Sequestration of imaging studies in MIDRC: controlling for ingenuous and disingenuous use of sequestered data

Dylan Tang,[1] Heather M. Whitney,[1] Kyle J. Myers,[2] Maryellen L. Giger[1]
[1]Department of Radiology, The University of Chicago, Chicago, IL, United States of America
[2]Puente Solutions, LLC, Phoenix, AZ, United States of America

ABSTRACT

Evaluation of AI/ML algorithm performance on a sequestered test set may lead to ingenuous and disingenuous use of the dataset, even though the data are not accessible to the developer. In the 'ingenuous' case, the resulting algorithm's performance metric, for example the area under the receiving operator curve (AUC) for a classification algorithm, may unintentionally overestimate or underestimate the true algorithm performance. A developer may also attempt to learn from the sequestered test set through attempting to repeatedly evaluate the algorithm on subsets of the test set, i.e., a 'disingenuous' use that may lead to algorithm overfitting of the test set. Creating a metric that can be used to 'dial in' ideal data set sampling to avoid each of these issues is an important area of investigation by the Medical Imaging and Data Resource Center (MIDRC, midrc.org). Building upon our prior work to address the ingenuous case, we also now address disingenuous use of the test set through a hash-table implementation that incorporates the Thresholdout$_{\text{AUC}}$ algorithm, and subsequently use the load factor metric to indicate overfitting to the test data. Furthermore, we devise analytical relationships between load factor and Thresholdout$_{\text{AUC}}$ budget. Notably, the relationship between load factor and budget is dependent on a noise rate parameter. We unify these methods with our previous findings for ingenuous use of sequestered data, specifically the relationship between AUC variability and load factor via the use case of a classifier trained to predict COVID-19 severity. The results show that while AUC standard error is inversely related to the load factor, the budget parameter from Thresholdout$_{\text{AUC}}$ is directly related to the load factor and noise rate. Thus, we anticipate using the load factor as a 'dial' that controls the number of test subsets eligible for evaluation. Specifically, if the developer requests to operate at a particular Thresholdout$_{\text{AUC}}$ budget, a specific load factor and noise rate combination can be determined that limits AUC variation while meeting budget demand.

Keywords: MIDRC, evaluation metrics, algorithm overfitting, AUC standard error, load factor

1. INTRODUCTION

The Medical Imaging and Data Resource Center (midrc.org) supports algorithm development and evaluation by partitioning the data into an open data commons and a sequestered data commons. Specifically, the open data commons can be used for algorithm training and validation, while the sequestered data commons is restricted and reserved for the fair testing by MIDRC of a previously trained algorithm. In the testing phase, the algorithm is evaluated on subsets of patient images (i.e., 'test subsets') drawn from the sequestered data commons, followed by reporting performance metrics (e.g., the area under the receiver operating characteristic curve, AUC, and its uncertainty) to the developer. Note that with the sequestered data commons, these test subsets on which the algorithm is evaluated on are inaccessible to the developer. The characteristics of potential test subsets, particularly the number of images and prevalence selected for each test subset and the number of potential test subsets, can be characterized through the load factor as described in prior work.[1] We hypothesize that the load factor, a metric that summarizes pairings between images and test subsets, can serve as a 'dial' by which MIDRC identifies test subsets eligible for algorithm evaluation, allowing the identification of the number of times the developer can "go back to the well" to retest their algorithm .

When reporting algorithm performance from evaluation on a single test subset, the performance metric on a single evaluation may overestimate or underestimate the algorithm's true performance. This could be considered 'ingenuous' use, which was the focus of our previous work.[1] Separately, it is possible for developers to aim to learn from the sequestered

