




Accuracy Barrier (*ACCBAR*): A novel performance indicator for binary classification

Gürol Canbek
Pointr /
Middle East Technical University
Ankara, Turkey
 [0000-0002-9337-097X](https://orcid.org/0000-0002-9337-097X)

Tugba Taskaya Temizel
Informatics Institute
Middle East Technical University
Ankara, Turkey
 [0000-0001-7387-8621](https://orcid.org/0000-0001-7387-8621)

Seref Sagiroglu
Computer Engineering Department
Gazi University
Ankara, Turkey
 [0000-0003-0805-5818](https://orcid.org/0000-0003-0805-5818)

Abstract—Although several binary classification performance metrics have been defined, a few of them are used for performance evaluation of classifiers and performance comparison/reporting in the literature. Specifically, *F1* and Accuracy (*ACC*) are the most known and conventionally used metrics. Despite their popularity and easy-to-understand characteristics, those metrics exhibit critical robustness issues. This paper suggests a new instrument category named ‘performance indicators’ and proposes a novel indicator named *accuracy barrier* (*ACCBAR* for short) that works to uncover confounding problems in performance reporting of *ACC* metric. The given case study in mobile malware classification, which is a domain of cyber security, has shown that the indicator gives an accurate interpretation of the results presented in terms of *ACC*. This study also recommends that researchers should use *ACCBAR* to eliminate potential publication or confirmation bias in classification performance evaluation.

Keywords—*performance evaluation, performance measures, performance indicators, publication bias, confirmation bias*

I. INTRODUCTION

Metrics or measures are important, particularly for comparison of the performance of different binary classifiers. However, those performance instruments may be limited in terms of interpretability by end-users or researchers. In particular, nonlinear or limitless measures such as Odds Ratio (*OR*) in $[0, \infty)$ are hard to interpret [1]. As examined from the general perspective by Texel [2], “measures”, “metrics”, and “indicators” refer to different but dependent concepts. In parallel with the semantic distinction among instruments [3], 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. Fig. 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 [4], [5].

Indicator is the new category of performance instruments as proposed and described in this and previous study [4]. Addressing a high-level research question “how to comprehend using, representing, reporting, learning, and teaching binary-classification performance instruments?”, this study also introduces a novel indicator (“Accuracy Barrier”) that specifically enhances usage and reporting of performance

instruments. Those enhancements are demonstrated via a case study where previously reported binary-classification performances in the literature are re-evaluated by the novel indicator.

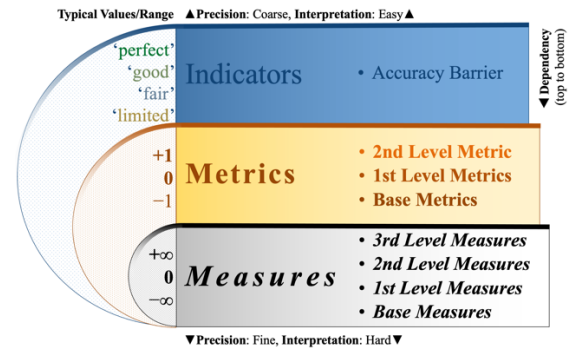


Fig. 1. Dependency and relative characteristics of performance evaluation instrument types. The attached semicircles on the left show the typical values or ranges for each instrument type. For binary-classification performance measures and metrics, the ranges are usually $[0, \infty)$ and $[0, 1]$ respectively whereas indicators have nominal values (see [4], [5] for more information about “measures” and “metrics”)

The rest of the paper is organized as follows. Section II introduces “indicators” as a new category for performance instruments apart from “performance measures” and “performance metrics”. Section III proposes a novel indicator named “Accuracy Barrier” (*ACCBAR*) as the first example of this new performance instrument category. Section IV defines *ACCBAR*. Section V provides a case study that re-evaluates the accuracy values of the published classification performance of 28 studies in the literature via the new performance indicator *ACCBAR*. The last section discusses the publication and confirmation biases in performance reporting and recommends adding *ACCBAR* to eliminate them. The section also summarizes the study and its contributions and provides future works.

