RLHR: A Framework for Driving Dynamically Adaptable Questionnaires and Profiling People Using Reinforcement Learning

Ciprian Paduraru[1] Catalina Camelia Patilea[1] and Alin Stefanescu[1,2]
[1]*Department of Computer Science, University of Bucharest, Romania*
[2]*Institute for Logic and Data Science, Romania*

Keywords: Reinforcement Learning, Bias Removal, Time Series, Classification, Behaviors, Profiling.

Abstract: In today's corporate landscape, the creation of questionnaires, surveys or evaluation forms for employees is a widespread practice. These tools are regularly used to check various aspects such as motivation, opportunities for improvement, satisfaction levels and even potential cybersecurity risks. A common limitation lies in their generic nature: they often lack personalization and rely on predetermined questions. Our research focuses on improving this process by introducing AI agents based on reinforcement learning. These agents dynamically adapt the content of surveys to each person based on their unique personality traits. Our framework is open source and can be seamlessly integrated into various use cases in different industries or academic research. To evaluate the effectiveness of the approach, we tackle a real-life scenario: the detection of potentially inappropriate behavior in the workplace. In this context, the reinforcement learning-based AI agents function like human recruiters and create personalized surveys. The results are encouraging, as they show that our decision algorithms for content selection are very similar to those of recruiters. The open-source framework also includes tools for detailed post-analysis for further decision making and explanation of the results.

1 INTRODUCTION

The goal of this work is to create a framework that contains the tools needed to conduct large-scale adaptive surveys and to thoroughly analyze the results after the survey. The main component is a software-based virtual HR agent that can behave like a real human during a survey and adapt the sequence of questions asked to the individuals being assessed and their previous responses. We refer to this adaptive way of asking questions as *dynamic survey*. With the proposed software agents, each person in an organization could be individually assessed at minimal cost. A limited and targeted number of questions must be asked to maintain respondent engagement. In this context, the agent strategically determines the sequence of questions. By optimizing this sequence, the goal is to better assess individuals based on the survey objectives, with the number of questions comparable to that of a typical fixed survey.

We summarize our contribution below:

1. The first deep reinforcement learning (DRL) method mimics HR professionals to drive questionnaires in real-time.
2. Improved methods for detecting and subtracting bias in responses (due to user over- or under-response to questions over time) using time series techniques. (Xia et al., 2015).
3. A method of augmenting existing datasets (which are usually small) to create large synthetic datasets that mimic the original datasets.
4. We make our work available to industry and academia as an open-source framework called *RLHR* (Reinforcement Learning Human Resources) at https://github.com/unibuc-cs/AIForProfilingHumans.

2 RELATED WORK

The work that comes closest to ours is (Paduraru. et al., 2024), which has similar goals but different methods. We compare their methods with ours using a new, larger anonymized dataset. The methods proposed in our current work have several technical features to improve the state of the art. First, we found that the *Pathfinding AI* method in (Paduraru. et al., 2024) suffers from biased selection (Wang and Singh, 2021) as it always selects the closest possible cluster (with limited random explorations). To improve the results, in this work we use a deep reinforcement learning method (Mnih et al., 2016) by adding bet-

Paduraru, C., Patilea, C. and Stefanescu, A.
RLHR: A Framework for Driving Dynamically Adaptable Questionnaires and Profiling People Using Reinforcement Learning.
DOI: 10.5220/0012752800003753

In *Proceedings of the 19th International Conference on Software Technologies (ICSOFT 2024)*, pages 359-366
ISBN: 978-989-758-706-1; ISSN: 2184-2833

ter explorations, latent encapsulation of individuals by deep neural networks, and temporal information understanding with gated recurrent unit (GRU) neural networks (Cho et al., 2014). We also improve the constant type of bias removal from the original method with time series (Benvenuto et al., 2020), which leads to better results. The available tools for post-survey analysis have also been improved. We are also addressing the issue of creating synthetic data that approximates real human profiles so that clients can evaluate and customize their survey definitions before sending them to people.