Medical Imaging 2025: Image Perception, Observer Performance, and Technology Assessment,
edited by Mark A. Anastasio, Jovan G. Brankov, Proc. of SPIE Vol. 13409,
134090Y · © 2025 SPIE · 1605-7422 · doi: 10.1117/12.3047230

test set through repeat evaluation on subsets of the sequestered set, a 'disingenuous' use that could result in the developer learning from the sequestered data (i.e., "train to the test"). In this paper, we aim to (1) incorporate the load factor as a metric for test subset selection in the context of repeat evaluations that could be 'disingenous' by the developer and (2) combine our findings with our previous work, resulting in a single metric that can be used to design test subset sampling to control for both ingenuous and disingenuous use. To do this, we employ hash table principles[2], and specifically use the load factor as a metric for evaluating the potential for overfitting via the use-case of an image-derived classifier trained on COVID-19 severity. Furthermore, we adapt Thresholdout$_{\text{AUC}}$, an algorithm originally developed for adaptive machine learning tasks[3], to the hash table implementation to control for algorithm learning from the test and establish several analytical relationships between its parameters and load factor. Lastly, we unify the Thresholdout$_{\text{AUC}}$ algorithm with our previous work on 'ingenuous' use.

2. METHODS

2.1 Overview

Figure 1 outlines a 'roadmap' for controlling for both ingenuous and disingenuous use of sequestered data via characterizing test subsets using the load factor. The first step when a developer submits a classification algorithm for evaluation is to identify the size and class imbalance of the sequestered test dataset for the task (i.e., image types, patient inclusion, criteria, etc.). Then, the minimum test subset size for sufficient statistical power in differentiating the area under the receiver operating characteristic curve (AUC) from chance is determined[4]. Next, as described in the following section, the parameters of Thresholdout$_{\text{AUC}}$ are implemented and subsequently analyzed as a function of load factor. These include budget (the number of times a discrepancy between AUC of the test subset and the entire dataset is allowed to exceed a certain threshold), noise rate (which controls for the perturbation distribution of the test subset AUC), and statistical power. Lastly, these findings are combined with an evaluation of AUC variation based on mathematical derivation and clinical data to altogether identify a range of load factors that can minimize AUC variation (addressing ingenuous use) while controlling for the parameters in Thresholdout$_{\text{AUC}}$ (addressing disingenuous use). Note that in this study there is one image per patient.

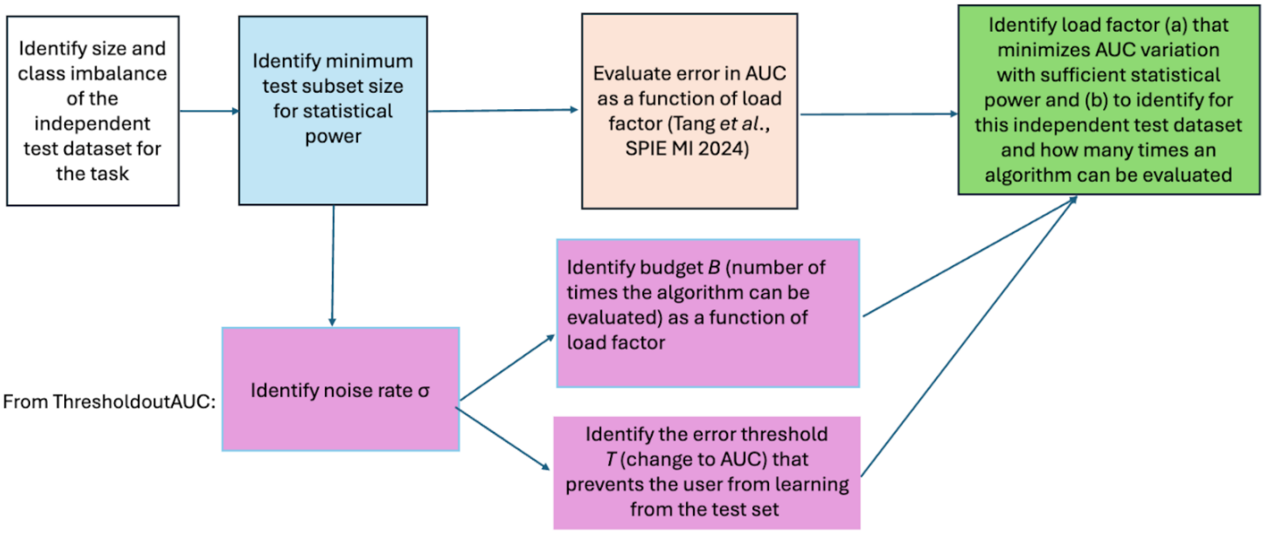
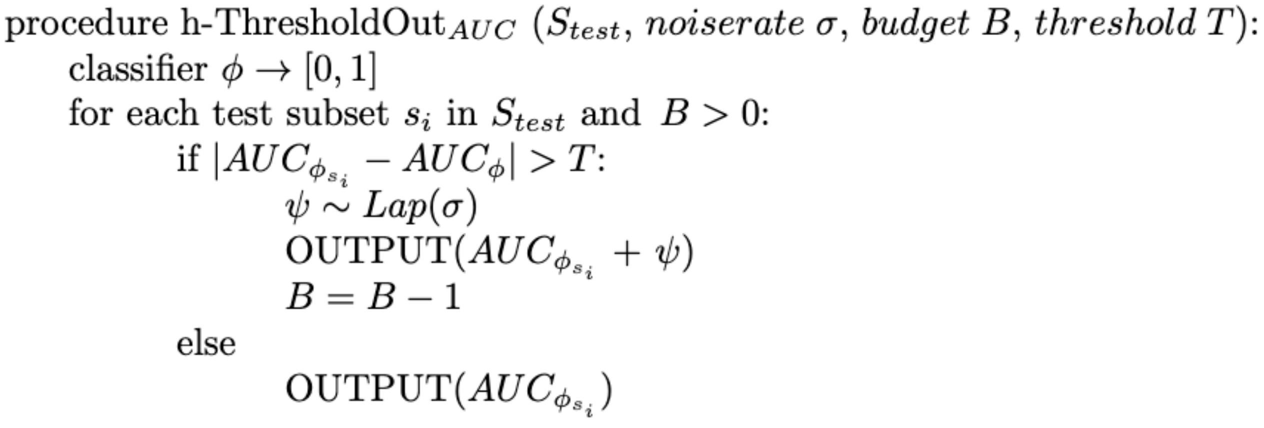

