RiskFix: Supporting Expert Validation of Predictive Timeseries Models in High-Intensity Settings

G. Morgenshtern[1,5], A. Verma[2], S. Tonekaboni[2,3,4], R. Greer[3], J. Bernard[1,5], M. Mazwi[3], A. Goldenberg[2,3,4], and F. Chevalier[2]

[1] University of Zurich, Institute of Informatics, Switzerland
[2] University of Toronto, Department of Computer Science, Canada
[3] The Hospital for Sick Children, Canada
[4] Vector Institute, Canada
[5] Digital Society Initiative, Zurich, Switzerland

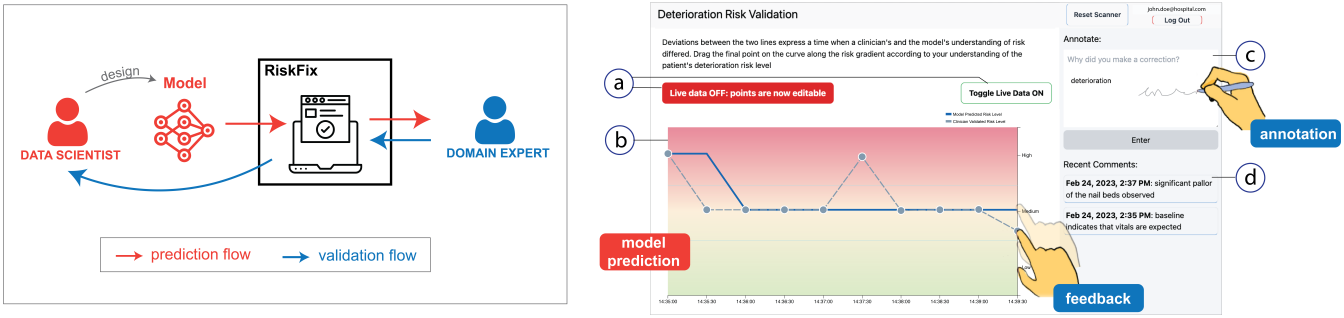

Figure 1: *RiskFix is a package supporting domain expert validation of predictive timeseries models designed by data scientists.* ***Left****: Underlying conceptual framework formalizing the information flows necessary for model validation: communication of predictions from the model to the domain expert, i.e. the* ***prediction flow****, and communication of input feedback from the expert to the data scientist who can further calibrate and validate the model, i.e. the* ***validation flow****.* ***Right****: RiskFix implements a user interface for the domain expert to visualize predictions (b), and input feedback and annotations used to validate the model. This screenshot shows a visualization of risk predictions of cardiac arrest in patients, which a clinician expert corrects by providing feedback through a drag-and-drop gesture (b), and further annotates by providing a comment (c). Past expert-validated data are visualized as a dashed line in the line chart, whereas model predictions are visualized as a solid line (b). Past annotations are also visible in the 'recent comments' widget (d). While entering comments, the live streaming of novel predictions is paused (a), to avoid bias and distraction while the expert enters free-form explanations and observations.*

Abstract

Many real-world machine learning workflows exist in longitudinal, interactive machine learning (ML) settings. This longitudinal nature is often due to incremental increasing of data, e.g., in clinical settings, where observations about patients evolve over their care period. Additionally, experts may become a bottleneck in the workflow, as their limited availability, combined with their role as human oracles, often leads to a lack of ground truth data. In such cases where ground truth data is small, the validation of interactive machine learning workflows relies on domain experts. Only those humans can assess the validity of a model prediction, especially in new situations that have been covered only weakly by available training data. Based on our experiences working with domain experts of a pediatric hospital's intensive care unit, we derive requirements for the design of support interfaces for the validation of interactive ML workflows in fast-paced, high-intensity environments. We present RiskFix, a software package optimized for the validation workflow of domain experts of such contexts. RiskFix is adapted to the cognitive resources and needs of domain experts in validating and giving feedback to the model. Also, RiskFix supports data scientists in their model-building work, with appropriate data structuring for the re-calibration (and possible retraining) of ML models.