II. PERFORMANCE INDICATORS AS A NEW CATEGORY OF PERFORMANCE INSTRUMENTS

In general, indicators facilitate the comprehension and comparison of the metrics and measures; therefore, they are recommended for reporting. The outputs of an indicator are mostly 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 [6], only one attempt of binary classification performance metric categorization is found where Cohen's Kappa (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 [7, p. 165] who stated that the divisions were arbitrary and provided for benchmarking.

III. FROM ACCURACY PARADOX TO ACCURACY BARRIER

Although it is one of the most popular metrics, Accuracy (ACC) results can be high even for a random classifier. Therefore, it is essential to define a minimum performance that should be expected from a binary classifier. Null Error Rate (NER) and No Information Rate (NIR), which are not well-known or reported [8, p. 35], [9, p. S9], are two measures that can be used to define that limit as shown in Eq. (2), NER specifies the minimum successful classification rate of a classifier without a classification model that always labels a given instance with “Negative”. 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” [10] from which this study introduces and formally defines the “Accuracy Barrier” indicator according to the following equations:

$$ACC \cong NIR \geq NER \quad (1)$$

$$\frac{TC}{Sn} \cong \frac{\max(P, N)}{Sn} \geq \frac{N}{Sn} \quad (2)$$

$$TC \cong \max(P, N) \geq N \quad (3)$$

Via Eq. (3), the accuracy barrier states a limit to overcome to avoid the accuracy paradox. A classifier with a reasonably high ACC where True Classification ($TC = TP + TN$) is close to the number of “Positives” ($TC \approx P$) or “Negatives” ($TC \approx N$) cannot overcome the Accuracy Barrier.

Table I 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). The table shows the confusion matrix (TP : true positives, FP : false positives, FN : false negatives, and TN : true negatives) as well as grand-truth class sizes (P and N : number of positives/negatives), classifier outcome

class sizes (OP and ON : number of outcome positives/negatives), and number true classifications (TC). When the performance is reported with only the ACC metric, both classifiers achieve notable performances (ACC values are 0.916 and 0.868). Nevertheless, their ACC s are very close to the ACC of an ordinary classifier (0.818) whose outcome is always “Negative” ($N \gg P$). In the first case, $ACC = 0.916$ but the classification is “very close” to Accuracy Barrier. In the second case, $ACC = 0.868$ but the classification “hit” the Accuracy Barrier. Note that when the classification performance is reported in terms of other metrics such as $F1$, CK , and Matthews Correlation Coefficient (MCC), the results are lower than ACC as shown in Table I.

TABLE I. ACCURACY BARRIERS AND OTHER METRICS ON TWO EXAMPLE HYPOTHETICAL CLASSIFIERS

Classifier-1:

Very close Accuracy Barrier

<i>OP</i>	<i>TP</i>	<i>FP</i>	base measures
265	240	25	
<i>ON</i>	<i>FN</i>	<i>TN</i>	
1935	160	1775	

<i>TC</i>	<i>P</i>	<i>N</i>
2015	400	1800

$TC \approx N$

<i>ACC</i>	Barrier	<i>NIR</i>	<i>NER</i>
0.916		0.818	0.818

$ACC \approx NIR$

	<i>CK</i>	<i>MCC</i>
<i>F1</i>	0.675	0.695
0.722	0.837	0.847 *

ACCBAR	<i>Very close</i>	Over Close Very close Not
--------	-------------------	---------------------------

ACCBAR delta (Δ) 0.098

Unit step length (θ) 0.05

Classifier-2:

Hit Accuracy Barrier

<i>OP</i>	<i>TP</i>	<i>FP</i>	base measures
190	150	40	
<i>ON</i>	<i>FN</i>	<i>TN</i>	
2010	250	1760	

