MARS: Assisting Human with Information Processing Tasks Using Machine Learning

CONG SHEN, University of Virginia
ZHAOZHI QIAN and ALIHAN HUYUK, University of Cambridge
MIHAELA VAN DER SCHAAR, University of Cambridge and University of California

This article studies the problem of automated information processing from large volumes of unstructured, heterogeneous, and sometimes untrustworthy data sources. The main contribution is a novel framework called Machine Assisted Record Selection (MARS). Instead of today's standard practice of relying on human experts to manually decide the order of records for processing, MARS learns the optimal record selection via an online learning algorithm. It further integrates algorithm-based record selection and processing with human-based error resolution to achieve a balanced task allocation between machine and human. Both fixed and adaptive MARS algorithms are proposed, leveraging different statistical knowledge about the existence, quality, and cost associated with the records. Experiments using semi-synthetic data that are generated from real-world patients record processing in the UK national cancer registry are carried out, which demonstrate significant (3 to 4 fold) performance gain over the fixed-order processing. MARS represents one of the few examples demonstrating that machine learning can assist humans with complex jobs by automating complex triaging tasks.

CCS Concepts: • **Applied computing → Health informatics;**

Additional Key Words and Phrases: Data entry, online learning, human-in-the-loop decision support system

ACM Reference format:
Cong Shen, Zhaozhi Qian, Alihan Huyuk, and Mihaela van der Schaar. 2022. MARS: Assisting Human with Information Processing Tasks Using Machine Learning. *ACM Trans. Comput. Healthcare* 3, 2, Article 21 (February 2022), 19 pages.
https://doi.org/10.1145/3494582

1 INTRODUCTION

It is widely believed that with the advances of **artificial intelligence** (**AI**) and **machine learning** (**ML**), machines are taking over human jobs [3, 27]. Closer studies have argued that this is not entirely true—lower-level jobs that can be rigorously streamlined are much more susceptible to be replaced by machines than jobs that require "intelligence" and "creativity", such as artists, architects, IT specialists, and public relations professionals [1, 7]. However, the views should be also taken with a grain of salt, as the boundaries between such low-level and high-level jobs are constantly shifting and becoming much more obscure with AI and ML [2].

CS acknowledges the funding support by the US National Science Foundation (NSF) under Grants ECCS-2029978, ECCS-2033671, and CNS-2002902. The work of MvdS was supported in part by US ONR and NSF.
Authors' addresses: C. Shen, University of Virginia, Charlottesville, VA, 22904; email: cong@virginia.edu; Z. Qian and A. Huyuk, University of Cambridge, The Old Schools, Trinity Ln, Cambridge CB2 1TN, United Kingdom; emails: zq224@cam.ac.uk, ah2075@cam.ac.uk; M. van der Schaar, University of Cambridge, The Old Schools, Trinity Ln, Cambridge CB2 1TN, United Kingdom and University of California, Los Angeles, 57-127 Engineering IV Building, Los Angeles, CA 90095; email: mv472@cam.ac.uk.

2637-8051/2022/02-ART21 $15.00
https://doi.org/10.1145/3494582

Historically, tools were invented to assist humans to complete tasks, and today's machines should be no different, even with much more advanced computation and reasoning capabilities. It is, however, a new and challenging problem to design the *framework* that integrates machines with humans so that the former can seamlessly assist the latter to complete a task, one that can accommodate the ever-increasing "intelligence" of machines. Solving this problem has significant societal ramifications, as it is central to the ongoing concern of job losses to AI [23].

This article aims at developing one such framework, using data entry systems as a particular example. The objective of a data entry system is to construct and maintain a structured database that contains accurate information of required fields from possibly heterogeneous, unstructured, and sometimes untrustworthy data sources. Among the many tasks in a data entry system, the "low intelligence" processing tasks have already been automated with **computer vision** (**CV**) and **natural language processing** (**NLP**) tools. Nevertheless, the "high intelligence" tasks of control, that determines *which* record to process next among the large volume of unstructured data records, is significantly harder as it requires understanding the quality of the data sources and making educated predictions. In today's reality, this is exclusively done by human experts, and there is a strong desire to develop ML algorithms to automate the record selection process [16]. Doing so would further reduce the workload of data entry clerks and allow them to focus on other high-value tasks.

However, achieving this goal faces several unique challenges. Data records are heterogeneous as they come from different sources and may contain different sets of reports with overlapping information. They can also be unstructured, leading to a large variation of processing cost. Last but not the least, the quality of different "raw" records may vary significantly. Record content may be missing or inconsistent due to data collection and conversion errors. Recognizing processing errors alone is often complex enough that domain expert knowledge is an indispensable requirement in contemporary data processing systems.

In this article, we aim at addressing the aforementioned challenges by proposing a new **Machine Assisted Record Selection** (**MARS**) paradigm to automate the record selection process in data entry systems. We use clinical data registry as an example to highlight the proposed paradigm, but the methodology is general enough to be applied to other applications in healthcare data processing and beyond. To handle heterogeneous reports with a large variation of processing cost, MARS has two sets of algorithms, **fixed-order greedy** (**FOG**) and **varying-order greedy** (**VOG**), that sequentially select the next record and the subset of features to be processed. MARS is principled—there is no "black box" in the decision making. This is because we leverage two critical features (that are known to the domain expert). First, the cost can be reduced by removing the realized features from future report examination. Second, the set of features that are yet to be extracted vary based on the past observations, and the optimal selection of the next report should be adjusted in an adaptive manner. When no prior knowledge about the records is given, *online* MARS is proposed that simultaneously learns such knowledge and selects the reports.

In addition, identifying errors or inconsistency in the heterogeneous patient records for clinical data registration often requires medical domain knowledge or societal background information, which can be better handled by a well-trained cancer registration staff than today's AI agents. On the other hand, by allowing machines to make decisions on record selection, human experts have a reduced workload, which can lead to fewer human errors and smaller task delay. The overall MARS paradigm demonstrates that pursuing performance gains does not necessarily lead to unfairness among humans and machines, as our design achieves "the best of both worlds'—they both exploit their strengths and avoid shortcomings, leading to improved system efficiency and fairness. This is further validated in the semi-synthetic experiments, which build on a data entry processing pipeline with data generated from statistics that are obtained from real-world patient records and data entry clerk processing in the UK national cancer registry. We see significant (3–4 fold) performance improvement of adapting the order of report examination based on past observations over the fixed-order algorithm.

The rest of this article is organized as follows. The proposed MARS system, together with the problem formulation is described in Section 2. The proposed algorithmic framework is presented in Sections 3 and 4. Theoretical

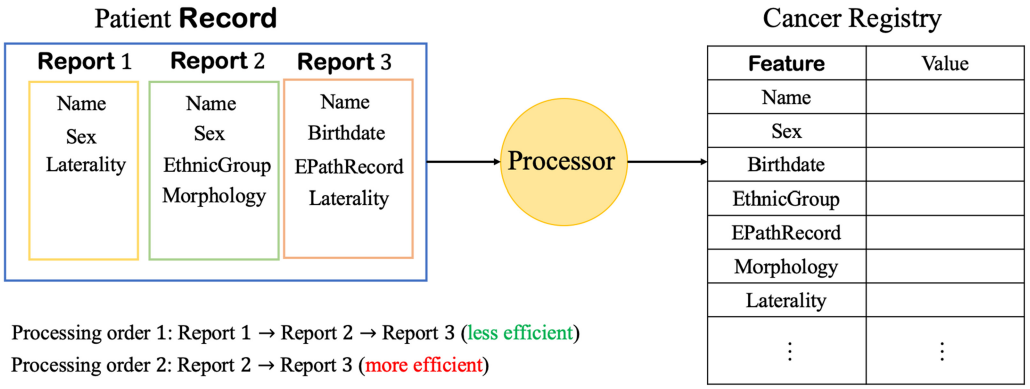

Fig. 1. Exemplary record with 3 reports containing relevant features for cancer registry.

performance analysis of MARS is given in Section 5. Numerical experiments are reported in Section 6, and related literature is discussed in Section 7. Finally, Section 8 concludes the article.