Profiling people for content recommendations, such as news recommendations, is a long-standing practice (Mannens et al., 2013). Automatic detection of fraudulent profiles on social media platforms such as Instagram and Twitter is another common application for the creation of people profiles using data mining and clustering techniques (Khaled et al., 2018). In (Ni et al., 2017), social media data extracted from WeChat [1] is used to create individual profiles and group them based on their occupational field, using similar NLP techniques to those previously mentioned. The research in (Schermer, 2011) discusses the use of data mining in automated profiling processes, with a focus on ethics and potential discrimination. Use cases include security services or internal organizations that create profiles to assess various characteristics of their employees. Profiling and grouping individuals using data mining and NLP techniques to extract information from text data is a common topic in the literature. In (Wibawa et al., 2022), the authors use AI methods such as traditional NLP to process application documents for job openings, which enables automatic filtering, evaluation and prioritization of candidates.

3 SURVEY SETUP

3.1 Survey Formalization

Our aim is to present a survey in as generalized a form as possible. In doing so, we rely on our experience with the clients of *vorteXplore* and on the experience we have gained with later versions of the framework. The proposed high-level presentation method consists of a limited number of questions (configurable on the client side, on average between 15-25) that are either general in nature or related to an asset shown in the form of an image, video or extracted text (e.g. articles, SMS messages, emails, etc.). The formal specifications and components of a survey are explained below.

Groups. Every asset and question that is asked is part of a group. Examples of groups from the use case: *Awareness*, *Prevalence*, *Sanction*, *Inspiration*, *Factual*, *Sensitivity*. In our experience, this has proven to be very useful for characterizing people from multiple perspectives and organizing assets and questions. It also has implications for reusability and makes it easier to maintain the dataset.

Assets. A collection of assets representing video files, media posts, SMS, etc. Asset indices also have an optional dependency specification, i.e. the client can specify that an asset should depend on a previously displayed set of other assets: $Deps(A_i) = \{A_j\}_{j \in 1..|Assets|}$. For example, a video or image asset could only make sense as a sequence of previous assets.

Question. The set of textual questions is denoted by Q. Each element $Q_i \in Q$ has two categories of properties:

1. Structural properties.
   - The set of assets that are compatible with this question: $Compat(Q_i) = \{A_j \in Assets\}_j$. The idea of compatibility is that some of the questions make sense for each type of asset shown. Others do not, e.g. video-based assets with a concrete action demonstration.
   - Dependencies on previous questions. Internally, the dependencies between questions take the form of a directed acyclic graph, where each node Q_i has a set of dependencies $Deps(Q_i) = \{Q_j\}_j$. This set represents a restriction that Q_i can only be asked as a follow-up question to a previous question $Q_k \in Deps(Q_i)$.

   b) Scoring properties.
   - Attributes. For the use case of IB recognition, some examples: *Team interaction*, *Offensive language*, *Rumors*, *Personal boundaries*, *Leadership Style* (the full list can be found in a table in our repository). These are customizable in the framework, are usually set by the organizations prior to the surveys and are not visible to the respondents. Generally, the client organization strategically uses these inherent characteristics to gain the insights they are looking for in the post-survey analysis. The $Attr$ set represents the collection of attributes used by an application. For each question Q_i, a vector of all attributes ordered by indices is given, representing the relative importance of each attribute to the question. The value range is $[0-1]$, where

[1] WeChat.com

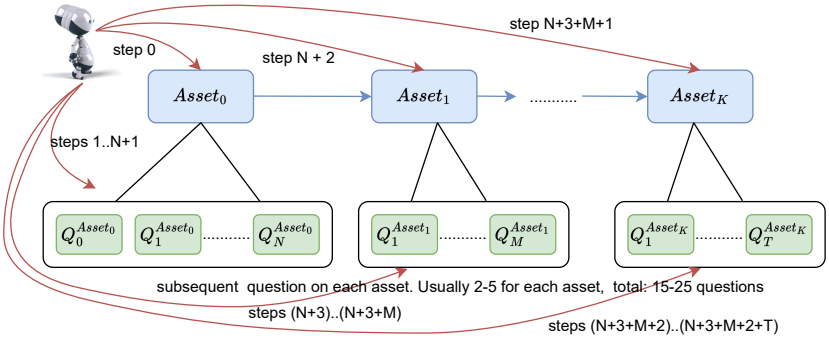

Figure 1: Example of the RLHR agent first selecting an asset and then asking a series of related questions, taking into account constraint dependencies until the end of the interview. For each asset, the typical number of questions asked by clients ranged from 1-5, and the total number of questions was 15-25.

