ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10889646

A Unified Metric for Simultaneous Evaluation of Error Rate and Annotation Cost

Mark Lindsey
Content Analytics Division
Probity, Inc.
Herndon, VA, USA
mlindsey@probity.com

Francis Kubala
Content Analytics Division
Probity, Inc.
Herndon, VA, USA
fkubala@probity.com

Richard M. Stern
Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA, USA
rms@cmu.edu

***Abstract*—Pattern classification systems have traditionally been trained using a set of labeled training data and subsequently evaluated using different testing data. The cost of labeling the training data is typically substantial. Online Human-In-The-Loop (HITL) algorithms present an alternate approach that enables useful classification for many real-world applications using much less labeled data. These classifiers begin with a very small amount of training data and iteratively improve their performance by labeling a selected small number of utterances manually. Unfortunately, there is no unified evaluation metric that considers both classifier performance and annotation cost, which makes it difficult to evaluate these algorithms objectively. Furthermore, the lack of such a metric restricts the evaluation of online learning algorithms to prequential evaluation (before the classifier is adapted to the newly-labeled evaluation data), which does not realistically reflect the algorithm's ability to adapt to the data stream in real time. This paper introduces the Interactive Machine Learning Metric (IMLM), a new unified evaluation metric that makes the combination of performance and annotation cost for binary classification tasks far less arbitrary. This metric is well suited for the evaluation of online HITL algorithms and also allows for fair comparison of different algorithms after adapting to the evaluation data. The value and appropriateness of IMLM is demonstrated by evaluating a series of Online Active Learning algorithms on a Spoken Language Verification task.**

***Index Terms*—online learning, human-computer interaction, active learning, detection tasks, evaluation metrics.**

I. Introduction

Conventional Machine Learning (ML) algorithms learn from a set of manually annotated training data and are subsequently evaluated on test data or deployed for operational use. In many use cases, annotation of sufficiently large training corpora is costly, and the algorithm struggles to generalize to unseen patterns that appear in the test set. Online Human-In-The-Loop (HITL) algorithms address both of these issues with conventional ML approaches by learning incrementally from minimal annotated samples pulled directly from the evaluation data stream, where the annotations are provided by a human operator who is also a domain expert.

Examples of existing online HITL algorithms include Online Active Learning (OAL) [1], Online Reinforcement Learning [2], Online Apprenticeship Learning [3], and many others. In OAL, an algorithm identifies a small subset of incoming samples from the data stream that have been estimated to be the most informative, and then queries the human user for the associated class labels. The machine then learns from these labeled samples, and the process is repeated as more data from the stream is presented. Online Reinforcement Learning and Online Apprenticeship Learning also operate by obtaining information from a human about the data stream, but the machine learns directly from provided environmental conditions or demonstrated behavior rather than explicitly annotated samples to adapt to incremental changes. These algorithms have been applied to a number of Natural Language Processing systems [4] including dialogue learning [5]–[7] and topic modeling [8], [9], spam filtering [10], network protocol identification [11], object classification [12], and various detection and monitoring applications [13]–[16].

Funding provided by Probity, Inc. Much of this work was completed as part of the first author's doctoral thesis in the Department of Electrical and Computer Engineering at Carnegie Mellon University.

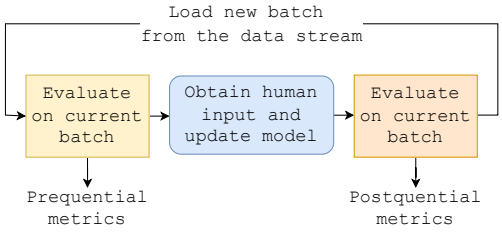

Fig. 1. The order of prequential evaluation and postquential evaluation in the pipeline of a typical online HITL algorithm.

Online HITL algorithms are powerful tools for classifying massive data streams. As the data streams grow in volume and complexity, human input becomes more valuable as a reliable source of knowledge and course correction for the algorithm. Human input for large data streams comes at a high cost, however. Therefore, the primary goal in engineering online HITL algorithms is to train high-performance classifiers that learn from a small number of human annotations. In other words, efficient online HITL algorithms jointly minimize classification error and annotation cost over the data stream.

