A Fraud Detection Dataset for Learning to Defer

Jean V. Alves
Feedzai
Universidade de Lisboa
Lisboa, Portugal
jean.alves@feedzai.com

Diogo Leitão
IDK
Lisboa, Portugal
idon'tknow

Sérgio Jesus
Feedzai
Lisboa, Portugal
sergio.jesus@feedzai.com

Marco O. P. Sampaio
Feedzai
Lisboa, Portugal
marco.sampaio@feedzai.com

Pedro Saleiro
Feedzai
Lisboa, Portugal
pedro.saleiro@feedzai.com

Mário A. T. Figueiredo
Feedzai
Universidade de Lisboa
Lisboa, Portugal
mario.figueiredo@tecnico.ulisboa.pt

Pedro Bizarro
Feedzai
Lisboa, Portugal
pedro.bizarro@feedzai.com

ABSTRACT

Public dataset limitations have significantly hindered the development and benchmarking of *learning to defer* (L2D) algorithms, which aim to optimally combine human and AI capabilities in hybrid decision-making systems. In such systems, human availability and domain-specific concerns introduce complexity, while obtaining human predictions for training and evaluation is costly. Financial fraud detection is a high-stakes setting where often algorithms and human experts work in tandem, however, there are no publicly available datasets for L2D concerning this key application of human-AI teaming. To fill this gap in L2D research, we introduce a synthetic bank account fraud detection dataset, containing the predictions of a team of 50 highly complex and varied synthetic fraud analysts, with adjustable bias and feature dependence. We also provide a realistic definition of human work capacity constraints, an aspect of L2D systems which is often overlooked, allowing for extensive testing of assignment systems under real-world conditions. We use our dataset to develop a capacity aware L2D method and rejection learning approach under realistic data availability conditions, benchmarking these baselines under an array of 300 distinct testing scenarios. We believe that this dataset will serve as a pivotal instrument in facilitating a systematic, rigorous, reproducible, and transparent evaluation and comparison of L2D methods, thereby fostering the development of more synergistic human-AI collaboration in decision-making systems. The instantiated public dataset and detailed synthetic expert information are available at: https://anonymous.4open.science/r/openl2d-7BD3

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY
© 2023 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

KEYWORDS

learning to defer, human-ai collaboration

ACM Reference Format:
Jean V. Alves, Diogo Leitão, Sérgio Jesus, Marco O. P. Sampaio, Pedro Saleiro, Mário A. T. Figueiredo, and Pedro Bizarro. 2023. A Fraud Detection Dataset for Learning to Defer. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX).* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Recently, an increasing body of research has been dedicated to studying human-AI collaboration (HAIC), with several authors arguing that humans have complementary sets of strengths and weaknesses to those of AI [11, 12]. Collaborative systems have demonstrated that humans are able to rectify model predictions in specific instances [11], and have shown that humans, in collaboration with ML models, may achieve synergistic performance - a higher performance than the expert or the model on their own [21].

The state-of-the-art approach to manage assignments in human-AI collaboration is *learning to defer* (L2D) [5, 18, 30–34]. These are algorithms that choose whether to assign an instance to a human or a ML model, aiming to take advantage of their complementary strengths. L2D algorithms require large amounts of data on human decisions: some require multiple human predictions per instance [33, 34], while others often require human predictions to exist for every single training instance [18, 30, 32, 38]. Due to the unavailability of large datasets containing human predictions, and the cost of obtaining large amounts of data annotated by human experts, these methods are frequently developed with small datasets, containing limited human predictions, or by using synthetic human subjects. The synthesized expert behavior is often simplistic, and varies significantly between authors. Consequently, research into L2D is lacking in robust benchmarking of different methods.

Financial fraud detection is a high-stakes use case where human-AI collaboration is often applied. Machine Learning models can be used in anti-money laundering, where an automated system monitors transactions, raising alerts that are then reviewed by human-experts [28]. In e-commerce transaction fraud detection, ML models' advice may help improve the accuracy of human decision-makers, as well as expedite the decision making process [2]. However, research into applying L2D in fraud prevention settings is lacking, possibly due to a lack of adequate public datasets in this domain.

To address this issue, we present the OpenL2D Bank Account Fraud Dataset, which includes the predictions of a team of 50 highly complex synthetic experts, generated in order to simulate a wide variety of human behaviours. We use a novel approach to generate complex synthetic experts, with control over performance, feature dependence and bias towards a protected attribute; and define capacity constraints limiting the amount of instances that can be deferred to each expert. We also create a version of our dataset simulating realistic data availability conditions (only one expert prediction per instance) during training, thus providing realistic training and testing scenarios for L2D research. Subject to these conditions, we develop a capacity aware L2D algorithm, and benchmark two versions of our method as well as a capacity aware version of *rejection learning*, by testing their performance and fairness under 300 different testing scenarios. We hope to bolster research into development and testing of L2D methods subject to real-world problems, such as changes in human availability and limited amounts of human prediction data. Our dataset and the code utilized to generate it is available at: https://anonymous.4open.science/r/openl2d-7BD3

2 BACKGROUND AND RELATED WORK

In this section we discuss the most commonly used datasets in HAIC research, methods of synthetic expert generation in L2D research, and the current state-of-the-art L2D approaches.