Identify size and class imbalance of the independent test dataset for the task
Identify minimum test subset size for statistical power
Evaluate error in AUC as a function of load factor (Tang et al., SPIE MI 2024)
Identify load factor (a) that minimizes AUC variation with sufficient statistical power and (b) to identify for this independent test dataset and how many times an algorithm can be evaluated
From ThresholdoutAUC:
Identify noise rate σ
Identify budget B (number of times the algorithm can be evaluated) as a function of load factor
Identify the error threshold T (change to AUC) that prevents the user from learning from the test set

Figure 1: Roadmap identifying a load factor that minimizes the potential for both ingenuous and disingenuous misuse of sequestered datasets. The tan-colored box denotes tasks related to controlling for ingenuous use, and pink-colored boxes denote tasks related to controlling for disingenuous use.

2.2 Procedures

First, the minimum test subset size for statistical power (β) to distinguish AUCs from chance from the sequestered dataset is determined at a given prevalence of the minority class. This sets a bound on the minimum dataset size that can be used for all subsequent steps. Then, a hash table is used to implement Thresholdout$_{\text{AUC}}$, which takes in threshold *T*, budget *B*, perturbation random noise ψ, and noise rate σ. A hash table was used to track images (identified by image IDs) and test subsets (identified by test subset IDs). The hash table can thus be understood as a mapping of test subset IDs to image IDs, where the image IDs serve as the keys that make the test subset IDs (which constitute the values) accessible.

