, C., Sung-Hyuk, C., & Tappert, C. C. (2010). A Survey of Binary Similarity and Distance Measures. *Journal of Systemics, Cybernetics & Informatics*, *8*(1), 43–48.

Shah, S. A. R., & Issac, B. (2018). Performance comparison of intrusion detection systems and application of machine learning to Snort system. *Future Generation Computer Systems*, *80*, 157–170. https://doi.org/10.1016/j.future.2017.10.016

Shepperd, M. (2013). Assessing the Predictive Performance of Machine Learners in Software Defect Prediction Function. In *The 24th CREST Open Workshop (COW), on Machine Learning and Search Based Software Engineering (ML&SBSE)* (pp. 1–16). London: Centre for Research on Evolution, Search and Testing (CREST). Retrieved Aug 15th, 2019 from http://crest.cs.ucl.ac.uk/cow/24/slides/COW24_Shepperd.pdf

Siegerink, B., & Rohmann, J. L. (2018). Impact of your results: Beyond the relative risk. *Research and Practice in Thrombosis and Haemostasis*, *2*(4), 653–657. https://doi.org/10.1002/rth2.12148

Sokolova, M. (2006). Assessing invariance properties of evaluation measures. *Proceedings of the Workshop on Testing of Deployable Learning and Decision Systems, the 19th Neural Information Processing Systems Conference (NIPS 2006)*.

Sokolova, M., Japkowicz, N., & Szpakowicz, S. (2006). Beyond accuracy, F-Score and ROC: A family of discriminant measures for performance evaluation. *Advances in Artificial Intelligence*, *4304*, 1015–1021. https://doi.org/10.1007/11941439_114

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, *45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Straube, S., & Krell, M. M. (2014). How to evaluate an agent's behavior to infrequent events? Reliable performance estimation insensitive to class distribution. *Frontiers in Computational Neuroscience*, *8*(April), 1–6. https://doi.org/10.3389/fncom.2014.00043

Texel, P. P. (2013). Measure, metric, and indicator: An object-oriented approach for consistent terminology. In *Proceedings of IEEE Southeastcon*. Jacksonville, FL: IEEE. https://doi.org/10.1109/SECON.2013.6567438

Tharwat, A. (2018). Classification assessment methods. *Applied Computing and Informatics*. https://doi.org/10.1016/j.aci.2018.08.003

Tulloss, R. E. (1997). Assessment of Similarity Indices for Undesirable Properties and a new Tripartite Similarity Index Based on Cost Functions. In *Mycology in Sustainable Development: Expanding Concepts, Vanishing Borders.* (pp. 122–143). Boone, North Carolina: Parkway Publishers.

Ubayawardana, G. M., & Karunaratna, D. D. (2019). Bug prediction model using code smells. In *18th International Conference on Advances in ICT for Emerging Regions, ICTer 2018* (pp. 70–77). IEEE. https://doi.org/10.1109/ICTER.8615550

Ulysses. (2019). *Machine Learning Techniques for Predicting Errors in Software (in Greek)*. National Technical University of Athens. Retrieved Aug 15th, 2019 from http://artemis.cslab.ece.ntua.gr:8080/jspui/bitstream/123456789/17325/1/diploma_thesis.pdf

Valverde-Albacete, F. J., & Peláez-Moreno, C. (2014). 100% classification accuracy considered harmful: The normalized information transfer factor explains the accuracy paradox. *PLoS ONE*, *9*(1). https://doi.org/10.1371/journal.pone.0084217

van Stralen, K. J., Stel, V. S., Reitsma, J. B., Dekker, F. W., Zoccali, C., & Jager, K. J. (2009). Diagnostic methods I: sensitivity, specificity, and other measures of accuracy. *Kidney International*, *75*(12), 1257–1263. https://doi.org/10.1038/ki.2009.92

van Wilgenburg, E., & Elgar, M. A. (2013). Confirmation Bias in Studies of Nestmate Recognition: A Cautionary Note for Research into the Behaviour of Animals. *PLoS ONE*, *8*(1), 1–8. https://doi.org/10.1371/journal.pone.0053548

Wang, S., & Yao, X. (2013). Relationships between diversity of classification ensembles and single-class performance measures. *IEEE Transactions on Knowledge and Data Engineering*, *25*(1), 206–219. https://doi.org/10.1109/TKDE.2011.207

Warrens, M. J. (2008). *Similarity Coefficients for Binary Data: Properties of Coefficients, Coefficient Matrices, Multi-way Metrics and Multivariate Coefficien*. Leiden University. Retrieved Aug 15th, 2019 from https://openaccess.leidenuniv.nl/bitstream/handle/1887/12987/Full?sequence=2

Wilks, D. S. (2006). *Statistical methods in the atmospheric sciences*. *International Geophysics Series* (2nd ed., Vol. 59). Elsevier. https://doi.org/10.1002/met.16

Yan, B., Koyejo, O., Zhong, K., & Ravikumar, P. (2018). Binary Classification with Karmic, Threshold-Quasi-Concave Metrics. In *Proceedings of the35th International Conference on Machine Learning (ICML)* (pp. 5527–5536). Stockholm, Sweden.

Yerima, S. Y., Sezer, S., & McWilliams, G. (2014). Analysis of Bayesian classification-based approaches for Android malware detection. *IET Information Security*, *8*(1), 25–36. https://doi.org/10.1049/iet-ifs.2013.0095

APPENDICES

APPENDIX A

PERFORMANCE INSTRUMENTS: CATEGORIES, LEVELS, NOTATION, AND FORMATING CONVENTIONS

Table A.1 Performance instruments: categories (measures, metrics, and indicators), abbreviations, names, alternative names, notations, and styles

<table>
<tr><td rowspan="8">**Canonicals**</td><td colspan="2">**Base Measures (*BM*)**</td><td colspan="2">(4 measures)</td><td>**2nd Level Measures** (10 measures)</td></tr>
<tr><td>***FN***</td><td>False Negative</td><td>***FP***</td><td>False Positive</td><td rowspan="7">*BIAS*: Bias, *CKc*: Cohen's Kappa Chance, ***DET***: Determinant, ***DPR***: D Prime, ***LRN***: Likelihood Ratio Negative, ***LRP***: Likelihood Ratio Positive, *NER*: Null Error Rate, *NIR*: No Information Rate (non-information rate), *PREV*: Prevalence, *SKEW*: (Class) Skew</td></tr>
<tr><td>***TN***</td><td>True Negative</td><td>***TP***</td><td>True Positive</td></tr>
<tr><td colspan="2">**1st Level Measures**</td><td colspan="2">(7 measures)</td></tr>
<tr><td>***N***</td><td>Negative</td><td>***P***</td><td>Positive</td></tr>
<tr><td>***ON***</td><td>Outcome Negative</td><td>***OP***</td><td>Outcome Positive</td></tr>
<tr><td>***FC***</td><td>False Classification</td><td>***TC***</td><td>True Classification</td></tr>
<tr><td>***Sn***</td><td>Sample Size</td><td></td><td></td></tr>
<tr><td colspan="4">**3rd Level Measures**</td><td colspan="2">(4 measures)</td></tr>
<tr><td colspan="6">***DP***: Discriminant Power, *HC*: Class Entropy, *HO*: Outcome Entropy, ***OR***: Odds Ratio</td></tr>
<tr><td colspan="3">**Base Metrics** (14 metrics)</td><td colspan="3">**1st Level Metrics** (9 metrics)</td></tr>
<tr><td colspan="3">*ACC*: Accuracy (efficiency, rand index), *AUC*: Area Under Curve, *CRR*: (Correct) Rejection Rate, *DR*: Detection Rate, *FDR*: False Discovery Rate, *FNR*: False Negative Rate, *FOR*: False Omission Rate (imprecision), *FPR*: False Positive Rate, *HOC*: Joint Entropy, *MCR*: Misclassification Rate, *MI*: Mutual Information, *NPV*: Negative Predictive Value, *PPV*: Positive Predictive Value (precision, confidence), *TNR*: True Negative Rate (inverse recall, specificity), *TPR*: True Positive Rate (recall, sensitivity, hit rate, recognition rate)</td><td colspan="3">*BACC*: Balanced Accuracy (strength), *CK*: Cohen's Kappa (Heidke skill score, quality index), *Fm*: F-metrics, *F1*: F1 (F-score, F-measure, positive specific agreement), *G*: G-metric (G-mean, Fowlkes-Mallows index), *INFORM*: Informedness (Youden’s index, delta P', Peirce skill score), *MARK*: Markedness (delta P, Clayton skill score, predictive summary index), *nMI*: Normalized Mutual Information, *WACC*: Weighted Accuracy</td></tr>
<tr><td colspan="3">**2nd Level Metric** (1 metric)</td><td colspan="3">**Indicators** (1 indicator)</td></tr>
<tr><td colspan="3">*MCC*: Matthews Correlation Coefficient (Phi correlation coefficient, Cohen’s index, Yule phi)</td><td colspan="3">*ACCBAR*: Accuracy Barrier</td></tr>
</table>

Table A.2 Color palette (red, green, blue (RGB) color codes in hexadecimal format for background and text colors) for performance instrument types and canonical measures

Type	Level/Type Style	Background	Text	Measure	Abbreviation /Style	Background	Text	
Measures	Base	#A6A6A6	#000000	True Positive	***TP***	#FFCCCC	#CC0000	Canonicals
				False Positive	***FP***	#CCFFCC	#7D3F3F	
				False Negative	***FN***	#FFCCCC	#274927	
				True Negative	***TN***	#CCFFCC	#009900	
	1st Level	#BFBFBF	#000000	Positive	***P***	#990000	#FF5050	
				Negative	***N***	#006600	#33CC33	
				Outcome Positive	***OP***	#CC9999	#FFCCCC	
				Outcome Negative	***ON***	#99CC99	#CCFFCC	
				True Classification	***TC***	#77CCCC	#117777	
				False Classification	***FC***	#FFCCFF	#7030A0	
				Sample Size	***Sn***	#999966	#424100	
	2nd Level	#D9D9D9	#000000					
	3rd Level	#F2F2F2	#000000					
Metrics	Base	#FED96F	#974715					
	1st Level	#FEE59D	#BD581A					
	2nd Level	#FFF1CE	#E46A21					
Indicators	Indicator	#77AADD	#114477					

APPENDIX B

PERFORMANCE INSTRUMENTS EQUATIONS WITH DUALS AND COMPLEMENTS

Table B.1 Measure Equations (numbered in curly braces according to PToPI shown in Figure C.2)

$P = TP + FN \quad \{5\}$	$N = TN + FP \quad \{6\}$	$OP = TP + FP \quad \{7\}$
$ON = TN + FN \quad \{8\}$	$TC = TP + TN \quad \{9\}$	$FC = FP + FN \quad \{10\}$
$Sn = TP + FP + FN + TN = P + N = OP + ON = TC + FC \quad \{11\}$		
$Sn = P + N$	$Sn = OP + ON$	$Sn = TC + FC$
$PREV = \frac{P}{Sn} = BIAS^{*} \quad \{12\}$	$IMB = \frac{\max(P,N)}{\min(P,N)} \quad \{12'\}$	$BIAS = \frac{OP}{Sn} = PREV^{*} \quad \{13\}$
$SKEW = N{:}P \quad \{14\}$	$NIR = \frac{\max(P,N)}{Sn} \quad \{15\}$	$NER = \frac{N}{Sn} = \overline{PREV} \quad \{16\}$
$CKc = \frac{P \cdot OP + N \cdot ON}{Sn^2} \quad \{17\}$	$DPR = \mathrm{Z}(TPR) - \mathrm{Z}(FPR) \quad \{18\}$	
$LRP = \frac{TPR}{FPR} = \frac{TP \cdot N}{FP.P} \quad \{19\}$	$LRN = \frac{FNR}{TNR} = \frac{FN \cdot N}{TN.P} \quad \{20\}$	
$DET = TP \cdot TN - FP \cdot FN \quad \{21\}$		
$HC = -\sum_{m=PREV,1-PREV} m \log_2 m \quad \{22\}$	$HO = -\sum_{m=BIAS,1-BIAS} m \log_2 m \quad \{23\}$	
$OR = \frac{LRP}{LRN} = \frac{TPR \cdot TNR}{FPR \cdot FNR} = \frac{TP \cdot TN}{FP \cdot FN} \quad \{24\}$		
$DP = \frac{\sqrt{3}}{\pi}\left(\log \frac{TPR \cdot TNR}{FPR \cdot FNR}\right) = \frac{\sqrt{3}}{\pi} \log OR = \frac{\sqrt{3}}{\pi} \log \frac{TP \cdot TN}{FP \cdot FN} \quad \{25\}$		
Correction 1. ***OR*** and ***DP*** *are undefined (NaN) due to the zero division by zero (*0/0*) in case of* ***TP·FP***=0 *and* ***FP·FN***=0. *Therefore, they should be 0 (zero) for these cases which means an arbitrary classifier.*		