One major issue in developing online HITL algorithms is the lack of a metric that can measure this joint minimization. Of the many existing metrics that measure classification performance (including classification error, precision and recall, the F-measure [17], the Receiver Operating Characteristic (ROC) and the Area Under the [ROC] Curve (AUC) [18], along with cost functions like the Detection Cost Function (DCF) [19]), none consider the effort required of the human in the loop to achieve the performance being evaluated. Devising an applicable metric is a challenging problem here because annotation cost is usually measured in terms of time or number of samples rather than a cost or a rate. This makes it difficult to determine whether an algorithm that achieves a lower error rate using more annotations is more optimal than an alternative algorithm that sacrifices accuracy for a lower annotation cost.

Another issue in evaluating online algorithms is the question of which point in the training pipeline is better for the evaluation to take place (see Fig. 1). In many cases, evaluation is completed before the model adapts to the most recent batch from the data stream. This form of "prequential" evaluation is done to avoid mixing training data with testing data [20]. While this approach is well motivated, it does not reflect the real operational ability of the algorithm to adapt to the data stream.

In this paper, we define the Interactive Machine Learning Metric (IMLM) [**?**], a new cost function that considers both the error rate and annotation cost for detection and verification tasks. IMLM reduces the arbitrariness associated with representing the impact of these two very different attributes of HITL decision making by viewing both in terms of time cost. This general approach allows for fair comparison of algorithms that are trained on a non-identical subset of annotated samples from the data stream. We show how IMLM can be used to compare the efficiency of different online algorithms using an illustrative example. In the example, we apply OAL algorithms to the Spoken Language Verification (SLV) task and demonstrate the use of IMLM and "postquential" evaluation to determine which algorithm is objectively the most efficient.

This paper is focused specifically on the problem of detecting events in long-running data streams while also leveraging the domain knowledge of a human operator who benefits from the automatic detection process. We also focus on metrics that are operationally realistic in the sense that they force an algorithm to choose a specific set of operating parameters without explicit knowledge of how the parameters will affect performance, unlike metrics like ROC, AUC, or Average Precision. One existing evaluation metric that is operationally realistic and intended for detection tasks is DCF, which requires a set of predictions from the classifier in order to calculate false negative and false positive rate. However, DCF does not consider the cost of requiring the human operator to provide annotations throughout the data stream, which is key for evaluating HITL algorithms for long data streams. IMLM directly accounts for this cost in its formulation, giving it a unique advantage over other existing evaluation metrics. We are not aware of any other metric that combines the cost of annotation with a metric tied to error rate.

II. The Interactive Machine Learning Metric

The challenge of creating a single unified metric for online HITL detection algorithms lies in justifying the combination of two values from different modalities. Our approach to this issue is to view both annotation cost and detection errors in terms of *time cost*.

Annotation cost is already typically measured in terms of time, where each sample takes some amount of time, a, to annotate. For example, to annotate a five-second segment of audio as either speech or non-speech for the Speech Activity Detection (SAD) task, an annotator would have to listen to that whole segment and make a quick binary annotation. This means a would be 5 seconds at the least. So, to annotate a whole dataset without the aid of an algorithm, the required time cost would be C_b in Eq. 1, where N is the total number of samples in the dataset.

$$C_b = a \cdot N \tag{1}$$

Converting false positive errors to units of time is also straightforward. In the context of detection and validation tasks, the value of an algorithm can be measured by how much time the algorithm saves. For every non-target sample (i.e., not an event of interest) correctly classified by the algorithm, the time spent by the user observing unimportant data is reduced. Conversely, for every false positive prediction, the user expends a similar amount of effort as required to annotate the sample (i.e., the user observes the sample and moves on). Thus, the cost of a false positive error is also a.