The modified version of Thresholdout$_{\text{AUC}}$, which we call h-Thresholdout$_{\text{AUC}}$, is implemented as follows: (1) N images are randomly selected per test subset by image ID at the specified prevalence, . Then, the hash table is inverted (thus mapping test subsets to images), and (4) for each test subset s_i in the hash table, if the absolute difference between the AUC performance of the test subset s_i and the AUC performance of the entire dataset exceeds a threshold *T*, then the AUC of s_i is perturbed by random noise ψ before being reported to the developer, and the budget is decremented. The budget is defined as the number of times the absolute difference between AUC of the test subset can exceed the AUC of the entire dataset by threshold *T*, with respect to the load factor. ψ is randomly sampled from a Laplace probability density function that depends on noise rate σ, similar to the procedure in Thresholdout$_{\text{AUC}}$[3]. (5) If the absolute difference between AUC performance of s_i and the AUC performance of the entire dataset does not exceed *T*, the AUC performance of s_i is reported with no perturbation. Pseudocode for the h-Thresholdout$_{\text{AUC}}$ used in this work is shown in Figure 2.



```
procedure h-ThresholdOut$_{AUC}$ ($S_{test}$, $noiserate\ \sigma$, $budget\ B$, $threshold\ T$):
    classifier $\phi \rightarrow [0,1]$
    for each test subset $s_i$ in $S_{test}$ and $B > 0$:
        if $|AUC_{\phi_{s_i}} - AUC_{\phi}| > T$:
            $\psi \sim Lap(\sigma)$
            OUTPUT($AUC_{\phi_{s_i}} + \psi$)
            $B = B - 1$
        else
            OUTPUT($AUC_{\phi_{s_i}}$)
```

Figure 2: Pseudocode detailing the h-Thresholdout$_{\text{AUC}}$ procedure, modified for the hash table implementation.

2.3 Analytical Relationship between Load Factor and Budget

An analytical expression of budget *B* in terms of load factor can be derived. From the hash table implementation, the load factor is a ratio of values to keys and serves as a metric that describes the average number of images drawn per test subset. Specifically, given the number of images in each test subset (N), the number of test subsets (ρ), and the number of images in the dataset (φ), the load factor ξ can be written as: $\xi = \frac{N\cdot\rho}{\phi}$. Adapting an expression from Dwork *et al.*[5], given a relevant noise rate (a scaling parameter that controls for the variation in error), the number of images in the test subset, and the probability that an algorithm overfits a random dataset $\gamma = e^{2\sigma^2\phi}$, then Thresholdout$_{\text{AUC}}$ satisfies $\left(\epsilon = \frac{B}{\sigma\phi},\ \delta = 0\right)$-if the budget is set according to $B = \frac{\sigma^5\left(\frac{\xi\phi}{\rho}\right)^2}{512\left(\ln(8)+2\sigma^2\left(\frac{\xi\phi}{\rho}\right)\right)}$ (Eq. 3).

An algorithm M satisfies $(\epsilon,\ \delta)$-differential privacy if given adjacent datasets (datasets which differ by at most one entry) *x*, *y*, and $\forall S \subseteq Range(M)$, then $\Pr(M(x) \in S) \ \leq e^{\epsilon} \Pr(M(y) \in S) \ + \delta$.[6] This relationship between load factor and budget can ultimately serve as a potential upper bound for the number of test subsets the developer can request evaluations for their algorithm.

2.4 Clinical Use Case

A dataset of 1047 deidentified chest radiograph (CXR) images from COVID-19+ patients had been acquired between March and September of 2020. Each image was associated with the corresponding status of the patient as admitted to the intensive care unit (ICU) or not within 24 hours of imaging. A previously-trained deep learning/artificial intelligence model using DenseNet121 architecture had been used to predict patients' need for intensive care within 24 hours of imaging. The AUC performance of the model on the entire dataset was previously reported to be 0.81.[7] A value of σ = 0.124 was determined by setting the full width half maximum (FWHM) of the Laplace distribution equal to the endpoints of an *a posteriori* bootstrapped 95% CI for AUC of the entire dataset. For this study, an error threshold T = 0.02 was used, based on the simulation studies of Gossman *et al*.[3] To allow for the construction of test subsets of larger sizes, sampling was conducted with replacement.

3. RESULTS

Figure 3 shows the relationship between power of the test, load factor, and number of test subsets as derived in Section 2.3. Load factor was varied from values 0 to 2, and the number of test subsets was varied from values 0 to 100 subsets. The corresponding test subset size was calculated for ten randomly selected points. For example, to achieve statistical power greater than 0.7 at a load factor of 1.6, then at least 73 test subsets are needed of size n = 23, i.e., 23 images each.