0 means that the respective attribute is insignificant for the asset, while a value of 1 represents a strong correlation between the attribute and the asset. Formally, each question is given a specification vector (by the client): $At(Q_i) = \{At_1, At_2, ..., At_{NAt}\}$, where $NAt = |Attr|$. A getter function, $Imp(Q_i, At_i)$, is used later in the paper to determine the degree of importance for each attribute in relation to the question.

- Importance of the questions. The function $\text{Imp}(Q_i)$ is used to determine the relevance of the question for the survey from the client's point of view. The values are numerical floating point numbers in the range $[0-1]$, where 0 has no relevance, while 1 represents a high interest in the user responses to the question.
- Baseline and tolerance values. $\text{Base}(Q_i)$ represents the *standard* expected response of respondents to this question by the organization. $\text{Tol}(Q_i)$ is used to denote the *accepted* deviations in the responses.
- $\text{Amb}(Q_i)$: each question has an ambiguity factor. This is not set from the beginning (no one would intentionally create ambiguous questions); it is regressed from the results of the post-survey or participant feedback. Rather than changing the entire survey structure, the client can increase this factor to remove potential ambiguity and mitigate the value of divergent responses. The value range is between $[0-1]$, where a value of 1 means no ambiguity, while 1 is the maximum.

Internally, the answers to the questions were mapped to floating point numbers in the range $[1-7]$, either in binary format, as point values within a range, etc. The same range of values is used for baselines and tolerances.

Profiles Specification. The autonomous survey aims to categorize a person into a specific profile that corresponds as closely as possible to an HR professional who would interview the person face-to-face. Intuitively, people's responses to a survey's assets and questions according to the factors mentioned above (i.e. baselines and tolerance from the client's perspective, importance of questions and ambiguities) contribute to the aggregate score for each attribute. These scores are used to calculate the match with one or the other profile.

A profile is specified as a multivariate Gaussian distribution (Gutiérrez et al., 2023), with a total of *NAt* (number of attributes) dimensions, i.e. one for each attribute. The set of all profiles is denoted by $Profiles$. The reasons for using this type of distribution are explained below, while the technical details can be found in Section 4:

- It enables the natural modeling of an individual by the properties hidden in the question. Intuitively, the HR defines the mean, the μ vector, as the expected values of the deviations for the observed inherent attributes for each of the profiles.
- The covariance matrix, Σ, can be used to indicate both the tolerance (variance) of these attributes for each of the profiles and the correlations between the attributes. Of course, some attributes are correlated with each other and cannot be treated separately. At the beginning of a project with new attributes and no previous data set, the client has no information about correlations, so it uses a diagonal matrix Σ. However, after data has been collected, as in our use case, the RLHR framework has tools to calculate the correlation between the attributes based on the Pearson correlation (Benesty et al., 2008).

4 METHODS FOR EVALUATING SURVEYS

This section introduces the common evaluation functions and the internal accounting of the statistics to prepare the inputs for the RLHR agent discussed in Section 5.

4.1 Deviations

The root of the scoring process begins with the calculation of the *deviations* for each question. After each question Q_i in the survey, the user responds with a numerical sliding value in the range $[1-7]$ (Section 3.1), denoted by $R(Q_i)$. As shown in Eq. (1), this value is compared with the base value and the tolerance values. Then, the importance of the question (or severity) in the range $[0-1]$ is added to mitigate deviations from questions that are irrelevant to the final classification from the client's perspective. Finally, the reported and agreed ambiguities in the range $[0-1]$ are added to the equation to mitigate questions that have been found to be ambiguous, and depending on the degree, the deviations become inversely proportionally less important.