2.1 Current HAIC Datasets

A Dataset suitable for L2D training and evaluation has to comply with a few requirements. Firstly, the dataset must contain a sizeable amount of predictions from each member of the expert team, to enable modeling of the human behavior. The human which made each prediction must be identifiable, allowing for individual modelling of each expert's behavior. Finally, in testing, we must have a set of each expert's predictions for every instance in the test set, as the assignment system may query any expert on a given instance. To the best of our knowledge, there are only two public real-world datasets suitable for training multi-expert human-AI assignment systems, which we now describe. The NIH Clinical Center X-ray dataset [39], used by Hemmer et al. [18], is a computer vision dataset aimed at detecting airspace opacity. For each X-ray image, there are recorded predictions from an ensemble of 22 experts, and a golden label created by an independent team of 3 radiologists. The main drawback of this dataset is its size: it contains only 4,374 X-ray images. The Hate Speech and Offensive Language Detection dataset enriched by Keswani et al. [25], consists of a subset of 1,471 tweets from the original dataset [10], annotated by a total of 170 crowd-sourcing workers according to the presence of hate speech or offensive language. Each tweet was labelled by an average of 10 workers, meaning that each worker labelled an average of 87 instances. The low volume of instances hinders the capacity to model individual expertise conditioned to the input space.

2.2 Simulation of Human Experts

Due to the lack of adequate real-world datasets, several authors have resorted to synthesizing expert behavior for datasets in the ML literature. Madras et al. [30] propose a *model-as-expert* technique, fitting a ML classifier to mimic expert behavior on two binary classification datasets (COMPAS [26] and Heritage Health [20]). They use the same ML algorithm used for the main task with extra features, in order to simulate access to exclusive information. These authors also introduce bias, with the goal of studying unfairness mitigation. Verma and Nalisnick [38] use a similar approach to produce an expert on the HAM10000 dataset [36]. In a *model-as-expert* approach, the modelling bias of the ML algorithm is the same for the main classifier and the synthetic experts. Furthermore, they are trained on data with a large fraction of features in common. This may lead to artificially large agreement between the classifier and experts, which is why we choose not to use it.

Other authors use a *label noise* approach to produce arbitrarily accurate expert predictions. Mozannar and Sontag use CIFAR-10 [27], where they simulate an expert with perfect accuracy on a fraction of the 10 classes, but random accuracy on the others (see also Verma and Nalisnick [38] and Charusaie et al. [5]). The main drawback of these synthetic experts is that their expertise is either feature-independent or only dependent on a single feature or concept. As such, the methods tested on these benchmarks are not being challenged to learn nuanced and varied types of expertise. This type of approach has been criticised. Zhu et al. [40] and Berthon et al. [3] argue that *instance-dependent label noise* (IDN) is more realistic, as human error is more likely to be dependent on the difficulty of a given task, and, as such, should also be dependent on the instance's features. Our approach will make use of *instance-dependent label noise.*

2.3 Current L2D Methods

One of the simplest deferral approaches in the literature is given by *rejection learning* (ReL) [6, 8]. In a human-AI collaboration setting, ReL defers instances from the model to humans [30, 33]. Its simplest implementation [19] produces uncertainty estimates for the model's prediction in each instance, ranking them, and rejecting to predict if the uncertainty is above a given threshold [6, 8].

Madras et al. [30] argue that ReL is sub-optimal because it does not consider the performance of the human(s) involved in the task, so they propose *learning to defer* (L2D). In the original L2D framework, the classifier and assignment system are jointly trained, taking into account a single model and a single human. Many authors have since contributed to the single-expert framework [32, 38]. Keswani et al. [25] observe that decisions can often be deferred to one or more humans out of a team, expanding L2D to the multi-expert setting [25], followed by Hemmer et al. [18], Verma et al. [37]. Most L2D approaches require predictions from every human team member, for every training instance, imposing significant, and often unrealistic, data requirements. Furthermore, the limited work capacity of each team member often goes unaddressed.

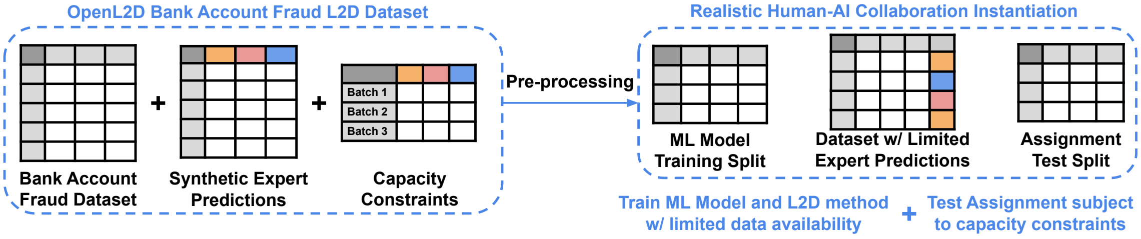
OpenL2D Bank Account Fraud L2D Dataset
Bank Account Fraud Dataset
+
Synthetic Expert Predictions
+
Batch 1
Batch 2
Batch 3
Capacity Constraints
Pre-processing
Realistic Human-AI Collaboration Instantiation
ML Model Training Split
Dataset w/ Limited Expert Predictions
Assignment Test Split
Train ML Model and L2D method w/ limited data availability
+
Test Assignment subject to capacity constraints

Figure 1: OpenL2D Bank Account Fraud Dataset

In conclusion currently available L2D algorithms require realistic datasets for development and testing under realistic conditions. As it is currently unfeasible to collect real world expert predictions for large datasets, a promising avenue is to develop methods to generate synthetic expert predictions that look realistic.