Table B.2 Metric Equations (numbered in braces according to PToPI shown in Figure C.2)

$TPR = \frac{TP}{P} = PPV^{*}$ (1)	$FNR = \frac{FN}{P} = \overline{TPR}$ (2)	$TNR = \frac{TN}{N} = NPV^{*}$ (3)
$FPR = \frac{FP}{N} = \overline{TNR}$ (4)	$PPV = \frac{TP}{OP} = TPR^{*}$ (5)	$FDR = \frac{FP}{OP} = \overline{PPV}$ (6)
$FOR = \frac{FN}{ON} = \overline{NPV}$ (7)	$NPV = \frac{TN}{ON} = TNR^{*}$ (8)	
$HOC = -\sum_{m=TP,FP,FN,TN} \frac{m}{Sn}\log_2\frac{m}{Sn}$ (9)		
$MI = \frac{TP}{Sn}\log_2\frac{TP/Sn}{PREV \cdot BIAS} + \frac{FP}{Sn}\log_2\frac{FP/Sn}{(1-PREV)\cdot BIAS}$ $+\frac{FN}{Sn}\log_2\frac{FN/Sn}{PREV\cdot(1-BIAS)} + \frac{TN}{Sn}\log_2\frac{TN/Sn}{(1-PREV)\cdot(1-BIAS)}$ (10)		
$DR = \frac{TP}{Sn}$ (11)	$CRR = \frac{TN}{Sn}$ (12)	
$ACC = \frac{TC}{Sn}$ (13)	$MCR = \frac{FC}{Sn} = \overline{ACC}$ (14)	
$INFORM = TPR + TNR - 1 = \frac{TP\cdot N + TN\cdot P - P\cdot N}{P\cdot N} = \frac{TP\cdot N + TN\cdot P}{P\cdot N} - 1 = MARK^{*}$ (15)		
$MARK = PPV + NPV - 1 = \frac{TP\cdot ON + TN\cdot OP - OP\cdot ON}{OP\cdot ON} = \frac{TP\cdot ON + TN\cdot OP}{OP\cdot ON} - 1 = INFORM^{*}$ (16)		
$BACC = \frac{TPR + TNR}{2} = \frac{TP\cdot N + TN\cdot P}{2\cdot P\cdot N}$ (17)	$WACC = \mathrm{w}\cdot TPR + (1-w)\cdot TNR$ (17') w is in (0, 1)	
$G = \sqrt[2]{TPR\cdot TNR} = \sqrt{TP\cdot TN/_{P\cdot N}}$ (18)	$nMI = \frac{MI}{f(HO,HC,HOC)}$ (19)	
$nMI = nMI_{ari} = \frac{MI}{(HO+HC)/2}$ (19.1)	$nMI_{geo} = \frac{MI}{\sqrt[2]{HO\cdot HC}}$ (19.2)	
$nMI_{joi} = \frac{MI}{HOC}$ (19.3)	$nMI_{min} = \frac{MI}{\min(HO,HC)}$ (19.4)	
$nMI_{max} = \frac{MI}{\max(HO,HC)}$ (19.4)	$F_1 = \frac{2PPV\cdot TPR}{PPV + TPR} = \frac{2TP}{2TP + FC}$ (20)	
$F_\beta = (1+\beta^2)\frac{(PPV.TPR)}{(\beta^2 PPV) + TPR)} = \frac{(1+\beta^2).TP}{(1+\beta^2).TP + \beta^2.FN + FP}$ (20')		
$F_{0.5} = 1.25\frac{(PPV.TPR)}{(0.25\cdot PPV + TPR)}$ (21)	$F_2 = 5\frac{(PPV\cdot TPR)}{(4PPV) + TPR)}$ (22)	
$CK = \frac{ACC - CKc}{1 - CKc} = \frac{2(TP\cdot TN - FP\cdot FN)}{P\cdot ON + N\cdot OP} = \frac{DET}{(P\cdot ON + N\cdot OP)/2}$ (23) **Correction 2**. *CK is undefined (NaN) due to the zero division by zero (0/0) in case of* ***P***=0 *and* ***OP***=0 *or* ***N***=0 *or* ***ON***=0. *Therefore, CK should be 0 (zero) for these cases.*		
$MCC = \sqrt{INFORM\cdot MARK} = \frac{TP\cdot TN - FP\cdot FN}{\sqrt{P\cdot OP\cdot N\cdot ON}} = \frac{DET}{\sqrt{P\cdot OP\cdot N\cdot ON}}$ (24) **Correction 3**. *MCC is undefined (NaN) due to the zero division by zero (0/0) in case of* ***P***=0 *and/or* ***OP***=0 *and/or* ***N***=0 *and/or* ***ON***=0. *The possible cases are more than CK's. CK is 0 for them except the cases specified in* Correction 1 *above. Therefore, MCC should also be* 0 *for these cases.*		

APPENDIX C

PToPI: PERIODIC TABLE OF PERFORMANCE INSTRUMENTS (GEOMETRY POSITIONS AND FULL VIEW)

The following figure shows the positioning of the instruments according to instrument geometries. The full view of PToPI is shown in the next page.

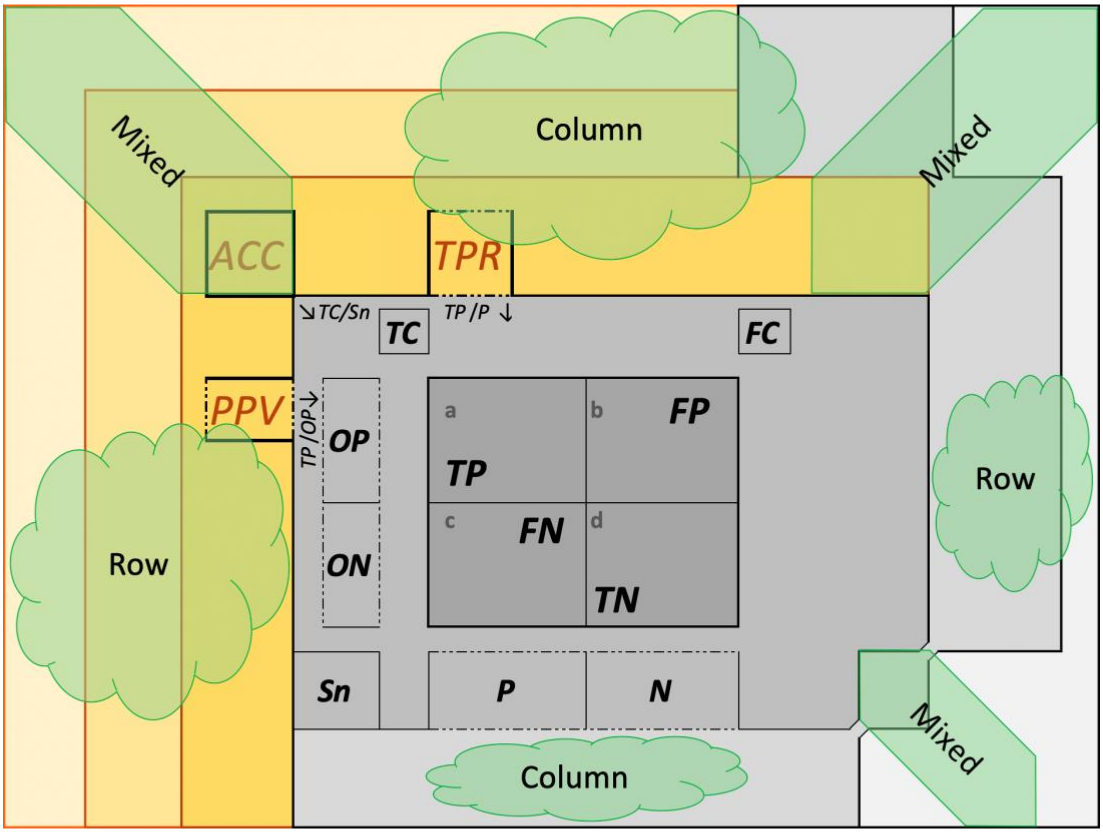


Figure C.1 PToPI instrument positioning according to geometry

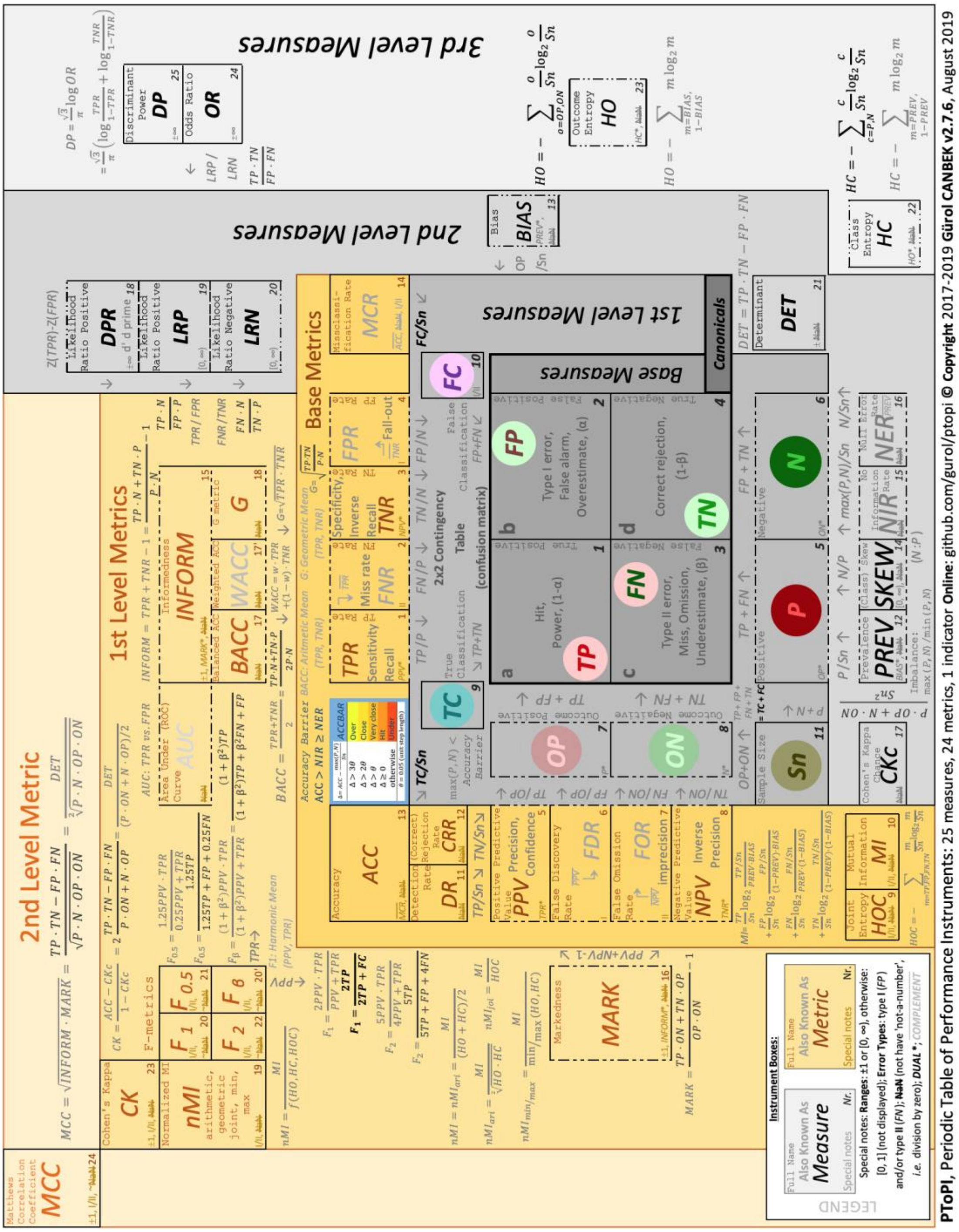


Figure C.2 The proposed binary-classification performance instruments exploratory table (PToPI) for 50 performance instruments including 25 measures, 24 metrics, and 1 indicator in full view.