<i>TC</i>	<i>P</i>	<i>N</i>
1910	400	1800

$TC \approx N$

<i>ACC</i>	Barrier	<i>NIR</i>	<i>NER</i>
0.868		0.818	0.818

$ACC \approx NIR$

	<i>CK</i>	<i>MCC</i>
<i>F1</i>	0.443	0.484
0.508	0.722	0.742 *

ACCBAR	<i>Hit</i>	Over Close Very close Not
--------	------------	---------------------------

ACCBAR delta (Δ) 0.050

Unit step length (θ) 0.05

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

IV. ACCURACY BARRIER (ACCBAR) INDICATOR

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 equations (4) and (5) calculate the proposed indicator called $ACCBAR$ along with the indicator categories. 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 Fig. 2 also depicts accuracy barrier categories for example delta values in the TasKar tool [11].

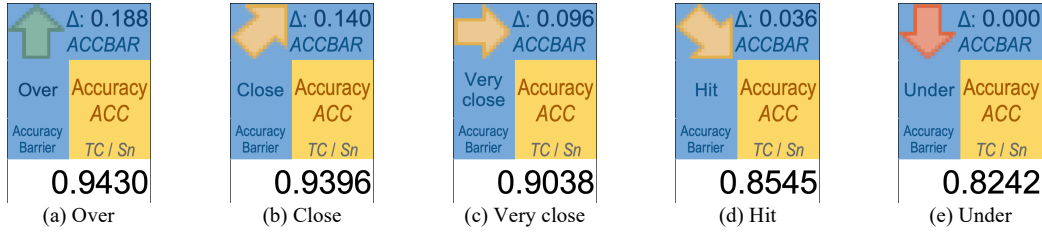


Fig. 2. Screenshots of the part in TasKar calculator showing the different accuracy barrier indicator categories (in (b) or (c), although Accuracy values are high 0.9396 and 0.9038, *ACCBAR* indicates that those classification applications “close” and “very close” to the accuracy barrier)

$$\Delta = ACC - \frac{\max(P, N)}{Sn} \quad (1)$$

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

ACCBAR can give notable insight into the classification performance by evaluating one metric (*ACC*) and one measure (*NIR*). The indicator is straightforward to calculate and clarify the vague condition interpretation of the Accuracy Paradox in the literature and provide an exact and explicit 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 I.

V. CASE STUDY: RE-EVALUATION OF CLASSIFICATION PERFORMANCE VIA *ACCBAR*

The ideal approach in ranking different classification studies for the same classification problem (e.g., machine learning (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 robust metric (e.g., *MCC* as suggested in [12]). However, this

approach could not be possible due to 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 datasets.

ACCBAR provides a means for checking classification performances expressed in terms of *ACC*. To show *ACCBAR* indicator usage, 28 surveyed studies that report their classification performances in terms of *ACC* are analyzed via *ACCBAR*. As there was more than one alternative classifier model published in most of the studies, the configurations yielding the highest *ACC* are chosen. The references of 28 studies are listed in Appendix A. The survey selection methodology was explained in [4].

A. Case Study Results

Table II 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. 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 the 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 #19 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,

TABLE II. PERFORMANCE RANKINGS OF CLASSIFICATIONS IN TERMS OF *ACC* METRIC ARE DIFFERENT WHEN *ACCBAR* IS TAKEN INTO ACCOUNT.