$$D(Q_i) = \left[\frac{|R(Q_i) - Base(Q_i)|}{Tol(Q_i)}\right]^2 \times \left[1 + Amb(Q_i)^{-1}\right] \times Imp(Q_i) \quad (1)$$

4.2 Removing Anchor and over- or under-scoring the Questions

Numerous biases can manifest themselves in a survey (Yan et al., 2018). The most common are anchors (influences or connections to a previously asked question) and the consistent over- or under-rating of answers. The identification of these are needed to obtain accurate statistics at the team and organizational level. Otherwise, the RLHR agent might misunderstand the situation. Figure 2 illustrates this behavior. The method is to find patterns in the deviation either in the entire survey or in short, consecutive sequences (Dee, 2006).

While the RLHR agent is conducting a survey, it has access to the answers to the questions asked in steps $[1...K-1]$ in each step K. To detect possible bias or anchor, our method looks for an initial position S in the range of steps so that a model can fit a predictor of biases for the range $[S..K-1]$. The model is an auto-regressive integrated moving average (ARIMA) (Benvenuto et al., 2020).

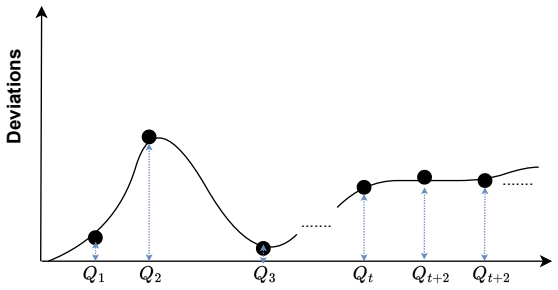

Figure 2: An example of user responses during a survey and deviation values calculated with Eq. (1). At the beginning, for the first three questions, there is no trend in the deviations. From step t onwards, however, it can be seen that the deviations gradually equalize, which means that the user could be over- or under-rating responses provided for a number of steps.

4.3 Scores Feature Vector

An important feature that the RLHR agent uses to categorize a user U into one of the defined profiles is the aggregated score of each inherent attribute in the set $Attr$ in relation to the questions asked and their answers. Assume that a survey is in progress and there are already t pairs of questions, answers and both types of deviations that can be computed using Eq. (1).

The set of inherent attributes and their scores represent the *features* of U used by the RLHR agent and for profile classification in each step. The calculation method of these scores is shown in Eq. (2), where the final result $Sc_t(U, At_k)$ represents the score vector for the feature (attribute) $At_k \in Attr$ of the user U during the survey after t questions have been asked. We further denote by $Sc_t^{nb}(Ui, At_k)$ the same score function without bias, $D^{nb}(Q_i)$ instead of $D(Q_i)$. The idea behind the calculations is that for each attribute At_k it iterates over all questions Q_i asked so far and aggregate their contribution to At_k (as an average) by using the deviations and the importance of questions, $Imp(Q_i, At_k)$ (Section 3.1). For simplicity, we use the vectorized notation of the scores of U at time t by $Sc_t(U) \in R^{NAt}$.

$$Sc_t(U, At_k) = \frac{\sum_{i=1}^{t} Imp(Q_i, At_k) \times D(Q_i)}{\sum_{i=1}^{t} \mathbb{1}(Imp(Q_i, At_k) > 0)} \quad (2)$$

As mentioned in Section 3.1, the profiles are defined using multivariate Gaussian distributions (Gutiérrez et al., 2023) around the set of inherent attributes by the expected mean and covariance (tolerance of each attribute and predicted correlations between them) Eq. (3). We denote the number of profiles with $NumPrf = |Profiles|$.

$$PrfDef_k = \mathcal{N}(\mu_k, \Sigma_k),\ \mu \in \mathcal{R}^{NAt},\ \Sigma \in \mathcal{R}^{NAt \times NAt},\ \forall PrfDef_k \in Profiles \tag{3}$$

To determine the probability that U is part of each profile at time t given the current score vector $Sc_t(U)$, the deviation scores calculated above are passed to the standard probability density function, as shown in Eq. (4).