2 SYSTEM DESCRIPTION AND PROBLEM FORMULATION

2.1 Data Entry from Heterogeneous Sources

In this work, we use cancer registration as an exemplary setting to illustrate the proposed MARS framework for general data processing systems. The goal of a cancer registry is to create a high-quality clinical database that consists of relevant **features** for a given tumour. To accomplish this goal, the data entry processor takes incoming patients records that are collected from participating hospitals and clinics. These records arrive irregularly and sometimes unpredictably at the cancer registry center, with varying degrees of data quality. Each **record** is assumed to consists of some **reports**, e.g., pathology report, clinical meeting report, and surgery report. All of these reports may or may not have the needed features for cancer registration, and it only becomes clear after the report has been processed. Figure 1 illustrates this problem with a simple example: The required features in the registry are scattered in multiple reports of a patient's record, and the order in which these reports are processed makes a big difference to the effectiveness and efficiency of data entry processing.

2.2 Probabilistic Model

We use N and D to denote the maximum number of features and the maximum number of reports in a patient's record. Associated with each record is a random **feature indicator matrix** $X \in \{0,1\}^{N\times D}$, with each element $X_{i,j}$ assumed to be distributed independently as $X_{i,j} \sim \text{Bernoulli}(p_{i,j})$, where $X_{i,j} = 1$ indicates that feature i exists in report j, and 0 otherwise. The column and row vectors of X_{ij} are denoted as $X_{\cdot j}$ and $X_{i\cdot}$, and for a subset, $\mathcal{I} \subseteq \{1,\ldots,N\}$, $X_{\mathcal{I},j}$ is a column vector whose ith element is X_{ij} if $i \in \mathcal{I}$ and 0 otherwise. In addition, each report j is associated with a *quality* parameter ϵ_j, which denotes the probability that any feature is populated with an incorrect data item. We thus also have a random **quality indicator matrix** $Y \in \{0,1\}^{N\times D}$, where $Y_{i,j} \sim \text{Bernoulli}(1-\epsilon_j)$ is 1 if the item is correct, and 0 otherwise. Lastly, processing each report j and examining whether feature i is correctly populated incurs costs (e.g., processing time), which is modeled as a bounded independent random variable $C_{i,j}$. We do not specify its distribution but assume $\mathbb{E}[C_{i,j}] = \alpha_{i,j}$, and $\mathbb{E}[|C_{\cdot j}|_1] = \sum_{i=1}^{N} \alpha_{i,j} \doteq \alpha_j$. We denote the random **cost matrix** $C = [C_{i,j}] \in \mathcal{R}_+^{N\times D}$ with mean $A = [\alpha_{i,j}]$.

Before proceeding to the system description, we emphasize that the statistical modeling of X, Y, and C reflects the unknown and varying characteristics of the patient records. The realizations x (what features actually exist in the report), y (what existing features are incorrectly populated), and c (the processing cost of extracting these

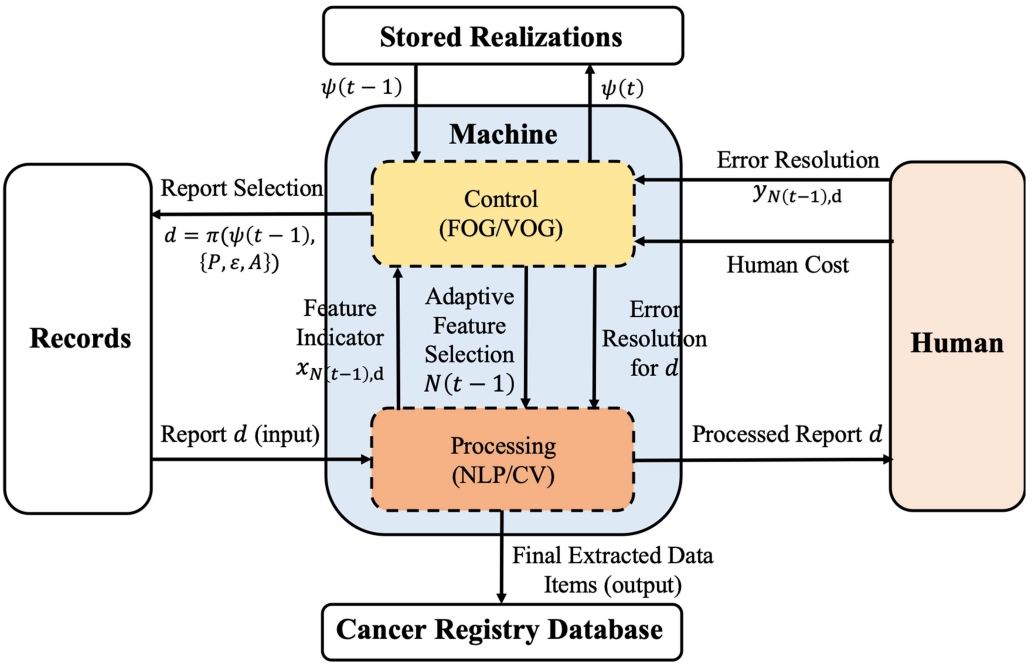

Fig. 2. The MARS framework for record selection in building cancer registry.

features and examining them) of a record are *unknown until they are processed by MARS*, with a control unit making the decisions while facing these uncertainties to assist the overall data entry work flow.

2.3 System Description

Figure 2 illustrates the relationships among various entities in the proposed data entry system and how they interact with each other in MARS. Note again that we focus on the decision problem of which report to examine in the control unit of Figure 2, and how the machine interacts with a human expert for error resolution. Specifically, the control unit adopts a policy π to determine which report to process next. At time step t, let $\tilde{\mathcal{D}}(t)$ be the set of reports that have not been examined yet in $\tilde{\mathcal{D}}$. We use $\mathcal{N}(t)$ to denote the set of features whose correct data items are still unknown after time t. The control unit selects record $\tilde{D}(t)$ and the processing unit extracts relevant features within $\mathcal{N}(t-1)$ from this report. A feature indicator matrix $x_{\mathcal{N}(t-1),\tilde{D}(t)}$ is revealed to the control unit, and the actual extracted features are sent for human examination, which returns the error indicator matrix $y_{\mathcal{N}(t-1),\tilde{D}(t)}$ together with the processing cost. The extracted features are then re-processed with error resolution, and the existing and correct data items are finally stored in the data registry, i.e.,

$$\tilde{x}_{\mathcal{N}(t-1),\tilde{D}(t)} = x_{\mathcal{N}(t-1),\tilde{D}(t)} \wedge y_{\mathcal{N}(t-1),\tilde{D}(t)}.$$

The current realization is stored and the processing continues to $t+1$.

2.4 Problem Formulation

The performance of policy π at time t is measured by the number of successfully collected features

$$\sigma(t) = N - |\mathcal{N}(t)|. \tag{1}$$

The goal for designing policy π is to maximize the expected number of successfully collected features at the end of budget T

$$\max_{\pi} \mathbb{E}\left[\sigma(T)\right],$$

with minimal expected total cost

$$\mathbb{E}\left[\sum_{t=1}^{T} \left\| c_{\mathcal{N}(t-1),\tilde{D}(t)} \right\|_1\right].$$

The report selection policy design can be posed in both fixed-order and adaptive settings, both of which assume that the probabilistic information of $X_{i,j}$, $Y_{i,j}$, and $C_{i,j}$ is available to the MARS control unit a priori. In a fixed-order setting, the selection of any future reports after t does not depend on the actual realizations x, y, and c in the past. Essentially, the fixed-order policy π_{fo} re-orders the set of report types $\mathcal{D}$ and obtains $\tilde{\mathcal{D}}$ using only the knowledge of $\{P, \epsilon, A\}$.

