HHAI 2023: Augmenting Human Intellect
P. Lukowicz et al. (Eds.)

doi:10.3233/FAIA230100

Value-Based Hybrid Intelligence

Burcu SAYIN [a,1], Jie YANG [b], Andrea PASSERINI [a] and Fabio CASATI [c]
[a] *University of Trento, Italy*
[b] *Delft University of Technology, Netherlands*
[c] *Servicenow, Santa Clara, USA*

Abstract. In this paper, we argue that the way we have been training and evaluating ML models has largely forgotten the fact that they are applied in an organization or societal context as they provide value to people. We show that with this perspective we fundamentally change how we evaluate and select machine learning models.

Keywords. machine learning, hybrid intelligence, selective classification

1. Introduction

Recently, Hybrid Intelligence (HI) [1,2] has become a compelling paradigm that combines human and machine intelligence to perform better than human-only or machine-only decisions [3], especially on complex tasks. It is a breakthrough for solving real-world tasks because HI can make more reliable decisions on complex functions. That is why designing and deploying HI systems becomes critical for many domains, such as medical diagnosis, speech recognition, and autonomous driving. Although the HI community paid attention to designing better HI systems, how to measure and assess the *value* that HI adds to real-world systems and society is still an open question.

HI is widely adopted in enterprise scenarios as selective classifiers [4,5], where machine learning (ML) models can abstain from making a prediction and resort to human judgment or a default path (as in Figure 1). We argue that current metrics to evaluate such systems (e.g. accuracy, F1, etc) do not measure the real "value" of HI systems and they often lead to wrong decisions on which is the best model to use in a certain situation.

In this paper, we highlight the need for designing novel metrics to evaluate the *value* of HI systems in real-world tasks. By saying *value* we mean the value HI systems add to the real-world system deployed, and it consists of several components, such as trust, complementarity, cost-sensitive learning, etc. This has been addressed in our recent papers [5,6,7,8] that propose a novel way of measuring the value of HI systems, and we show that the existing accuracy-based metrics are not compatible with HI systems. We further discuss open research questions to invite the HI community to pay enough attention to how and why HI models are used in practice, and to the aspects and metrics that are relevant to enterprises when they adopt and deploy a model.

[1]Corresponding Author: Burcu Sayin, burcu.sayin@unitn.it

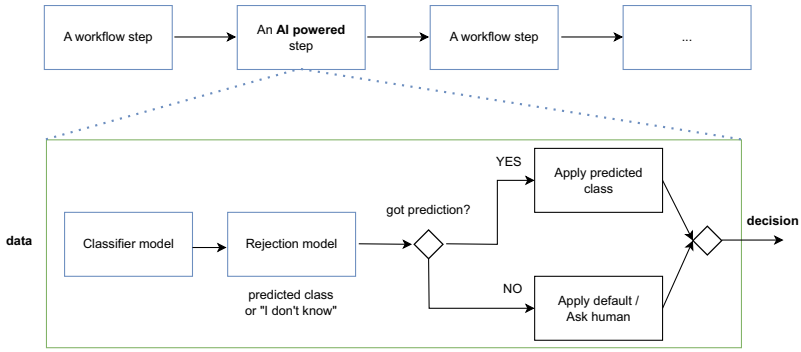

Figure 1. Typical implementation of ML models into an ML solution workflow - edited from [6,7]. A typical implementation of the rejection function is to filter based on a confidence threshold, assuming that the classifier is trained independently of the rejection logic. It is less general to have a classifier that is aware of the cost.