$$P_t^U(k) = P_t^U(PrfDef_k|Sc_t(U)) = p(Sc_t(U);\mu_k,\Sigma_k) = \frac{1}{(2\pi)^{NAt/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\right),\ \forall PrfDef_k \in Profiles \tag{4}$$

The predicted profile index at time step t for the user U results from the selection of the maximum from these results, Eq. (5).

$$Prf_t^{pred}(U) = argmax[P_t^U(PrfDef_k|Sc_t(U))]_{k\in[1...NumPrf]} \tag{5}$$

5 THE RLHR AGENT

The goal of the RLHR agent is to autonomously control the survey process, adapt to the content requested by the respondent, and provide a distribution of scores across profiles that matches the ground truth profile as closely as possible. The general ideas for applying the RL methodology and components to our objectives are detailed in this section and outlined in Figure 3

5.1 Synthetic Environments and Dataset

The environment represents the *world* in which the RLHR agent performs actions and receives feedback through partial observations and rewards. We have used the OpenAI Gym (Towers et al., 2023) interfaces and principles (more specifically, the updated Gymnasium library) so that our framework can be further used for experiments in the community.

Set up Virtual Users. With defined profiles, even without collecting real data, synthetic data can be created based on sampling methods. Specifically, N examples of virtual users, $VUsers$, can be created, with each $U \in VUsers$ following a two-step process:

1. Select a ground truth profile for U by drawing a uniform sample from the available set of profiles. Note that this is hidden from the observation of the RLHR agent and is only used for background evaluation mechanisms when interacting with the environment.
$$Prf^{gt}(U) = Uniform[1, NumPrf] \tag{6}$$
2. Sample a vector of inherent (ground truth) attributes, knowing the ground truth profile and its base distribution parameters from Eq. (3).
$$At^{gt}(U) \sim \mathcal{N}(\mu_{gt}, \Sigma_{gt}) \tag{7}$$

If accurate data is available from HR experts, annotated data for points 1. and 2. can be added to the database.

Simulation of Responses from Virtual Users When the RLHR agent asks the environment for an answer from the surveyed user U to a question Q_i, the value of the answer must be correlated with: (a) the inherent personality attributes, $At^{gt}(U)$, and (b) with the importance of the attributes in the questions, $At(Q_i)$. This correlation can be solved by a dot product between the two, Eq.s (8), which gives the normalized deviation value for Q_i in the range $[0-1]$. It must then be converted to the client range (in our use case, for example, the range $[1-7]$ is used, Section 3.1.

$$D(Q_i) = remap(At^{gt}(U) * At(Q_i)) \tag{8}$$

Finally, we substitute $D(Q_i)$ into Eq. (1) to determine the response value$R(Q_i)$. This results in the form shown in Eq. 9

$$R(Q_i) = Base(Q_i) + Tol(Q_i) \times \left[\frac{D(Q_i) \times Imp^{-1}(Q_i)}{1 + Amb^{-1}(Q_i)}\right]^{1/2} \tag{9}$$

5.2 Episodes, Actions and Observations

An in-progress survey of a user U is represented as a *trajectory*, τ, using the reinforcement learning policy-based algorithms (Sutton and Barto, 2018). In our case, a *episode* is the same as a trajectory from the beginning of a survey to its end.

At any time t in a survey, the state includes all assets displayed and the t questions asked. As shown in Figure 1, at each step (or *action* in RL terminology), the agent must either select a new asset to show or a follow-up question based on the currently presented asset. Suppose that t questions have been asked using multiple $NG-1$ completed groups of assets and associated questions, and the RLHR agent is deciding which asset or question to show for group NG. We denote the asset shown in group K by A^K, the i-th follow-up question by Q_i^k, and the total number of questions asked in group K by $N(G_K)$. Eq. (10) shows closed groups (indexed by k). Similarly, Eq. (11) defines an ongoing group that must select the next question $i+1$, while an empty group means that the next action of the RLHR agent should be to select an asset first, Eq. (12). Finally, Eq.(13) shows