In the adaptive setting as illustrated in Figure 2, policy π_{a} sequentially selects a non-repetitive report from $\mathcal{D}$. Let us define the *realization* $\psi(t)$ as the set of selected reports, observed feedback, and actual cost at the end of time t

$$\psi(t) = \left\{ \left(\tilde{D}(\tau), x_{\mathcal{N}(\tau-1),\tilde{D}(\tau)}, y_{\mathcal{N}(\tau-1),\tilde{D}(\tau)}, c_{\mathcal{N}(\tau-1),\tilde{D}(\tau)} \right)_{\tau=1}^{t} \right\}. \tag{2}$$

The policy for the adaptive setting is formally defined as a mapping from the past realizations to the next report to choose

$$\tilde{D}(t+1) = \pi_{\text{a}}\left(\psi(t), \{P, \epsilon, A\}\right). \tag{3}$$

In both settings, the learner has access to $\{P, \epsilon, A\}$ as an input of the problem. Such assumptions are reasonable because for certain use cases, these values can be estimated from, e.g., the existing registries, representing a quantitative interpretation of domain expert guidance. However, there are also situations where such prior domain knowledge does not exist, and the system is built entirely from a clean slate. When this information is unavailable a priori, an online setting is motivated where the proposed solution needs to simultaneously determine the future actions and learn the problem-dependent variables. This problem is addressed in Section 4.

3 REPORT SELECTION ALGORITHMS

Three categories of data entry algorithms are presented in this section: non-adaptive, adaptive, and online. The differences primarily lie in whether the a priori statistical knowledge is available, or whether the action can be based on actual realizations of the reports. This section focuses on developing both non-adaptive and adaptive policies for the data entry problem where the learner has access to the values of p_{ij} and ϵ, which can be estimated from existing cancer registries if available. The difference between both policies is that the adaptive algorithm takes into account the realization $\psi_k(t)$ when choosing the next report, while the non-adaptive algorithm does not. As a result the non-adaptive algorithm always outputs the same sequence of reports for all patients.

To facilitate the discussion, we summarize the main notations in Table 1. We also note that all proposed algorithms in Sections 3 and 4 have theoretical performance analyses that are presented in Section 5.

3.1 FOG

In the fixed-order setting, since no realization can be observed to facilitate the future decisions, we can absorb ϵ_j into the "effective" existence probability $\tilde{p}_{i,j} = p_{i,j}(1-\epsilon_j)$. In other words, only the probability that a *correct* feature is collected matters when making the report selection decision.

Generally, the optimal solution can be computed from an exhaustive search of all possible combinations, where for each chosen $\tilde{\mathcal{D}}$ of T reports, we have

$$\mathbb{E}\left[\sigma(T)\right] \doteq F(\tilde{\mathcal{D}}) = \sum_{n=1}^{N} F_n(\tilde{\mathcal{D}}), \tag{4}$$

Table 1. Table of Notations

Notation	Explanation
P	The probability matrix. Each element is denoted as $p_{i,j}$.
N	The number of different data items/features/fields.
D	The maximum number of reports.
$\mathcal{D}$	The set of all reports. The cardinality $\|\mathcal{D}\| = D$.
T	Maximum number of reports that can be examined for a given patient.
ϵ_d	The probability that any feature in report d is populated with incorrect data.
α_d	The inverse of the mean cost.
η	The minimum number of correct features a successful registry must contain.
$X_{i,j}$	The Bernoulli random variable with success probability $p_{i,j}$.
Y_d	The Bernoulli random variable with success probability $1-\epsilon_d$ indicating which fields are error-free.
C_d	The random variable representing the cost for processing report d. $\mathbb{E}(C_d) = 1/\alpha_d$.
π	The policy to decide which reports to choose.
x	A realization of X representing the fields registered from a report. It potentially contains error.
y	A realization of Y indicating which fields are free of errors.
$\tilde{x}$	The registered fields with errors removed. $\tilde{x} = x \wedge y$.
$\mathcal{D}_k$	The set of records available to patient k, $\mathcal{D}_k \subseteq \mathcal{D}$.
$\tilde{\mathcal{D}}_k$	A reordering of $\mathcal{D}_k$. The order is determined by the policy.
$\tilde{\mathcal{D}}_k(t)$	The set of reports that have not been examined yet in $\tilde{\mathcal{D}}_k$ at time t.
$\sigma(t)$	The cumulative reward up to time t. This is the number of fields collected by time t.
$\gamma(t)$	The cumulative cost up to time t.
$\psi(t)$	The realizations up to time t. It contains the set of selected reports, observed feedback, and cost.
$\mathcal{N}(t)$	The set of fields whose correct data items are still unknown after time t. It can be derived from $\psi(t)$.
$F(\tilde{\mathcal{D}})$	Expected total reward after examining reports in $\tilde{\mathcal{D}}$.
$\Delta(d\|\psi(t-1))$	Expected gain for choosing report d at time t after observing realization $\psi(t-1)$.

where

$$F_n(\tilde{\mathcal{D}}) = 1 - \prod_{d \in \tilde{\mathcal{D}}} (1 - \tilde{p}_{n,d}) \,. \tag{5}$$

For a given set of reports $\mathcal{D}$, adding a new report d would result in an increased expected reward vector for all features

$$\Delta(d|\mathcal{D}) = [F_1(\{d \cup \mathcal{D}\}) - F_1(\{\mathcal{D}\}), \ldots, F_N(\{d \cup \mathcal{D}\}) - F_N(\{\mathcal{D}\})] \,, \tag{6}$$

while increasing the average total cost by α_d. The FOG algorithm sequentially selects a new report d that maximizes the marginal sum increase normalized by the average additional cost

$$\frac{||\Delta(d|\mathcal{D})||_1}{\alpha_d}.$$

FOG is formally presented in Algorithm 1.

ALGORITHM 1: FOG Algorithm

Input: P; ϵ, and A.
Initialize: $\mathcal{D}_{\text{fog}} = \varnothing$; $\mathcal{D}_{\text{res}} = [D]$; $t = 1$.
1: **while** $t \leq T$ **do**
2: Compute $d = \arg\max_{j \in \mathcal{D}_{\text{res}}} \frac{1}{\alpha_j} ||\Delta(j|\mathcal{D})||_1$;
3: Remove d from $\mathcal{D}_{\text{res}}$ and add it to the end of $\mathcal{D}_{\text{fog}}$;
4: $t = t + 1$.
5: **end while**
Output: Ordered set $\mathcal{D}_{\text{fog}}$.

3.2 VOG

The difference between adaptive and non-adaptive settings is that the adaptive one takes into account the realization $\psi_k(t)$ when choosing the next report. This also allows for potential error resolution before the next decision is taken.

We use $\mathcal{D}(t)$ and $\mathcal{D}_{\text{res}}(t)$ to denote the set of examined reports and the set of remaining reports after time step t, respectively. At time t, the algorithm has access to not only $\mathcal{D}(t-1)$, but also the realization $\psi(t-1)$. The importance of the actual realization is two-fold. First, we can now apply the greedy procedure only to those features that still have reward 0 (meaning that the features are unknown, due to either no or incorrect realizations in previous steps). Second, after the current decision d is made, we can observe the feature realizations, thus only keeping the realized features with the correct data, with the help of domain experts.

The complete VOG algorithm is presented in Algorithm 2. We first define $\bar{\Delta}(d|\psi(t-1))$ as the conditional normalized expected gain for choosing report d at time t. For each $d \in \mathcal{D}_{\text{res}}(t-1)$, we have

$$\bar{\Delta}(d|\psi(t-1)) = \frac{(1-\epsilon_d)\mathbb{E}\left[\sum_{i\in\mathcal{N}(t-1)} X_{i,d}\right]}{\mathbb{E}\left[\sum_{i\in\mathcal{N}(t-1)} C_{i,d}\right]} = \frac{\sum_{i\in\mathcal{N}(t-1)} \tilde{p}_{i,d}}{\sum_{i\in\mathcal{N}(t-1)} \alpha_{i,d}}. \tag{7}$$

Note that in Equation (7), the numerator represents the expected number of *additional* correct features registered after examining report d, while the denominator represents the expected cost associated with it; both are only based on fields that have not been registered before t. The adaptive algorithm then greedily chooses report with the greatest $\bar{\Delta}(d|\psi(t-1))$ at time t (step 2), and the extract features from the selected report, resulting in a realized feature vector $x_{\mathcal{N}(t-1),d}$ (step 4).