2. Formalizing the "Value"

As seen in Figure 1, the majority of AI deployments in enterprise context pursue a hybrid workflow: *there is always some human in some "loop"* when we apply a model, with very few exceptions. That is why, ML models are deployed as *selective classifiers* [3] in enterprise scenarios, as the cost of errors is often more than the cost of not predicting and following a safe, default route. From this simple description, we can draw two observations [6]: (i) the way in which the system is deployed depends on the "cost" of machine errors as compared to the cost of involving a human, and (ii) a well-calibrated model with *arbitrarily bad accuracy* can still provide *value*. Based on these observations, we formalize the *value* of an ML model as follows [7]. Given a classifier g that operates on items $x \in \mathscr{D}$ and returns a predicted class $y \in \mathscr{Y}$ or a special label y_r ("rejected"), the average value per the prediction of applying a model g over $\mathscr{D}$ can be computed as:

$$V(g,\mathscr{D}) = \rho V_r + (1-\rho)(\alpha V_c + \Sigma_{ij}[\Omega \odot V_w]_{ij}) \tag{1}$$

where ρ is the proportion of items in $\mathscr{D}$ that are rejected by g (classified as y_r), α is the accuracy for predictions above the threshold, V_r and V_c are the value of rejecting an item and classifying it correctly respectively, Ω is a matrix denoting the proportion of predictions (above threshold) in each cell of the confusion matrix, and V_w is a matrix with the cost for each type of error (set to zero on the main diagonal corresponding to correct predictions), and $\odot$ denotes the Hadamard (element-wise) product, of which we take the summation across all elements ij. Notice that ρ, α, Ω all depend on $\mathscr{D}$ and g, and we omit the indices to simplify notation. Also, if our classification problem has $|\mathscr{Y}|$ classes, then Ω and V_w are $|\mathscr{Y}| \times |\mathscr{Y}|$ (y_r is not included here). An alternative representation would be to just say that $V(g,\mathscr{D}) = \Omega' V'$, where the confusion and value matrices incorporate the reject class. This would allow us to model the case where the value of rejections and correct predictions is also class-dependent. Instead, if we only consider costs based on what we misclassify then Ω and V_w become vectors, and in the most common case where all wrong predictions are considered equally bad in a first approximation, then Ω and V_w are scalar, and $\Omega = 1-\alpha$, so in this case, the formula becomes:

$$V(g,\mathscr{D}) = \rho V_r + (1-\rho)(\alpha V_c + (1-\alpha)V_w) \tag{2}$$

At this point, we simplify the notation to remove dimensionality and arrive at a formulation that is digestible for process owners, for whom it may be hard to come up with the three cost parameters/vectors. None of the above simplifications change the concepts presented. We define as baseline the case where we do not have ML, or, equivalently, where we reject any prediction. We set this baseline at 0 ($V_r = 0$); making it easier to evaluate a model in terms of (i) whether it improves on the baseline or not, and (ii) whether we should adopt AI for a given problem. We also express V_w in terms of V_c, as in $V_w = -kV_c$, where k is a constant telling us how bad is an error with respect to getting the correct prediction. When reasoning about an AI-powered solution workflow, V_c should be considered as a scaling factor in terms of value "per unit of V_c dollars", or equivalently the magnitude of V_c. From now on we, therefore, focus on "value per dollar unit of rejection cost" $V' = V/V_c$. We avoid introducing a new symbol and, without loss of generality with respect to the above equations, we set $V_c = 1$ and get Equation 3 which captures the same concepts as Equation 1 but simplifies our presentation.

$$V(g, \mathscr{D}) = (1-\rho)(\alpha - k(1-\alpha)) \tag{3}$$

Filtering the predictions with confidence threshold. Here we focus on the most common situation observed in practice; the model selectivity is applied by thresholding confidence values and rejecting predictions that have confidence c less than a threshold τ. If we assume perfect calibration (the expected accuracy for a prediction of confidence c is c) [9], then we know that the threshold is at the point where the value of accepting a prediction is greater than 0 and $\alpha_\tau = \tau$. To have $V(g, \mathscr{D}) > 0$ we need $\tau - k + k\tau > 0$:

$$\tau > k/(k+1) \tag{4}$$

This conforms to intuition: if k is large, it never makes sense to predict, better go with the default. If k=0 (no cost for errors), we might always predict since there is no penalty for wrong predictions. Perhaps paradoxically, this is the case where we want to use *accuracy* because wrong predictions are harmless. If k=1 (errors are the mirror image of correct predictions), then our threshold is 0.5. In deriving the threshold we assumed that all errors have equal cost, but the derivation can be easily extended to cost-sensitive settings [7].