PToPI in full view presents all the information such as canonical and/or high-level dependency equations. See the legend for the details represented in PToPI. The full-resolution and up-to-date version of PToPI with other extra information can be accessed online at https://github.com/gurol/PToPI.

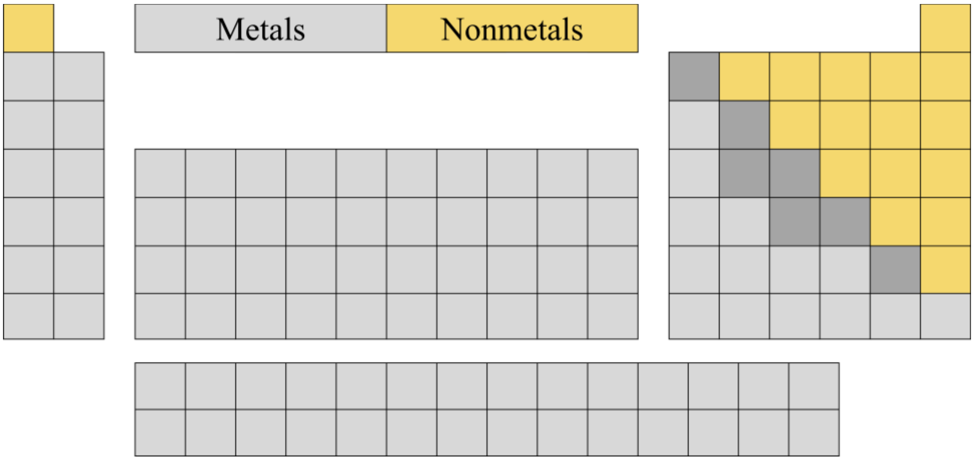
APPENDIX D

ANALOGY BETWEEN PToPI AND PERIODIC TABLE OF ELEMENTS

Table D.1 lists the similarities identified from the familiar source domain (periodic table of elements) to the unfamiliar target domain (PToPI). The similarities could be in corresponding attributes and/or relations.

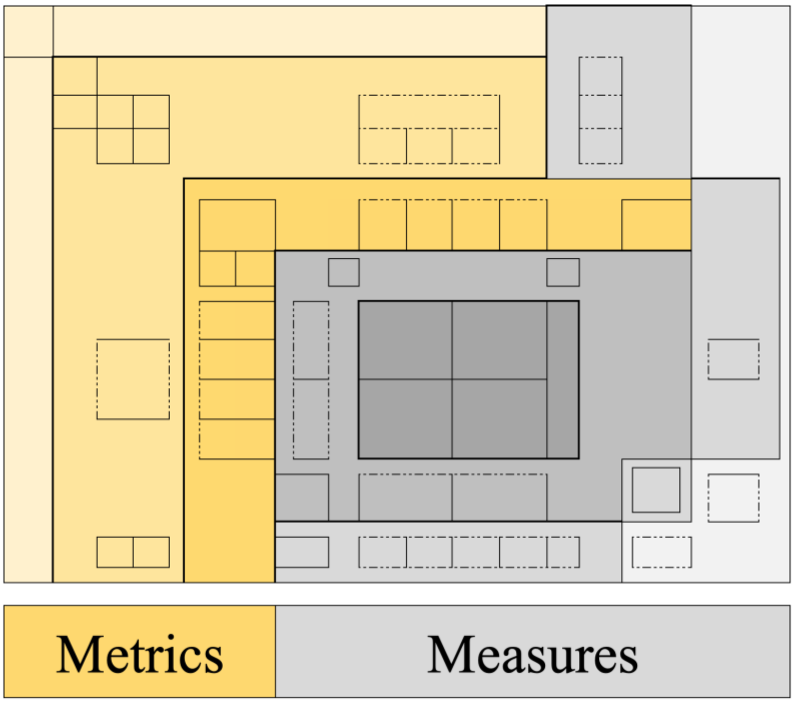
Table D.1 Analogical similarities from the periodic table of elements to PToPI

	Periodic Table (Source Domain)		**PToPI (Target Domain)**	
Description	Tabular display of the chemical elements		Tabular display of the classification instruments	
Types	Metallic	(Left Bottom)	Measures	(Right Bottom)
	Non-Metallic	(Right Top)	Metrics	(Left Top)



(a) Types in Periodic Table: Metals / Nonmetals

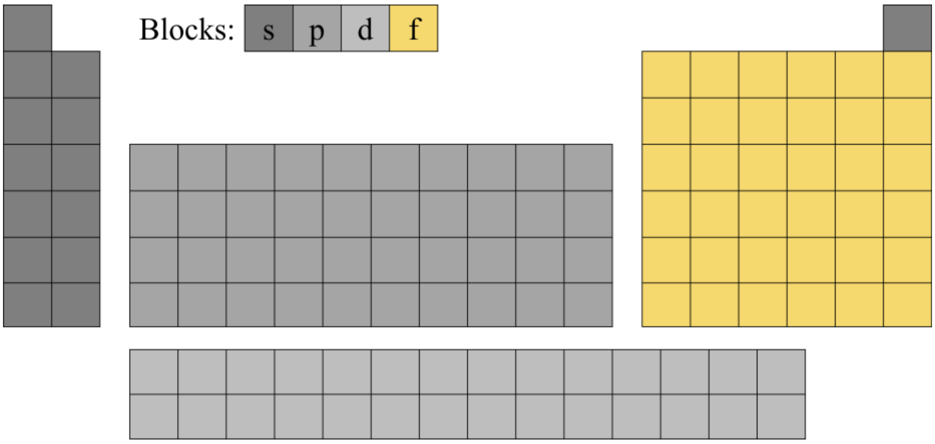
Table D.1 Analogical similarities from the periodic table of elements to PToPI (*continued*)



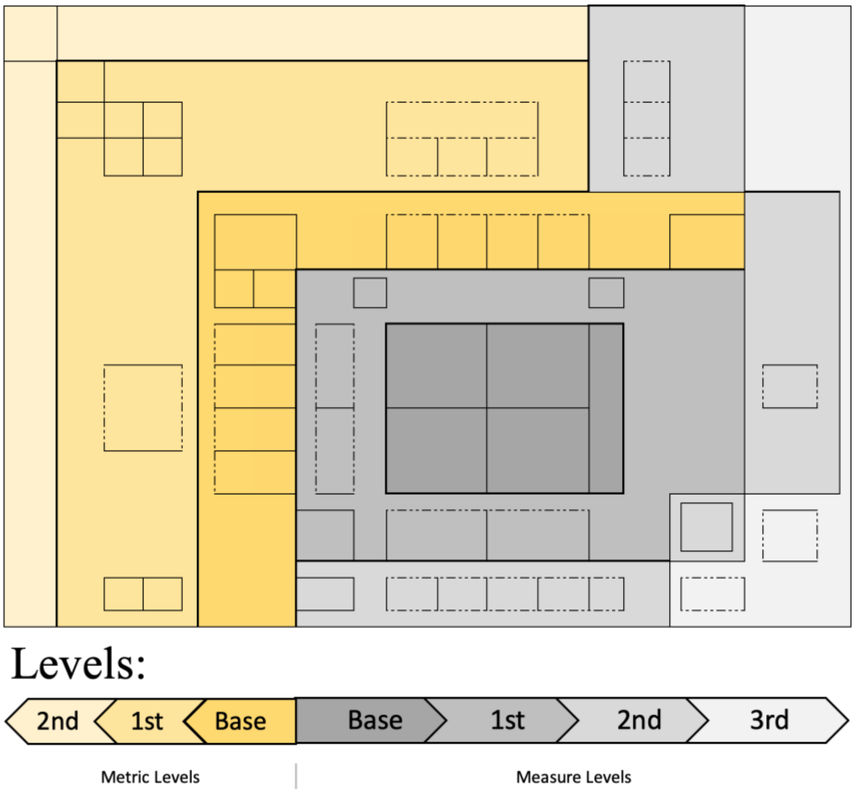
(b) Types in PToPI: Measures / Metrics

	Periodic Table (Source Domain)	PToPI (Target Domain)
Numbering	Atomic number (the total number of protons in the atomic nucleus)	Instrument number (sequence per instrument type started from low-level to high-level, from column, row, to mixed geometry in the same level, and according to the location of dependent instruments)
Instrument Size and Origins	Natural elements (The first 94 elements all occur naturally)	50 measures and metrics
	Synthesized elements (Elements 95 to 118 have only been synthesized in laboratories or nuclear reactors)	Indicators (*ACCBAR*)
Spatial	Periods (periodic trends in element properties such as melting point, density, hardness)	Geometries (from column, row, to mixed)
Grouping	Blocks (4 blocks): Groups having predominantly characterized by the highest energy electrons in the same atomic orbital type.	Levels (7 levels): Similar dependencies in the same instrument type

Table D.1 Analogical similarities from the periodic table of elements to PToPI (*continued*)



(c) Grouping in Periodic Table: Blocks (s, p, d, f)



(d) Grouping in PToPI: Levels (base, 1st, 2nd, 3rd)

	Periodic Table (Source Domain)		**PToPI (Target Domain)**	
Properties	*Metallic*	Hard	*Measures*	Hard to interpret
	Non-Metallic	Soft	*Metrics*	Easy to interpret
	Metallic	High density	*Measures*	High precision
	Non-Metallic	Low density	*Metrics*	Low precision

APPENDIX E

SURVEY SELECTION METHODOLOGY

This study surveys the following 78 academic studies that model some machine learning Android malware classifiers and reports the performance evaluation within the last seven years (2012–2018). The references are given in Appendix F below. Additional to 35 symposia, conference and journal articles published between 2012 and 2018 that had already been reviewed by me, 43 articles were included by the following methodology:

- Selecting the relevant journal articles by searching the IEEE academic database with having "((Android AND malware) AND (accuracy OR precision OR "True Positive" OR "False Positive") AND (Classification OR Detection))" words in articles' title, abstract, or body on 27 March 2018.

- Selecting the relevant conference/journal articles by searching the Google Scholar with matching the same keywords above and reviewing the first 10 relevant articles per year from 2012 to 2018 on May 2018 excluding the patents.

Among the relevant 78 studies, all of the articles were included in performance evaluation terminology findings where available. For other statistics, only the applicable studies are included as specified in Appendix F. For example, when analyzing Accuracy Barrier effect, covered 28 studies have been covered by discarding

- the ones based on malware family detection only, dynamic malware analysis, repackaged application detection, and machine learning evasion, because the goals, datasets, features, and/or metric levels are different from pure static-malware detection domain and

- the articles not reporting *ACC* metric.