CCS Concepts

• ***Computing methodologies*** → ***Model verification and validation;*** • ***Human-centered computing*** → *Open source software;* • ***Applied computing*** → *Health care information systems;*

DOI: 10.2312/evs.20231036

1. Introduction and Background

Interactive machine learning (ML) processes deployed in-the-wild are typically longitudinal endeavors. In many cases, deployed models are never "finished" calibrating; more high-quality labels could always be collected. Also, dataset sizes increase as the real-world case develops. In addition to (slowly) growing training data sizes, the situation is often impeded by the fact that real-world settings may not necessarily offer the possibility of acquiring many ground-truth labels from an involved expert group. This does not only pose challenges for model-building, but also for the *validation* of models. The validation of models solely based on statistical performance metrics does not suffice for its deployment context, for two reasons. First, these models initially need to be trained on some limited number of reliable training data with high-quality labels, or else by some proxy labels. Second, at least in the early stages of the longitudinal process, it cannot be guaranteed that all relevant real world phenomena are already captured well, at representative scale. For these both reasons, the validation of models must rely on domain experts.

Without a validation set, validation of a model refers to the task of comparing the model output with its real-life scenario. Validation by domain experts can only happen in the very moment when a prediction occurs, before additional predictions are made, the situation is resolved, or hindsight bias sets in. Additionally, a domain expert validating a model within a fast-paced, high-intensity context must stay aware of both the information the model is presenting, and simultaneously, how this information compares to the real-life scenario they are observing. If the main goal is for experts to enter useful and usable feedback on predictions of models deployed into their contexts, it is imperative that the cognitive load of validation tasks is reduced.

In this paper, we look at how this validation process can be implemented. We propose a conceptual framework, and an open-source package that implements an interface allowing domain experts to perform validation tasks in real-time.

Past efforts have made software openly available for easier handling of model augmentation [WKN*22, PNL*19, LBG*19]. Other recent human-model cooperation approaches have been presented solely as theoretical concepts, without an open-source artifact available [GA20, Shn20], and while these present helpful guidance, it is challenging to imagine such frameworks in practice. Here, we contribute both conceptual underpinnings (section 2), as well as an open-source, generalizable implementation of these concepts, to encourage replication and exploration of our methods.

Past approaches focused on continuous feedback, similarly to our approach, have been targeted on model calibration [WKN*22, BBH*20, LST20], rather than validation of the model in-the-wild. We conducted our own explorations into design requirements for such interfaces, as these approaches did not make their implementations available [BBH*20], concerned themselves with *ex-situ* validation scenarios [LBG*19, PNL*19, SSK*20], or did not deal with timeseries models [LST20, CRH*19, DAFC20, GA20, ERLO19], making their application challenging. Additionally, to meet user validation needs in expert settings, it is imperative to involve experts in the development process of any approach attempted [SSZ*17, WKN*22, dHLH*22, ZEM*22, Shn20, XDGG23].

Our motivating example is a clinical use case within the cardiac intensive care unit (ICU) of a pediatric hospital. This is our running example, due to the complexity of making correct predictions of patient status. The fast-paced, high-intensity, and high-stakes deployment environment of the ICU presents heterogeneous clinical pathways to a given predicted condition. For accurate model validation, clinicians cannot label a patient scenario that has already been resolved; by then, much of the necessary context is forgotten. Also, the lack of ground truth labels available, due to the heterogeneity of possible pathways leading to a clinical scenario, means a lack of available statistical validation metrics. Thus, a human expert is the only possible validator for a model deployed in this environment. Comparing a prediction against this real-life scenario adds significant workload to the ICU domain expert, both in terms of tasks performed and the attention required. This is a consideration that must be taken into account when building tools to support the validation process.

Our work makes the following contributions: (i) we formalize a conceptual framework for support interfaces enabling model validation in high-paced, high-intensity environments, (ii) we present RiskFix, a software solution that implements an interface embodying identified design requirements, released as an open-source *npm* package, to increase its accessibility to the community.

