

Bridging the Gap in Hybrid Decision-Making Systems

Federico Mazzoni¹, Roberto Pellungrini², and Riccardo Guidotti¹✉

¹University of Pisa, Italy, federico.mazzoni@phd.unipi.it,
riccardo.guidotti@unipi.it, ²Scuola Normale Superiore, Pisa, Italy,
roberto.pellungrini@sns.it

Abstract. We introduce BRIDGET, a novel human-in-the-loop system for hybrid decision-making, aiding the user to label records from an unlabeled dataset, attempting to “bridge the gap” between the two most popular Hybrid Decision-Making paradigms: those featuring the human in a leading position, and the other with a machine making most of the decisions. BRIDGET understands when either a machine or a human user should be in charge, dynamically switching between two statuses. In the different statuses, BRIDGET still fosters the human-AI interaction, either having a machine learning model assuming *skeptical* stances towards the user and offering them suggestions, or towards itself and calling the user back. We believe our proposal lays the groundwork for future synergistic systems involving a human and a machine decision-makers.

Keywords: Human-Centered AI · Hybrid Decision Maker · Skeptical Learning · Incremental Learning · Learning-to-Defer · Explainable AI

1 Introduction

Automated decision-making processes based on Machine Learning (ML) are still not widely adopted for high-stakes decisions such as medical diagnoses or court decisions [14]. In these fields, humans are aided but not replaced by Artificial Intelligence (AI), resulting in Hybrid Decision-Makers (HDM) [7].

As the literature keeps growing, the term “Hybrid Decision-Makers” has been used as an umbrella word for various different kinds of algorithms, often with a different focus, and a proper consensus has not been reached yet. Punzi et al. [8] notes two major HDM paradigms: *Learning-to-Abstain* where under certain conditions an ML model refuses to make a decision, and *Learning Together*, where the human can interact with the training process of the ML model. Two of the most representative approaches of the two paradigms are, respectively, *Learning-to-Defer* (LtD) [4] systems, where the machine plays the primary role, deferring decisions on records with a high degree of uncertainty to an external human supervisor, and *Skeptical Learning* (SL), where an ML model learns “in parallel” to the decisions taken by a human and queries them if it is “skeptical” of the human decision [15, 16]. The two have vastly different scopes. LtD assumes a cost to query the human user and aims to minimize that, leaving most

of the decisions to the machine, while SL assumes the human user is always in control but needs to be helped by a machine model to stay consistent over time. Whereas SL posits a human expert who always has the final say but who can albeit get confused and thus in need of the machine’s help, LtD assumes that some decisions are better suitable either to the user or the machine. As such, the model is trained not only to classify instances but to defer them [8]. Another key difference is the training phase. SL training is continuous, i.e., the model is always learning something from the user’s final decision, acting as ground truth, whereas LtD employs a stationary dataset, thus resulting in a stationary model, with different ground truths for the classification and the deferral policy. Therefore, HDM systems can also be classified either following the role of the primary decision-maker or their training process. Although Explanatory Interactive ML systems have been proposed [12], HDMs mostly focus on providing decisions rather than explanations. With that said, CINCER can provide explanations as contrastive and influential counterexamples [2], whereas FRANK can show the model logic and provide them with real and synthetic examples and counterexamples [6]. Both are based on SL.

Our proposal aims to “fill the gap” between human-driven HDMs, such as SL, and machine-led ones, such as LtD, effectively creating a “bridge”, hence the name BRIDGET, between the two paradigms, and an interpretable system able to adapt to different scenarios. Following [2, 6], BRIDGET employs an *Incremental Learning* (IL, or Continual Learning) model. IL is a ML paradigm where the model is continuously trained on small data batches, potentially only one data point, instead of the entirety of the training set [5, 13]. Moreover, BRIDGET shares a focus on explanation with other interactive ML methods [1, 6, 11, 12].

2 Setting the Stage

In the following we report a brief overview of concepts necessary to understand our proposal. We indicate with H and M the Human user and the Machine of the system, and with X, Y a dataset where $X \in \mathcal{X}$ is a set of n records in feature space \mathcal{X} , while $Y \in \mathcal{Y}$ is the set of the target variable in the target space \mathcal{Y} . For classification problems, $y_i \in \{1, \dots, l\} = L$, where L is the set of different class labels and l , is the number of the classes. We indicate a trained decision-making model with a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that maps data instances x from the feature space \mathcal{X} to the target space \mathcal{Y} . We represent the user decision process as an analogous function $h : \mathcal{X} \rightarrow \mathcal{Y}$. We write $f(x_i) = \tilde{y}_i$ to denote the decision \tilde{y}_i taken by f , $h(x_i) = \hat{y}_i$ to denote the decision \hat{y}_i taken by h .