In analyzing publication/confirmation biases, 43 studies are covered that are applicable by eliminating

- the articles based on malware family detection only, dynamic malware analysis, repackaged application detection, and machine learning evasion, because the goals, datasets, features, and/or metric levels are different from pure static-malware detection domain and

- the articles where their confusion matrix (base measures) could not be calculated by the reported instruments (e.g. reporting only Accuracy metric).

Note that reviewing and extracting the relevant information from the surveyed studies was long and tiresome because each study describes their methodology and reports the result in different ways and orders.

APPENDIX F

REFERENCES FOR THE ANDROID MALWARE CLASSIFICATION STUDIES SURVEYED

Table F.1 shows the reference information for the surveyed 78 studies described in Appendix E above which is also provided in online data. The table also shows which studies are applicable in the following analysis conducted in this study:

I. Survey 1: Included for performance evaluation reporting analysis? (69 of 78)

II. Survey 1: Included for performance measures or metrics terminology usage? (55 of 78)

III. Survey 1: Included for alternative terms usage for individual metrics? (78 of 78) (*see* Section 2.3.3 for Survey 1)

IV. Case Study 1: Included for Accuracy Barrier (*ACCBAR*) indicator analysis (*i.e.* is *ACC* reported)? (28 of 78) (*see* Section 3.11.1)

V. Case Study 2: Included for publication/confirmation biases case study? (43 of 78) (*see* Appendix H)

Table F.1 Surveyed binary classification studies

Study	References	I	II	III	IV	V
#1	Aafer, Y., Du, W., & Yin, H. (2013). DroidAPIMiner: Mining API-level features for robust malware detection in Android. In *9th International Conference on Security and Privacy in Communication Networks (SecureComm)* (pp. 86–103). Sydney, NSW, Australia: Springer International Publishing	Yes	N/A	Yes	Yes	Yes
#2	Aonzo, S., Merlo, A., Migliardi, M., Oneto, L., & Palmieri, F. (2017). Low-resource footprint, data-driven malware detection on Android. *IEEE Transactions on Sustainable Computing*, *3782*, 1–1. https://doi.org/10.1109/TSUSC.2017.2774184.	Yes	Others	Yes	Yes	Yes

Table F.1 Surveyed binary classification studies (*continued*)

Study	**References**	**I**	**II**	**III**	**IV**	**V**
#3	Apvrille, L., & Apvrille, A. (2015). Identifying unknown android malware with feature extractions and classification techniques. In *14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)* (Vol. 1, pp. 182–189). https://doi.org/10.1109/Trustcom.2015.373	Yes	N/A	Yes	N/A	Yes
#4	Arp, D., Spreitzenbarth, M., Hübner, M., Gascon, H., & Rieck, K. (2014). DREBIN: Effective and explainable detection of Android malware in your pocket. In *Network and Distributed System Security (NDSS) Symposium*. San Diego, California: Internet Society. https://doi.org/10.14722/ndss.2014.23247	Yes	Others	Yes	N/A	N/A
#5	Aswini, A. M., & Vinod, P. (2014). Droid permission miner: Mining prominent permissions for Android malware analysis. In *The 5th International Conference on the Applications of Digital Information and Web Technologies (ICADIWT)* (pp. 81–86). Bangalore, India: IEEE. https://doi.org/10.1109/ICADIWT.2014.6814679	Yes	Both	Yes	Yes	Yes
#6	Canfora, G., De Lorenzo, A., Medvet, E., Mercaldo, F., & Visaggio, C. A. (2015). Effectiveness of opcode ngrams for detection of multi family Android malware. In *10th International Conference on Availability, Reliability and Security (ARES)* (pp. 333–340). https://doi.org/10.1109/ARES.2015.57	Yes	Metrics	Yes	Yes	N/A
#7	Canfora, G., Mercaldo, F., & Visaggio, C. A. (2013). A classifier of malicious Android applications. In *The 8th International Conference on Availability, Reliability and Security (ARES)* (pp. 607–614). Regensburg: IEEE. https://doi.org/10.1109/ARES.2013.80	Yes	Metrics	Yes	N/A	Yes
#8	Cen, L., Gates, C., Si, L., & Li, N. (2015). A probabilistic discriminative model for Android malware detection with decompiled source code. *IEEE Transactions on Dependable and Secure Computing*, *12*(4), 400–412. https://doi.org/10.1109/TDSC.2014.2355839	Yes	Metrics	Yes	N/A	Yes
#9	Damshenas, M., Dehghantanha, A., Choo, K.-K. R., & Mahmud, R. (2015). M0Droid: An Android behavioral-based malware detection model. *Journal of Information Privacy and Security*, *11*(3), 141–157. https://doi.org/10.1080/15536548.2015.1073510	Yes	N/A	Yes	N/A	N/A
#10	Dash, S. K., Suarez-Tangil, G., Khan, S., Tam, K., Ahmadi, M., Kinder, J., & Cavallaro, L. (2016). DroidScribe: Classifying Android malware based on runtime behavior. In *IEEE Symposium on Security and Privacy Workshops (SPW)* (pp. 252–261). https://doi.org/10.1109/SPW.2016.25	Yes	Metrics	Yes	N/A	N/A
#11	Demme, J., Maycock, M., Schmitz, J., Tang, A., Waksman, A., Sethumadhavan, S., & Stolfo, S. (2013). On the feasibility of online malware detection with performance counters. *ACM SIGARCH Computer Architecture News*, *41*(3), 559. https://doi.org/10.1145/2508148.2485970	Yes	Metrics	Yes	N/A	N/A
#12	Demontis, A., Melis, M., Biggio, B., Maiorca, D., Arp, D., Rieck, K., … Roli, F. (2017). Yes, machine learning can be more secure! A case study on Android malware detection. *IEEE Transactions on Dependable and Secure Computing*, *PP*(99), 1–14. https://doi.org/10.1109/TDSC.2017.2700270	Yes	Measures	Yes	N/A	N/A
#13	Deshotels, L., Notani, V., & Lakhotia, A. (2014). DroidLegacy: Automated familial classification of Android malware. In *3rd ACM SIGPLAN on Program Protection and Reverse Engineering Workshop (PPREW)* (pp. 1–12). San Diego, CA, USA: ACM.	Yes	Measures	Yes	Yes	N/A

Table F.1 Surveyed binary classification studies (*continued*)

Study	References	I	II	III	IV	V
	https://doi.org/10.1145/2556464.2556467					
#14	Dimjasevic, M., Atzeni, S., Ugrina, I., & Rakamaric, Z. (2016). Evaluation of Android malware detection based on system calls. In *International Workshop on Security and Privacy Analytics (IWSPA@CODASPY)* (pp. 1–8). New Orleans, LA: ACM. https://doi.org/10.1145/2875475.2875487	Yes	Measures	Yes	N/A	N/A
#15	Du, Y. A. O., Wang, J., & Li, Q. I. (2017). An Android malware detection approach using community structures of weighted function call graphs. *IEEE Access*, *5*, 17478–17486. https://doi.org/10.1109/ACCESS.2017.2720160	Yes	Metrics	Yes	N/A	Yes
#16	Elish, K. O., Shu, X., Yao, D., Ryder, B. G., & Jiang, X. (2015). Profiling user-trigger dependence for Android malware detection. *Computers and Security*, *49*(540), 255–273. https://doi.org/10.1016/j.cose.2014.11.001	Yes	Others	Yes	N/A	N/A
#17	Fan, M., Liu, J., Luo, X., Chen, K., Tian, Z., Zheng, Q., & Liu, T. (2018). Android malware familial classification and representative sample selection via frequent subgraph analysis. *IEEE Transactions on Information Forensics and Security*, *13*(8), 1890–1905. https://doi.org/10.1109/TIFS.2018.2806891	Yes	Metrics	Yes	N/A	N/A
#18	Fan, M., Liu, J., Wang, W., Li, H., Tian, Z., & Liu, T. (2017). DAPASA: Detecting Android piggybacked apps through sensitive subgraph analysis. *IEEE Transactions on Information Forensics and Security*, *12*(8), 1772–1785. https://doi.org/10.1109/TIFS.2017.2687880	Yes	Metrics	Yes	N/A	N/A
#19	Feizollah, A., Badrul, N., & Salleh, R. (2017). AndroDialysis: Analysis of Android intent effectiveness in malware detection. *Computers & Security*, *65*, 121–134. https://doi.org/10.1016/j.cose.2016.11.007	Yes	Others	Yes	N/A	Yes
#20	Feng, Y., Anand, S., Dillig, I., & Aiken, A. (2014). Apposcopy: Semantics-based detection of Android malware through static analysis. In *22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2014)* (pp. 576–587). Hong Kong: ACM. https://doi.org/10.1145/2635868.2635869	Yes	N/A	Yes	N/A	N/A
#21	Garcia, J., Hammad, M., & Malek, S. (2018). Lightweight, obfuscation-resilient detection and family identification of Android malware. *ACM Transactions on Software Engineering and Methodology*, *26*(3), 1–29. https://doi.org/10.1145/3162625	Yes	Metrics	Yes	N/A	Yes
#22	Gascon, H., Yamaguchi, F., Rieck, K., & Arp, D. (2013). Structural detection of Android malware using embedded call graphs. In *ACM Workshop on Artificial Intelligence and Security* (pp. 45–54). New York, New York, USA: ACM. https://doi.org/10.1145/2517312.2517315	Yes	Measures	Yes	N/A	N/A
#23	Ge, H., Ting, L., Hang, D., Hewei, Y., & Miao, Z. (2014). Malicious code detection for Android using instruction signatures. In *8th International Symposium on Service Oriented System Engineering (SOSE)* (pp. 332–337). Oxford, UK: IEEE. https://doi.org/10.1109/SOSE.2014.48	Yes	N/A	Yes	N/A	Yes
#24	Glodek, W., & Harang, R. (2013). Rapid permissions-based detection and analysis of mobile malware using random decision forests. In *Military Communications Conference (MILCOM)* (pp. 980–985). San Diego, CA: IEEE. https://doi.org/10.1109/MILCOM.2013.170	Yes	N/A	Yes	N/A	Yes
#25	Ham, H.-S., & Choi, M.-J. (2013). Analysis of Android malware detection performance using machine learning classifiers. In *International Conference on ICT Convergence*	Yes	Metrics	Yes	N/A	N/A

Table F.1 Surveyed binary classification studies (*continued*)