2. Conceptual Underpinnings

In the realm of expert-driven model calibration and validation, typically two stakeholders exist: **domain experts** and **data scientists**. Domain experts should be able to give feedback regarding the model performance, and do so in such a way that data scientists developing the models obtain usable and useful data for model validation and calibration. This outlines two core information flows our solution must support (Figure 1, left):

- **Prediction flow (→)**: the predictive output of the model
- **Validation flow (←)**: the feedback input of the domain expert

The solution should enable this two-way communication in a structured way that allows both stakeholders to achieve their goals. For domain experts: an expert-validated model. For data scientists: a domain-calibrated model.

To achieve these two goals, our solution must enable the following capabilities, each with their own requirements (**Ri**) to support:

Communication of model output (→). Presenting too much information (e.g. showing a new prediction every second, or showing a precise predictive value when a broad categorization would suffice) can be overwhelming, and practically impossible for the domain expert to thoughtfully validate. Conversely, presenting too little information (e.g. showing only the last prediction, or too broad a categorization for a prediction) can deprive the expert out of otherwise important context about the model which could help with their judgment and correction. To support an effective prediction flow, the solution must provide:

- Appropriate visual support for model output data type (**R1**).
- Appropriate recency and update frequency of new predictions, based on *domain expert*'s validation needs (**R2**).

Collection of feedback input (←). Collecting too much information (e.g., many different types of feedback data types, or giving feedback on too many data points at once) can be overwhelming

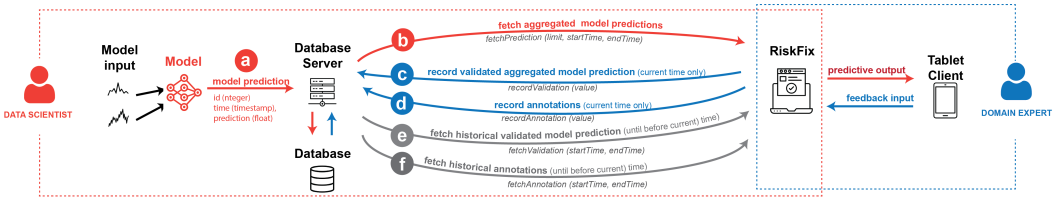

Figure 2: *Schematic view of RiskFix functionality supporting information flows. The* ***prediction flow*** *consists of (a) storing model predictions to a database, then (b) aggregating and fetching, for communication to the domain expert. The* ***validation flow*** *encompasses the feedback input provided by the domain expert through the RiskFix interface, which consists of either (c) validated aggregated model prediction, or (d) annotations, sent to the database for later retrieval by the data scientist. In gray, historical validations (e) and annotations (f) which are not directly part of either of the information flows, are fetched from the database for context in the visual interface (see Figure 1b)*

to the domain expert in a fast-paced environment. Conversely, collecting too little information (e.g., not enough richness in the feedback, not enough corrections to significantly refine model behavior) would reduce the effectiveness of the validation process. To support the validation flow, the following requirements must be satisfied:

- Appropriate input modality for feedback data type (e.g., numerical, categorical, textual), based on device used, and *domain expert and data scientist* needs (**R3**)
- Appropriate timing of feedback collection, based on considerations for usability and bias reductions for *domain experts*, and the data richness needs of *data scientists* (**R4**)

3. RiskFix: Design & Specifications

Building off the requirements (**R1–R4**), we designed RiskFix, a package which implements an interface supporting the information flows necessary for model calibration and validation (Figure 1). Figure 2 shows a schematic overview of the data flows relevant to our package's functionality. Here, we describe the design and specifications of RiskFix and its interface.

3.1. Model specifications

Our package supports predictive models which output timeseries data as new input data becomes available (i.e., a continuous or discrete score generated at regular time intervals). For instance, a model estimating a risk score based on continuous monitoring of clinical instrument measurements; or a model predicting gain based on continuous monitoring of stock markets.

While RiskFix is model-agnostic, its compatibility to a given model is predicated on a backend architecture that stores new predictions into a database as soon as the model outputs them (Figure 2a). The database should store the timestamp of when the prediction was made, and the predicted value (typically $\in [0, 1]$).