3. Experimental observations

We tested our hypothesis on a number of NLP benchmarks and evaluated leaderboard models together with simple models like logistic regression or shallow multi-layer perceptrons. Our experimental evaluation, described in detail in [7], highlights four relevant issues with the current way in which ML models are evaluated and deployed:

Accuracy and F1-score are poor indicators of model value Models with very reasonable accuracy/F1 (beyond 0.8) have a value that decreases substantially with the increase of the cost factor k. Many models achieve *negative* value (i.e., it is better to simply ignore them) for $k = 4$ or $k = 8$. These are definitely realistic cost factors in high-stake applications ($k = 4$ means that "being wrong is 4 times as bad as it is good to be right"). Another major finding is that accuracy or F1 are quite poor proxies of value even in relative terms. The best-performing model is largely dependent on the cost factor, and accuracy (or F1) quickly becomes unreliable as a metric to identify the most appropriate model to employ.

Cost-sensitive error is a poor indicator of model value in cost-sensitive settings In cost-sensitive settings (i.e. different types of errors have different costs), cost-sensitive error [10], which computes a weighted sum of errors with the weights given by the corresponding cost, is a popular alternative to standard cost-insensitive metrics. However, the cost-sensitive error is still a poor indicator of model value for high costs and often detects as best performing models that actually achieve *negative* value. The problem is not how it treats the costs of different errors, but the fact that it does not assume a selective classifier and a corresponding cost-sensitive rejection threshold, which is the main practical contribution of our "value" definition. This also implies that cost-sensitive learning [11] (i.e. training classifiers to minimize cost-sensitive error) should be coupled with learning to reject mechanisms [12] to be effective in optimizing the value of the learned models.

Lack of calibration substantially affects model value The threshold in Eq. 4 assumes that models are perfectly calibrated, which is often not true for trained models, and deep learning models in particular [13]. Applying a simple but effective recalibration technique (e.g. temperature scaling [13]) substantially improves the value of models, and almost completely eliminates the degenerate behavior of models with negative value ("useless" models receive 0 value, as expected). These results suggest that learning models should always be recalibrated before being incorporated into practical workflows. On the other hand, accuracy and F1 are still poor indicators of value, with the best-performing model being still largely dependent on the cost factor. In domain adaptation scenarios, simple logistic regression (LogReg) consistently outperforms all other models for large values of k, most likely because of its inherent calibration capabilities [14] as compared to more complex (and hard to recalibrate) models. The lively research area of calibration in ML and especially deep learning can provide useful solutions to this problem [9].

Operating in an out-of-distribution setting substantially affects model value The lack of calibration in ML models is known to be particularly harmful when the model operates in an out-of-distribution (OOD) setting [15,16]. Indeed the most recent large pretrained language models (LLM) tend to perform well on domain adaptation tasks, better than leaderboard models in those tasks, most likely because being trained on massive datasets, few examples are out of distribution in terms of language. However, even for these models, accuracy is a poor proxy of value when k is large, with frequent swaps in the leaderboard and simple linear models occasionally outperforming these powerful (and very expensive to employ) LLM.

4. Conclusions and Future Works

All in all, we see this work as providing evidence of a serious problem in how we evaluate machine learning models, and of its consequences on the utility of these models in practical applications. We take this opportunity to invite the HI community to pay attention to formalizing the *value* that hybrid intelligence adds to deployed real-world systems, and how these systems can be adapted to maximize this value.

Acknowledgements. This research was partially supported by TAILOR, a project funded by the EU Horizon 2020 research and innovation program under GA No 952215. The work of Burcu Sayin and Andrea Passerini was partially supported by the project AI@Trento (FBK-Unitn).