Skeptical Learning. Given a ML model f and a dataset X , the user is tasked to assign a label y_i to each record $x_i \in X$. In SL, the user assigns the label \hat{y}_i and, independently from them, f assigns the label \tilde{y}_i . The ML model f can be pre-trained on a small training set. If $\hat{y}_i \neq \tilde{y}_i$ and f is *skeptical* (see below), the user is asked if they want to accept \tilde{y}_i as y_i . If they do, y_i takes the value \tilde{y}_i . If the user refuses, if $\hat{y}_i = \tilde{y}_i$ or if the model is not skeptical, y_i is assigned \hat{y}_i . f is then incrementally trained on x_i and y_i .

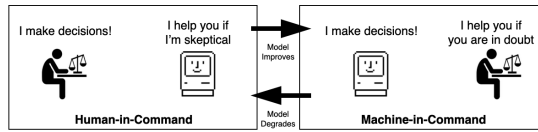


Fig. 1. BRIDGET looping between its potential phases.

The model’s *skepticality* is related to the model’s *epistemic uncertainty* [11], which is independent of the notion of prediction probability towards a certain decision. As the model is exposed to enough data, its epistemic uncertainty, i.e., its *ignorance*, should be minimized, assuming consistency in the labels [3]. Since few models allow access to epistemic uncertainty [2,3], it has been approximated by SL with the *Empirical Accuracy* (EA) of past predictions, both of the user and the model, i.e., the ratio between the number of times a label has been proposed by the user/predicted by the model, and the times it has been accepted as y_i [15]. Let $X_p \subseteq X$ be the set of past-seen instances and $Y_p \subseteq Y$ the respective ground truths. Let $\delta_{f(x),y}$ be the Kronecker Delta measuring accurate prediction of $f(x)$ w.r.t ground truth y . We calculate Skepticality as:

$$skp(x_i, \tilde{y}_i, \hat{y}_i, X_p, Y_p) = c_f(x_i, \tilde{y}_i) \cdot ea_f(\tilde{y}_i, X_p, Y_p) - c_f(x_i, \hat{y}_i) \cdot ea_h(\hat{y}_i, X_p, Y_p)$$

where $c_f(x_i, \tilde{y}_i)$ and $c_f(x_i, \hat{y}_i)$ are the model prediction probabilities towards \tilde{y}_i and \hat{y}_i , and $ea_f(\tilde{y}_i, X_p, Y_p) = \frac{1}{|X_p|} \sum_{\{x_p \in X_p | f(x_p) = \tilde{y}_i\}} \delta_{f(x_p), y_p}$ is the empirical accuracy of the model f toward label \tilde{y}_i , and $ea_h(\hat{y}_i, X_p, Y_p) = \frac{1}{|X_p|} \sum_{\{x_p \in X_p | h(x_p) = \hat{y}_i\}} \delta_{h(x_p), y_p}$ is the empirical accuracy of the user function h toward label \hat{y}_i . Thus, each possible label $l \in L$ has two EA values – following the user’s and the model’s track record (e.g., in binary classification with $|L| = 2$, we have 4 EA values, 2 for M and 2 for H) [15].

3 A Bridget System

BRIDGET, whose pseudocode is reported in Algorithm 1, assumes two potential statuses, depicted in Figure 1:

- A *Human-in-Command* (HiC) phase where the human H and the machine M are into a *co-evolutionary relationship* [6]. H takes all the decisions, and M offering suggestions and explanations if skeptical.
- A *Machine-in-Command* (MiC) phase where M takes most of the decisions, but it can call H back if uncertain, and it is able to explain why.

The BRIDGET system can loop between the two phases, accommodating the user’s needs, potential fallouts in the model’s accuracy, or novelties in the data.