3.2. Communication of the model output

RiskFix employs a traditional line chart representation to convey the model prediction to the domain expert, chosen so that the interface comprises itself of components already within most users' data literacy level (**R1**) (see Figure 1b). The temporal resolution of the x-axis, i.e., the number of historical predictions displayed, and frequency at which new predictions are communicated to the expert, should be set to satisfy the expert's needs and constraints for performing validation (**R2**). That is, it must account for the cognitive load associated with *assessment*, which encompasses forming one's own judgement of the current situation, processing the model's prediction, and comparing the two; as well as the time needed to *provide feedback*, if any. Typically, several dozens of seconds are required, at the very least.

Since models are often capable of outputting new predictions at a high frequency, RiskFix was designed with the assumption that several predictions would be stored in the database in the time between refreshes of the visualization. The function that fetches a new prediction to display, `fetchPrediction` (Figure 2b), therefore expects data scientists to define what is the most appropriate value to show, i.e., the newest available prediction, or some aggregate of predictions generated since the last displayed value. Parameters necessary for the data scientist to write their custom database query are included, i.e. the time window covered by the visualization (`startTime`, `endTime`) and number of predictions plotted over that time window (`limit`), derived from constants `timeWindow` and `fetchInterval` in the main source code. When a new aggregated prediction is fetched, RiskFix automatically updates the chart, sliding the temporal axis to fit the new data point.

Similarly, for models that output prediction values on a continuous scale, discretization into meaningful ranges could be more interpretable to the domain expert than abstract, but precise, scores (**R1**). Our current implementation discretizes (aggregated) prediction values into three bins: high (numerically rounded as 1.0), medium (rounded as 0.5), low (rounded as 0). The domain expert is presented with a visualization of predictions plotted at these discrete categories (solid line in Figure 1b). To communicate that the predictions are more nuanced, RiskFix overlays the line chart on a gradient background. The published package does not include wrapping API functions to further customize discretization of the aggregated prediction values and the visualization's background easily. For custom background and y-axis specifications, modification of the original source code is necessary.

3.3. Feedback input specifications

The domain expert is able to provide two types of feedback for model validation: ordinal prediction validation values, and free-text annotations. The expert-**validated values** map to the model's output domain; and the **free-text annotations** provide additional con-

text that may be useful to data scientists trying to better understand when, and why, the model fails (**R3**).

3.4. Collection of feedback input

Expert-validated values are collected through (1) the pre-bias input prompt; and (2) direct manipulation of the expert-validated line, represented by a dashed line in the visualization (Figure 1b). If their expert-validated assessment differs from the model aggregated prediction, a drag-and-drop interaction allows experts to modify the plotted value from the default (the model prediction), to their assessment value (**R3**). The interaction paradigm we employ is such that the expert is in agreement with the model, unless otherwise specified. That is, expert input is only required when there is disagreement with the model. This is to minimize the burden of validation effort that would otherwise come with explicit feedback on every single prediction (**R4**). Further, because validation must happen in-the-moment, while the context for assessment is still available, RiskFix restricts expert-validated assessment to the present time– the only rightmost point of the dashed line can be modified. This is to avoid hindsight bias: experts reconsidering past assessments due to information that was not available at the time prediction was made (**R4**). Experts can also enter free-form textual annotations, by typing or writing these with a stylus (Figure 1c). Unlike expert-validated values, annotations can be submitted at any time. The expert can use these to provide general comments and observations, as well as constructive explanations for why they disagreed with the model. Free-form annotations support rich input necessary for the data scientist to understand what piece of information the model may be missing, as well as the expert's reasoning behind an assessment (**R3**).

Historical validations are fetched from the database to build the dashed line (Figure 2e). Expert-validated values are then stored each time a new aggregated prediction is fetched for display (see Figure 2c). Annotations are posted to the database at submission (Figure 2d), and then fetched from the database for display on the interface (Figure 1d; Figure 2f).

3.5. Additional features and implementation

RiskFix incorporates features that enhance its usability in real-world settings: secure log-in and a QR reader allow for secure identification of users, data sources to be used as model input, and communication of data source identifiers to the database; a pre-bias input prompt, allowing for collection of an expert's initial assessment *prior to* reveal of the model predictions; and visual design features to support meaningful display of missing predictions.