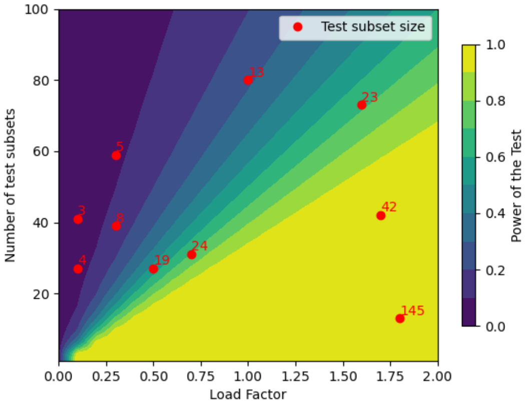


Figure 3: Contour plot depicting the analytical relationship between load factor, number of test subsets and power of the test. The ten randomly selected combinations of load factor and number of test subsets are shown as red data points. This figure demonstrates that choosing to operate at a specific number of test subsets and load factor can control for statistical power.

Figure 4 plots the absolute difference between AUC of the specific test subset and the AUC of the entire dataset, visualizing instances where the difference exceeds an error threshold of $T = 0.02$ for two load factors, 10 and 70. This comparison is repeated for 100 test subsets, where each subset is distinguished by a unique test subset ID from 1 to 100. When the difference exceeds the error threshold, the random noise ψ is randomly selected from a Laplace distribution with noise rate σ = 0.124, as consistent with the procedure described in Section 2.1.

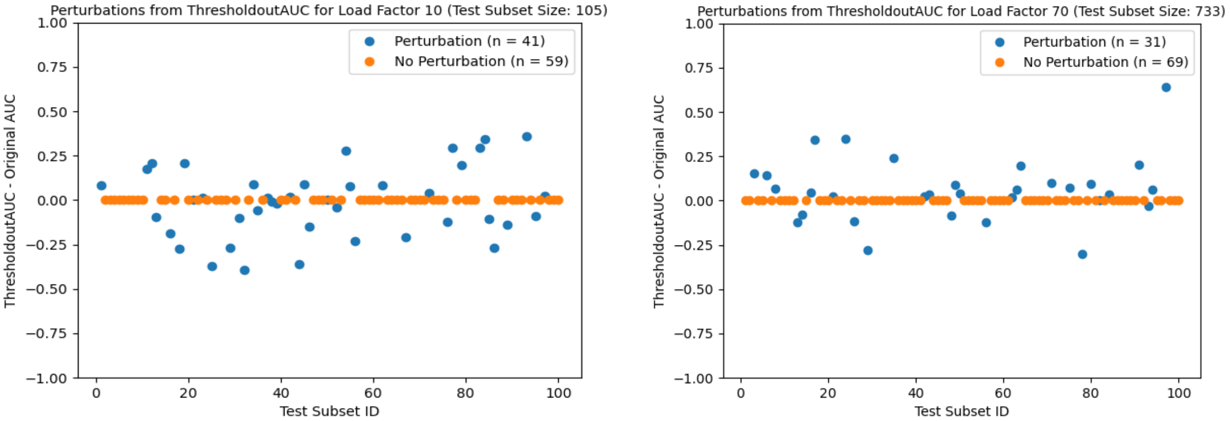


Figure 4: Plot depicting the perturbation to AUC for each test subset constructed at load factors 10 (left) and 70 (right). Test subsets with no added perturbation from Thresholdout$_{\mathrm{AUC}}$ are shown in orange, while test subsets with added perturbation are shown in blue. The y-axis measures the direction and magnitude of the perturbation.

Figure 5 shows the relationship between variation of AUC and h-Thresholdout$_{\mathrm{AUC}}$ budgets evaluated at noise rates σ = 0.4, 0.6, 0.8, and 1.0. The AUC interquartile ranges (IQRs) were calculated from load factors 10 to 590 in increments of 20 following a procedure similar to the previous study. The analytical expression for AUC standard error (SE) had also been previously determined.[1,8] Thus, if the developer requests to operate at a given h-Thresholdout$_{\mathrm{AUC}}$ budget, a particular load factor and noise rate combination can be determined that controls for AUC variation while meeting the budget demand.