The design for error resolution relies on a human-in-the-loop action, as it typically requires expert knowledge to recognize errors or inconsistencies in the reports. At this stage, a registration staff examines the populated features and flags the ones with error (step 5).[1] The adaptive algorithm thus only keeps the features that are populated with the correct data for future evaluation (step 6). The total cost of extracting features from the selected report, expert examination, and feature correction is then observed by the control unit in MARS (step 7).

ALGORITHM 2: VOG Algorithm

Input: P; ϵ, and A.
Initialize: $\psi = \varnothing$; $\mathcal{D}_{\text{res}} = [D]$; $\mathcal{N} = [N]$; $t = 1$; $y_{i,j} = 1$ for $i \in [N]$ and $j \in [D]$.
1: **while** $t \leq T$ **do**
2: Compute $d = \arg\max_{j\in\mathcal{D}_{\text{res}}} \bar{\Delta}(j|\psi)$ according to (7); // Select Report
3: Remove d from $\mathcal{D}_{\text{res}}$;
4: Select report d and observe feature realization $x_{\mathcal{N}(t-1),d}$; // Feature Extraction
5: Expert validates realized feature i and sets $y_{i,d}$ to 0 if error is found; // Expert Examination
6: $\tilde{x}_{\mathcal{N}(t-1),d} = x_{\mathcal{N}(t-1),d} \wedge y_{\mathcal{N}(t-1),d}$; // Feature Correction
7: Observe individual costs $c_{\mathcal{N}(t-1),d}$;
8: $\psi = \psi \cup \{d, x_{\mathcal{N}(t-1),d}, y_{\mathcal{N}(t-1),d}, c_{\mathcal{N}(t-1),d}\}$;
9: $t = t + 1$.
10: **end while**
Output: $\{d(t), \tilde{x}_{\mathcal{N}(t-1),d(t)}\}_{t=1}^{T}$.

[1]The human expert involvement for addressing error resolution is the current state-of-the-art to the best of the authors' knowledge. It is, however, conceivable that this task can be automated with advanced ML and AI techniques in the future, which is itself an interesting research question that is worth investigation.

We remark that the VOG design for the adaptive record selection problem not only allows both machine and human expert to exploit their strengths but also improves the *interpretability* of the final outcome from an AI algorithm via human involvement, not just at the last stage of decision making but throughout the entire process. In addition, *fairness* can also be easily incorporated, as domain experts can influence the extracted features by nulling some data items. The singular criterion of error resolution can be extended to incorporate some fairness criteria to achieve a more balanced objective.

3.3 Queuing Delay

The expected queueing delay of a patient record is proportional to the expected total cost for different algorithms. For FOG, the set of features to be extracted remains the same, i.e., $\mathcal{N}(t) = [N]$ regardless of t. This means that the expected total cost will be $\sum_{j \in \mathcal{D}_{\text{fog}}} \alpha_j$. On the other hand, the VOG algorithm has an expected total cost $\mathbb{E}[\sum_{t=1}^{T} \|c_{\mathcal{N}(t-1), \tilde{D}(t)}\|_1]$, which is difficult to analyze but we can show that it is superior to FOG for the same set of selected reports. Intuitively, the adaptive algorithm prioritizes the reports that, on average, generate more useful (unknown) information while incurring less cost. Hence, for a given amount of budget for the total cost, more records can be processed, thus reducing the queueing delay. This aspect will be further evaluated in the experiment.

4 ONLINE ALGORITHMS

The online setting is arguably a more important and challenging use case, because an accurate estimate of the report "quality" may not always be available a priori. In fact, such issues are not limited to AI algorithms—senior data entry clerks often know better from their experiences which reports or sources are more likely to provide the needed information than junior clerks, who have to gain such knowledge by trial and error. This aspect can be tackled by a new online learning-based control unit design, which is the focus of this section.

As it is clear from Algorithms 1 and 2, only the *effective existence probability* $\tilde{p}_{i,j}$ impacts the algorithms. Hence, without loss of generality and to simplify the notation, we ignore ϵ and only focus on estimating P. In addition, the cost distribution A needs to be learned online as well for the VOG algorithm (it is not needed in FOG).

The online algorithm is designed as a clean-slate solution with no prior knowledge of the quality and processing cost of the records, so that the learner determines policies over multiple episodes. In each episode, it selects T reports dictated by a policy and observes the realizations of the reports that are selected as feedback. Decisions of the control engine are based on the feedback that it has received in previous episodes.

4.1 Online FOG

It is possible to have a direct estimate of the unknown $p_{i,j}$ from the selected reports and collected data items. However, since future decisions are based on these estimates, the algorithm needs to account for the estimation uncertainty. For simplicity, we consider the cost fixed and be the same for all reports. With this assumption, we note that the fixed-order online problem can be considered a particular instance of the probabilistic maximum coverage problem described in [4]. By applying the principle of [4] to Algorithm 1, we have the fixed-order online greedy record processing in Algorithm 3.

For each element of the Bernoulli distribution matrix, Algorithm 3 maintains an average of the previous observations ($\hat{p}_{i,j}$) and an exploration bonus ($\rho_{i,j}$) based on the number of such observations ($K_{i,j}$). In each episode, the exploration bonuses are added to the averages to obtain an "optimistic" estimate of P. Then, Algorithm 1 is run with these optimistic estimates as input to determine a policy.

4.2 Online VOG

The online problem can be viewed as a variation of the adaptive submodular maximization in a bandit setting, as discussed in [9]. The difference is that [9] deals with binary or categorical realizations for items, while our work

ALGORITHM 3: Online FOG Algorithm

Input: N, D, Algorithm 1.
Initialize: Set $\hat{p}_{i,j}, K_{i,j}$ for all $i \in [N], j \in [D]$ by selecting each report at least once.

for $\tau = 1, 2, \ldots$ **do**
  $\rho_{i,j} \leftarrow \sqrt{1.5 \log t / K_{i,j}}$ for all $i \in [N], j \in [D]$;
  $\bar{p}_{i,j} \leftarrow \hat{p}_{i,j} + \rho_{i,j}$ for all $i \in [N], j \in [D]$;
  $\mathcal{D} \leftarrow$ Algorithm 1 $(\bar{P})$;
  Run policy $\mathcal{D}$, observe $x_{\cdot,j}$ for all $j \in [D]$;
  for $i \in [N], j \in [D]$ **do**
    $\hat{p}_{i,j} \leftarrow (x_{i,j} + \hat{p}_{i,j} K_{i,j})/(1 + K_{i,j})$;
    $K_{i,j} \leftarrow 1 + K_{i,j}$.
  end for
end for

handles vectorial realizations of reports. We adapt the OASM algorithm in [9] as Algorithm 4. Differently from Algorithm 3, this algorithm also maintains averages for reciprocal costs, that is $\beta_{i,j} \doteq 1/\alpha_{i,j}$. For simplicity, we assume that all costs are lower bounded by 1, but our results can be generalized to any positive lower bound on costs. This assumption guarantees that $\beta_{i,j} \in (0, 1]$.

ALGORITHM 4: Online VOG Adaptive Algorithm

Input: N, D, Algorithm 2
Initialize: Set $\hat{p}_{i,j}, \hat{\beta}_{i,j}, K_{i,j}$ for all $i \in [N], j \in [D]$ by selecting each report at least once

for $\tau = 1, 2, \ldots$ **do**
  $\rho_{i,j} \leftarrow \sqrt{1.5 \log t / K_{i,j}}$ for all $i \in [N], j \in [D]$
  $\bar{p}_{i,j} \leftarrow \min\{1, \hat{p}_{i,j} + \rho_{i,j}\}$ for all $i \in [N], j \in [D]$
  $\bar{\beta}_{i,j} \leftarrow \min\{1, \hat{\beta}_{i,j} + \rho_{i,j}\}$ for all $i \in [N], j \in [D]$
  $\pi \leftarrow$ Algorithm 2 $(\bar{P}, \bar{A})$
  Run policy π, obtain final observations ψ
  for $i \in [N], j \in [D] : (j, x_{i,j}, c_{i,j}) \in \psi$ **do**
    $\hat{p}_{i,j} \leftarrow (x_{i,j} + \hat{p}_{i,j} K_{i,j})/(1 + K_{i,j})$
    $\hat{\beta}_{i,j} \leftarrow (1/c_{i,j} + \hat{\beta}_{i,j} K_{i,j})/(1 + K_{i,j})$
    $K_{i,j} \leftarrow 1 + K_{i,j}$
  end for
end for

5 PERFORMANCE ANALYSIS

The performance study in this section is carried out to provide insight into the proposed algorithms. In particular, we see that different principles (monotone submodularity for FOG and VOG, and upper confidence bound for online algorithms) that exist in the algorithms help to bound their performances.