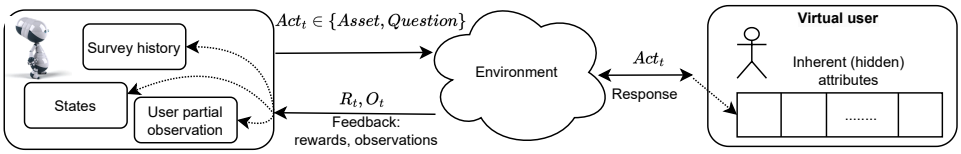

Figure 3: Relationship between the RLHR agent (left), the environment (center) and the virtual user being interviewed. The agent sends actions to the environment and asks to display a new asset or ask a new question about the current state. In return, the environment simulates a response that correlates with the user's ground truth profile. Each response updates the inherent attributes. The environment sends feedback as a reward for the last action performed and the new partial observation of the user, which models the agent's belief in the user's inherent attributes. The dashed lines represent the updates made internally.

the formalized relationship between the step (actions) and the parameter t.

$$G_k = \left(A^k, Q_1^k \dots Q_{N(G_k)}^k\right) \tag{10}$$

$$G_k^{i+1} = \left(A^k, Q_1^k \dots, Q_i^k, \mathbf{Q_{i+1}^k} = ?\right) \tag{11}$$

$$G_k^{\emptyset} = \left(\mathbf{A^k} = ?\right) \tag{12}$$

$$t = \sum_{j=k}^{NG-1} N(G^k) \tag{13}$$

The trajectory for a running survey is displayed in Eq. (14). It is parameterized by three parameters: (a) t-the total number of questions asked so far, (b) NG-the index of the current group and (c) k-the number of questions asked so far in the group NG, which can be $\emptyset$ if no question has been asked yet, i.e. if an asset is expected. In order not to overcomplicate the equations, we omit the typical pair (state, action, reward) at each step and keep only the state and action to be performed next (with an exclamation mark). The actions are formally discussed in Section 5.2, while the rewards are taken after each action and defined in Section 5.3.

$$\tau_{(t,NG,k)} = \left[G_1, \dots, G_{NG-1}, \mathbf{G_{NG}} = \mathbf{G_{NG}^{k+1}} \textbf{ or } \mathbf{G_{NG}^{\emptyset}}\right] \tag{14}$$

At each step during surveying a user U at time t, the observation of the RLHR agent returned by the environment, O_t^U, is composed of two components:

(a) the trajectory τ, which consists of the history of pairs of groups, assets and questions asked.

(b) the score of the user's attributes after each action, $Sc_t(U)$, which is calculated as in Eq. (2).

The state of the agent is given by Eq. (15)). It includes the observation, the set of valid questions VQ_t and the assets VA_t at time t due to the course of the survey and the contextual dependencies.

$$S_t^U = \left(\tau_{(t,..)}, Sc_t^U, \{VQ_t, VA_t\}\right) \tag{15}$$

Actions and environment constraints . There are also hard constraints that must be fulfilled along the trajectory (or episode) in relation to the actions:

1. In the first step, an asset must be shown.
2. If at any time t the RLHR agent decides to ask a new question Q_{new}, it must comply with two main rules. First, it must satisfy the dependencies on the previously asked question (or no dependencies at all), i.e. $Deps(Q_{new}) = \emptyset$, or $Q_t \in Deps(Q_{new})$. In addition, the new question must be compatible with the current asset, i.e. $A(t) \in Compat(Q_{new})$.
3. A maximum number of follow-up questions can be asked about a currently presented asset, represented by the parameter *MaxQPerAsset* (in our example *MaxQPerAsset*=5). Once this threshold is reached, a hard constraint to show a new asset is added to the RLHR agent's observation. Note that the agent can switch to a new asset even if this threshold is not reached.
4. When a new asset is shown, it must satisfy the dependency on a previous asset, similar to questions.
5. The episode ends when: (a) the number of steps reaches a threshold *MaxSteps* (in our example *MaxSteps*=30 - intuitively set for a maximum of 25 questions and five or more assets), or (b) when there is no remaining question or asset that can be shown to satisfy the dependencies and structural requirements (e.g., an asset must be shown, but there is no longer one that satisfies the dependencies). Note in this context that the number of questions may vary between surveys depending on the user's choices and answers. We think this is natural human behavior.
6. To handle the case of *general questions* where no asset needs to be shown, we consider a special NULL asset that does not visually display anything other than the following general questions.
7. The minimum number of questions in a group is 1.