3 DATASET AND HAIC SCENARIO

The OpenL2D Bank Account Fraud Dataset is comprised by three components, represented in Figure 1: a base dataset, which contains each instances features; a table containing each synthetic expert's predictions for each instance (see Section 3.2); and a set of capacity constraints, detailing the amount of instances that each expert can process in a given time interval (see Section 3.4). Using our dataset, we instantiate a scenario simulating the development and testing of a L2D system under realistic conditions (see Section 3.5).

3.1 Base Dataset

As the base dataset, we choose to use the publicly available bank-account-fraud tabular dataset [23]. This dataset is sizeable (one million rows) and boasts other key properties that are relevant for our use case. The data was generated by applying tabular data generation techniques on an anonymized, real-world bank account opening fraud detection dataset. Each instance represents a bank account opening application, with features containing information about the application and the applicant, and a label that denotes if the instance is a fraudulent (1) or a legitimate (0) application.

The task of a decision maker (automated or human) is to either accept (predicted negative) or reject (predicted positive) it. A positive prediction results in a declined application. As such, false positives in account opening fraud can significantly affect a person's life (with no possibility to open a bank account or to access credit). This is thus a cost-sensitive problem, where the cost of a false positive must be weighed against the cost of a false negative. The optimization goal is to maximize recall at a fixed FPR (we use 5%), which implicitly establishes a relationship between the costs. This task also entails fairness concerns, as ML models trained on this dataset tend to raise more false fraud alerts for older clients ($\geq$ 50 years), thus reducing their access to a bank account.

3.2 Decision Generation Method

Our expert generation approach is based on *instance-dependent noise*, in order to obtain more realistic experts, whose probability of error varies with the properties of each instance. We generate synthetic predictions by flipping each label y_i with probability $\mathbb{P}(m_{i,j} \neq y_i | \boldsymbol{x}_i, y_i)$. In some HAIC systems, the model score for a given instance may also be shown to the expert [2, 11, 29], so an expert's decision may also be dependent on an ML model score $m(\boldsymbol{x}_i)$. We define the expert's probabilities of error, for a given instance, as a function of its features, $\boldsymbol{x}_i$, and the model score $m(\boldsymbol{x}_i)$,

$$\begin{cases} \mathbb{P}(m_{i,j} = 1 | y_i = 0, \boldsymbol{x}_i, M) = \sigma\left(\beta_0 - \alpha \frac{\mathbf{w} \cdot \boldsymbol{x}_i + w_M M(\boldsymbol{x}_i)}{\sqrt{\mathbf{w} \cdot \mathbf{w} + w_M^2}}\right) \\ \mathbb{P}(m_{i,j} = 0 | y_i = 1, \boldsymbol{x}_i, M) = \sigma\left(\beta_1 + \alpha \frac{\mathbf{w} \cdot \boldsymbol{x}_i + w_M M(\boldsymbol{x}_i)}{\sqrt{\mathbf{w} \cdot \mathbf{w} + w_M^2}}\right), \end{cases}$$

$$M(\mathbf{x}_i) = \begin{cases} \frac{m(\boldsymbol{x}_i) - t}{2t}, & m \leq t \\ \frac{m(\boldsymbol{x}_i) - t}{2(1-t)}, & m > t, \end{cases}$$

where σ denotes a sigmoid function and M is a transformed version of the original model score $m \in [0, 1]$. Each expert's probabilities of the two types of error are parameterized by five variables: $\beta_0, \beta_1, \alpha, \mathbf{w}$ and w_M. The weight vector $\mathbf{w}$ embodies a relation between the features and the probability of error. To impose a dependence on the model score, we can set $w_M \neq 0$. The feature weights are normalized so that we can separately control, with the α parameter, the overall magnitude of the variation of the probability of error due to the instance's features. The values of β_1 and β_0 control the base probability of error.

3.3 Human Decision-Making Properties

In this section we list the characteristics of human decision-making that we aim to capture with our approach, in order to make our synthetic experts as realistic as possible.

Feature and AI assistant dependence: When a decision is made by an expert, it is assumed that they will base themselves on information related to the instance in question. Therefore, we expect experts to be dependent on the instance's features.

In some real world deferral systems [2, 11], the instance's features are accompanied by an AI model's score, representing the model's estimate of the probability that said instance belongs to the positive class. The aim of presenting the model score to an expert is to provide them with extra information, as well as possibly expediting the decision process. It has been shown that, in this scenario, expert's performance can be impacted by presenting the model's score when deferring a case to an expert [2, 11, 29]. We should then consider that our experts are impacted by the ML classifier.