3.1 Human-in-Command

BRIDGET starts in the HiC phase and requires a set of records X , to be label one by Once a new record x_i is received, H makes its decision as well as M ,

Algorithm 1: BRIDGET

Input : X - records, α - skeptical threshold, β - belief threshold

```

1  $X_p, Y_p, \tilde{Y}, \hat{Y}, f, k, p, phase \leftarrow initialize;$  // sets initialization
2  $x_i \leftarrow receive\_record(X);$  // receive a new un-label record
3 if  $phase = HiC$  then // if Human-in-Command
4    $\hat{y}_i \leftarrow h(x_i); \tilde{y}_i \leftarrow f(x_i);$  // get user decision and model prediction
5   if  $\hat{y}_i \neq \tilde{y}_i \wedge skp(x_i, \tilde{y}_i, \hat{y}_i, X_p, Y_p) > \alpha$  then // if clash & skepticism
6     if  $is\_expl\_desired(x_i, \tilde{y}_i)$  then // if an explanation for  $\tilde{y}_i$  is desired
7        $e_i \leftarrow get\_and\_show\_expl(x_i, \tilde{y}_i, f, X_p);$  // return explanation  $e_i$ 
8     if  $accept\_label\_change(x_i, \tilde{y}_i)$  then  $y_i \leftarrow \tilde{y}_i;$  //  $\tilde{y}_i$  is accepted
9     else  $y_i \leftarrow \hat{y}_i;$  //  $\tilde{y}_i$  is refused
10  else  $y_i \leftarrow \hat{y}_i;$  // otherwise  $\hat{y}_i$  is accepted
11   $\tilde{Y}, \hat{Y}, f, p \leftarrow update;$  // model and parameter updates
12  if  $check(h, f, p)$  then  $phase = MiC, p = 0;$  // phase change
13 else // if Machine-in-Command
14    $\tilde{y}_i \leftarrow f(x_i); b \leftarrow compute(f, x_i, \tilde{y}_i, Y, \tilde{Y});$  //  $\tilde{y}_i$  and  $M$ 's belief towards it
15   if  $b < \beta$  then // if  $b$  is low, user is called back
16      $u_i \leftarrow get\_and\_show\_unr(x_i, \tilde{y}_i, f, X_p);$  // show explanation  $u_i$ 
17      $\hat{y}_i \leftarrow h(x_i); y_i \leftarrow \tilde{y}_i;$  //  $\hat{y}_i$  accepted
18   else if  $random\_check(b)$  then  $y_i \leftarrow \tilde{y}_i;$  // probabilistic check,  $\tilde{y}_i$  accepted
19   else // if user is called back
20     if  $is\_expl\_desired(x_i, \tilde{y}_i)$  then // if an explanation for  $\tilde{y}_i$  is desired
21        $e_i \leftarrow get\_and\_show\_expl(x_i, \tilde{y}_i, f, X_p);$  // return explanation  $e_i$ 
22      $\hat{y}_i \leftarrow h(x_i); y_i \leftarrow \hat{y}_i;$  //  $\hat{y}_i$  accepted
23      $h, f, p \leftarrow update;$  // model and parameter updates
24     if  $check(h, f, p)$  then  $phase = HiC, k = 0;$  // phase change
25  $X_p, Y_p \leftarrow update(x_i, y_i);$  // recording final decision

```

following SL (line 4 of Alg. 1). To compute skepticism, we propose *Fading Empirical Accuracy* (FEA) as a replacement for EA. While computing M and H 's track record, instead of assigning the same weight to each previously-seen record and the respectively assigned label, our metric weights each record w.r.t. its temporal distance to the current one (remember that \tilde{Y}, \hat{Y}, Y and the other sets are ordered w.r.t. their appearance). In other words, older records weigh less. This is consistent with the idea of EA as an ever-evolving, more accurate proxy for the model's epistemic uncertainty, i.e., its *current* status. For example, with FEA the model's early errors are de-emphasized. FEA is employed for H as well, as their behavior might also change over time. Then, we define the *Fading Skepticity* of M towards the user decision \hat{y}_i as:

$$skp(x_i, \tilde{y}_i, \hat{y}_i, X_p, Y_p) = c_f(x_i, \tilde{y}_i) \cdot fea_f(x_i, \tilde{y}_i, X_p, Y_p) - c_f(x_i, \hat{y}_i) \cdot fea_h(x_i, \hat{y}_i, X_p, Y_p)$$

where $fea_f(x_i, \tilde{y}_i, X_p, Y_p) = \frac{1}{|X_p|} \sum_{\{x_p \in X_p | f(x_p) = \tilde{y}_i\}} \delta_{f(x_p), y_p} d(x_p, x_i)$ and by analogy $fea_h(x_i, \hat{y}_i, X_p, Y_p) = \frac{1}{|X_p|} \sum_{\{x_p \in X_p | h(x_p) = \hat{y}_i\}} \delta_{h(x_p), y_p} d(x_p, x_i)$. Here $d(x_p, x_i)$ is a distance function between the current data point x_i and previously seen data

point x_p . If $fskp$ is higher than a certain threshold α , M is skeptical of H 's decision (line 5), and the user is prompted if they want to change them.