Study	References	I	II	III	IV	V
	(ICTC) (pp. 490–495). Jeju: IEEE. https://doi.org/10.1109/ICTC.2013.6675404					
#26	Jerome, Q., Allix, K., State, R., & Engel, T. (2014). Using opcode-sequences to detect malicious Android applications. In *Communication and Information Systems Security Symposium (IEEE ICC 2014)* (pp. 914–919). https://doi.org/10.1109/ICC.2014.6883436	Yes	Both	Yes	N/A	Yes
#27	Kirubavathi, G., & Anitha, R. (2018). Structural analysis and detection of android botnets using machine learning techniques. *International Journal of Information Security*, *17*(2), 153–167. https://doi.org/10.1007/s10207-017-0363-3	Yes	Both	Yes	Yes	N/A
#28	Li, J., Sun, L., Yan, Q., Li, Z., Srisa-an, W., & Ye, H. (2018). Significant permission identification for machine learning based Android malware detection. *IEEE Transactions on Industrial Informatics*, *14*(7), 3216–3225. https://doi.org/10.1109/TII.2017.2789219	Yes	Both	Yes	Yes	Yes
#29	Liang, S., & Du, X. (2014). Permission-combination-based scheme for Android mobile malware detection. In *IEEE International Conference on Communications (ICC)* (pp. 2301–2306). Sydney, NSW, Australia: IEEE. https://doi.org/10.1109/ICC.2014.6883666	Yes	N/A	Yes	N/A	Yes
#30	Liu, X., & Liu, J. (2014). A two-layered permission-based Android malware detection scheme. In *2nd International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)* (pp. 142–148). Oxford, UK: IEEE. https://doi.org/10.1109/MobileCloud.2014.22	Yes	Metrics	Yes	Yes	Yes
#31	Lu, Y., Zulie, P., Jingju, L., & Yi, S. (2013). Android malware detection technology based on improved Bayesian classification. In *The 3rd International Conference on Instrumentation, Measurement, Computer, Communication and Control (IMCCC)* (pp. 1338–1341). Shenyang: IEEE. https://doi.org/10.1109/IMCCC.2013.297	Yes	N/A	Yes	Yes	Yes
#32	Mahindru, A., & Singh, P. (2017). Dynamic permissions-based Android malware detection using machine learning techniques. In *10th Innovations in Software Engineering Conference (ISEC)* (pp. 202–210). Jaipur, India: ACM. https://doi.org/10.1145/3021460.3021485	Yes	Both	Yes	N/A	Yes
#33	Martinelli, F., Mercaldo, F., & Saracino, A. (2017). BRIDEMAID: An hybrid tool for accurate detection of Android malware. In *Asia Conference on Computer and Communications Security (ASIA CCS)* (pp. 899–901). Abu Dhabi, United Arab Emirates: ACM. https://doi.org/10.1145/3052973.3055156	Yes	N/A	Yes	Yes	N/A
#34	Matsudo, T., Kodama, E., Wang, J., & Takata, T. (2012). A proposal of security advisory system at the time of the installation of applications on Android OS. In *International Conference on Network-Based Information Systems* (pp. 261–267). Melbourne, VIC: IEEE. https://doi.org/10.1109/NBiS.2012.110	Yes	N/A	Yes	N/A	Yes
#35	Meng, G., Xue, Y., Xu, Z., Liu, Y., Zhang, J., & Narayanan, A. (2016). Semantic modelling of Android malware for effective malware comprehension, detection, and classification. In *25th International Symposium on Software Testing and Analysis (ISSTA)* (pp. 306–317). Saarbrücken, Germany: ACM. https://doi.org/10.1145/2931037.2931043	Yes	Others	Yes	N/A	Yes
#36	Milosevic, N., Dehghantanha, A., & Choo, K.-K. R. (2017). Machine learning aided Android malware classification. *Computers and Electrical Engineering*, *61*, 266–274. https://doi.org/10.1016/j.compeleceng.2017.02.013	Yes	Both	Yes	N/A	Yes

Table F.1 Surveyed binary classification studies (*continued*)

Study	References	I	II	III	IV	V
#37	Muttik, I., Yerima, S. Y., & Sezer, S. (2015). High accuracy Android malware detection using ensemble learning. *IET Information Security*, *9*(6), 313–320. https://doi.org/10.1049/iet-ifs.2014.0099	Yes	Metrics	Yes	Yes	Yes
#38	Narayanan, A., Chandramohan, M., Chen, L., & Liu, Y. (2017). Context-aware, adaptive, and scalable Android malware detection through online learning. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *1*(3), 157–175. https://doi.org/10.1109/TETCI.2017.2699220	Yes	Metrics	Yes	Yes	Yes
#39	Narudin, F. A., Feizollah, A., Anuar, N. B., & Gani, A. (2016). Evaluation of machine learning classifiers for mobile malware detection. *Soft Computing*, *20*(1), 343–357. https://doi.org/10.1007/s00500-014-1511-6	Yes	Both	Yes	N/A	N/A
#40	Pan, J. S., Yang, C. N., & Lin, C. C. (2013). Performance evaluation on permission-based detection for Android malware. *Advances in Intelligent Systems & Applications, Smart Innovation, Systems and Technologies (SIST)*, *21*, 111–120. https://doi.org/10.1007/978-3-642-35473-1	Yes	Metrics	Yes	N/A	Yes
#41	Peiravian, N., & Zhu, X. (2013). Machine learning for Android malware detection using permission and API calls. In *IEEE 25th International Conference on Tools with Artificial Intelligence (ICTAI)* (pp. 300–305). Herndon, VA: IEEE. https://doi.org/10.1109/ICTAI.2013.53	Yes	N/A	Yes	Yes	Yes
#42	Rahman, M. (2013). DroidMLN: A Markov logic network approach to detect android malware. In *Proceedings - 2013 12th International Conference on Machine Learning and Applications, ICMLA 2013* (Vol. 2, pp. 166–169). https://doi.org/10.1109/ICMLA.2013.184	Yes	N/A	Yes	N/A	N/A
#43	Rahman, M., Rahman, M., Carbunar, B., & Chau, D. H. (2017). Search rank fraud and malware detection in Google Play. *IEEE Transactions on Knowledge and Data Engineering*, *29*(6), 1329–1342. https://doi.org/10.1109/TKDE.2017.2667658	Yes	N/A	Yes	Yes	Yes
#44	Sahs, J., & Khan, L. (2012). A machine learning approach to Android malware detection. In *European Intelligence and Security Informatics Conference (EISIC)* (pp. 141–147). Odense: IEEE. https://doi.org/10.1109/EISIC.2012.34	Yes	Measures	Yes	N/A	Yes
#45	Sanz, B., Santos, I., Laorden, C., Ugarte-Pedrero, X., Bringas, P. G., & Alvarez, G. (2013). PUMA: Permission usage to detect malware in Android. In *International Joint Conference CISIS-ICEUTE-SOCO Special Sessions* (pp. 289–298). Ostrava, Czech Republic: Springer Berlin Heidelberg.	Yes	Others	Yes	Yes	Yes
#46	Sanz, B., Santos, I., Laorden, C., Ugarte-Pedrero, X., Nieves, J., Bringas, P. G., & Marañón, G. Á. (2013). MAMA: Manifest analysis for malware detection in Android. *Cybernetics and Systems*, *44*(6–7), 469–488. https://doi.org/10.1080/01969722.2013.803889	Yes	Both	Yes	Yes	Yes
#47	Sen, S., Aysan, A. I., & Clark, J. A. (2018). SAFEDroid: Using structural features for detecting Android malwares. In *Security and Privacy in Communication Networks (SecureComm 2017) - Workshop on Security and Privacy on Internet of Things (SePrIoT)* (pp. 255–270). Niagara Falls, Canada: Springer International Publishing. https://doi.org/10.1007/978-3-319-78816-6_18	Yes	Metrics	Yes	Yes	Yes
#48	Sheen, S., Anitha, R., & Natarajan, V. (2015). Android based malware detection using a multifeature collaborative decision fusion approach. *Neurocomputing*, *151*(P2), 905–912. https://doi.org/10.1016/j.neucom.2014.10.004	Yes	Measures	Yes	N/A	Yes

Table F.1 Surveyed binary classification studies (*continued*)

Study	References	I	II	III	IV	V
#49	Shen, F., Vecchio, J. Del, Mohaisen, A., Ko, S. Y., & Ziarek, L. (2017). Android malware detection using complex-flows. In *37th International Conference on Distributed Computing Systems (ICDCS)* (pp. 2430–2437). Atlanta, GA, USA: IEEE. https://doi.org/10.1109/ICDCS.2017.190	Yes	Metrics	Yes	Yes	Yes
#50	Shen, Z., Hsu, C.-W., & Shieh, S. W. (2017). Security semantics modeling with progressive distillation. *IEEE Transactions on Mobile Computing*, *16*(11), 3196–3208. https://doi.org/10.1109/TMC.2017.2690425	Yes	N/A	Yes	N/A	N/A
#51	Suarez-Tangil, G., Dash, S. K., Holloway, R., Ahmadi, M., Giacinto, G., Kinder, J., & Cavallaro, L. (2017). DroidSieve: Fast and accurate classification of obfuscated Android malware. In *7th ACM Conference on Data and Application Security and Privacy (CODASPY)* (pp. 309–320). Scottsdale, Arizona: ACM. https://doi.org/10.1145/3029806.3029825	Yes	Metrics	Yes	Yes	Yes
#52	Talha, K. A., Alper, D. I., & Aydin, C. (2015). APK Auditor: Permission-based Android malware detection system. *Digital Investigation*, *13*, 1–14. https://doi.org/10.1016/j.diin.2015.01.001	Yes	N/A	Yes	Yes	Yes
#53	Tao, G., Zheng, Z., Guo, Z., & Lyu, M. R. (2017). MalPat: Mining patterns of malicious and benign Android apps via permission-related APIs. *IEEE Transactions on Reliability*, *67*(1), 355–369. https://doi.org/10.1109/TR.2017.2778147	Yes	Both	Yes	N/A	Yes
#54	Tian, K., Yao, D., Ryder, B. G., & Tan, G. (2016). Analysis of code heterogeneity for high-precision classification of repackaged malware. In *IEEE Symposium on Security and Privacy Workshops (SPW)* (pp. 262–271). San Jose, CA, USA: IEEE. https://doi.org/10.1109/SPW.2016.33	Yes	N/A	Yes	N/A	N/A
#55	Tong, F., & Yan, Z. (2017). A hybrid approach of mobile malware detection in Android. *Journal of Parallel and Distributed Computing*, *103*, 22–31. https://doi.org/10.1016/j.jpdc.2016.10.012	Yes	N/A	Yes	N/A	N/A
#56	Wang, S., Yan, Q., Chen, Z., Yang, B., Zhao, C., & Conti, M. (2018). Detecting Android malware leveraging text semantics of network flows. *IEEE Transactions on Information Forensics and Security*, *13*(5), 1096–1109. https://doi.org/10.1109/TIFS.2017.2771228	Yes	Both	Yes	Yes	N/A
#57	Wang, W., Li, Y., Wang, X., Liu, J., & Zhang, X. (2018). Detecting Android malicious apps and categorizing benign apps with ensemble of classifiers. *Future Generation Computer Systems*, *78*, 987–994. https://doi.org/10.1016/j.future.2017.01.019	Yes	Measures	Yes	Yes	Yes
#58	Wang, W., Wang, X., Feng, D., Liu, J., Han, Z., & Zhang, X. (2014). Exploring permission-induced risk in Android applications for malicious application detection. *IEEE Transactions on Information Forensics and Security*, *9*(11), 1828–1842. https://doi.org/10.1109/TIFS.2014.2353996	Yes	Measures	Yes	N/A	Yes
#59	Wei, T.-E., Mao, C.-H., Jeng, A. B., Lee, H.-M., Wang, H. T., & Wu, D.-J. (2012). Android malware detection via a latent network behavior analysis. In *Proc. of the 11th IEEE Int. Conference on Trust, Security and Privacy in Computing and Communications, TrustCom-2012 - 11th IEEE Int. Conference on Ubiquitous Computing and Communications, IUCC-2012* (pp. 1251–1258). https://doi.org/10.1109/TrustCom.2012.91	Yes	Metrics	Yes	N/A	N/A
#60	Wu, D.-J., Mao, C.-H., Wei, T.-E., Lee, H.-M., & Wu, K.-P. (2012). DroidMat: Android malware detection through manifest and API calls tracing. In *The 7th Asia Joint Conference on Information Security (Asia JCIS)* (pp. 62–69). Tokyo: IEEE. https://doi.org/10.1109/AsiaJCIS.2012.18	Yes	Metrics	Yes	N/A	Yes

Table F.1 Surveyed binary classification studies (*continued*)