False negative errors are often much more costly than false positives in detection and verification tasks because of the relative rarity and importance of the target class. The time cost of false negatives could be viewed as the effort required to circle back and find the target samples manually, or as the time spent dealing with the real-world harm of missing the information in a target sample. Here, the time cost can be difficult to quantify, so, for simplicity, we assume that false negatives are q times more costly than false positives on average, $q \cdot a$. Putting all these costs together, we can express the total time cost after applying an algorithm as C_a below. Here, N_{ann} is the number samples labeled for the purpose of training or fine-tuning the algorithm, and N_{fp} and N_{fn} are the number of false positive and false negative errors, respectively.

$$C_a = a(N_{\text{ann}} + N_{\text{fp}}) + qaN_{\text{fn}} \tag{2}$$

Because of variable dataset sizes, it is useful to express the time cost of a detection algorithm as the relative cost savings compared to manual analysis. Thus, IMLM can be defined and simplified as in Eq. 3.

$$\text{IMLM} = \frac{C_a}{C_b} = \frac{a(N_{\text{ann}} + N_{\text{fp}}) + qaN_{\text{fn}}}{aN}$$

$$\text{IMLM} = \frac{N_{\text{ann}} + N_{\text{fp}} + qN_{\text{fn}}}{N} \tag{3}$$

A. Determining the Cost of a False Negative Error

The time cost of a false negative error can vary in severity based on the application. As such, the value of q should be set to the needs of the user. If it is desired to match the typical definition of DCF where false negatives are three times as expensive as false positives [21], one could set $q = 3$. However, it should be noted that this DCF weighting was determined "at the request of the participants"[1] and is just as arbitrary as any other weighting. As a less arbitrary alternative, we provide here a logically-motivated rule of thumb for setting q based on the prevalence of the target class.

In many applications, false negatives become more costly as the target class becomes more scarce. This is because a rare class is difficult to find (manually or algorithmically), so circling back to find missing information becomes especially time consuming. This characteristic of detection tasks can be reflected in the weight of false negative errors by setting it to a ratio of target class prevalence, as shown in Eq. 4.

$$q = \frac{N}{N_{\text{target}}} \tag{4}$$

Substituting Eq. 4 for q in Eq. 3 and simplifying yields

$$\text{IMLM} = \frac{N_{\text{ann}} + N_{\text{fp}}}{N} + \frac{N_{\text{fn}}}{N_{\text{target}}}. \tag{5}$$

Of course, there are situations in which false positive errors are more detrimental than false negatives (e.g. Automatic Speaker Verification [22]), or where the target class is more prevalent than the non-target class. In these cases, q should be adjusted to meet the needs of the task.

B. Intuitive Interpretation

Eq. 5 is an intuitively satisfying representation of relative time cost reduction. Since the IMLM is formulated as a ratio, the range is $[0, \infty)$. (Note that the value can approach infinity if a large volume of annotations outside of the data stream are used for pre-training.) If the resulting value is less than 1, the algorithm saved time; if it is greater than or equal to 1, the algorithm was no better than manually reviewing the whole data stream. Furthermore, the fractions in Eq. 5 are nearly identical to the false negative and false positive rate

[1] Evaluation Plan for NIST OpenSAD15

expressions used to calculate DCF. Thus, the IMLM can be viewed as a DCF that considers annotation cost.

III. Postquential Evaluation

Online HITL algorithms can be meaningfully evaluated at two points in the pipeline, as depicted in Fig. 1: before online adaptation ("test-then-train" or "prequential") and after online adaptation ("train-then-test" or "postquential").

Prequential evaluation has long been the preferred method, since it guarantees that different test runs will be evaluated on identical evaluation data. Prequential evaluation has also been shown to converge to evaluation on a held-out test set under certain conditions [20]. In a laboratory setting, these are useful properties that allow for objectively fair comparison in the traditional sense.

Postquential evaluation, on the other hand has been largely overlooked, only appearing outside of this work in one singular piece of literature [23]. Despite its lack of widespread use, postquential evaluation has an advantage over prequential evaluation in that it captures the effect of model adaptation using the data from the current batch. Not only is this informative from an engineering perspective, but it also provides a more realistic view of the performance of the model, since the predictions from the most up-to-date adapted model are expected to be the most accurate in real operational settings.

It is also notable that it is not necessary to use only one evaluation order at a time; prequential and postquential evaluation can be used together to form a larger picture of the behavior and performance of an online HITL algorithm. For example, the difference between the pre- and postquential metrics over time illustrates how efficiently the algorithm utilizes the data from the current sessions compared to previous sessions.