Sorted: <i>ACC</i> ↓ <i>ACCBAR</i> ↓, Δ↑, <i>ACC</i> ↓			Change at Rank bottom			Reported metrics/measures			#Study reference (see Appendix)		
<i>N</i>	<i>P</i>	<i>ACC</i>	Initial Rank	Δ Rank	change/ top	Δ	<i>ACCBAR</i>				
<small><i>ACC</i>: Accuracy, <i>AUC-ROC</i>: Area-Under-ROC-Curve, <i>BM</i>: Base Measures (<i>TP</i>, <i>FP</i>, <i>FN</i>, <i>TN</i>), <i>CK</i>: Cohen's Kappa, <i>F1</i>, <i>FNR</i>: False Negative Rate, <i>FPR</i>: False Positive Rate, <i>MCC</i>: Matthews Correlation Coefficient, <i>PPV</i>: Positive Predictive Value, <i>TNR</i>: True Negative Rate, <i>TPR</i>: True Positive Rate</small>											
8,000	400	0.9860	7	28	the last	-21	▼	0.03	Hit	<i>ACC</i> , <i>BM</i> , <i>TPR</i> , <i>FPR</i> , <i>PPV</i>	#8
99,037	10,581	0.9982	↓ 1	24	5th last	-23	▼	0.09	Very close	<i>TPR ACC</i> , , <i>FPR</i> , <i>F1</i>	#19
107,327	8,701	0.9949	3	26	3rd last	-23	▼	0.07		<i>ACC</i> , <i>TPR</i> , <i>FPR</i> , <i>PPV</i> , <i>F1</i>	#22
122,176	9,756	0.9906	4	27	2nd last	-23	▼	0.06		<i>ACC</i> , <i>TPR</i> , <i>FPR</i> , <i>PPV</i> , <i>F1</i> , <i>CK</i> , <i>MCC</i>	#6
1,853	6,909	0.8828	26	25	4th last	1	▼	0.09		<i>ACC</i> , <i>BM</i> , <i>TNR</i>	#20
9,804	2,794	0.9970	↓ 2	22	7th last	-20	▼	0.22	Over	Only <i>ACC</i>	#10
16,000	3,987	0.9900	5	23	6th last	-18	▼	0.19		<i>ACC</i> , <i>TPR</i>	#1
7,494	7,494	0.9890	6	1	first	5	▲	0.49		<i>ACC</i> , <i>FPR</i> , <i>FNR</i>	#2
1,260	1,260	0.9840	8	2	second	6	▲	0.48		<i>ACC</i> , <i>FPR</i>	#17
480	743	0.9787	9	17		-8		0.37		<i>ACC</i> , <i>TPR</i> , <i>PPV</i> , <i>F1</i>	#5
3,938	2,925	0.9750	10	13		-3		0.40		<i>ACC</i> , <i>TPR</i> , <i>TNR</i> , <i>FPR</i> , <i>FNR</i> , <i>PPV</i> , <i>AUC-ROC</i>	#11
12,026	5,264	0.9740	11	20		-9		0.28		<i>ACC</i> , <i>TPR</i> , <i>FPR</i>	#23
3,938	2,925	0.9720	12	14		-2		0.40		<i>ACC</i> , <i>TPR</i> , <i>TNR</i> , <i>FPR</i> , <i>FNR</i> , <i>AUC-ROC</i>	#25
1,250	610	0.9688	14	19		-5		0.30		<i>ACC</i> , <i>TPR</i> , <i>PPV</i> , <i>AUC-ROC</i>	#13
5,560	5,560	0.9688	13	3	third	10	▲	0.47	Only <i>ACC</i>	#4	

- 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**.

#10 study is reduced from 2nd position to 7th from last and #22 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 sorting instrument *ACCBAR* and the secondary sorting instrument *ACC* (*e.g.*, the sorting of #10, #1, #2, *etc.* studies in “Over” accuracy barrier) in Table II explain this condition. In the “Over” group, #10 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 the accuracy barrier) than the values of the preceding two studies (#2 and #17 with 0.49 and 0.48, respectively). Hence, the #2 and #17 studies are expected to be the first and second, respectively even though their *ACC*s were lower (*i.e.* the achieved accuracy can be considered more credible).

The possible reduction is not limited to the top-performing classifications. *ACCBAR* can also spot the underestimates in classification performance. Classification with a relatively lower *ACC* can move to the higher ranks as observed for the #4 study that goes up from rank 13th to 3rd via *ACCBAR* indicator correction. Note that the details and complete data are also provided online [dataset].