5.1 FOG and VOG

For a deterministic set of costs, which could be different across reports, the analysis is a simple application of the budgeted maximum coverage problem in [13], and we arrive at the following result leveraging monotone submodularity.

Theorem 1. *$\mathcal{D}_{\text{fog}}$ obtained from Algorithm 1 achieves:*

$$F(\mathcal{D}_{\text{fog}}) \geq \left(1 - \frac{1}{e}\right) \max_{\tilde{\mathcal{D}} \subseteq \mathcal{D}: |\tilde{\mathcal{D}}| \leq T} F(\tilde{\mathcal{D}}). \tag{8}$$

Proof. This is an application of the well-known $(1 - 1/e)$-approximation [8] once we verify that function $F_n(\cdot)$ is monotone submodular, which is straightforward using the definition. □

The analysis for Algorithm 2 is much more involved due to the simultaneous impacts of sequential rewards and costs. If we assume the same constant cost for all reports, we have Theorem 2.

Theorem 2. *The expected reward $f_{\text{avg}}(\pi)$ of Algorithm 2 achieves:*

$$f_{\text{avg}}(\pi) \geq \left(1 - \frac{1}{e}\right) f_{\text{avg}}(\pi^*), \tag{9}$$

where π^* is the optimal adaptive policy and all costs are constant and the same.

Proof. We first define two operations on policies. The **policy truncation** $\pi_{[i]}$ denotes a new policy obtained by running policy π for at most i steps. The **policy concatenation** $\pi_1@\pi_2$ denotes a new policy obtained from first running π_1 to its finish and then running π_2 without considering the realizations collected from π_1, i.e., running π_2 from a fresh start. If π_2 chooses a report d that has been examined previously by π_1, it gets the same realization $x_{\cdot,d}$ as before.

For the optimal adaptive policy π^* that is allowed to run T steps, we have the following inequality for the greedy policy π:

$$\begin{aligned} f_{\text{avg}}(\pi^*) &\leq f_{\text{avg}}(\pi_{[i]}@\pi^*) \\ &\leq f_{\text{avg}}(\pi_{[i]}) + T\left(f_{\text{avg}}(\pi_{[i]}@\pi^*_{[1]}) - f_{\text{avg}}(\pi_{[i]})\right) \\ &\leq f_{\text{avg}}(\pi_{[i]}) + T\left(f_{\text{avg}}(\pi_{[i+1]}) - f_{\text{avg}}(\pi_{[i]})\right). \end{aligned} \tag{10}$$

The first inequality follows from the monotonicity of σ: examining more reports never decreases the reward. The second inequality is due to the submodularity of σ. The third inequality holds because π is a greedy policy.

Let us define $\delta_i \doteq f_{\text{avg}}(\pi^*) - f_{\text{avg}}(\pi_{[i]})$. After rearranging the terms in the above inequality, we obtain $\delta_{i+1} \leq (1 - \frac{1}{T})\delta_i$. Without loss of generality, we assume the greedy policy is also allowed to run T steps. In this case, we have $\delta_T \leq (1 - \frac{1}{T})^T \delta_0 < (1 - \frac{1}{e})\delta_0$, where for this last inequality we have used the fact that $1 - x < e^{-x}$ for all $x > 0$. Further rearranging the terms proves Theorem 2. □

5.2 Online Learning

The online algorithms estimate P and A while simultaneously selecting reports, which can lead to suboptimal performance. The performance is measured by comparing its cumulative reward with the cumulative reward that could have been achieved by the algorithms with full knowledge of P and A from the very beginning of the process, and define the expected difference between the two quantities as *regret*. For the online FOG case, the regret up to episode $\tilde{T}$ can be written as

$$\text{Reg}_{\text{fog}}(\tilde{T}) = \tilde{T}\left(1 - \frac{1}{e}\right) \max_{\tilde{\mathcal{D}}} F(\tilde{\mathcal{D}}) - \mathbb{E}\left[\sum_{\tau=1}^{\tilde{T}} F(\mathcal{D}_\tau)\right],$$

where $\mathcal{D}_\tau$ is the policy determined by the learner at episode τ. For the adaptive case, the regret up to episode $\tilde{T}$ can be written as

$$\mathrm{Reg}_a(\tilde{T}) = \tilde{T} f_{\mathrm{avg}}(\pi^g) - \mathbb{E}\left[\sum_{\tau=1}^{\tilde{T}} f_{\mathrm{avg}}(\pi_\tau)\right],$$

where π^g is the greedy policy given in Algorithm 2 and π_τ is the policy determined by the learner at episode τ.

Theorem 3 bounds the regret of Algorithm 3.

Theorem 3. *The regret of Algorithm 3 up to episode $\tilde{T}$ is bounded as*

$$Reg_{\mathrm{fog}}(\tilde{T}) \le \sum_{\substack{i\in[N],\\ j\in[D]:\\ \Delta^{i,j}_{\min}>0}} \frac{12N^2D^2\log\tilde{T}}{\Delta^{i,j}_{\min}} + \left(1+\frac{\pi^2}{3}\right) ND\Delta_{\max},$$

where $\Delta^{i,j}_{\min} = (1-1/e)\max_{\tilde{\mathcal{D}}} F(\tilde{\mathcal{D}}) - \max_{\tilde{\mathcal{D}}:j\in\tilde{\mathcal{D}}} F(\tilde{\mathcal{D}})$ and $\Delta_{\max} = (1-1/e)\max_{\tilde{\mathcal{D}}} F(\tilde{\mathcal{D}}) - \min_{\tilde{\mathcal{D}}} F(\tilde{\mathcal{D}})$.

The proof of Theorem 3 can be derived by noting that it is a special case of Theorem 1 in [4]. For more details, see Section 4.2 in [4], which describes how Theorem 1 is applied to the PMC problem.

Theorem 4 bounds the adaptive regret of Algorithm 4.

Theorem 4. *The adaptive regret of Algorithm 4 up to episode $\tilde{T}$ is bounded as*

$$Reg_a(\tilde{T}) \le N^2T\sum_{j=1}^{D}\ell_j + \frac{1}{3}N^2D(D+1)T,$$

where $\Delta_j = \min_{\psi:j\neq\pi^g(\psi)}(\bar{\Delta}(\pi^g(\psi)|\psi) - \bar{\Delta}(j|\psi))$ and $\ell_j = \lceil 6(N+N^2)^2\log\tilde{T}/\Delta_j^2\rceil$.

The proof of Theorem 4 is technically very complicated and is deferred to Appendix A.

6 EXPERIMENTS

6.1 Setup

In the experiment we have new patient records (e.g., the national, local, and other feeds in Scottish Cancer Registry [15]) arrive daily following a Poisson arrival process with rate λ (patients per day).[2] The records are generated, stored, and processed as described in Section 2. If the sum of the cost in one day reaches a predefined budget (8 working hours), the processor stops the action for that day.[3] The remaining reports in the queue will wait to be processed in subsequent days.

The control unit in the data entry processor employs the algorithms detailed in previous sections to determine the report processing order. In addition, we allow for an early exit when the successfully collected features exceed a predefined threshold η. This is a meaningful setting and widely adopted in practice, particularly for varying reports, because not all data may be available for each patient. We assume that the processor can examine at most T reports for each patient. If the number of registered features does not reach η, this particular processing attempt is marked as a failure.

As discussed before, the registered features may contain errors. We randomly generate the error probability ϵ_d from a uniform distribution in $[0, 1]$. In the constant cost case, the cost for examining each report α_d is fixed

[2]For example, the national cancer registry in the United Kingdom receives approximately 6,000 to 10,000 records per week for 40 tumour types.