Before making the final decision, H can request an explanation for the suggestion, such as (counter-)exemplar records, local decision rules or an overview of the model's logic (lines 6-7). As in [6], the explanations are generated following the latest updated version of the model and the training data, and thus evolving over time. By exploiting in BRIDGET the CAIPI model [12], H can also teach M if the decision is right but for the wrong reason. After a decision is taken, M 's internal model f and the various sets are updated (line 11), following H 's decision, who always holds full veto power. This co-evolutionary phase ensures a profitable human-machine interaction both for H , as they might receive useful suggestions, and M , since the model is progressively updated.

As mentioned, for traditional EA, both M and H have as many FEA values as $|L|$, i.e., 2 in a binary classification task. As SL assumes an expert user, the average of the model's FEA values can provide an esteem of its overall closeness to the user at any given point in time. As such, FEA also plays a key role in transitioning towards the machine-led phase, and after each labelled record, BRIDGET checks if a transition towards the MiC phase is possible (line 12). The transition happens if all the following set of conditions t_M is met:

1. during the co-evolutionary phase, more than k_{max} records have been seen;
2. the model's average FEA is higher than a certain threshold;
3. H designates M as the primary decision-maker.

The first point employs SL's co-evolutionary phase as the traditional ML training step. Compared to LtD training, it assumes only one user, i.e., H , who is deemed reliable and who provides knowledge to M . The second point focuses on the quality of the model itself, i.e., to what extent it is aligned with the final decisions taken by H at the current time. Combined, those two points lead to avoiding training a separate deferral system for the MiC phase, as it is assumed M is a good approximation of H . With that said, the third point ensures H willingly agrees with putting the machine in command.

3.2 Machine-in-Command

If t_M is triggered, BRIDGET transitions to a state where M is in command, labelling incoming records individually. As soon as a new record x_i is received, M 's f computes its *belief*, i.e., its prediction probability, $b \in [0, 1]$ towards its prediction (line 14), following the first part of the fading skepticism:

$$b(x_i, \tilde{y}_i, Y, \tilde{Y}) = c_f(x_i, \tilde{y}_i) \cdot fea_f(x_i, \tilde{y}_i, X_p, Y_p)$$

If b is lower than a user-defined threshold β , M 's stance towards x_i is considered unreliable, and H is immediately called back to make their own decision \hat{y}_i (lines 15 and 17). In this state, M can explain why it is unreliable by providing, for instance, a small set of real or synthetic records with a low prediction probability similar to that achieved on x_i (line 16).

Otherwise, if $b > \beta$, in order to prevent machine overreliance [9], i.e., a common drawback of MiC systems, BRIDGET engages the user to check the model prediction on randomly selected record, not only those with a low belief. As a simple implementation, BRIDGET draws a random number $r \in [0, 1]$. If $r < \beta$, M 's prediction \tilde{y} is accepted as x_i 's label (line 18), otherwise the user H is called back (line 19). Also, in the MiC phase, due to the reasonable level of reliability according to the high b , BRIDGET can provide the user with the same forms of explanations as in the HiC phase (lines 20-21).

Finally, the behaviour of M 's model f after accepting a label differs between the HiC and MiC phases. Whereas f is always updated after each decision originated either from H or M in the HiC phase, in the MiC phase f and the various sets are only updated when the user is called back (line 23). This allows BRIDGET to notify the user if it reaches a critical status, i.e., if at least one of the following conditions is met (line 24):

1. the user has been called back due to M 's low b for more than p_{max} times;
2. the model's average FEA is now lower than a certain threshold.

The user can then decide to come back in command, reverting to the HiC phase. These checks indicate a decrease in model reliability, potentially implying concept-drift. Effectively, this doubles as a function commonly found in LtD systems, i.e., rejection of novelties and ambiguities [8].