Online Material:

You can test different classification results and see the accuracy barrier outputs in the online extra material (including an open-source R script developed for *ACCBAR*) provided at <https://github.com/gurol/PToPI> [5] as well as using the developed calculator tool TasKar [11] provided online at <https://github.com/gurol/TasKar>. Case study data in Section V is also provided online [dataset].

VI. DISCUSSION AND CONCLUSION

The researchers might have not known robust metrics (such as *MCC* is the most robust metric as suggested in [12]) or the robustness issues of the metrics of choice (see [12] for Accuracy robustness issues). They also follow the conventions in choosing a performance metric for a specific application domain. 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 the academic publication process such as journals, reviewers, and editors [13, 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.

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) [13, p. 54]. The high expectations for an experiment can affect many phases including interpreting and reporting the results [14, p. 1].

As shown in Fig. 3, this study provides a novel performance indicator (*ACCBAR*) as a convenient method to investigate the presence of confirmation bias in ML-based classification studies in a broad range of application domains. The case study evaluated via the *ACCBAR* also has demonstrated that mobile malware detection studies appear to be prone to confirmation biases. It is expected that this method will be applied in different domains to see whether such biases do exist.

In a conclusion, this study

- suggests a new classification performance instrument category called “indicator”, and
- proposes a novel performance indicator named “Accuracy Barrier” (*ACCBAR*) to assess whether the performance of a classifier is close to ordinary classification (*e.g.*, labeling only positive or negative).

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* are misleading whereas the studies with lower *ACC*s had appeared to achieve more reliable performance.

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

- Accuracy barrier effect (as demonstrated in the case study)
- Presence of publication and confirmation biases (as discussed above)

in other classification domains in the literature.

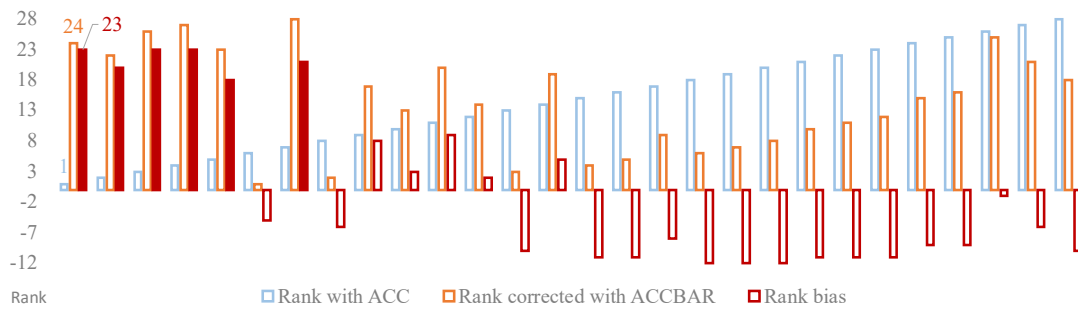


Fig. 3. Performance rank biases of the classification performances of 28 surveyed studies. Blue bars show the ranks among 28 studies calculated in terms of *ACC* values (1st: the best performance, 2nd, 3rd, ..., 28th: the worst performance). Orange bars depict the corrected rank via the proposed *ACCBAR* indicator. Red bars show the rank biases (corrected rank – published rank). For the top 5 performing classifications the biases are extreme whereas the least performing classifications at the right should be in the better ranks (*e.g.*, in the left most study, the classification rank is 1st in terms of *ACC* whereas it is 24th when corrected with *ACCBAR*.)

Despite the known and newly discovered robustness issues [12] of the Accuracy metric, researchers tend to use it in their performance reporting. Because end users are also familiar with the Accuracy metric. To avoid such biases and misleading results, we suggest the researchers add the *ACCBAR* category along with the *ACC* value when they publish and compare the classifier performances. For example, *ACC* = 0.944 (*ACCBAR* or accuracy is “over” the barrier).