[3]In theory, computers do not have to be bounded by the limited budget. However, since the MARS framework have a human component and in order to have a fair comparison, we have enforced the same budget for different methods.

Table 2. Basic Experiment Configuration

Parameter	Sim. Setting	Real-World Statistics
# Iterations	20	5 years
# Patients	60,000	60,765 per month
# Features: N	64	63 on average
# Report Types: D	16	15 in total
Patient Arrival Rate: λ	2,000	2,026 per day
Success Threshold: η	38	38 complete fields
Cost α: Constant	1	1 working hour
Cost α: Varying	~Exp(1)	~Exp(1) hour

as one while in the varying cost case it is generated from an Exponential distribution with mean one. When the VOG algorithm makes use of the human-in-the-loop error resolution, it will incur 20% additional (human) cost, which is obtained from practice [15].

Whether a feature is registered from a report or not depends on the underlying probability matrix P. We generate the matrix P in a way that only a small number of entries in each column is non-zero. Hence, only a small set of features is likely to be registered from a particular type of report. This is a reasonable assumption because medical reports fall into different "topics" and each topic may contain very different information. For example, medical test reports are likely to contain bio-markers but they are very unlikely to cover treatment plans or diagnosis codes. We also make sure that the features have a certain level of overlapping across reports. This situation is very common in reality. For example, gender and age information is present in almost all medical reports.

6.2 Baselines

We compare the proposed FOG algorithm and VOG algorithm with two baseline approaches. The first baseline (**Random**) goes through the reports in a random order. The algorithm stops when either η features have been registered or it has gone through all T reports. This baseline mimics a real-world case where inexperienced data entry staff (possibly without rigorous training) randomly process the incoming patient reports.

The second baseline algorithm (**Naive Order**) is designed to resemble a one-size-fits-all report selection guideline. This baseline mimics a real-world situation where a detailed, step-by-step data entry processing procedure is provided to the staff and they are asked to strictly follow the documented instructions. It orders reports by the expected number of features per unit cost i.e., $\sum_i P_{ij}/\alpha_j$. The algorithm does not adapt to the partial realization $\psi(t)$ obtained during processing.

6.3 Main Results

Our experiments are based on an anonymized dataset collected from the UK national cancer registry. The dataset contains the detailed information about the operation of the cancer registry including (1) each patient record's arrival time and processing time, (2) the number of fields (features) successfully extracted from each report, and (3) if the patient is deemed successfully registered. The dataset covers the period spanning five years from 2014 to 2018. Due to the highly sensitive nature of the dataset, we extracted the key statistics which faithfully represent the real-world cancer registry (shown in Table 2). Based on these statistics, we perform two representative experiments, and the results are reported in this section. Table 2 lists the configuration for FOG and VOG adaptive algorithms, while Table 3 is for online algorithms. Both settings closely mirror the real scenario. In the first experiment, a total of 60,000 patients are added to the queue at an average rate of 2,000 patients per day. We evaluate the performance under both constant cost and varying random cost. This simulation is run independently for 20 times.

Table 3. Experiment Configuration for Online Algorithms

Parameter	Value
# Iterations	20
# Patients	60,000
# Features: N	64
# Report Types: D	64
Patient Arrival Rate: λ	2,000
Success Threshold: η	50

Table 4. Summary of the First Experiment Result

Cost	Algorithm	Capacity	Average Cost	Wait Time
Constant	Random	1.2 ± 0.1	155.9 ± 0.15	1,134.1 ± 2.8
	NaiveOrder	3.9 ± 0.1	166.1 ± 0.03	1,200.6 ± 0.6
	FOG	9.1 ± 0.1	68.6 ± 0.26	486.8 ± 2.6
	VOG	**25.9 ± 0.1**	**22.8 ± 0.05**	**136.8 ± 0.7**
Varying	Random	1.2 ± 0.1	159.3 ± 0.22	1,150.8 ± 2.4
	NaiveOrder	5.6 ± 0.1	113.9 ± 0.23	816.8 ± 2.8
	FOG	15.4 ± 0.1	39.8 ± 0.21	268.2 ± 2.2
	VOG	**63.0 ± 0.3**	**9.2 ± 0.03**	**26.3 ± 0.7**

Average Cost is measured in Minutes. Wait Time is measured in Days.
The bold values indicate the best performance.

In Table 4, we present the results averaged across 20 independent runs together with the standard deviations. We measure the processing **capacity** for various algorithms as the average number of successfully registered patients in a day. We are also interested to know how long a patient's record has to wait before getting processed. This is reflected in the metric **wait time** (days). Last but not least, the metric **average cost** tracks the average time (minutes) for processing each patient. A successful algorithm should have a high capacity, low average cost, and low waiting time.

The VOG algorithm achieves the best performance among all the algorithms being studied even when it takes 20% more cost due to human error resolution. Compared with the second-best algorithm, FOG, the processing capacity for VOG is increased by 3 to 4 folds. We see a similar order of reduction in its average cost and waiting time. This confirms our proposition that an efficient algorithm should adapt to the previous realizations and update the policy accordingly.

Turning to a comparison of the constant and varying cost cases, we see that the AI-powered algorithms do even better in the varying cost cases. The algorithms are able to exploit the fact that some reports have lower cost and higher marginal values than others.

The first experiment does not exhibit strong heterogeneity; all the patient records have the same number of reports. In reality, as discussed earlier in the article, patient reports often vary significantly. For example, if the physical condition of a particular patient does not allow for invasive medical tests, the corresponding reports will not be available. In the second experiment, we focus on the record heterogeneity and only allow a random subset of reports to be available for each patient. The cardinality of the random subset is assumed to be uniformly distributed between $D_{min} = 64$ and $D_{max} = 128$ so that at most 50% of reports are unavailable for a given patient record. Table 5 presents the experiment results. We observe that VOG again achieves top performance, mainly because it can react to the adversarial situation where the reports could be randomly missing.

Table 5. Summary of the Experiment Result When at Most 50% of Reports are Unavailable

Algorithm	Daily Capacity	Average Cost	Wait Time
Random	5.1 ± 0.1	98.0 ± 0.19	873.4 ± 1.8
NaiveOrder	10.4 ± 0.1	47.4 ± 0.16	417.0 ± 2.2
FOG	26.7 ± 0.2	18.5 ± 0.12	136.7 ± 1.4
VOG	**81.6 ± 0.5**	**5.9 ± 0.04**	**11.6 ± 0.7**

Average Cost is measured in Minutes. Wait Time is measured in Days.
The bold values indicate the best performance.

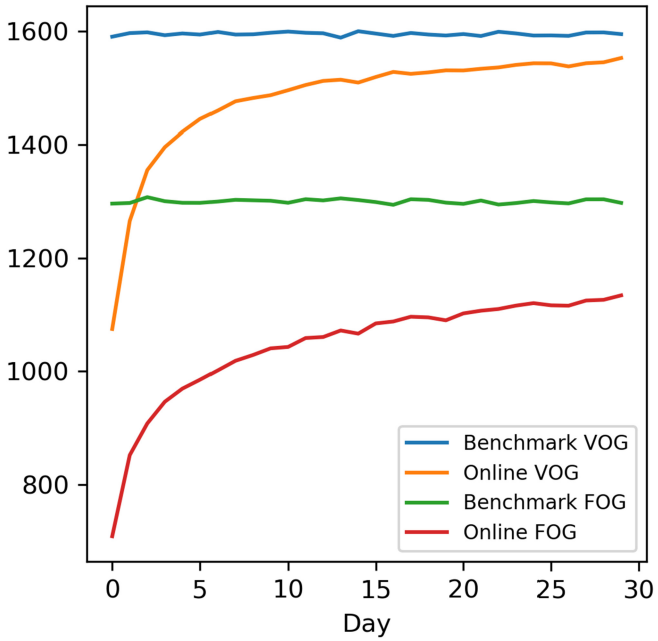

Fig. 3. The daily processing capacity for online and benchmark algorithms. Benchmark algorithms are given the true P.