A. Addressing Concerns with Postquential Evaluation

The primary reason that postquential evaluation has not been used in the past is likely that this kind of evaluation causes the training set to be mixed with the test set. "Testing on the training data", of course, only tests an algorithm's ability to memorize data and is not a valid form of evaluating algorithms in a traditional sense. However, this becomes less of a concern when viewed form the perspective of an online HITL algorithm. All HITL algorithms are essentially designed with two classifiers built in—the machine classifier and the human in the loop. As such, as long as the cost of using each type of classifier is accounted for (by using a metric like IMLM), "testing on the training data" in the context of postquential evaluation should be both fair and informative.

IV. Illustrative Example

In this section, we illustrate the application of the evaluation methods introduced above by evaluating algorithms applied to the SLV task. The algorithms and the task are described below, followed by evaluation and analysis of the results.

A. Spoken Language Verification

SLV is the binary classification task of identifying whether a given utterance is in a pre-determined target language. This is different from Language Identification (LID), which is a multi-class classification task with the goal of determining which language from a list of possible languages is being spoken.

The dataset used for these experiments is composed of utterances from the South Asian and Southeast Asian languages found in the Common Voice Corpus [24]. The training and development splits of Common Voice were used where offline training was required, and all online functions were evaluated on the test split. Since Common Voice is not organized in any particular order, we define the order in which the samples are presented in the simulated data stream by leveraging a Dirichlet distribution. This method of simulating data streams is common in the Test-Time Adaptation literature, since real data streams are rarely truly uniform in their temporal class distribution [25].

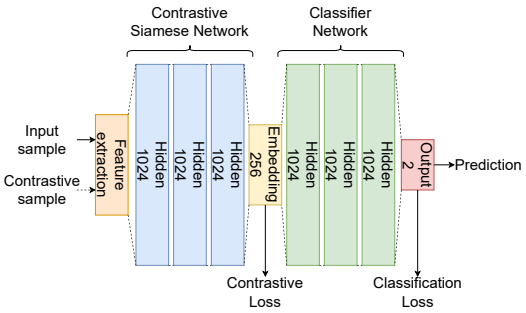

Fig. 2. The constrastive classifier used in the experiments in this paper.

There are 13 Southeast Asian and 20 South Asian languages in Common Voice. We chose 9 target languages that composed 1-5% of each dataset, including those shown in Table I.

Language	Abbr.	Train size	Test size	Test prevalence
Cantonese	yue	2,068,763	90,720	2.86
Indonesian	id	2,068,763	90,720	3.97
Vietnamese	vi	2,068,763	90,720	1.39
Dhivehi	dv	894,222	68,400	3.22
Hindi	hi	894,222	68,400	4.59
Malayam	ml	894,222	68,400	1.03
Marathi	mr	894,222	68,400	2.55
Saraiki	skr	894,222	68,400	1.47
Odia	or	894,222	68,400	1.01

TABLE I
Common Voice Southeast Asian and South Asian target languages, dataset sizes, and target prevalence.

B. Algorithms

The algorithms evaluated in this demonstration are neural networks trained using a conventional Passive Machine Learning (PML) paradigm and using a set of OAL paradigms with varying query budgets. More details are described below.

1) Neural Network Architecture: The neural network used here is the contrastive classifier depicted in Fig. 2 [26]. This network is composed of a pre-trained feature extractor followed by a Siamese network and a classifier composed of linear layers. The feature extractor is an ECAPA-TDNN trained on the VoxLingua107 dataset for LID from SpeechBrain[2]. The overall loss of the network is a weighted combination of the contrastive loss [27] for the Siamese network and Cross-Entropy for the classifier, where the weight on the classification loss is 0.09 that of the contrastive loss.