AI assistance and algorithmic bias: Should the generated experts use an AI assistant, we expect experts not to be in perfect agreement with the model, due to the assumption that humans and models have complementary strengths and weaknesses [11, 12].

As such, we would assume humans and AI perform better than one another in separate regions of the feature space. The degree of "algorithmic bias" [1] varies between humans, measured as the model's impact on a human's performance [21, 22]. As such, our synthetic experts also exhibit varying levels of model dependence.

Varied Expert Performance In order for our team of experts to be realistic, it is important that these exhibit varying levels of overall performance. Experts within a field have been shown to have varying degrees of expertise, with some being outperformed by ML models [14, 16]. As such human decision processes can be expected to be varied even amongst a team of experts.

Predictability and Consistency It is a common assumption that, when making a decision, experts follow an internal process based on the available information. However, it is also known that even experts are still subject to flaws that are inherent to human decision making processes, one of these being inconsistency. When presented with similar cases, at different times, experts may perform drastically different decisions [9, 15]. Therefore we can expect a human's decision making process not to be entirely deterministic.

Human Bias and Unfairness It is also important to consider the role that the assignment system can play in mitigating unfairness. If an expert can be determined to be particularly unfair with respect to a given protected attribute, the assignment system can learn not to defer certain cases to that expert. In order to test the fairness of the system as a whole, it is useful to create a team of individuals with varying propensity for unfair decisions.

To simulate a wide variety of human behavior, we created four distinct expert groups. All groups have similar performance as measured by their TPR and FPR, with a fraction of the team performing worse than the ML Model. The first is a *Standard* expert group: on average as unfair as the model, dependent on model score and, on average, twelve different features. The three other types of expert are variations on the *Standard* group. i) Unfair: experts which are more likely to incorrectly reject an older customer's application. ii) *Model agreeing*: experts which are heavily swayed by the model score. iii) *Sparse*: experts which are dependent on fewer features.

3.4 Definition of Capacity Constraints

Firstly, we formalize how to define capacity constraints. Humans are limited in the number of instances they may process in any given time period (e.g., work day). In real-world systems, human capacity must be applied over batches of instances, not over the whole dataset at once (e.g. balancing the human workload over an entire month is not the same as balancing it daily). A real-world assignment system must then process instances taking into account the human limitations over a given "batch" of cases, corresponding to a pre-defined time period. We divide our dataset into several batches and, for each batch, define the maximum number of instances that can be processed by each expert. In any given dataset comprised of N instances, capacity constraints can be represented by a vector $\boldsymbol{b}$, where component b_i denotes which batch instance $i \in \{1, ..., N\}$ belongs to, as well as a human capacity matrix H, where element $H_{b,j}$ is a non-negative integer denoting the number of instances expert j can process in batch b.

To define the batch vector, we have to define the number of cases in each batch, then distribute instances throughout the batches. To define the capacity matrix, we consider 4 separate parameters. (1) Deferral_rate: maximum fraction of each batch that can be deferred to the human team; (2) Distribution *homogeneous* or *variable*. Should the distribution be *homogeneous*, every expert will have the same capacity; otherwise, each expert's capacity is sampled from $\mathcal{N}(\mu_d = \text{Deferral\_rate} \times \text{Batch\_Size}/N_{experts}, 0.2 \times \mu_d)$, chosen so that each expert's capacity fluctuates around the value corresponding to an homogeneous distribution; (3) Absence rate, defined as the fraction of experts that are absent in each batch. This allows for testing several different team configurations without generating new experts, or scenarios where not all experts work in the same time periods. (4) Expert Pool, defined as which types of experts (standard, unfair, sparse or model agreeing) can be part of the team.

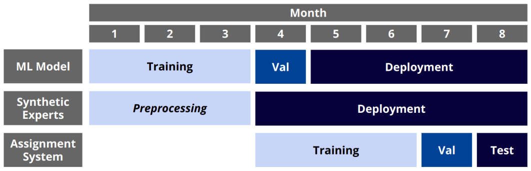

Figure 2: Temporal Splits for L2D Development

To allow for extensive testing, we create a vast set of capacity constraints. In Table 1, under "Scenario Properties", we list the different combinations of settings used. For each combination, several seeds were set for the batch, expert absence, and capacity sampling.

3.5 HAIC Setup

We choose temporal splits of the dataset that aim to emulate a realistic scenario as close as possible, represented in figure 2. To do so, we first train a fraud detection ML classifier. This model is trained on the first three months of the dataset and validated on the fourth month. We utilize the LightGBM [24] algorithm, due to its proven high performance on tabular data [4, 35]. The ML model yields a recall of 57.9% in validation, for a threshold of $t = 0.051$, chosen in validation to obtain 5% FPR.

Our simulated experts are assumed to act alongside the ML model on the period ranging from the fourth to the eighth month. There are several possible ways for models and humans to cooperate. In L2D testing, it is often assumed any instance can be deferred to either the model or the expert team. However, in a real world setting, it is common to use an ML model to raise alerts that are then reviewed by human experts [11, 17]. Without an assignment method, the decision system would function as follows: a batch of instances is processed by the model, a fraction of the highest scoring instances are flagged for human review, and, finally, these instances are randomly assigned to experts, who make the final decision. The rest of the instances are automatically accepted.