6.4 Performance of Online Algorithms

Online FOG and VOG algorithms are similarly evaluated in the experiment. Figure 3 illustrates how the processing capacity of online algorithms grows over time as their estimate of P becomes better. We see that the online algorithms gradually converge to their corresponding benchmarks. Furthermore, the relatively slow convergence shown in this experiment highlights the need for prior knowledge. In reality, the statistical estimate does not necessarily come from the online operation. Leveraging the data entry systems from other similar tasks may lead to a good initial estimate, which can simplify the online operation and speed up the convergence.

6.5 Impact of Human Error Resolution

In this section, we compare the performance of the adaptive algorithm with and without human-in-the-loop error resolution. When the algorithm decides to register a report d, it will get a realization x_d. However, this realization will be subject to error. The features in x_d might be invalid for a probability ϵ_d. If no error correction is made, the algorithm will take the potential erroneous realization as the truth and proceed as usual. As a result, the features it registered may contain errors. When a data entry clerk performs error correction, all invalid

Table 6. Performance Comparison of Adaptive Algorithms with and without Human Involvement

Error Correction	Capacity	Average Cost	Success Rate
Yes	63.0 ± 0.3	9.2 ± 0.03	95.3% ± 0.2%
No	46.2 ± 0.5	7.1 ± 0.02	54.0% ± 0.4%

Average Cost is measured in Minutes.

Table 7. Summary of the Experiment Result when at Most 50% of Reports are Unavailable

Algorithm	Daily Capacity	Average Cost	Wait Time
Random	5.1 ± 0.1	98.0 ± 0.19	873.4 ± 1.8
NaiveOrder	10.4 ± 0.1	47.4 ± 0.16	417.0 ± 2.2
FOG	26.7 ± 0.2	18.5 ± 0.12	136.7 ± 1.4
VOG	**81.6 ± 0.5**	**5.9 ± 0.04**	**11.6 ± 0.7**

Average Cost is measured in Minutes. Wait Time is measured in Days.
The bold values indicate the best performance.

features in x_d will be removed, and the algorithm will be presented with the correct information only. However, this process will incur additional costs.

When evaluating whether a patient is successfully registered, we will remove all the invalid features in the data registry. If the number of registered features still exceeds the threshold η, we mark it as a success. We calculate the average success rate in addition to the capacity and average cost defined previously.

The performance summary is given in Table 6. Without error correction, the algorithm tends to terminate prematurely because it is over-optimistic about the number of features registered so far. This means that the algorithm can process more patient records but at the cost of a lower success rate. However, even considering both effects, the algorithm with error resolution still outperforms the one without in terms of the number of daily successful registration.

6.6 Experiment Results with Missing Reports

In practice, not all types of reports will be available for each patient. For example, if the physical condition of the patient does not allow for an invasive medical test, the test reports will not be acquired. Here, we test how well the algorithms adapt to the unavailable report situation. We assume that only a random subset of reports are available for each patient. The cardinality of the random subset is assumed to be uniformly distributed between D_{min} and D_{max}. In this simulation at most 50% of reports are unavailable for some patients. Table 7 reports the performance. VOG achieves top performance again because it can react to the adversarial situation where the reports are missing by design.

7 RELATED WORKS

Information processing from data is an active research area. For example, how to extract useful features from big data has been extensively studied [10, 14, 20]. Handling "big data" becomes very important in recent years due to the explosion of generated data [12, 18]. In particular, processing unstructured medical reports using NLP is relevant to our work and has seen significant advances over the past decade. The methods have evolved from the bag-of-words models [5] to more sophisticated deep-learning-based models [21] [6]. Validation studies [22] [11] have shown that NLP-based data registration tools have achieved human-level performance on certain tasks. Our work is different from these information processing techniques in that the focus is on automated decision making with potential human expert involvement. In other words, these specific information processing techniques are complimentary to MARS.

The implication of ML and AI on job creation, losses, and transfers have been among the most debated topics due to its significant societal and economical ramifications. The authors of [2] discuss the key implications of AI for the workforce by drawing on the rubric of AI capabilities. It is recognized that although the capabilities of ML systems have had an impressive growth over the past decade, it is still far from being suitable for all tasks [23]. The proposed MARS framework is a concrete example of the philosophy to integrate AI and ML to assist humans to complete tasks.

The MARS framework is also broadly related to the **human-in-the-loop machine learning** (**HITL-ML**) [24] and human-machine collaboration. The main principle of HITL-ML is to let humans track changes and intermediate results of the iterative process in ML, thus accelerating the iteration and providing quick responsive feedback. This design principle has led to successful applications in image classification [26] and text analytics [25]. Human-machine collaboration [19] has been discussed in the application of object detection [17], where CV models and multiple human inputs are integrated to generate object annotations. However, the focus of this line of research is on reducing the iteration delay and improving the performance of ML workflows, while our work focuses on the close collaboration between humans and AI.

8 CONCLUSIONS

We have shown that with today's ML capabilities, especially in the online learning setting, it is possible to automate the complex triaging tasks and allow machines to assist humans with "high intelligence" jobs. The MARS framework was proposed using information processing in data entry systems as the particular example, which improves the efficiency and quality of sophisticated report selection tasks (such as cancer registration) that involve heterogeneous records which can be highly unstructured with erroneous or inconsistent data items. The machine intelligence is principled (no "black box") – they rely on the monotone and submodular properties of the success probabilities as well as upper confidence bound in the online setting. The performance improvement of the proposed MARS framework was evaluated through experiments using synthetic data that mimic the real-world cancer patient records. Probably more importantly, we demonstrated that MARS not only improves the overall system performance but achieves a balanced human-machine interaction that could have important social benefit.

APPENDIX

A PROOF OF THEOREM 4

First, we re-write the adaptive regret as

$$\begin{aligned}
\mathrm{Reg}_a(\tilde{T}) &= \mathbb{E}\left[\sum_{\tau=1}^{\tilde{T}}\left(f_{avg}(\pi^g) - f_{avg}(\pi_\tau)\right)\right] \\
&\leq \mathbb{E}\left[\sum_{\tau=1}^{\tilde{T}}\sum_{j=1}^{D}\sum_{t=1}^{T}\mathbb{I}_{j,t,\tau}\left(f_{avg}\left(\pi^g\right) - f_{avg}(\pi_\tau)\right)\right] \\
&\leq \mathbb{E}\left[N\sum_{\tau=1}^{\tilde{T}}\sum_{j=1}^{D}\sum_{t=1}^{T}\mathbb{I}_{j,t,\tau}\times\mathbb{I}\{\exists i\in\mathcal{N}_\tau(t-1): K_{i,j}(\tau)\leq\ell_j\}\right] \qquad (11) \\
&+ \mathbb{E}\left[N\sum_{\tau=1}^{\tilde{T}}\sum_{j=1}^{D}\sum_{t=1}^{T}\mathbb{I}_{j,t,\tau}\times\mathbb{I}\{\forall i\in\mathcal{N}_\tau(t-1), K_{i,j}(\tau)>\ell_j\}\right], \qquad (12)
\end{aligned}$$

where $\mathbb{I}_{j,t,\tau}$ is the indicator of the event that policy π_τ selects report j in time step t in episode τ whereas π^g would have selected another report, and $\mathcal{N}_\tau(t)$ denotes the unknown features at the end of time step t in episode τ.