Eq. (16) formalizes the action that the RLHR agent can take if a group NG is in progress in the current trajectory and k (possibly $\emptyset$) questions have already

been asked (Eq. (14)). The possible actions are: (a) displaying a new asset when the agent decides or is forced to start the next group $NG+1$, and (b) ask the new question $K+1$ in the current group.

$$Act_{NG}^{K} \in \{A_{\{NG\,or\,NG+1\}}^{new}, \quad Q_{NG}^{new}\} \tag{16}$$

5.3 Rewards

The aim of the RLHR agent is to drive the survey using the actions defined in Eq. (16) so that the user U is classified as close as possible to their known ground truth profile $Prf^{gt}(U)$ (Eq. (6)) at the end.

As in Eq. 4, at any time t during a survey, the probability that a user U belongs to a profile k is given by the values of the inherent attributes, $Sc_t(U)$. In this representation, the main idea is to display assets and ask corresponding questions to find the attribute scores that lead to the correct classification.

With this in mind, the system models the reward function at time t, i.e. with two main components:

(a) *OverallScore*. The agent is penalized for having attribute scores that do not yet approach the ones defined by the ground truth, Eq. (17). Intuitively, the maximum of this component is 0 if the inherent attributes have scores that are close to the predefined mean value of the ground truth μ_{gt}, and taking into account the associated covariances Σ_{gt}.

$$OverallSc = P_t^U(gt) - 1.0 \tag{17}$$

(b) The agent is penalized for not performing an action that moves the classification in the right direction. As shown in Eq. (18), the idea is to calculate the velocity of the last action in relation to the classification probability of the ground truth.

$$VelSc = \begin{cases} P_t^U(gt) - P_{t-1}^U(gt), & t > 1 \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

Eq. (19) shows the final reward function after t questions asked, with the same correlations to the current group NG and the number of questions asked k NG as in Eq. (13) is shown. The two components defined above are averaged with configurable weights. In our use case, we set the total reward component as $W_{ov} = 0.8$ and the velocity component as $W_{vel} = 0.2$.

$$Reward_t = OverallSc * W_{ov} + VelSc * W_{vel} \tag{19}$$

After some evaluation, we decided to use the asynchronous actor-critic method, more precisely A2C(Mnih et al., 2016) from the class of policy-based methods.

Table 1: Comparative results between HR professionals, PathfindingAI (Paduraru. et al., 2024), and our proposed method **RLHR**. Accuracy 1st indicates how many predictions of the person's profiles match the HR, which is considered the ground truth. The Accuracy 2nd for the two methods indicates how many of the incorrect predictions were placed at the 2nd position in the probability distributions of the output. The last column shows the average error between the probability assigned to the ground truth profile and the probability of the predicted profile.

Evaluation method	*Accuracy 1st*	*Accuracy 2nd*	*Avg. Error 1st to 2nd*
HR	100% (69)	0	0
PathfindingAI	62.3% (43)	6.9% (10)	~ 0.221
RLHR	**74% (51)**	**21.7% (15)**	~ 0.127

6 EVALUATION

The framework is evaluated from several perspectives. First, quantitative and qualitative assessments are presented to understand the ease of use from the user's perspective and the credibility of the methods. Then, the computational effort required to conduct scale surveys and retrain the RLHR agent is presented to understand the practical usability. Finally, this section presents post-survey analysis tools and lessons learned from prototype development and previous efforts.

Setup. Quantitative Evaluation. First, we try to evaluate the correctness of the methods proposed in this work by comparing them with an evaluation performed in parallel by HR experts and the algorithm *PathfindingAI* in (Paduraru. et al., 2024).