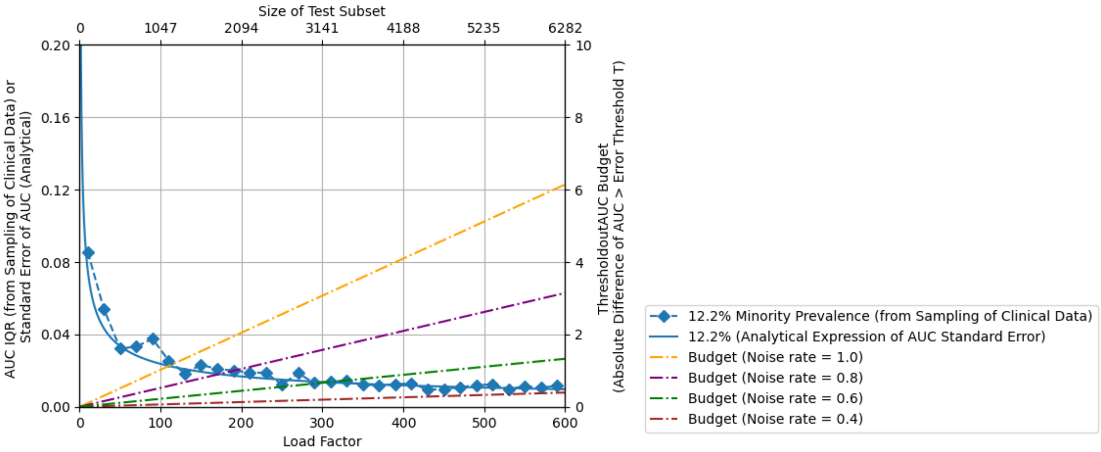


Figure 5: AUC interquartile range (solid blue), standard error of AUC (dashed blue), and analytical expressions for h-Thresholdout$_{\mathrm{AUC}}$ budget for four different noise rates.

In our analysis of the clinical dataset, a load factor of 400 was used with 100 test subsets, which resulted in a test subset size of 4188 images. Images were sampled with replacement. The AUC standard error corresponding to this load factor was 0.012, and the allocated Thresholdout$_{\text{AUC}}$ budget was determined to be 2.0.

Figure 5 combines the two endpoints by firstly showing (1) the analytical relationship between load factor and standard error of AUC and the empirical relationship between load factor and AUC interquartile ranges from sampling of clinical data. The figure also depicts (2) the analytical relationship between load factor, budget, and the noise rate parameter. Thus, if the developer chooses to operate at a particular Thresholdout$_{\text{AUC}}$ budget for a given noise rate, an appropriate load factor that minimizes variation in AUC can subsequently be determined.

4. DISCUSSION

In this work, we have shown that through the mapping of images to test subsets via a hash table implementation, load factor can serve as a metric that supports the establishment of task-based test subsets such that (1) the potential for over-/under-estimation of algorithm performance (ingenuous use) is minimized and (2) the potential overfitting to the test when repeated evaluations of the algorithm are conducted (disingenuous use) is also minimized. In regards to the first endpoint, an analytical relationship between load factor and standard error (SE) of AUC was established and subsequently compared to AUC interquartile ranges (IQRs) calculated from test subsets stored in the hash table, repeated at various load factors.

For the second endpoint, a series of benchmarks were established that integrated Thresholdout$_{\text{AUC}}$, an algorithm devised by Gossman *et al*. The implementation of Thresholdout$_{\text{AUC}}$ in this work, while similar to previous works, was modified for the testing of an algorithm on a fixed dataset, rather than adaptive machine learning with a varying test set. Once sufficient statistical power is reached, Thresholdout$_{\text{AUC}}$ is evaluated for each test subset in the hash table. The hash table is constrained by the analytical relationship between load factor and Thresholdout$_{\text{AUC}}$ budget (Eq. 3). The load factor is also constrained by the noise rate parameter σ, which is the scaling factor of the Laplace distribution from which the perturbation ψ is sampled.