We open-source our standalone web-based package at `https:github.com/vermaarn/riskfix`. RiskFix is built with Typescript, using the React.js, D3.js, React-QR-Scanner, and TailwindCSS libraries. This package is published via *npm* at: `https:npmjs.com/package/riskfix`.

4. Deployment & Validation

RiskFix is the result of an iterative co-design process including HCI and ML researchers (data scientists) and clinicians (domain experts), and feedback from external evaluators. We have applied our solution to a model which predicts risk of cardiac arrest in patients, based on physiological signals [TML*18]. Figure 1b shows a screenshot of our use case, where ten aggregated predictions are displayed, i.e. one per 30-second interval in the last 5 minutes of patient history, as defined by clinicians who indicated that this time interval corresponds to the time it takes them to evaluate the patient condition. We validated our approach with six clinicians external to our research team, in a usability study. Participants were asked to validate the model by using the RiskFix workflow in several clinical scenarios set up in a simulated ICU environment, with a mannequin patient undergoing various status and vital sign changes. The scenarios' predicted risk scores were such that they were sometimes concordant and sometimes discordant, to provide participants the experience of both agreeing and disagreeing with the model output. Then, a focus group served as a debrief for the usability study. The package described in this paper is the refined package after integrating feedback from this study. A video demonstrating RiskFix is available as supplemental material.

We now plan to plug RiskFix into a large scale silent trial at our partner institution [TMA*22]. This will allow us to observe how effective the tool is at facilitating human validation of the institution's risk prediction model, and whether the tool allows clinicians to feel more confident in the validation process.

5. Disscussion and Conclusion

Our work sheds light on the hard human-factor constraints existing in support of model validation tasks in fast-paced, high-intensity environments. Here, traditional validation approaches are not possible, and predictions can only be validated in-the-moment, and by domain experts. But domain experts cannot exert time or effort beyond what is typically allocated to assessment of a situation in their practice. Thus, even if a model is fast and precise, validation is bottle-necked by the *human* expert's needs and abilities. We identify two core information flows necessary for supporting expert validation in high-intensity settings: that of a model communicating predictions to the expert; and that of the expert providing feedback for model validation. We contribute a conceptual framework defining the requirements presented by these flows, and present RiskFix, an open-source package implementing these requirements.

While we evaluate our workflow on the specific case of assessing risk of cardiac arrest in patients, our approach is model-agnostic, and generalizable to models which output timeseries data. Refinements to the visual interface may be necessary to accommodate other experts, and domain' requirements (e.g. several time series).

Our research raises an important question: if the model and human validator do not operate at the same level or precision or frequency of prediction output, how applicable is the validation feedback to the models? It would not be reasonable to scale down the resolution of a model to match that of a human's processing abilities. How aggregated validation is to be reconciled with a model remains a challenge, and its resolution calls for extensive, in-the-moment, data collection from experts. We begin this effort by applying RiskFix to a real-world clinical scenario, and hope to see other researchers deploying our approach to their contexts as well, in an effort to expedite the conquering of this challenge.

References

[BBH*20] BEEDE E., BAYLOR E., HERSCH F., IURCHENKO A., WILCOX L., RUAMVIBOONSUK P., VARDOULAKIS L. M.: A human-centered evaluation of a deep learning system deployed in clinics for the detection of diabetic retinopathy. In *CHI '20: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2020), ACM, pp. 1–12. 2

[CRH*19] CAI C. J., REIF E., HEGDE N., HIPP J., KIM B., SMILKOV D., WATTENBERG M., VIEGAS F., CORRADO G. S., STUMPE M. C., OTHERS: Human-centered tools for coping with imperfect algorithms during medical decision-making. In *CHI '19: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2019), ACM, pp. 1–14. 2

[DAFC20] DE-ARTEAGA M., FOGLIATO R., CHOULDECHOVA A.: A case for humans-in-the-loop: Decisions in the presence of erroneous algorithmic scores. In *CHI '20: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2020), ACM, pp. 1–12. 2

[dHLH*22] DE HOND A. A., LEEUWENBERG A. M., HOOFT L., KANT I. M., NIJMAN S. W., VAN OS H. J., AARDOOM J. J., DEBRAY T. P., SCHUIT E., VAN SMEDEN M., OTHERS: Guidelines and quality criteria for artificial intelligence-based prediction models in healthcare: a scoping review. *NPJ Digital Medicine 5*, 1 (2022), 2. Publisher: Nature Publishing Group UK London. 2