A sample of 69 people was selected by HR professionals and interviewed in a similar way to that described in the study, but face-to-face. After six months, with no major post-survey interventions or actions, we assessed the same individuals using the proposed RLHR agent. Note that the dataset of assets and questions used by the HR and RLHR agents matched, but the questions and assets that were originally asked were replaced to avoid any bias. There were a total of 1498 responses to the questions. The results of the observed comparison follow:

Table 1 shows the results obtained by comparing the *supposed* ground truth assessment of HR professionals in the client organization with the PathfindingAI and RLHR agents. The key observation is that the RLHR agent implemented in our proposed framework performs better than the state-of-the-art PathfindingAI method. Moreover, in many cases, the RLHR

agent successfully classified the missing cases of the 1st ground truth profile at the 2nd position in the output probability distribution. It left only three out of 69 classified individuals at the 3rd and 4th positions, compared to the PathfindingAi, which left 16 individuals. Furthermore, the error of the RLHR is significantly lower for the misclassified examples, i.e. the entropy between the ground truth profile and the predicted profile is high.

The method of removing bias based on time series improved the final results, as shown in Table 1. More specifically, compared to the previous method for identifying constant bias in (Paduraru. et al., 2024), the new method improved the *Accuracy 1st* from 48 to 51 correctly predicted individuals, while the *Accuracy 2nd* increased from 12 to 15.

7 CONCLUSIONS

The purpose of the *RLHR* framework is not to replace the experts in the HR departments of companies. Its main purpose is to create another layer between individuals and HR departments. The intermediate layer we propose would improve the HR department's survey processes and interventions and focus on the available resources where they are most needed.

ACKNOWLEDGEMENTS

This research was supported by European Union's Horizon Europe research and innovation programme under grant agreement no. 101070455, project DYNABIC.

REFERENCES

Benesty, J., Chen, J., and Huang, Y. (2008). On the importance of the pearson correlation coefficient in noise reduction. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(4):757–765.

Benvenuto, D. et al. (2020). Application of the arima model on the covid-2019 epidemic dataset. *Data in Brief*, 29:105–340.

Cho, K. et al. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Dee, D. (2006). Bias and data assimilation. *In Proceedings of the ECMWF Workshop on Bias estimation and correction in data assimilation.*, pages 1–20.

Gutiérrez, F. et al. (2023). Differentiating abnormal, normal, and ideal personality profiles in multidimensional spaces. *Journal of Individual Differences*.

Khaled, S. et al. (2018). Detecting fake accounts on social media. In *IEEE International Conference on Big Data (Big Data)*, pages 3672–3681.

Mannens, E. et al. (2013). Automatic news recommendations via aggregated profiling. *Multimedia Tools and Applications - MTA*, 63.

Mnih, V. et al. (2016). Asynchronous methods for deep reinforcement learning.

Ni, X. et al. (2017). Behavioral profiling for employees using social media: A case study based on wechat. In *Chinese Automation Congress (CAC)*, pages 7725–7730.

Paduraru., C., Cristea., R., and Stefanescu., A. (2024). Adaptive questionnaire design using ai agents for people profiling. In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence - Volume 3: ICAART*, pages 633–640.

Schermer, B. W. (2011). The limits of privacy in automated profiling and data mining. *Computer Law and Security Review*, 27(1):45–52.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA.

Towers, M. et al. (2023). Gymnasium: An API standard for single-agent reinforcement learning environments, with popular reference environments and related utilities (formerly gym) https://gymnasium.farama.org/.

Wang, Y. and Singh, L. (2021). Analyzing the impact of missing values and selection bias on fairness. *International Journal of Data Science and Analytics*, 12(2):101–119.

Wibawa, A. D. et al. (2022). Text mining for employee candidates automatic profiling based on application documents. *EMITTER International Journal of Engineering Technology*, 10:47–62.

Xia, P., Zhang, L., and Li, F. (2015). Learning similarity with cosine similarity ensemble. *Information sciences*, 307:39–52.

Yan, T., Keusch, F., and He, L. (2018). The impact of question and scale characteristics on scale direction effects. *Survey Practice*.