4 Conclusion and Future Works

We have presented BRIDGET, an approach designed to bridge the two main Hybrid Decision-Making paradigms. BRIDGET uses a co-evolutionary process to train a ML model to closely mimic the user's behavior. This interactive dynamic between the human and machine agents allows for continual system parameter updates, resulting in alternating phases where either the human or the machine takes the lead. We plan to extensively test BRIDGET against stand-alone LtD systems and also consider in the implementation different data types, such as time series easily providing alternative types of explainability [10], and more in-depth functions such as fairness checks [6]. Moreover, we reckon BRIDGET should give a deeper focus on concept drift during phase transitions, as shifts in data are common reasons for a model's fall down. Lastly, the current iteration of BRIDGET was designed around two principles – employing FEA values as a model-agnostic proxy of the model's current status, and avoiding training an independent deferral system. Other approaches are possible, e.g., supplanting FEA values with the model's internal epistemic uncertainty or comparing the number of leaves of an incremental decision tree at two different points in time to assess the model's changes. Moreover, the user-provided decisions could effectively be used to train a deferral system at the end of the co-evolutionary phase.

Acknowledgment. This work is partially supported by the EU NextGenerationEU programme under the funding schemes PNRR-PE-AI FAIR, PNRR-SoBigData.it - Prot. IR0000013, H2020-INFRAIA-2019-1: Res. Infr. G.A. 871042 *SoBigData++*, TANGO G.A. 101120763, and ERC-2018-ADG G.A. 834756 *XAI*.

References

1. D. Banerjee, S. Teso, B. S. Grunel, and A. Passerini. Learning to guide human decision makers with vision-language models. *arXiv preprint arXiv:2403.16501*, 2024.
2. A. Bontempelli, S. Teso, et al. Learning in the wild with incremental skeptical gaussian processes. In *IJCAI*, pages 2886–2892. ijcai.org, 2020.
3. E. Hüllermeier and W. Waegeman. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *ML*, 110(3):457–506, 2021.
4. S. Joshi, S. Parbhoo, and F. Doshi-Velez. Pre-emptive learning-to-defer for sequential medical decision-making under uncertainty. *CoRR*, abs/2109.06312, 2021.
5. M. D. Lange et al. A continual learning survey: Defying forgetting in classification tasks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(7):3366–3385, 2022.
6. F. Mazzoni, R. Guidotti, and A. Malizia. A frank system for co-evolutionary hybrid decision-making. In *IDA (2)*, volume 14642 of *Lecture Notes in Computer Science*, pages 236–248. Springer, 2024.
7. K. L. Mosier et al. Human decision makers and automated decision aids: Made for each other? In *Automation and human performance*, pages 201–220. CRC, 2018.
8. C. Punzi, R. Pellungrini, M. Setzu, F. Giannotti, and D. Pedreschi. Ai, meet human: Learning paradigms for hybrid decision making systems. *arXiv preprint arXiv:2402.06287*, 2024.
9. C. Rastogi, Y. Zhang, D. Wei, K. R. Varshney, A. Dhurandhar, and R. Tomsett. Deciding fast and slow: The role of cognitive biases in ai-assisted decision-making. *Proceedings of the ACM on Human-computer Interaction*, 6(CSCW1):1–22, 2022.
10. F. Spinnato et al. Understanding any time series classifier with a subsequence-based explainer. *ACM TKDD*, 18(2):36:1–36:34, 2024.
11. S. Teso, A. Bontempelli, F. Giunchiglia, and A. Passerini. Interactive label cleaning with example-based explanations. In *NeurIPS*, pages 12966–12977, 2021.
12. S. Teso and K. Kersting. Explanatory interactive machine learning. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 239–245, 2019.
13. L. Wang et al. A comprehensive survey of continual learning: Theory, method and application. *CoRR*, abs/2302.00487, 2023.
14. T. Wang and M. Saar-Tsechansky. Augmented fairness: An interpretable model augmenting decision-makers’ fairness. *CoRR*, abs/2011.08398, 2020.
15. M. Zeni et al. Fixing mislabeling by human annotators leveraging conflict resolution and prior knowledge. *ACM IMWUT*, 3(1):32:1–32:23, 2019.
16. W. Zhang. Personal context recognition via skeptical learning. In *IJCAI*, pages 6482–6483. ijcai.org, 2019.