Study	References	I	II	III	IV	V
#61	Wu, W.-C., & Hung, S.-H. (2014). DroidDolphin: A dynamic Android malware detection framework using big data and machine learning. In *Conference on Research in Adaptive and Convergent Systems (RACS)* (pp. 247–252). Towson, Maryland: ACM. https://doi.org/10.1145/2663761.2664223	Yes	Both	Yes	N/A	N/A
#62	Xiao, X., Wang, Z., Li, Q., Xia, S., & Jiang, Y. (2017). Back-propagation neural network on Markov chains from system call sequences: A new approach for detecting Android malware with system call sequences. *IET Information Security*, *11*(1), 8–15. https://doi.org/10.1049/iet-ifs.2015.0211	Yes	Others	Yes	N/A	Yes
#63	Xu, K., Li, Y., & Deng, R. H. (2016). ICCDetector: ICC-based malware detection on Android. *IEEE Transactions on Information Forensics and Security*, *11*(6), 1252–1264. https://doi.org/10.1109/TIFS.2016.2523912	Yes	Metrics	Yes	Yes	Yes
#64	Yang, C., Xu, Z., Gu, G., Yegneswaran, V., & Porras, P. A. (2014). DroidMiner: Automated mining and characterization of fine-grained malicious behaviors in Android applications. In *European Symposium on Research in Computer Security (ESORICS)* (pp. 163–182). Wrocław, Poland: Springer. https://doi.org/10.1007/978-3-319-11203-9_10	Yes	Metrics	Yes	N/A	Yes
#65	Yerima, S. Y., Sezer, S., & McWilliams, G. (2014). Analysis of Bayesian classification-based approaches for Android malware detection. *IET Information Security*, *8*(1), 25–36. https://doi.org/10.1049/iet-ifs.2013.0095	Yes	Both	Yes	Yes	Yes
#66	Yerima, S. Y., Sezer, S., & Muttik, I. (2014). Android malware detection using parallel machine learning classifiers. In *The 8th International Conference on Next Generation Mobile Apps, Services and Technologies (NGMAST)* (pp. 37–42). Oxford, United Kingdom: IEEE. https://doi.org/10.1109/NGMAST.2014.23	Yes	Metrics	Yes	Yes	Yes
#67	Yerima, S. Y., Sezer, S., McWilliams, G., & Muttik, I. (2013). A new Android malware detection approach using Bayesian classification. In *27th International Conference on Advanced Information Networking and Applications (AINA)* (pp. 121–128). Barcelona, Spain: IEEE. https://doi.org/10.1109/AINA.2013.88	Yes	Both	Yes	Yes	Yes
#68	Yuan, Z., Lu, Y., & Xue, Y. (2016). Droiddetector: Android malware characterization and detection using deep learning. *Tsinghua Science and Technology*, *21*(1), 114–123. https://doi.org/10.1109/TST.2016.7399288	Yes	N/A	Yes	Yes	N/A
#69	Yuan, Z., Lu, Y., Wang, Z., & Xue, Y. (2014). Droid-Sec: Deep learning in Android malware detection. In *ACM Conference on SIGCOMM* (pp. 371–372). Chicago, Illinois, USA: ACM. https://doi.org/10.1145/2619239.2631434	Yes	Others	Yes	Yes	N/A
#70	Abawajy, J., & Kelarev, A. (2017). Iterative classifier fusion system for the detection of Android malware. *IEEE Transactions on Big Data*, *5*(3), 1–1. https://doi.org/10.1109/TBDATA.2017.2676100	N/A	Both	Yes	N/A	N/A
#71	Azmoodeh, A., Dehghantanha, A., & Choo, K.-K. R. (2018). Robust malware detection for Internet Of (Battlefield) Things devices using deep eigenspace learning. *IEEE Transactions on Sustainable Computing*, *3782*(c), 1–1. https://doi.org/10.1109/TSUSC.2018.2809665	N/A	Metrics	Yes	N/A	N/A
#72	Dini, G., Martinelli, F., Matteucci, I., Petrocchi, M., Saracino, A., & Sgandurra, D. (2016). Risk analysis of Android applications: A user-centric solution. *Future Generation Computer Systems*, *80*, 505–518. https://doi.org/10.1016/j.future.2016.05.035	N/A	N/A	Yes	N/A	N/A

Table F.1 Surveyed binary classification studies (*continued*)

Study	**References**	**I**	**II**	**III**	**IV**	**V**
#73	Grace, M., Zhou, Y., Zhang, Q., Zou, S., & Jiang, X. (2012). RiskRanker: Scalable and accurate zero-day Android malware detection categories and subject descriptors. In *International Conference on Mobile Systems, Applications, and Services (MobiSys)* (pp. 281–294). Low Wood Bay, Lake District: ACM. https://doi.org/10.1145/2307636.2307663	N/A	Others	Yes	N/A	N/A
#74	Peng, H., Gates, C., Sarma, B., Li, N., Qi, Y., Potharaju, R., … Molloy, I. (2012). Using probabilistic generative models for ranking risks of Android apps. In *19th Conference on Computer and Communications Security (CCS)* (pp. 241–252). New York, New York, USA: ACM. https://doi.org/10.1145/2382196.2382224	N/A	N/A	Yes	N/A	N/A
#75	Sarma, B., Li, N., Gates, C., Potharaju, R., Nita-Rotaru, C., & Molloy, I. (2012). Android permissions: A perspective combining risks and benefits. In *17th Symposium on Access Control Models and Technologies (SACMAT)* (pp. 13–22). New York, New York, USA: ACM. https://doi.org/10.1145/2295136.2295141	N/A	N/A	Yes	N/A	N/A
#76	Schmidt, A., Bye, R., Schmidt, H., Clausen, J., & Kiraz, O. (2009). Static analysis of executables for collaborative malware detection on Android. In *IEEE International Conference on Communications* (pp. 1–5). Dresden, Germany: IEEE. https://doi.org/10.1109/ICC.2009.5199486	N/A	Others	Yes	N/A	N/A
#77	Sun, M., Li, X., Lui, J., & Ma, R. (2017). MONET: A user-oriented behavior-based malware variants detection system for Android. *IEEE Transactions on Information Forensics and Security*, *12*(5), 1103–1112. https://doi.org/10.1109/TIFS.2016.2646641	N/A	N/A	Yes	N/A	N/A
#78	Zhang, M., Duan, Y., Yin, H., & Zhao, Z. (2014). Semantics-aware Android malware classification using weighted contextual API dependency graphs. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)* (pp. 1105–1116). Scottsdale, Arizona, USA: ACM. https://doi.org/10.1145/2660267.2660359	N/A	N/A	Yes	N/A	N/A

APPENDIX G

SUMMARY OF BENCHMETRIC RESULTS

Table G.1 shows the summary of BenchMetric results per binary-classification performance metrics according to the criteria in three stages.

Table G.1 Summary of BenchMetric results

Robustness Rank	Metrics	Stage-1			Stage-2											Stage-3						
		1.1	1.2	1.3	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8	2.9	2.10	2.11	3.1	3.2	3.3	3.4	3.5	3.6	3.7
1	*MCC*										🟥											
2	*CK*										🟧	🟧	🟧							🟧		
3	*BACC*	🟥			🟧		🟧				🟥									🟧		
4	*INFORM*	🟥			🟧		🟧				🟥									🟧		
5	*MARK*	🟥			🟧		🟧				🟥									🟧		
6	*G*	🟥			🟧		🟧				🟥											🟥
7	*ACC*				🟥	🟥	🟧											🟥	🟥			🟧
8	*F1*			🟧			🟨			🟧	🟧					🟨	🟧	🟧	🟧		🟧	🟧
9	*TNR*	🟧		🟧	🟧	🟧	🟧				🟥					🟧		🟧	🟧		🟧	🟧
10	*TPR*	🟧		🟧	🟧	🟧	🟧				🟥					🟧		🟧	🟧		🟧	🟧
11	*NPV*	🟧		🟧	🟧	🟧	🟧				🟥					🟧	🟥	🟧	🟧		🟧	🟧
12	*PPV*	🟧		🟧	🟧	🟧	🟧				🟥					🟧	🟥	🟧	🟧		🟧	🟧
13	*nMI*	🟥	🟧					🟧	🟧		🟧	🟧	🟧	🟧	🟧	🟥			🟥	🟥	🟥	

Robustness degree 🟥 -1 🟧 $-1/2$ 🟨 $-1/4$ (White) Robust

Table G.2 Summary of robustness issues in metrics (in alphabetic order)

Metric (Rank)	Robustness issues
ACC (7th)	NaN in some extreme cases, Missing class or outcome coverage, Insensitive to one or more base measures, Low granular output coverage in metric-spaces, Less smooth metric-spaces, Less consistent with other metrics
BACC (3rd)	(+) Same mean, median, and mode in metric-space, Missing class or outcome coverage, Insensitive to one or more base measures, Has monotonicity violations, Completely consistent with *INFORM*
CK (2nd)	NaN in some extreme cases, Unsymmetrical metric-space, Imbalanced or low correlations with individual base measures, Has minor monotonicity violations, Less consistent with other metrics
F1 (8th)	NaN in some extreme cases, Unsymmetrical metric-space, Accumulation at zero, Insensitive to one or more base measures, Inconsistency in swapping base measures, Imbalanced or low correlations with individual base measures, Correlated with *PREV*, Low granular output coverage in metric-spaces, Less consistent with other metrics
G (6ht)	Unsymmetrical metric-space, Accumulation at zero, Missing class or outcome coverage, Insensitive to one or more base measures, Imbalanced or low correlations with individual base measures, Less consistent with other metrics, The least discriminating metric
INFORM (4th)	(+) Same mean, median, and mode in metric-space, Missing class or outcome coverage, Insensitive to one or more base measures, Has monotonicity violations, Completely consistent with *BACC*
MARK (5th)	(+) Same mean, median, and mode in metric-space, Missing class or outcome coverage, Insensitive to one or more base measures, Has monotonicity violations, Less consistent with other metrics
MCC (1st)	NaN in some extreme cases
nMI (13th)	High values when ***FP*** and ***FN*** are higher than ***TP*** and ***TN***, Unsymmetrical metric-space, Inconsistency in swapping base measures, Highly right-skewed metric-space, The lowest correlation with individual base measures, Less smooth metric-spaces, Has considerable monotonicity violations, The least consistent with other metrics, The most discriminating metric
NPV (11th)	Missing class or outcome coverage, Single-class-only (*P*-only or *N*-only), Insensitive to one or more base measures, Imbalanced or low correlations with individual base measures, Correlated with *PREV*, Low granular output coverage in metric-spaces, Less consistent with other metrics
PPV (12th)	Missing class or outcome coverage, Single-class-only (*P*-only or *N*-only), Insensitive to one or more base measures, Imbalanced or low correlations with individual base measures, Correlated with *PREV*, Low granular output coverage in metric-spaces, Less consistent with other metrics
TNR (9th)	Missing class or outcome coverage, Single-class-only (*P*-only or *N*-only), Insensitive to one or more base measures, Imbalanced or low correlations with individual base measures, Low granular output coverage in metric-spaces, Less consistent with other metrics
TPR (10th)	Missing class or outcome coverage, Single-class-only (*P*-only or *N*-only), Insensitive to one or more base measures, Imbalanced or low correlations with individual base measures, Low granular output coverage in metric-spaces, Less consistent with other metrics

APPENDIX H

CASE STUDY 2: REPORTING BIAS IN CLASSIFICATION PERFORMANCE REPORTING

In the last decade, few studies have been conducted to criticize performance evaluation approaches in different domains. For example, Shepperd points out that ML researchers in software engineering especially concentrate on repeating experiments on new data until getting a better result for their classifier and publishing them (Shepperd, 2013, p. 9). His extensive survey reveals that classifiers actually perform poorly if their performances are expressed by *MCC*:

- *MCC* is even negative for 4.3% of the classifiers and *MCC* < 0.1 for 25%.

- Only, three percent of the publications reviewed had a performance greater than 0.7 when the reported performances were expressed with *MCC* (Shepperd, 2013, p. 22).