Future work will evaluate the performance values of other metrics such as Balanced Accuracy (*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 II. Note that an open-source R script developed for *ACCBAR* is provided at <https://github.com/gurol/PToPI>. Another important matter to resolve for future studies is defining a performance indicator for limitless measures such as Discriminant Power (*DP*)

REFERENCES

- [dataset] G. Canbek, T. T. Temizel, and S. Sagioglu, “Binary-Classification Performance Evaluation Reporting Survey Data with the Findings”, Mendeley Data, V3, doi: 10.17632/5c442vbjzg.3
- [1] C. O. Schmidt and T. Kohlmann, “When to use the odds ratio or the relative risk?,” *Int. J. Public Health*, vol. 53, no. 3, pp. 165–167, 2008, doi: 10.1007/s00038-008-7068-3.
- [2] P. P. Texel, “Measure, metric, and indicator: An object-oriented approach for consistent terminology,” 2013, doi: 10.1109/SECON.2013.6567438.
- [3] G. Canbek, S. Sagioglu, T. T. Temizel, and N. Baykal, “Binary classification performance measures/metrics: A comprehensive visualized roadmap to gain new insights,” in *2017 International Conference on Computer Science and Engineering (UBMK)*, Oct. 2017, pp. 821–826, doi: 10.1109/UBMK.2017.8093539.
- [4] G. Canbek, “Multi-Perspective Analysis and Systematic Benchmarking for Binary-Classification Performance Evaluation Instruments,” Middle East Technical University, 2019.
- [5] G. Canbek, T. T. Temizel, and S. Sagioglu, “PToPI: A comprehensive review, analysis, and knowledge representation of binary classification performance measures/metrics (Accepted),” *SN Comput. Sci.*, 2022.
- [6] S. S. Mayya, A. D. Monteiro, and S. Ganapathy, “Types of biological variables,” *J. Thorac. Dis.*, vol. 9, no. 6, pp. 1730–1733, 2017, doi: 10.21037/jtd.2017.05.75.
- [7] J. R. Landis and G. G. Koch, “The Measurement of Observer Agreement for Categorical Data,” *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977, doi: 10.2307/2529310.
- [8] I. García-Magariño, L. Chittaro, and I. Plaza, “Bodily sensation maps: Exploring a new direction for detecting emotions from user self-reported data,” *Int. J. Hum. Comput. Stud.*, vol. 113, no. January, pp. 32–47, 2018, doi: 10.1016/j.ijhcs.2018.01.010.
- [9] R. R. Bond *et al.*, “Automation bias in medicine: The influence of automated diagnoses on interpreter accuracy and uncertainty when reading electrocardiograms,” *J. Electrocardiol.*, vol. 51, no. 6, pp. S6–S11, 2018, doi: 10.1016/j.jelectrocard.2018.08.007.
- [10] F. J. Valverde-Albacete and C. Peláez-Moreno, “100% classification accuracy considered harmful: The normalized information transfer factor explains the accuracy paradox,” *PLoS One*, vol. 9, no. 1, pp. 1–10, 2014, doi: 10.1371/journal.pone.0084217.
- [11] G. Canbek, T. Taskaya Temizel, and S. Sagioglu, “TasKar: A research and education tool for calculation and representation of binary classification performance instruments,” 2021, doi: 10.1109/ISCTURKEY53027.2021.9654359.
- [12] G. Canbek, T. Taskaya Temizel, and S. Sagioglu, “BenchMetrics: A systematic benchmarking method for binary-classification performance metrics,” *Neural Comput. Appl.*, vol. 33, no. 21, pp. 14623–14650, 2021, doi: 10.1007/s00521-021-06103-6.
- [13] M. Porta, Ed., *A Dictionary of Epidemiology*, 6th ed. New York: Oxford University Press, 2014.
- [14] E. van Wilgenburg and M. A. Elgar, “Confirmation Bias in Studies of Nestmate Recognition: A Cautionary Note for Research into the

Behaviour of Animals,” *PLoS One*, vol. 8, no. 1, pp. 1–8, 2013, doi: 10.1371/journal.pone.0053548.