2) Learning Paradigms: The primary learning paradigm that is analyzed in the following section is the OAL paradigm. In this paradigm, a neural network classifier that is initialized on a minimal bootstrap corpus from the training set data (comprised of 8 utterances in these experiments), is adapted to and then evaluated on incoming batches from a data stream. Each batch contains 720 utterances. For every batch, the algorithm is allowed to query the human about the

[2]https://huggingface.co/speechbrain/lang-id-voxlingua107-ecapa

class label of M utterances. In the experiments below, we use IMLM to determine how many queries per session will be the most cost effective, so we define M as the independent variable. The values of M explored here are 2, 4, 6, 8, 10, 12, 14, and 16. In other words, only 0.3%, 0.6%, 0.8%, 1.1%, 1.4%, 1.7%, 1.9%, and 2.2% of the evaluation data are used to adapt the algorithm.

As an additional comparison, we train the same neural network using a conventional PML paradigm. This training is done offline and uses a much larger training pool (see the training split sizes in Table I).

All training is done using the Adam optimizer and early stopping with a patience of 15 epochs. Every experiment used a learning rate of 10^{-4} and weight decay of 10^{-5}. Additional details are on Github[3].

C. Results and Analysis

The primary results of the experiments are shown in Fig. 3. Here, we report aggregated OAL results across all SLV languages and report the scores of different query allotments in the left column panels. Considering the DCF and query allotment separately, it may be difficult to determine objectively whether the minimum DCF score achieved using 12 queries per session is more efficient than the slightly higher score achieved using only 6 queries per session. The IMLM metric makes objective evaluation easier by combining error rate and annotation cost into a single number. Here, we can see that the minimum IMLM score indeed appears at 12 queries per session, so the minimum DCF point is most efficient.

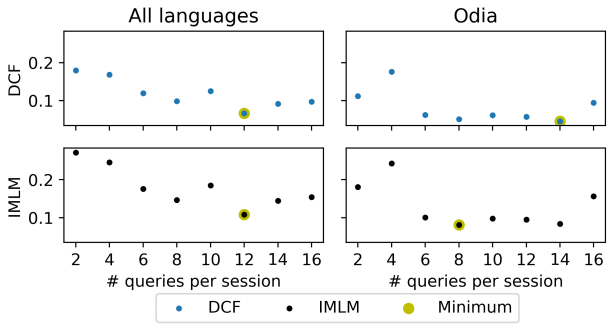

Fig. 3. Performance of OAL algorithms with different query allotments aggregated across all target languages (left column) and for the Odia language alone (right column). Performance is measured by DCF (top row) and IMLM (bottom row). Minima are highlighted in yellow.

Note that, in the left column of Fig. 3, the two metrics choose the same point even though the weightings are drastically different—DCF uses the standard 3:1 weighting while IMLM uses $q \approx 55$ based on Eq. 4. An example of an individual language where the DCF and IMLM did not agree on an optimal operating point is Odia, in the right column. For this specific instance, the metric indicates that the algorithm with fewer queries and marginally higher DCF is more efficient than the algorithm that achieves the absolute minimum DCF with more queries.

IMLM can be used to compare online algorithms like OAL to conventional PML algorithms by considering the cost of annotating the training dataset. The results of the PML algorithm (shown in Table II), are worse than the OAL algorithm in many cases in terms of DCF. This happens because, despite learning from over 1,000 times as much training data, PML does not adapt directly to the data stream like OAL does. The IMLM shows an even larger difference than DCF because of the disparity in training resources. Specifically, the IMLM value of the PML run indicates that, given the number of manually labeled samples, the whole process took nearly 17 times longer than it would have to manually review only the evaluation set. On the other hand, every OAL algorithm speeds up the process significantly because they achieve lower error at a fraction of the annotation cost.

[3]https://github.com/markrl/oalcf

Paradigm	# annotations	DCF	IMLM
PML	12,465,843	0.281	16.975
OAL	11,448	0.066	0.108

TABLE II
LABEL COUNT, DCF, AND IMLM AGGREGATED ACROSS ALL SLV LANGUAGES FOR THE PML AND OAL (12 QUERIES) PARADIGMS.

We demonstrate the utility of postquential evaluation in Fig. 4. Here, we show the aggregated DCF and IMLM scores throughout the South Asian data stream and observe an expected pattern: the postquential scores are always better than the prequential scores because the former evaluates the adapted model while the latter does not. We do observe, however, that the gap between the two curves on each plot narrows as the model learns from the data stream. This indicates that adaptation is crucial for a new model, but a mature model may be able to perform reasonably well without utilizing adaptation data.