- The classifiers in two published studies have −0.50 and −0.47 *MCC* performances and one study reporting its performance with *TPR*, *PPV,* and *ACC* metrics as 0.68, 0.62, 0.64, respectively has actually 0.29 *MCC*.

From a general research perspective, it is possible to encounter binary classification studies that did not report the confusion matrix. Thus, we cannot know their performances in terms of other metrics. For example, what if we could re-evaluate existing classification studies in terms of *MCC* as the most robust metric determined in BenchMetric. Specifically, preferring a metric among the possible ones may cause confirmation and/or publication biases in the literature.

Note that this chapter makes contributions addressing the following research questions:

- What are the problems in performance evaluation reporting? (**RQ1**)
- How to enhance comprehending, using, representing, reporting, learning, and teaching binary-classification performance instruments? (**RQ3**)
- What should be reported for expressing classification performance? (**RQ4**).

Equations Revealing Confusion Matrix

This thesis introduces the equations to reveal confusion matrix or base measures with given performance instruments. Having base measures allows calculating any performance instrument including the ones that are not reported in the original study. Note that the equations that are given below and some additional facilities in a developed R script (TasKarMissing.R) are provided online at https://github.com/gurol/TasKar.

The followings list 18 equations to calculate ***TP***, ***FP***, ***FN***, and ***TN*** for 8 different combinations of given measures and metrics. To the best of my knowledge, such equations are provided for the first time in the literature. Most of the combinations address the cases found in articles reporting classification performance as reviewed in the case study domain (Android mobile malware detection) summarized in Table 2.2.

*I) Given **P**, **N**, TPR, and FPR*

$$TP = TPR.P \tag{H.1}$$

$$FP = FPR.N \tag{H.2}$$

$$FN = P - TP \tag{H.3}$$

$$TN = N - FP \tag{H.4}$$

*II) Given **P**, **N**, TPR, and PPV*

$$TP = TPR.P \tag{H.5}$$

$$FP = TP\left(\frac{1}{PPV} - 1\right) \tag{H.6}$$

FN via (H.3) and ***TN*** via (H.4)

*III) Given **P**, **N**, TPR, and ACC*

TP via (H.5) and ***FN*** via (H.7)

$$TN = ACC(P + N) - TP \tag{H.7}$$

$$FP = N - TN \tag{H.8}$$

IV) Given ***P****,* ***N****, ACC, and FPR*

$$FP = FPR.N \tag{H.9}$$

TN via (H.4)

$$TP = ACC(P + N) - TN \tag{H.10}$$

FN via (H.3)

V) Given ***P****,* ***N****, ACC, and F1*

$$TP = \left(\frac{(P + N).(1 - ACC).F1}{2.(1 - F1)}\right) \tag{H.11}$$

TN via (H.7), ***FN*** via (H.3), and ***FP*** via (H.8)

VI) Given ***P****,* ***N****, BIAS, and TPR*

TP via (H.5)

$$FP = BIAS(P + N) - TP \tag{H.12}$$

FN via (H.3) and ***TN*** via (H.4)

VII) Given ***Sn****, FPR, FNR, and ACC*

$$FN = FNR.\frac{Sn.(1 - ACC - FPR)}{2.FPR} \tag{H.13}$$

$$FP = FPR.\left(Sn - \frac{FN}{FNR}\right) \tag{H.14}$$

$$P = FNR.FN \tag{H.15}$$

$$N = FPR.FP \tag{H.16}$$

$$TP = P - FN \tag{H.17}$$

TN via (H.4)

VIII) Given ***P****, TPR, FPR, and ACC*

$$N = \frac{P.(TPR - ACC)}{ACC + FPR - 1} \tag{H.18}$$

Apply the equations in (I) or (III).

The calculated base measures are fractional and converted into integers by ensuring ***TP*** + ***FN*** is equal to given ***P*** and ***TN*** + ***FP*** is equal to given ***N*** value. Rounding the calculated base measures can cause under or over totals in classes. For example, for given ***P*** = 25, the calculated ***TP*** = 12.25 and ***FN*** = 12.39 yield ***TP*** + ***FN*** = 12+12 = 24 < ***P*** = 25. Therefore, I designed and developed a procedure to handle different cases. Refer to the documentation in TasKarMissing.R script for more information.

Using provided script, researchers can easily test the classification studies in a domain they study and reveal the confusion matrix of the studies to analyze further (*e.g.*, checking whether a publication bias and/or confirmation bias exist).

Table H.1 Classification report information for an example of classification studies to reveal confusion matrix

Study	Config	*N*	*P*	*TP*	*FP*	*FN*	*TN*	*TPR*	*TNR*	*FPR*	*FNR*	*ACC*	*PPV*	*NPV*	*F1*
s01	1	261	180					0.956	0.621						
s01	2	261	180					0.467	0.13						
s02	1	500	500					0.8				0.75			

"Study" column represents the individual studies to be surveyed that are typically related to an article. "Config" column is the order number determining the specific configuration of a classifier. For example, in a single study (*e.g.*, "s01"), one configuration (Config = 1) belongs to a decision tree classifier whereas the other configuration (Config = 2) belongs to a support vector machines classifier. "***N***" and "***P***" depicts the number of negative and positive class samples. Other columns specify any measure or metric reported by the studies. In both of the configurations of the first study, for example, only *TPR* and *TNR* were reported whereas the second study reported *TPR* and *ACC*.

The following is the code snippet to reveal the confusion matrix in R by sourcing the provided script (TasKarMissing.R).

```
# Copy the values in the spreadsheet provided like in Table H.1.
survey <- rclip()
# Set problematic metrics as NA
#   (for example, the ones cause exceptions in initParsedMetrics)
#   survey$F1[44] <- NA
# Reveal confusion matrixes
parsed_base_metrics <- revealConfusionMatrixes(survey)
## Or reveal confusion matrixes by excluding mismatching Sns
parsed_base_metrics <- revealConfusionMatrixes(survey, FALSE)
```

Results

The equations introduced in this chapter are tested in a case study by running the provided API. The case study domain is Android mobile malware detection as surveyed in Section 2.3. The base measures of 43 surveyed studies listed in Appendix F. Table H.2 lists the highly varied distributions of individual and combination of metrics reported in 43 studies.

Table H.2 The distribution of metrics/combinations of metrics reported in 43 binary classification studies surveyed

Individual Metrics		**23 Different Metric Combinations**					
TPR	84%	*TPR, PPV, F1*	14%	*TPR, PPV*	5%	*ACC, F1*	2%
FPR	65%	*TPR, FPR*	14%	*TPR, PPV, F1, TNR, NPV*	2%	*ACC, TNR*	2%
ACC	47%	*TPR, FPR, ACC*	9%	*TPR, FPR, ACC, FNR*	2%	*FPR, ACC*	2%
PPV	42%	*TPR, FPR, ACC, FNR, TNR*	7%	*TPR, FPR, ACC, PPV*	2%	*FPR, FNR*	2%
F1	33%	*TPR, FPR, ACC, PPV, FNR, TNR*	5%	*TPR, FPR, FNR, TNR*	2%	*TPR, ACC*	2%
FNR	23%	*TPR, FPR, ACC, PPV, F1*	5%	*TPR, ACC, PPV*	2%	*TPR, TNR*	2%
TNR	21%	*TPR, FPR, PPV, F1*	5%	*TPR, FPR, F1*	2%	*F1*	2%
NPV	2%	*FPR, ACC, FNR*	5%	*TPR, FPR, PPV*	2%		

The following steps are conducted in this case study:

- Prepare the classification report information list like Table H.1
- Using the provided API on the report information list
    - Reveal the base measures
    - Re-calculate unreported performance metrics based on the base measures
- Extract the maximum value of the metrics originally reported per surveyed study (M_{max})
- Compare the re-calculated *MCC'* (normalized *MCC* in [0, 1] range) as a robust metric and the maximum reported metric as the published metric.

Figure H.1 shows the results of the case study. The prepared graphic shows the difference between M_{max} and *MCC'*. The deltas ($M_{\text{max}} – MCC'$) are shown in Y-axis and the studies are sorted according to deltas in decreasing order.

The case study uncovers a critical issue in performance reporting. The findings suggest that some studies might report classification performances in terms of the metrics with amplified values. Among the studies, 23% reports a metric that is more than *MCC* above 0.05, which is a significant difference in classifications targeting top performance in [0.95, 1.00] range. The maximum difference (delta) is unexpectedly 0.37 following 0.26, 0.13, 0.12, and 0.09 in all the studies.

The researchers might have not known that *MCC* is the most robust metric and/or followed the conventions in choosing a performance metric. However, this could also be interpreted as a potential sign of reporting biases such as publication bias or confirmation bias that should be avoided in any case.

Publication bias is a tendency of the researchers to preferentially include in their study reports findings conforming to their preconceived notions, or outcomes preferred by the other parties around academic publication process such as journals, reviewers, and editors (Porta, 2014, p. 230). Authors who may feel the need to achieve high performance to be able to publish their studies could use metrics with higher outcomes.

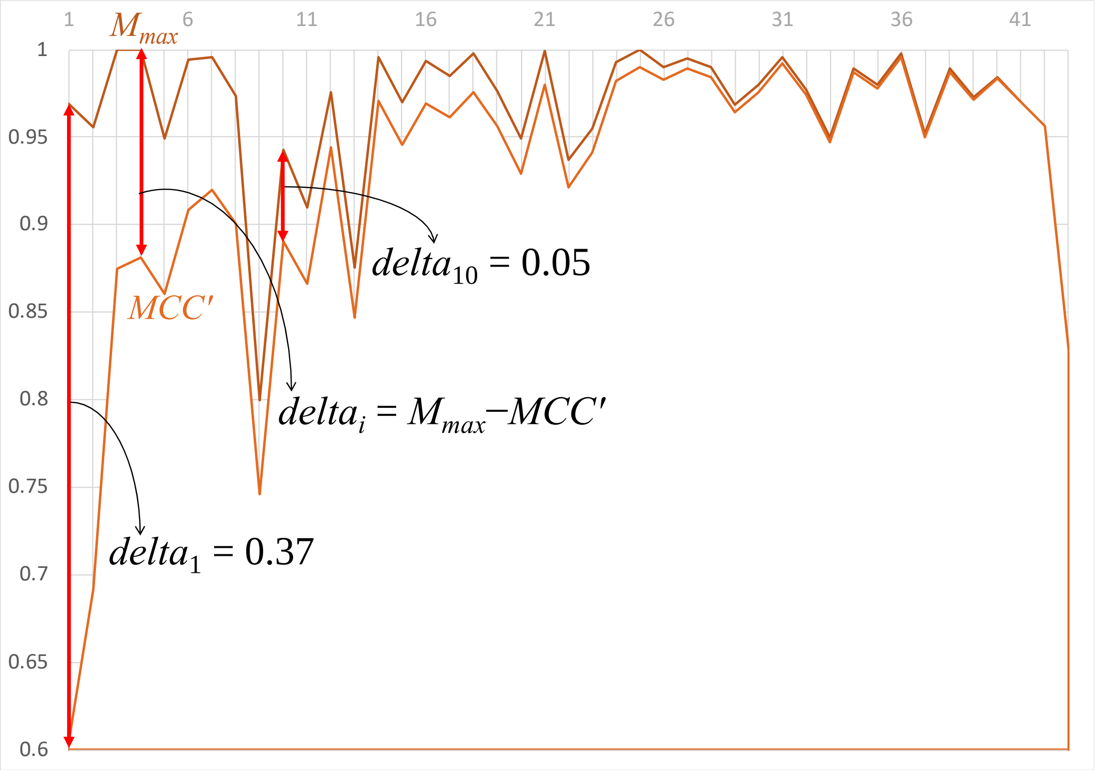


Figure H.1 Performance exaggeration via non-robust metrics reporting demonstrated on the surveyed 43 studies. Differences between the maximum of the metrics reported ($M_{\max}$, *e.g.*, *TPR* among *TPR* and *ACC*) and revealed *MCC'* (normalized *MCC*).