APPENDIX A

The case study references of 28 classification applications (referred to as Study #1, #2, etc.) in Android mobile malware detection domain:

- #1 Y. Aafer, W. Du, and H. Yin, “DroidAPIMiner: Mining API-level features for robust malware detection in Android,” in *9th International Conference on Security and Privacy in Communication Networks (SecureComm)*, 2013, pp. 86–103.
- #2 S. Aonzo, A. Merlo, M. Migliardi, L. Oneto, and F. Palmieri, “Low-resource footprint, data-driven malware detection on Android,” *IEEE Trans. Sustain. Comput.*, vol. 3782, pp. 1–1, 2017, doi: 10.1109/TSUSC.2017.2774184.
- #3 A. M. Aswini and P. Vinod, “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)*, Feb. 2014, pp. 81–86, doi: 10.1109/ICADIWT.2014.6814679.
- #4 G. Canfora, A. De Lorenzo, E. Medvet, F. Mercaldo, and C. A. Visaggio, “Effectiveness of opcode ngrams for detection of multi family Android malware,” in *10th International Conference on Availability, Reliability and Security (ARES)*, 2015, pp. 333–340, doi: 10.1109/ARES.2015.57.
- #5 L. Deshotels, V. Notani, and A. Lakhota, “DroidLegacy: Automated familial classification of Android malware,” in *3rd ACM SIGPLAN on Program Protection and Reverse Engineering Workshop (PPREW)*, 2014, pp. 1–12, doi: 10.1145/2556464.2556467.
- #6 G. Kirubavathi and R. Anitha, “Structural analysis and detection of android botnets using machine learning techniques,” *Int. J. Inf. Secur.*, vol. 17, no. 2, pp. 153–167, 2018, doi: 10.1007/s10207-017-0363-3.
- #7 J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-an, and H. Ye, “Significant permission identification for machine learning based Android malware detection,” *IEEE Trans. Ind. Informatics*, vol. 14, no. 7, pp. 3216–3225, 2018, doi: 10.1109/TII.2017.2789219.
- #8 X. Liu and J. Liu, “A two-layered permission-based Android malware detection scheme,” in *2nd International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, Apr. 2014, pp. 142–148, doi: 10.1109/MobileCloud.2014.22.
- #9 Y. Lu, P. Zulie, L. Jingju, and S. Yi, “Android malware detection technology based on improved Bayesian classification,” in *The 3rd International Conference on Instrumentation, Measurement, Computer, Communication and Control (IMCCC)*, Sep. 2013, pp. 1338–1341, doi: 10.1109/IMCCC.2013.297.
- #10 F. Martinelli, F. Mercaldo, and A. Saracino, “BRIDEMAID: An hybrid tool for accurate detection of Android malware,” in *Asia Conference on Computer and Communications Security (ASIA CCS)*, 2017, pp. 899–901, doi: 10.1145/3052973.3055156.
- #11 I. Muttik, S. Y. Yerima, and S. Sezer, “High accuracy Android malware detection using ensemble learning,” *IET Inf. Secur.*, vol. 9, no. 6, pp. 313–320, 2015, doi: 10.1049/iet-ifs.2014.0099.
- #12 A. Narayanan, M. Chandramohan, L. Chen, and Y. Liu, “Context-aware, adaptive, and scalable Android malware detection through online learning,” *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 1, no. 3, pp. 157–175, 2017, doi: 10.1109/TETCI.2017.2699220.
- #13 N. Peiravian and X. Zhu, “Machine learning for Android malware detection using permission and API calls,” in *IEEE 25th International Conference on Tools with Artificial Intelligence (ICTAI)*, Nov. 2013, pp. 300–305, doi: 10.1109/ICTAI.2013.53.
- #14 M. Rahman, M. Rahman, B. Carbanar, and D. H. Chau, “Search rank fraud and malware detection in Google Play,” *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 6, pp. 1329–1342, 2017, doi: 10.1109/TKDE.2017.2667658.
- #15 B. Sanz, I. Santos, C. Laorden, X. Ugarte-Pedrero, P. G. Bringas, and G. Alvarez, “PUMA: Permission usage to detect malware in Android,” in *International Joint Conference CISIS-ICEUTE-SOCO Special Sessions*, 2013, pp. 289–298.
- #16 B. Sanz *et al.*, “MAMA: Manifest analysis for malware detection in Android,” *Cybern. Syst.*, vol. 44, no. 6–7, pp. 469–488, 2013, doi: 10.1080/01969722.2013.803889.