By assuming that the alert review system is employed from months four to seven, we can construct a dataset that would correspond to human predictions gathered over this period. Using this data, we train our assignment algorithms with the data of months four to six, validating them on the seventh month. Testing is done by creating a new deferral system, where the cases flagged by the ML classifier for review are distributed to the humans according to an assignment algorithm trained on the gathered data.

Table 1: Baseline Results. Intervals denote standard deviation. FPR disparity (FPR$_d$) standard deviations are omitted due to low variability. "Model Only" represents a fully automated baseline, with predictions made by the model, according to threshold t.

Scenario Properties					Model Only		ReL		ReL$_{\text{greedy}}$		ReL$_{\text{linear}}$	
Pool	Batch Size	Deferral Rate	Absence Rate	σ_d	Loss	PE	Loss	PE	Loss	PE	Loss	PE
all	250	0.2	0.0	0.2	918	0.33	**753**±12	0.29	780±10	0.31	780±9	0.32
all	250	0.2	0.0	0.0	918	0.33	**755**±14	0.30	781±8	0.31	789±8	0.32
all	250	0.2	0.5	0.2	918	0.33	**760**±11	0.29	788±9	0.31	782±9	0.32
all	250	0.2	0.5	0.0	918	0.33	**768**±10	0.29	788±11	0.31	786±10	0.32
all	250	0.5	0.0	0.2	918	0.33	**746**±17	0.29	788±7	0.34	766±9	0.36
all	250	0.5	0.0	0.0	918	0.33	**759**±14	0.29	790±4	0.34	765±13	0.36
all	250	0.5	0.5	0.2	918	0.33	**756**±13	0.29	779±7	0.32	782±8	0.36
all	250	0.5	0.5	0.0	918	0.33	**754**±11	0.29	783±6	0.32	783±5	0.36
all	5000	0.2	0.0	0.2	918	0.33	**752**±8	0.30	780±4	0.32	779±5	0.33
all	5000	0.2	0.0	0.0	918	0.33	**752**±12	0.30	778±3	0.32	782±5	0.33
all	5000	0.2	0.5	0.2	918	0.33	**762**±10	0.30	778±12	0.31	773±4	0.33
all	5000	0.2	0.5	0.0	918	0.33	**753**±9	0.30	776±11	0.31	776±3	0.33
all	5000	0.5	0.0	0.2	918	0.33	**749**±8	0.29	774±6	0.34	768±6	0.36
all	5000	0.5	0.0	0.0	918	0.33	**750**±11	0.29	776±8	0.34	768±1	0.36
all	5000	0.5	0.5	0.2	918	0.33	**759**±12	0.29	774±7	0.32	780±8	0.37
all	5000	0.5	0.5	0.0	918	0.33	**758**±8	0.29	773±6	0.33	781±7	0.37

Table 2: Varying Expert Pool Results. Nomenclature used is consistent with Table 1

Scenario Properties					ReL		ReL$_{\text{greedy}}$		ReL$_{\text{linear}}$	
Pool	Batch Size	Deferral Rate	Absence Rate	σ_d	Loss	PE	Loss	PE	Loss	PE
agreeing	250	0.2	0.0	0.0	813±8	0.37	873±7	0.37	**810**±3	0.34
agreeing	250	0.5	0.0	0.0	815±11	0.39	900±4	0.40	**783**±5	0.36
agreeing	5000	0.2	0.0	0.0	816±7	0.37	875±4	0.37	**808**±5	0.34
agreeing	5000	0.5	0.0	0.0	814±12	0.39	905±3	0.40	**784**±3	0.36
sparse	250	0.2	0.0	0.0	766±9	0.29	770±6	0.31	**755**±6	0.31
sparse	250	0.5	0.0	0.0	752±11	0.28	738±8	0.31	**737**±11	0.34
sparse	5000	0.2	0.0	0.0	752±4	0.29	767±5	0.31	**738**±3	0.32
sparse	5000	0.5	0.0	0.0	764±11	0.29	758±4	0.32	**737**±5	0.34
standard	250	0.2	0.0	0.0	**742**±13	0.30	782±12	0.32	788±7	0.33
standard	250	0.5	0.0	0.0	**739**±9	0.31	773±9	0.33	782±6	0.34
standard	5000	0.2	0.0	0.0	**739**±6	0.31	773±1	0.32	773±4	0.33
standard	5000	0.5	0.0	0.0	**731**±12	0.31	757±2	0.33	777±4	0.35
unfair	250	0.2	0.0	0.0	736±8	0.22	721±6	0.24	**714**±1	0.25
unfair	250	0.5	0.0	0.0	722±9	0.19	708±3	0.21	**687**±8	0.23
unfair	5000	0.2	0.0	0.0	724±11	0.23	726±2	0.25	**711**±2	0.26
unfair	5000	0.5	0.0	0.0	712±7	0.20	712±3	0.22	**682**±5	0.24