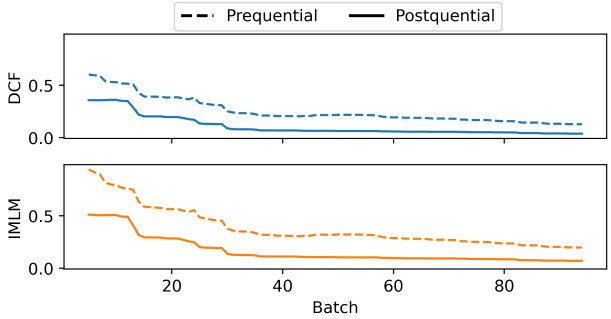

Fig. 4. Prequential and postquential trends aggregated across all South Asian languages in terms of DCF and IMLM. For both, the postquential scores are better than the prequential scores, but the gap narrows over time.

V. CONCLUSIONS

In this paper, we introduce IMLM as a new method for evaluating online HITL algorithms. IMLM allows consideration of both annotation cost and error rate in a single metric by viewing both in terms of time cost. Since IMLM considers annotation cost, it also enables postquential evaluation—evaluation of online methods after classifier adaptation rather than before—which provides a more realistic view of a classifier's ability to adapt. We demonstrate the use of IMLM and postquential evaluation with an illustrative example using conventional PML and various forms of the OAL paradigm to perform the SLV task on languages from Common Voice. Based on its intuitive nature and applicability to a real-world online learning task, we prove that IMLM is an important and necessary metric for evaluation of online HITL algorithms.

The IMLM introduced here is limited to evaluating the error rate and annotation cost of binary classifiers. Future work might extend the metric to handle multi-class classifiers and regressors. IMLM could also be adjusted to consider computational cost or operating cost, since online learning can be computationally intensive, and HITL operations can be financially costly due to the need of human operators. Despite these limitations, the introduction of IMLM marks an important step toward better evaluation of online HITL algorithms.

References

[1] D. Cacciarelli and M. Kulahci, "Active Learning for Data Streams: A Survey," *Machine Learning*, vol. 113, no. 1, pp. 185–239, 2024.

[2] P. M. Pilarski, M. R. Dawson, T. Degris, F. Fahimi, J. P. Carey, and R. S. Sutton, "Online Human Training of a Myoelectric Prosthesis Controller via Actor-Critic Reinforcement Learning," in *2011 IEEE International Conference on Rehabilitation Robotics*, 2011, pp. 1–7.

[3] L. Shani, T. Zahavy, and S. Mannor, "Online Apprenticeship Learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, no. 8, 2022, pp. 8240–8248.

[4] Z. J. Wang, D. Choi, S. Xu, and D. Yang, "Putting Humans in the Natural Language Processing Loop: A Survey," *arXiv preprint arXiv:2103.04044*, 2021.

[5] B. Hancock, A. Bordes, P.-E. Mazare, and J. Weston, "Learning from Dialogue after Deployment: Feed Yourself, Chatbot!" *arXiv preprint arXiv:1901.05415*, 2019.

[6] J. Li, A. H. Miller, S. Chopra, M. Ranzato, and J. Weston, "Dialogue Learning With Human-In-The-Loop," 2017. [Online]. Available: https://arxiv.org/abs/1611.09823

[7] B. Liu, G. Tur, D. Hakkani-Tur, P. Shah, and L. Heck, "Dialogue Learning with Human Teaching and Feedback in End-to-End Trainable Task-Oriented Dialogue Systems," 2018. [Online]. Available: https://arxiv.org/abs/1804.06512

[8] H. Kim, D. Choi, B. Drake, A. Endert, and H. Park, "TopicSifter: Interactive Search Space Reduction through Targeted Topic Modeling," in *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2019, pp. 35–45.