- #17 S. Sen, A. I. Aysan, and J. A. Clark, "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)*, 2018, pp. 255–270, doi: 10.1007/978-3-319-78816-6_18.
- #18 F. Shen, J. Del Vecchio, A. Mohaisen, S. Y. Ko, and L. Ziarek, "Android malware detection using complex-flows," in *37th International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 2430–2437, doi: 10.1109/ICDCS.2017.190.
- #19 G. Suarez-Tangil *et al.*, "DroidSieve: Fast and accurate classification of obfuscated Android malware," in *7th ACM Conference on Data and Application Security and Privacy (CODASPY)*, 2017, pp. 309–320, doi: 10.1145/3029806.3029825.
- #20 K. A. Talha, D. I. Alper, and C. Aydin, "APK Auditor: Permission-based Android malware detection system," *Digit. Investig.*, vol. 13, pp. 1–14, 2015, doi: 10.1016/j.diin.2015.01.001.
- #21 S. Wang, Q. Yan, Z. Chen, B. Yang, C. Zhao, and M. Conti, "Detecting Android malware leveraging text semantics of network flows," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 5, pp. 1096–1109, 2018, doi: 10.1109/TIFS.2017.2771228.
- #22 W. Wang, Y. Li, X. Wang, J. Liu, and X. Zhang, "Detecting Android malicious apps and categorizing benign apps with ensemble of classifiers," *Futur. Gener. Comput. Syst.*, vol. 78, pp. 987–994, 2018, doi: 10.1016/j.future.2017.01.019.
- #23 K. Xu, Y. Li, and R. H. Deng, "ICCDetector: ICC-based malware detection on Android," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 6, pp. 1252–1264, 2016, doi: 10.1109/TIFS.2016.2523912.
- #24 S. Y. Yerima, S. Sezer, and G. McWilliams, "Analysis of Bayesian classification-based approaches for Android malware detection," *IET Inf. Secur.*, vol. 8, no. 1, pp. 25–36, Jan. 2014, doi: 10.1049/iet-ifs.2013.0095.
- #25 S. Y. Yerima, S. Sezer, and I. Muttik, "Android malware detection using parallel machine learning classifiers," in *The 8th International Conference on Next Generation Mobile Apps, Services and Technologies (NGMAST)*, 2014, pp. 37–42, doi: 10.1109/NGMAST.2014.23.
- #26 S. Y. Yerima, S. Sezer, G. McWilliams, and I. Muttik, "A new Android malware detection approach using Bayesian classification," in *27th International Conference on Advanced Information Networking and Applications (AINA)*, 2013, pp. 121–128, doi: 10.1109/AINA.2013.88.
- #27 Z. Yuan, Y. Lu, and Y. Xue, "Droiddetector: Android malware characterization and detection using deep learning," *Tsinghua Sci. Technol.*, vol. 21, no. 1, pp. 114–123, 2016, doi: 10.1109/TST.2016.7399288.
- #28 Z. Yuan, Y. Lu, Z. Wang, and Y. Xue, "Droid-Sec: Deep learning in Android malware detection," in *ACM Conference on SIGCOMM*, 2014, pp. 371–372, doi: 10.1145/2619239.2631434.