[ERLO19] EKER S., ROVENSKAYA E., LANGAN S., OBERSTEINER M.: Model validation: A bibliometric analysis of the literature. *Environmental Modelling & Software 117* (2019), 43–54. Publisher: Elsevier. 2

[GA20] GRØNSUND T., AANESTAD M.: Augmenting the algorithm: Emerging human-in-the-loop work configurations. *The Journal of Strategic Information Systems 29*, 2 (2020), 101614. Publisher: Elsevier. 2

[LBG*19] LÖNING M., BAGNALL A., GANESH S., KAZAKOV V., LINES J., KIRÁLY F. J.: sktime: A unified interface for machine learning with time series. *arXiv preprint arXiv:1909.07872* (2019). 2

[LST20] LERTVITTAYAKUMJORN P., SPECIA L., TONI F.: FIND: human-in-the-loop debugging deep text classifiers. *arXiv preprint arXiv:2010.04987* (2020). 2

[PNL*19] PHAM V., NGUYEN N., LI J., HASS J., CHEN Y., DANG T.: MTSAD: Multivariate Time Series Abnormality Detection and Visualization. In *2019 IEEE International Conference on Big Data* (2019), IEEE, pp. 3267–3276. 2

[Shn20] SHNEIDERMAN B.: Human-centered artificial intelligence: Reliable, safe & trustworthy. *International Journal of Human–Computer Interaction 36*, 6 (2020), 495–504. Publisher: Taylor & Francis. 2

[SSK*20] SON J., SHIN J. Y., KIM H. D., JUNG K.-H., PARK K. H., PARK S. J.: Development and validation of deep learning models for screening multiple abnormal findings in retinal fundus images. *Ophthalmology 127*, 1 (2020), 85–94. Publisher: Elsevier. 2

[SSZ*17] SACHA D., SEDLMAIR M., ZHANG L., LEE J. A., PELTONEN J., WEISKOPF D., NORTH S. C., KEIM D. A.: What you see is what you can change: Human-centered machine learning by interactive visualization. *Neurocomputing 268* (2017), 164–175. Publisher: Elsevier. 2

[TMA*22] TONEKABONI S., MORGENSHTERN G., ASSADI A., POKHREL A., HUANG X., JAYARAJAN A., GREER R., PEKHIMENKO G., MCCRADDEN M., CHEVALIER F., MAZWI M., GOLDENBERG A.: How to validate Machine Learning Models Prior to Deployment: Silent trial protocol for evaluation of real-time models at ICU. In *Conference on Health, Inference, and Learning* (2022), PMLR, pp. 169–182. 4

[TML*18] TONEKABONI S., MAZWI M., LAUSSEN P., EYTAN D., GREER R., GOODFELLOW S. D., GOODWIN A., BRUDNO M., GOLDENBERG A.: Prediction of cardiac arrest from physiological signals in the pediatric ICU. In *Machine Learning for Healthcare Conference* (2018), PMLR, pp. 534–550. 4

[WKN*22] WANG Z. J., KALE A., NORI H., STELLA P., NUNNALLY M. E., CHAU D. H., VORVOREANU M., WORTMAN VAUGHAN J., CARUANA R.: Interpretability, Then What? Editing Machine Learning Models to Reflect Human Knowledge and Values. In *KDD '22: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2022), ACM, pp. 4132–4142. 2

[XDGG23] XU W., DAINOFF M. J., GE L., GAO Z.: Transitioning to human interaction with AI systems: New challenges and opportunities for HCI professionals to enable human-centered AI. *International Journal of Human–Computer Interaction 39*, 3 (2023), 494–518. Publisher: Taylor & Francis. 2

[ZEM*22] ZHANG M., EHRMANN D., MAZWI M., EYTAN D., GHASSEMI M., CHEVALIER F.: Get To The Point! Problem-Based Curated Data Views To Augment Care For Critically Ill Patients. In *CHI '22: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2022), ACM, pp. 1–13. 2