[9] V. Kumar, A. Smith-Renner, L. Findlater, K. Seppi, and J. Boyd-Graber, "Why Didn't You Listen to Me? Comparing User Control of Human-in-the-Loop Topic Models," 2019. [Online]. Available: https://arxiv.org/abs/1905.09864

[10] D. Sculley, "Online Active Learning Methods for Fast Label-efficient Spam Filtering," in *CEAS*, vol. 7, 2007, p. 143.

[11] H. Zhang, W. Liu, L. Sun, L. Chen, Z. Ding, and Q. Liu, "Analyzing Network Traffic for Protocol Identification: An Ensemble Online Active Learning Method," in *2020 6th International Conference on Big Data and Information Analytics (BigDIA)*, 2020, pp. 167–172.

[12] A. Narr, R. Triebel, and D. Cremers, "Stream-based Active Learning for Efficient and Adaptive Classification of 3D Objects," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 227–233.

[13] S. Ghiasi, G. Pazzi, C. D. Grosso, G. D. Magistris, and G. Veneri, "Combining Thermodynamics-based Model of the Centrifugal Compressors and Active Machine Learning for Enhanced Industrial Design Optimization," 2023. [Online]. Available: https://arxiv.org/abs/2309.02818

[14] X. Yan, M. Sarkar, B. Lartey, B. Gebru, A. Homaifar, A. Karimoddini, and E. Tunstel, "An Online Learning Framework for Sensor Fault Diagnosis Analysis in Autonomous Cars," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 12, pp. 14 467–14 479, 2023.

[15] D. Manjah, D. Cacciarelli, B. Standaert, M. Benkedadra, G. R. De Hertaing, B. Macq, S. Galland, and C. De Vleeschouwer, "Stream-based Active Distillation for Scalable Model Deployment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4999–5007.

[16] N. Beck, S. Kothawade, P. Shenoy, and R. Iyer, "STREAMLINE: Streaming Active Learning for Realistic Multi-Distributional Settings," 2023. [Online]. Available: https://arxiv.org/abs/2305.10643

[17] P. Christen, D. J. Hand, and N. Kirielle, "A Review of the F-Measure: Its History, Properties, Criticism, and Alternatives," *ACM Comput. Surv.*, vol. 56, no. 3, oct 2023. [Online]. Available: https://doi.org/10.1145/3606367

[18] P. A. Flach, "The Geometry of ROC Space: Understanding Machine Learning Metrics Through ROC Isometrics," in *Proceedings of the 20th international conference on machine learning (ICML-03)*, 2003, pp. 194–201.

[19] A. Martin and M. Przybocki, "The NIST 1999 Speaker Recognition Evaluation—An Overview," *Digital Signal Processing*, vol. 10, no. 1, pp. 1–18, 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S105120049990355X

[20] J. Gama, R. Sebastiao, and P. P. Rodrigues, "Issues in Evaluation of Stream Learning Algorithms," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 329–338.

[21] F. Byers, J. Fiscus, Seyed, G. Sanders, and M. Przybocki, "Open Speech Analytic Technologies Pilot Evaluation OpenSAT Pilot," 2019-02-27 2019.

[22] H. jin Shim, J. weon Jung, T. Kinnunen, N. Evans, J.-F. Bonastre, and I. Lapidot, "a-dcf: an architecture agnostic metric with application to spoofing-robust speaker verification," 2024. [Online]. Available: https://arxiv.org/abs/2403.01355

[23] M. Khannouz and T. Glatard, "Dynamic Ensemble Size Adjustment for Memory Constrained Mondrian Forest," in *2022 IEEE International Conference on Big Data (Big Data)*, 2022, pp. 3358–3363.

[24] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common Voice: A Massively-multilingual Speech Corpus," *arXiv preprint arXiv:1912.06670*, 2019.

[25] L. Yuan, B. Xie, and S. Li, "Robust Test-time Adaptation in Dynamic Scenarios," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 15 922–15 932.

[26] M. Lindsey, N. R. Robinson, F. Kubala, and R. M. Stern, "Reducing the Cost of Spoof Detection Labeling using Mixed-Strategy Active Learning and Pretrained Models," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2023, pp. 1–7.

[27] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a Similarity Metric Discriminatively, with Application to Face Verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 539–546 vol. 1.