4 EXPERIMENTAL SETTING

4.1 Evaluating Assignments

As stated in Section 3.1, the optimization goal is to maximize the recall at a 5% FPR (Neyman-Pearson Criterion). When evaluating a set of assignments, values for the FPR may not be the same across experiments. This hinders our ability to directly compare the recall of different algorithms (*i.e.* If two methods obtain the same recall, the one with the lowest FPR is preferred). Implicitly, this optimization goal expresses a tradeoff between the misclassification costs of FP and FN mistakes, that is, a cost-sensitive problem. To evaluate performance, we can utilize a cost sensitive loss:

$$L = \lambda \mathrm{N}(\mathrm{FP}) + \mathrm{N}(\mathrm{FN}) \quad \text{with} \quad \lambda = \frac{t}{1-t}\,, \tag{1}$$

Where N(FP/FN) is the number of FP/FN errors. The parameter λ enforces a relationship between the cost of a FP and the cost of a FN. We now must define the relationship between our Neyman-Pearson criterion and the value of λ. Elkan [13] shows that the value of the ideal threshold t for a binary classifier and the misclassification costs are related according to Equation 1. When training our ML classifier, its threshold was chosen in alignment with the Neyman-Pearson criterion, so we set λ based on the model's threshold t.

4.2 Baselines

When searching for possible L2D baselines, we found no current method able to take individual capacity constraints into account. Therefore, we provide three baselines.

Rejection Learning (ReL) In this implementation we use the model score as a measure of model confidence. To apply rejection learning batch-wise, within our capacity constraints, we first order the cases within the batch by descending order of model score. The top 5% cases are automatically predicted positive (declined). Then, the following cases are randomly assigned to experts within our team until their capacity constraints are met. All left over cases, with the lowest model score, are classified negatively (accepted).

Human Expertise Aware Rejection Learning In this version of *rejection learning*, instead of randomly assigning the rejected cases throughout our expert team, we attempt to model each individual's behavior, in order to optimize assignments. To do so, we train a LightGBM model on the instance features and the *expert_id* to predict if the expert's prediction was a false positive (FP), false negative (FN), true positive (TP), or true negative (TN). For each instance, we then have a prediction for the probability that the expert will make either a FP, or a FN mistake, $\hat{\mathbb{P}}(\text{FP})$ and $\hat{\mathbb{P}}(\text{FN})$, respectively. We then calculate the predicted loss associated with deferring instance $\mathbf{x}_i$ to expert e:

$$L(\mathbf{X}_i, e) = \lambda\hat{\mathbb{P}}(\text{FP}) + \hat{\mathbb{P}}(\text{FN}) \ , \tag{2}$$

We present two versions of this algorithm:

- **Greedy ($\mathbf{ReL_{greedy}}$)**: The algorithm moves through the batch case by case, assigning each case to the expert with lowest predicted loss. Should an assignment violate capacity constraints, the algorithm tries to assign to the expert with second lowest loss, and so on. This is done until the experts' capacity constraints are met.
- **Integer Linear Programming ($\mathbf{ReL_{linear}}$)**: In this method, we minimize the loss over an entire batch by solving a linear programming problem subject to our capacity constraints, in order to find the optimal assignment over the entire batch.

For the context of fairness, we want to guarantee that the probability of wrongly declining a legitimate application is independent of the sensitive attribute value. Hence we measure the ratio between FPRs in each age group, i.e., predictive equality (PE) [7]. The ratio is calculated by dividing the FPR of the group with lowest observed FPR by the FPR of the group with the highest FPR.

4.3 Results

In Table 1 we show results for the discussed L2D baselines as well as a "Model only" system. The loss function is calculated according to Equation 1, with N(FP) and N(FN) counted over the test set. We can see how results for each of our L2D baselines vary with the generated human-AI collaboration environment (Scenario Properties) across the rows for our performance metric (Cost sensitive loss) and our fairness metric (Predictive Equality). We observe that, throughout all the considered scenarios, rejection learning performs best, despite our attempts to model human behavior. This may be due to the low volume of FNs and TPs in the training data, which may lead to poor probability estimates and ranking of the expert's probability of error for a given instance. In section D of the Appendix, Table 6 shows that our methods mostly mitigate FP errors, resulting in a lower FPR, but negatively impacting the recall as well. The mitigation of False Positives also leads to higher predictive equality, showing that our human expertise model was able to learn that experts tend to make more FP mistakes on older clients' applications. A drastic variation can be seen in the results for the $\text{ReL}_{\text{linear}}$ method. While it seems that expert absence has no significant effect on the loss for scenarios with a deferral rate of 20%, in the cases with 50% deferral rate, it seems that introducing expert absence significantly increases the loss. This illustrates the importance of testing the system under a wide variety of conditions.

In Table 2, we introduce variation in the pool of available experts. Here we can see that $\text{ReL}_{\text{linear}}$ outperforms ReL when the expert pool contains only agreeing, sparse or unfair experts. This may be due to the fact that these experts are simpler to model, as they have a clear dominating feature, or simpler feature dependencies. This illustrates the importance of considering variable complexity of human behavior when evaluating HAIC systems.

5 CONCLUSIONS AND FUTURE WORK

In this paper we introduce the OpenL2D Bank Account Fraud Dataset. To illustrate its use, we provide three L2D baselines tested under 300 different scenarios, in a financial fraud detection task. Our dataset enables comprehensive benchmarking of L2D algorithms, subject to real world constraints and scenarios. The main limitation of our work is that our baselines do not include any established methods in the L2D literature, as these do not currently consider the existence of capacity constraints.