References

[1] Dellermann D, Ebel P, Söllner M, Leimeiste JM. Hybrid Intelligence. Business & Information Systems Engineering; 61:637–643; 2019; doi: 10.1007/s12599-019-00595-2.

[2] Dellermann D, Calma A, Lipusch N, Weber T, Weigel S, Ebel PA. The Future of Human-AI Collaboration: A Taxonomy of Design Knowledge for Hybrid Intelligence Systems. In: Hawaii International Conference on System Sciences (HICSS); Hawaii, USA; 2019.

[3] Kamar E. Hybrid workplaces of the future. XRDS: Crossroads, The ACM Magazine for Students; 23:22–25; 2016; doi: 10.1145/3013488.

[4] Lakkaraju H, Kamar E, Caruana R, Horvitz E. Identifying unknown unknowns in the open world: representations and policies for guided exploration. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI'17); p. 2124–2132; 2017; AAAI Press.

[5] Sayin B, Krivosheev E, Ramírez J, Casati F, Taran E, Malanina V, Yang J. Crowd-powered hybrid classification services: Calibration is all you need. In: 2021 IEEE International Conference on Web Services (ICWS); p. 42–50; 2021.

[6] Sayin B, Yang J, Passerini A, Casati F. The science of rejection: A research area for human computation. In: The 9th AAAI Conference on Human Computation and Crowdsourcing (HCOMP); 2021; AAAI Press.

[7] Sayin B, Yang J, Chen X, Passerini A, Casati F. Rethinking and Recomputing the Value of ML Models. ArXiv; abs/2209.15157; 2022.

[8] Casati F, Noel P, Yang J. On the value of ML models. ArXiv; abs/2112.06775; 2021.

[9] Filho TMS, Song H, Perelló-Nieto M, Santos-Rodríguez R, Kull M, Flach PA. Classifier Calibration: How to assess and improve predicted class probabilities: a survey. ArXiv; abs/2112.10327; 2021.

[10] Elkan C. The Foundations of Cost-Sensitive Learning. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence; p. 973–978; 2001.

[11] He H, Ma Y. Imbalanced Learning: Foundations, Algorithms, and Applications. Wiley-IEEE Press; 2013; doi: 10.1002/9781118646106.

[12] Hendrickx K, Perini L, Plas DV, Meert W, Davis J. Machine Learning with a Reject Option: A survey. ArXiv; abs/2107.11277; 2021.

[13] Guo C, Pleiss G, Sun Y, Weinberger KQ. On Calibration of Modern Neural Networks. In: Proceedings of the 34th International Conference on Machine Learning (ICML'17) - Volume 70; Sydney, NSW, Australia; p. 1321–1330; 2017; doi: 10.48550/ARXIV.1706.04599.

[14] Kull M, Filho TMS, Flach PA. Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. Electronic Journal of Statistics 11:5052–5080; 2017.

[15] Tomani C, Buettner F. Towards Trustworthy Predictions from Deep Neural Networks with Fast Adversarial Calibration. In: AAAI Conference on Artificial Intelligence; 2019.

[16] Wu Y, Zeng Z, He K, Mou Y, Wang P, Xu W. Distribution Calibration for Out-of-Domain Detection with Bayesian Approximation. In: Proceedings of the 29th International Conference on Computational Linguistics; Gyeongju, Republic of Korea; p. 608–615; 2022.

[17] Ruder S, Plank B. Strong baselines for neural semi-supervised learning under domain shift. In: The 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018); p. 1044–1054; 2018.

[18] Heitmann M, Siebert C, Hartmann J, Schamp C. More than a feeling: Benchmarks for sentiment analysis accuracy. Communication & Computational Methods eJournal; 2020.

[19] Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V. Roberta: A robustly optimized bert pretraining approach. ArXiv; abs/1907.11692; 2019.

[20] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research; 12:2825–2830; 2011.