One limitation to the present study is that the analytical relationship between Thresholdout$_{\text{AUC}}$ budget and load factor may place overly strict constraints on either the number of test subsets or the size of the test subsets required. For example, given noise rate $\sigma = 1$, a load factor of 600 is required for a budget of 6. Then, if there are 100 test subsets sampled from 1047 chest radiograph images (as consistent with the clinical study), each test subset size must contain 6282 entries. For datasets with a small number of images, this requirement on test subset size may be difficult to satisfy. One possible solution would be to increase the number of test subsets required to achieve a specific load factor, ensuring that the size of each test subset is smaller. Another possible solution, which has been explored by Gossman *et al.*[9], is to relax the (ϵ, δ)-differential privacy constraint that dictates the relationship between Thresholdout$_{\text{AUC}}$ and load factor. This is a potential future direction for subsequent analyses.

5. CONCLUSION

In this paper, we extend the hash table data structure (a mapping from images in the test set to test subsets) to account for repeat algorithm evaluations of the test set via an implementation of Thresholdout$_{\text{AUC}}$. We propose that the load factor metric can control for several key parameters, notably power of the test and budget of Thresholdout$_{\text{AUC}}$. Lastly, by combining these findings with an analytical relationship between load factor and AUC standard error, load factor can serve as a parameter that measures the number of evaluations in which additional perturbation is added to the outputted AUC while controlling for over-/under-estimation of algorithm performance.

6. ACKNOWLEDGEMENTS

MIDRC is supported by the National Institutes of Health, National Institute of Biomedical Imaging and Bioengineering contract 75N92020D00021, and by ARPA-H. D.T. is additionally supported by the Jeff Metcalf Internship Program at the University of Chicago.

7. REFERENCES

[1] Tang, D., Whitney, H. M., Myers, K. J. and Giger, M., “Sequestration of imaging studies in MIDRC: using load factor to minimize algorithm performance overestimation and image reuse,” Med. Imaging 2024 Image Percept. Obs. Perform. Technol. Assess., Y. Chen and C. R. Mello-Thoms, Eds., 17, SPIE, San Diego, United States (2024).

[2] Maurer, W. D. and Lewis, T. G., “Hash Table Methods,” ACM Comput. Surv. **7**(1), 5–19 (1975).

[3] Gossmann, A., Pezeshk, A. and Sahiner, B., “Test data reuse for evaluation of adaptive machine learning algorithms: over-fitting to a fixed ‘test’ dataset and a potential solution,” Med. Imaging 2018 Image Percept. Obs. Perform. Technol. Assess., R. M. Nishikawa and F. W. Samuelson, Eds., 19, SPIE, Houston, United States (2018).

[4] Zhou, X.-H., Obuchowski, N. A. and McClish, D. K., [Statistical Methods in Diagnostic Medicine, 2nd Edition].

[5] Dwork, C., Feldman, V., Hardt, M., Pitassi, T., Reingold, O. and Roth, A., “The reusable holdout: Preserving validity in adaptive data analysis,” Science **349**(6248), 636–638 (2015).

[6] Aitsam, M., “Differential Privacy Made Easy,” arXiv:2201.00099 (2021).

[7] Li, H., Drukker, K., Hu, Q., Whitney, H. M., Fuhrman, J. D. and Giger, M. L., “Predicting intensive care need for COVID-19 patients using deep learning on chest radiography,” J. Med. Imaging **10**(04) (2023).

[8] Hanley, J. A. and McNeil, B. J., “The meaning and use of the area under a receiver operating characteristic (ROC) curve.,” Radiology **143**(1), 29–36 (1982).

[9] Gossmann, A., Pezeshk, A., Wang, Y.-P. and Sahiner, B., “Test Data Reuse for the Evaluation of Continuously Evolving Classification Algorithms Using the Area under the Receiver Operating Characteristic Curve,” SIAM J. Math. Data Sci. **3**(2), 692–714 (2021).