We want to emphasize that our synthetic experts can not replace, in any way, humans involved in HAIC systems, as it would be necessary to gather real expert data in order to train the system to be used in a real-world application. It can be argued that by using these synthetic experts for research purposes, we are affecting the livelihood of large scale annotation workers (i.e. MTurk), which are often utilized by researchers. However, in some use cases, where domain-specific expertise is needed, such annotation services may not be of use to practitioners, and it may be impossible/unfeasible to obtain real human expert data. In cases where human predictions are accessible and pertinent to the use case, researchers should prefer these over synthetic expert predictions, as these would constitute real human behavior. For these reasons we believe that our work motivates the use of more complex synthetic expert data when real human predictions are unattainable, without posing a threat to current existing annotation services. It is also important to emphasize that our synthetic experts may amplify biases, due to the fact that our synthetic agents establish a monotonic relationship between each feature and the probabilities of error. It is possible that by increasing the weight of a feature that is correlated with the protected attribute, an expert with a positive weight for said feature in the false positive probability would exhibit a higher bias against said protected group. As such, biases could be amplified by our synthetic agents, and a careful analysis of the final predictive equality of each expert is encouraged.

REFERENCES

[1] Saar Alon-Barkat and Madalina Busuioc. 2023. Human–AI interactions in public sector decision making:"automation bias" and "selective adherence" to algorithmic advice. *Journal of Public Administration Research and Theory* 33, 1 (2023), 153–169.

[2] Kasun Amarasinghe, Kit T Rodolfa, Sérgio Jesus, Valerie Chen, Vladimir Balayan, Pedro Saleiro, Pedro Bizarro, Ameet Talwalkar, and Rayid Ghani. 2022. On the Importance of Application-Grounded Experimental Design for Evaluating Explainable ML Methods. *arXiv preprint arXiv:2206.13503* (2022).

[3] Antonin Berthon, Bo Han, Gang Niu, Tongliang Liu, and Masashi Sugiyama. 2021. Confidence scores make instance-dependent label-noise learning possible. In *International conference on machine learning*. PMLR, 825–836.

[4] Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. 2022. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[5] Mohammad-Amin Charusaie, Hussein Mozannar, David A. Sontag, and Samira Samadi. 2022. Sample Efficient Learning of Predictors That Complement Humans. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (Eds.). PMLR, 2972–3005.

[6] C. K. Chow. 1970. On Optimum Recognition Error and Reject Tradeoff. *IEEE Trans. Inf. Theory* 16, 1 (1970), 41–46. https://doi.org/10.1109/TIT.1970.1054406

[7] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. 2017. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining*. 797–806.

[8] Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. 2016. Learning with Rejection. In *Algorithmic Learning Theory - 27th International Conference, ALT 2016, Bari, Italy, October 19-21, 2016, Proceedings (Lecture Notes in Computer Science, Vol. 9925)*, Ronald Ortner, Hans Ulrich Simon, and Sandra Zilles (Eds.). 67–82. https://doi.org/10.1007/978-3-319-46379-7_5

[9] Shai Danziger, Jonathan Levav, and Liora Avnaim-Pesso. 2011. Extraneous factors in judicial decisions. *Proceedings of the National Academy of Sciences* 108, 17 (2011), 6889–6892.

[10] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 11. 512–515.

[11] Maria De-Arteaga, Riccardo Fogliato, and Alexandra Chouldechova. 2020. A Case for Humans-in-the-Loop: Decisions in the Presence of Erroneous Algorithmic Scores. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–12. https://doi.org/10.1145/3313831.3376638

[12] Dominik Dellermann, Philipp Ebel, Matthias Soellner, and Jan Marco Leimeister. 2019. Hybrid Intelligence. *Business & Information Systems Engineering* 61, 5 (Oct. 2019), 637–643. https://doi.org/10.1007/s12599-019-00595-2 arXiv:2105.00691 [cs]

[13] Charles Elkan. 2001. The Foundations of Cost-Sensitive Learning. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI 2001, Seattle, Washington, USA, August 4-10, 2001*, Bernhard Nebel (Ed.). Morgan Kaufmann, 973–978.

[14] Sharad Goel, Ravi Shroff, Jennifer Skeem, and Christopher Slobogin. 2021. The accuracy, equity, and jurisprudence of criminal risk assessment. In *Research handbook on big data law*. Edward Elgar Publishing, 9–28.

[15] Stein Grimstad and Magne Jørgensen. 2007. Inconsistency of expert judgment-based estimates of software development effort. *Journal of Systems and Software* 80, 11 (2007), 1770–1777.

[16] Varun Gulshan, Lily Peng, Marc Coram, Martin C Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, et al. 2016. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Jama* 316, 22 (2016), 2402–2410.

[17] Jingguang Han, Yuyun Huang, Sha Liu, and Kieran Towey. 2020. Artificial intelligence for anti-money laundering: a review and extension. *Digital Finance* 2, 3-4 (2020), 211–239.