Confirmation bias may occur when evidence (*e.g.*, non-robust performance metrics) that supports one's preconceptions is evaluated more favorably than evidence that challenges these convictions (*e.g.*, robust metrics) (Porta, 2014, p. 54). The high expectations for an experiment can affect many phases including interpreting and reporting the results (van Wilgenburg & Elgar, 2013, p. 1).

This thesis provides a convenient method to investigate the presence of confirmation bias in ML-based classification studies in a broad range of application domains. It also has demonstrated that mobile malware detection studies seem to be prone to confirmation biases. It is expected that this method will be applied in different domains to see whether such biases exist.

VITA

Gürol CANBEK, was born in Malatya, owns a B.Sc. degree in Control and Computer Engineering from Istanbul Technical University in 1996, a M.Sc. degree in Computer Engineering from Gazi University in 2005. He had worked as a Software Specialist, Project Officer, Lecturer, Chief Software Engineer, Software Group Leader, Information Security Coordinator, Chief Cyber Security Engineer, and Senior Lead Engineer in Takasbank, Ministry of Defense, Inonu University, Elbit Systems, HAVELSAN, and ASELSAN. He had successfully taken part in several large-scale software projects, some of them international, in finance, avionics, and electronic warfare in his career.

His thesis for the degree of Master of Science was related to Activity Monitoring based on Keyloggers. He is the lead author of "Information and Computer Security: Spyware and Safeguarding Methods" book in Turkish and has published numerous peer-reviewed articles in optimization, information security, mobile/cyber security, and machine learning as seen below:

1. Canbek, G., Taskaya Temizel, T., & Sagiroglu, S. (2019). Multi-Perspective Analysis of Binary-Classification Performance Evaluation Instruments. *Information Processing and Management* (under review)

2. Canbek, G., & Sagiroglu, S. (2018). Akıllı Şebekelerde Stratejik Siber Güvenlik Bakışı {Strategic Cyber-Security Perspective in Smart Grids}. In *6th International Symposium on Digital Forensic and Security (ISDFS)* (pp. 1–6). Antalya, Turkey: IEEE. https://doi.org/10.1109/ISDFS.2018.8355346

3. Canbek, G., Sagiroglu, S., & Taskaya Temizel, T. (2018). New Techniques in Profiling Big Datasets for Machine Learning with a Concise Review of Android Mobile Malware Datasets. *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*, 117–121. https://doi.org/10.1109/ibigdelft.2018.8625275

4. Canbek, G. (2018). Cyber Security by a New Analogy: "The Allegory of the 'Mobile' Cave." *Journal of Applied Security Research*, *13*(1), 63–88. https://doi.org/10.1080/19361610.2018.1387838

5. Canbek, G., Baykal, N., & Sagiroglu, S. (2017). Clustering and visualization of mobile application permissions for end users and malware analysts. In *The 5th International Symposium on Digital Forensic and Security (ISDFS)* (pp. 1–10). Tirgu Mures: IEEE. https://doi.org/10.1109/ISDFS.2017.7916512

6. Canbek, G. (2017). Yeni siber düzen ve siber silahlanma: Ne yapılabilir? {New cyber order and cyber weaponry: What should we do?}. *Aljazeera Turk*. Retrieved Aug 15th, 2019 from http://www.aljazeera.com.tr/gorus/yeni-siber-duzen-ve-siber-silahlanma-ne-yapilabilir

7. Canbek, G., Sagiroglu, S., Taskaya Temizel, T., & Baykal, N. (2017). Binary classification performance measures/metrics: A comprehensive visualized roadmap to gain new insights. In *2017 International Conference on Computer Science and Engineering (UBMK)* (pp. 821–826). Antalya, Turkey: IEEE. https://doi.org/10.1109/UBMK.2017.8093539

8. Canbek, G. (2016). *Bilgi Varlıklarının Gizlilik Derecelerine Göre Sınıflandırılması Kriteri {Criteria for Classification of Information Assets According to Their Confidentiality Levels}* (No. TSE K 523 (ICS 01.040.35)). Ankara.

9. Canbek, G., Sagiroglu, S., & Baykal, N. (2016). New Comprehensive Taxonomies on Mobile Security and Malware Analysis. *International Journal of Information Security Science (IJISS), 5*(4), 106–138. Retrieved Aug 15th, 2019 from http://www.ijiss.org/ijiss/index.php/ijiss/article/view/227

10. Canbek, G. (2016). A Security-Privacy Story. Retrieved Aug 15th, 2019 from http://www.slideshare.net/gurol44/a-securityprivacy-story

11. Canbek, G. (2016). *Siber Güvenlik Panorama 2015 {Cyber Security Panorama 2015}. HAVELSAN Siber Güvenlik Bülteni* (Vol. 1). Ankara. https://doi.org/10.13140/RG.2.1.2367.7048

12. Canbek, G. (2015). *Neden Bilgi Güvenliği? {Why is Information Security?}*. Turkey: YouTube. Retrieved Aug 15th, 2019 from https://youtu.be/7HUudZgmSlk

13. Canbek, G. (2015). Siber savaşın eşiğinde: sıfırıncı gün {On the verge of cyber war: zero day}. *Aljazeera Turk*. Retrieved Aug 15th, 2019 from http://support.mendeley.com/customer/portal/topics/633601-copyright/articles

14. Canbek, G., Sağıroğlu, Ş., & Baykal, N. (2013). Bilgisayar Ağlarından Yazılıma: Bütüncül Siber Güvenlik Yaklaşımı {A Holistic Cyber Security Approach: from Computer Networks to Software}. In *The 1st International Symposium on Digital Forensics and Security (ISDFS)* (pp. 126–130). Elazig, Turkey. Retrieved Aug 15th, 2019 from https://www.researchgate.net/publication/303909609_Bilgisayar_Aglarindan_Yazilima_Butuncul_Siber_Guvenlik_Yaklasimi_A_Holistic_Cyber_Security_Approach_from_Computer_Networks_to_Software

15. Alkan, M., Atalay, A. H., Canbek, G., Bilirgen, C., İnceefe, M. A., Sağıroğlu, Ş., … Yazıcı, A. (2012). *Türkiye Ulusal Siber Güvenlik Stratejisi Önerisi {National Cyber Security Strategy Proposal of Turkey}*. Ankara. Retrieved Aug 15th, 2019 from https://www.researchgate.net/publication/303909502_Turkiye_Ulusal_Siber_Guvenlik_Stratejisi_Onerisi_National_Cyber_Security_Strategy_Proposal_of_Turkey

16. Sağıroğlu, Ş., & Canbek, G. (2009). Keyloggers - Increasing Threats to Computer Security and Privacy. *Technology and Society Magazine, IEEE, 28*(3), 10–17. https://doi.org/10.1109/MTS.2009.934159

17. Canbek, G., & Sağıroğlu, Ş. (2008). Casus Yazılımlar: Bulaşma Yöntemleri ve Önlemler {Spyware: Infection Methods and Preventive Measures}. *Journal of the Faculty of Engineering and Architecture of Gazi University*, *23*(1), 165–180. Retrieved from http://www.mmfdergi.gazi.edu.tr/2008_1/DERGI 2008 V23 NO1 _sayfa 165-180_.pdf

18. Canbek, G., & Sağiroğlu, Ş. (2008). Spyware: Infection methods and preventive measures. *Journal of the Faculty of Engineering and Architecture of Gazi University, 23*(1).

19. Canbek, G., & Sağıroğlu, Ş. (2008). Kişisel Gizlilik ve Yasal Düzenlemelere Kötücül Yazılımlar Açısından Bakış {A Perspective to Personal Privacy and Legal

Regulations in Terms of Malicious Software}. *Kara Harp Okulu Dergisi {The Journal of Defense Sciences}*, *7*(2), 119–139. Retrieved Aug 15th, 2019 from http://www.kho.edu.tr/akademik/enstitu/savben_dergi/72.htm

20. Canbek, G., & Sağıroğlu, Ş. (2007). Kötücül ve Casus Yazılımlar: Kapsamlı bir Araştırma {Malware and Spyware: A Comprehensive Review}. *Journal of the Faculty of Engineering and Architecture of Gazi University*, *22*(1), 121–136. Retrieved Aug 15th, 2019 from http://www.mmfdergi.gazi.edu.tr/article/view/1061000244

21. Canbek, G., & Sağıroğlu, Ş. (2007). Kötücül ve Casus Yazılımlara Karşı Elektronik İmzanın Sağlamış Olduğu Korunma Düzeyi {The Level of Protection of E-Signature against Malware and Spyware}. In E. Akyıldız, M. Alkan, & Ş. Sağıroğlu (Eds.), *Information Security & Cryptology Conference (ISCTurkey)* (pp. 263–269). Ankara. Retrieved Aug 15th, 2019 from https://www.iscturkey.org/bildiriler/2007/2007-39.pdf

22. Canbek, G., & Sağıroğlu, Ş. (2007). Bilgisayar Sistemlerine Yapılan Saldırılar ve Türleri: Bir İnceleme {Attacks against Computer Systems and Their Types: A Review Study}. *Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, *23*(1–2), 1–12. Retrieved Aug 15th, 2019 from https://dergipark.org.tr/download/article-file/252301

23. Canbek, G., & Sağiroğlu, Ş. (2007). Malware and spyware: A comprehensive review. *Journal of the Faculty of Engineering and Architecture of Gazi University*, *22*(1).

24. Canbek, G., & Sağıroğlu, Ş. (2007). Çocukların ve Gençlerin Bilgisayar ve İnternet Güvenliği {Computer and Internet Security for Children and Teenagers}. *Politeknik Dergisi*, *10*(1), 33–39. https://doi.org/10.2339/2007.10.1.33-39

25. Canbek, G., & Akcayol, M. A. (2006). Implementation of real-time optimization of page layout of internet newspaper using simulated annealing. *Journal of the Faculty of Engineering and Architecture of Gazi University*, *21*(2).

26. Canbek, G., & Sağıroğlu, Ş. (2006). İş Yeri Gözetleme ve Etkinlik İzleme Sistemleri {Workplace Surveillance and Activity Monitoring Systems}. *Standard, Economical and Technical Journal*, 102–109. Retrieved Aug 15th, 2019 from https://www.researchgate.net/publication/303907552_Is_Yeri_Gozetleme_ve_Etkinlik_Izleme_Sistemleri_Workplace_Surveillance_and_Activity_Monitoring_Systems

27. Canbek, G., & Sağıroğlu, Ş. (2006). *Bilgi ve Bilgisayar Güvenliği: Casus Yazılımlar ve Korunma Yöntemleri {Information and Computer Security: Spyware and safeguarding Methods}*. Ankara: Grafiker Yayıncılık. Retrieved from http://canbek.com/BBG/english.htm

28. Canbek, G., & Sağıroğlu, Ş. (2006). Bilgi, Bilgi Güvenliği ve Süreçleri Üzerine Bir İnceleme {A Review on Information, Information Security and Security Processes}. *Politeknik Dergisi*, *9*(3), 165–174. Retrieved Aug 15th, 2019 from http://www.politeknik.gazi.edu.tr/index.php/PLT/article/download/299/295

29. Canbek, G., & Akcayol, M. A. (2006). İnternet Gazetesi Sayfa Düzeninin Gerçek Zamanlı Eniyilemesinin Benzetilmiş Tavlama Algoritmasıyla Gerçekleştirilmesi {Implementation of Real-Time Optimization of Page Layout of Internet Newspaper using Simulated Annealing}. *Journal of the Faculty of Engineering and Architecture of Gazi University*, *21*(2), 341–348. Retrieved Aug 15th, 2019 from http://www.mmfdergi.gazi.edu.tr/article/view/1061001339