[18] Patrick Hemmer, Sebastian Schellhammer, Michael Vössing, Johannes Jakubik, and Gerhard Satzger. 2022. Forming Effective Human-AI Teams: Building Machine Learning Models That Complement the Capabilities of Multiple Experts. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, Luc De Raedt (Ed.). ijcai.org, 2478–2484. https://doi.org/10.24963/ijcai.2022/344

[19] Dan Hendrycks and Kevin Gimpel. 2017. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

[20] Inc. Heritage Provider Network. 2011. Heritage Health Prize. https://kaggle.com/competitions/hhp.

[21] Kori Inkpen, Shreya Chappidi, Keri Mallari, Besmira Nushi, Divya Ramesh, Pietro Michelucci, Vani Mandava, Libuše Hannah Vepřek, and Gabrielle Quinn. 2022. Advancing Human-AI Complementarity: The Impact of User Expertise and Algorithmic Tuning on Joint Decision Making. *arXiv preprint arXiv:2208.07960* (2022).

[22] Maia Jacobs, Melanie F Pradier, Thomas H McCoy Jr, Roy H Perlis, Finale Doshi-Velez, and Krzysztof Z Gajos. 2021. How machine-learning recommendations influence clinician treatment selections: the example of antidepressant selection. *Translational psychiatry* 11, 1 (2021), 108.

[23] Sérgio Jesus, José Pombal, Duarte Alves, André F Cruz, Pedro Saleiro, Rita P Ribeiro, João Gama, and Pedro Bizarro. 2022. Turning the Tables: Biased, Imbalanced, Dynamic Tabular Datasets for ML Evaluation. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2022*.

[24] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 3146–3154.

[25] Vijay Keswani, Matthew Lease, and Krishnaram Kenthapadi. 2021. Towards Unbiased and Accurate Deferral to Multiple Experts. In *AIES '21: AAAI/ACM Conference on AI, Ethics, and Society, Virtual Event, USA, May 19-21, 2021*, Marion Fourcade, Benjamin Kuipers, Seth Lazar, and Deirdre K. Mulligan (Eds.). ACM, 154–165. https://doi.org/10.1145/3461702.3462516

[26] Lauren Kirchner and Jeff Larson. 2017. How We Analyzed the COMPAS Recidivism Algorithm. https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm.

[27] Alex Krizhevsky and Geoffrey Hinton. 2009. Learning Multiple Layers of Features from Tiny Images. (2009).

[28] Dattatray Vishnu Kute, Biswajeet Pradhan, Nagesh Shukla, and Abdullah Alamri. 2021. Deep learning and explainable artificial intelligence techniques applied for detecting money laundering–a critical review. *IEEE access* 9 (2021), 82300–82317.

[29] Ariel Levy, Monica Agrawal, Arvind Satyanarayan, and David Sontag. 2021. Assessing the impact of automated suggestions on decision making: Domain experts mediate model errors but take less initiative. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–13.

[30] David Madras, Toni Pitassi, and Richard Zemel. 2018. Predict Responsibly: Improving Fairness and Accuracy by Learning to Defer. In *Advances in Neural Information Processing Systems*, Vol. 31. Curran Associates, Inc.

[31] Hussein Mozannar, Hunter Lang, Dennis Wei, Prasanna Sattigeri, Subhro Das, and David Sontag. 2023. Who Should Predict? Exact Algorithms For Learning to Defer to Humans. *arXiv preprint arXiv:2301.06197* (2023).

[32] Hussein Mozannar and David A. Sontag. 2020. Consistent Estimators for Learning to Defer to an Expert. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 7076–7087.

[33] Maithra Raghu, Katy Blumer, Greg Corrado, Jon M. Kleinberg, Ziad Obermeyer, and Sendhil Mullainathan. 2019. The Algorithmic Automation Problem: Prediction, Triage, and Human Effort. *CoRR* abs/1903.12220 (2019). arXiv:1903.12220

[34] Maithra Raghu, Katy Blumer, Rory Sayres, Ziad Obermeyer, Bobby Kleinberg, Sendhil Mullainathan, and Jon Kleinberg. 2019. Direct uncertainty prediction for medical second opinions. In *International Conference on Machine Learning*. PMLR, 5281–5290.

[35] Ravid Shwartz-Ziv and Amitai Armon. 2022. Tabular data: Deep learning is not all you need. *Information Fusion* 81 (2022), 84–90.

[36] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. 2018. The HAM10000 Dataset, a Large Collection of Multi-Source Dermatoscopic Images of Common Pigmented Skin Lesions. *Scientific data* 5, 1 (2018), 1–9.

[37] Rajeev Verma, Daniel Barrejón, and Eric Nalisnick. 2023. Learning to Defer to Multiple Experts: Consistent Surrogate Losses, Confidence Calibration, and Conformal Ensembles. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 11415–11434.

[38] Rajeev Verma and Eric T. Nalisnick. 2022. Calibrated Learning to Defer with One-vs-All Classifiers. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (Eds.). PMLR, 22184–22202.

[39] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. 2017. Chestx-Ray8: Hospital-scale Chest x-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2097–2106.

[40] Zhaowei Zhu, Tongliang Liu, and Yang Liu. 2021. A second-order approach to learning with instance-dependent label noise. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10113–10123.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009