Then, Equation (11) can be bounded trivially as

$$(11) \leq N^2 T \sum_{j=1}^{D} \ell_j \ .$$

In order to bound Equation (12), we need to introduce some new notation. Let $\Psi_{j,t}$ denote the set of realizations ψ such that $\mathbb{I}_{j,t,\tau} = 1$ if $\psi_\tau(t-1) = \psi$, where $\psi_\tau(t)$ denotes the realization observed at the end of time step t in episode τ. Given a realization ψ, let $\mathcal{N}\langle\psi\rangle$ be the corresponding set of features that are still unknown and $j^*\langle\psi\rangle$ be the report that would be selected by π^g. Finally, let

$$f(p_{\cdot j}, \beta_{\cdot j}|\psi) \doteq \frac{\sum_{i\in\mathcal{N}\langle\psi\rangle} p_{i,j}}{\sum_{i\in\mathcal{N}\langle\psi\rangle} 1/\beta_{i,j}} \ .$$

Note that $f(p_{\cdot j}, \beta_{\cdot j}|\psi)$ is monotonic, meaning $f(p_{\cdot j}, \beta_{\cdot j}|\psi) < f(p'_{\cdot j}, \beta'_{\cdot j}|\psi)$ if $p_{i,j} < p'_{i,j}$ and $\beta_{i,j} < \beta'_{i,j}$ for all $i \in [N]$. It is also Lipschitz continuous, meaning

$$|f(p_{\cdot j}, \beta_{\cdot j}|\psi) - f(p'_{\cdot j}, \beta'_{\cdot j}|\psi)| \leq (N + N^2) \max_{i\in\mathcal{N}\langle\psi\rangle} \max\left\{|p_{i,j} - p'_{i,j}|, |\beta_{i,j} - \beta'_{i,j}|\right\} .$$

Then, for all reports j and time steps t, we have

$$\begin{aligned}\sum_{\tau=1}^{\tilde{T}} \mathbb{I}_{j,t,\tau}\mathbb{I}\{\forall i \in \mathcal{N}_\tau, K_{i,j}(\tau) > \ell_j\} \leq \sum_{\tau=\ell_j+1}^{\tilde{T}} \mathbb{I}\{\exists \psi \in \Psi_{j,t} : f(\bar{p}_{\cdot j}(\tau), \bar{\beta}_{\cdot j}(\tau)|\psi) \\ \geq f(\bar{p}_{\cdot j^*\langle\psi\rangle}(\tau), \bar{\beta}_{\cdot j^*\langle\psi\rangle}(\tau)|\psi) \wedge \forall i \in \mathcal{N}\langle\psi\rangle, K_{i,j}(\tau) > \ell_j\}.\end{aligned}$$

Note that the event we are left with implies that, for some $\psi \in \Psi_{j,t}$, at least one of the following events must be true:

$$\begin{aligned}\exists i \in \mathcal{N}\langle\psi\rangle : \hat{p}_{i,j^*\langle\psi\rangle}(\tau) &\leq p_{i,j^*\langle\psi\rangle}(\tau) - \rho_{i,j^*\langle\psi\rangle}(\tau) \\ \exists i \in \mathcal{N}\langle\psi\rangle : \hat{\beta}_{i,j^*\langle\psi\rangle}(\tau) &\leq \beta_{i,j^*\langle\psi\rangle}(\tau) - \rho_{i,j^*\langle\psi\rangle}(\tau) \\ \exists i \in \mathcal{N}\langle\psi\rangle : \hat{p}_{i,j}(\tau) &\geq p_{i,j}(\tau) + \rho_{i,j}(\tau) \\ \exists i \in \mathcal{N}\langle\psi\rangle : \hat{\beta}_{i,j}(\tau) &\geq \beta_{i,j}(\tau) + \rho_{i,j}(\tau) \\ f(p_{\cdot j^*\langle\psi\rangle}(\tau), \beta_{\cdot j^*\langle\psi\rangle}(\tau)|\psi) &< f(p_{\cdot j}(\tau), \beta_{\cdot j}(\tau)|\psi) + 2(N + N^2)\sqrt{\frac{3\log\tau}{2\ell_j}} \ . \end{aligned} \tag{13}$$

In order to see why, assume all these events except Equation (13) is false. Then, we have

$$\begin{aligned} f(p_{\cdot j^*\langle\psi\rangle}(\tau), \beta_{\cdot j^*\langle\psi\rangle}(\tau)|\psi) &< f(\bar{p}_{\cdot j^*\langle\psi\rangle}(\tau), \bar{\beta}_{\cdot j^*\langle\psi\rangle}(\tau)|\psi) && (14) \\ &\leq f(\bar{p}_{\cdot j}(\tau), \bar{\beta}_{\cdot j}(\tau)|\psi) \\ &\leq f(p_{\cdot j}(\tau), \beta_{\cdot j}(\tau)|\psi) + (N + N^2) \max_{i\in\mathcal{N}\langle\psi\rangle} 2\rho_{i,j}(\tau) && (15) \\ &\leq f(p_{\cdot j}(\tau), \beta_{\cdot j}(\tau)|\psi) + 2(N + N^2)\sqrt{\frac{3\log\tau}{2\ell_j}} \ , \end{aligned}$$

where Equation (14) is due to the monotonicity of f and Equation (15) is due to the Lipschitz continuity of f. The probabilities of all these events except Equation (13) can be bounded using the Hoeffding's inequality. We have

$$\begin{aligned}
&\mathbb{P}\left(\exists\psi \in \Psi_{j,t}, \exists i \in \mathcal{N}\langle\psi\rangle : \hat{p}_{i,j^*\langle\psi\rangle}(\tau) \le p_{i,j^*\langle\psi\rangle}(\tau) - \rho_{i,j^*\langle\psi\rangle}(\tau)\right)\\
&\le \sum_{i=1}^{N}\sum_{j^*=1}^{D}\sum_{k=1}^{\tau}\mathbb{P}\left(\hat{p}_{i,j^*}(\tau) \le p_{i,j^*}(\tau) - \rho_{i,j^*}(\tau)|K_{i,j^*}(\tau) = k\right)\\
&\le \sum_{i=1}^{N}\sum_{j^*=1}^{D}\sum_{k=1}^{\tau}\tau^{-3}\\
&\le ND\tau^{-2}
\end{aligned}$$

and

$$\begin{aligned}
&\mathbb{P}\left(\exists\psi \in \Psi_{j,t}, \exists i \in \mathcal{N}\langle\psi\rangle : \hat{p}_{i,j}(\tau) \ge p_{i,j}(\tau) + \rho_{i,j}(\tau)\right)\\
&\le \sum_{i=1}^{N}\sum_{k=1}^{\tau}\mathbb{P}\left(\hat{p}_{i,j}(\tau) \ge p_{i,j}(\tau) + \rho_{i,j}(\tau)|K_{i,j}(\tau) = k\right)\\
&\le \sum_{i=1}^{N}\sum_{k=1}^{\tau}\tau^{-3}\\
&\le N\tau^{-2}\,.
\end{aligned}$$

The remaining two events can be bounded in the same way.

When $\ell_j = \lceil 6(N+N^2)^2\log\tilde{T}/\Delta_j^2\rceil$, the event is given in Equation (13) is not possible at all. For all $\psi \in \Psi_{j,t}$, we have

$$\begin{aligned}
&f(p_{\cdot j^*\langle\psi\rangle}(\tau), \beta_{\cdot j^*\langle\psi\rangle}(\tau)|\psi) - f(p_{\cdot j}(\tau), \beta_{\cdot j}(\tau)|\psi) - 2(N+N^2)\sqrt{\frac{3\log\tau}{2\ell_j}}\\
&= \bar{\Delta}(j^*\langle\psi\rangle|\psi) - \bar{\Delta}(j|\psi) - 2(N+N^2)\sqrt{\frac{3\log\tau}{2\ell_j}}\\
&\ge \bar{\Delta}(j^*\langle\psi\rangle|\psi) - \bar{\Delta}(j|\psi) - \Delta_j\\
&\ge 0\,.
\end{aligned}$$

Then, Equation (12) can simply be bounded as

$$\begin{aligned}
(12) &\le \sum_{j=D}\sum_{t=1}\sum_{\tau=1}^{\infty}2N(1+D)\tau^{-2}\\
&\le \frac{1}{3}\pi^2 ND(D+1)T\,.
\end{aligned}$$

This completes the proof of Theorem 4.

REFERENCES

[1] Hasan Bakhshi, Carl B. Frey, and Michael Osborne. 2015. Creativity vs. robots. *The Creative Economy and The Future of Employment.* Nesta, London (2015).

[2] Erik Brynjolfsson and Tom Mitchell. 2017. What can machine learning do? Workforce implications. *Science* 358, 6370 (2017), 1530–1534.

[3] Stephen Cave and Seán S. ÓhÉigeartaigh. 2019. Bridging near- and long-term concerns about AI. *Nature Machine Intelligence* 1, 1 (2019), 5–6.
[4] Wei Chen, Yajun Wang, and Yang Yuan. 2013. Combinatorial multi-armed bandit: General framework and applications. In *Proceedings of the 30th International Conference on Machine Learning*. 151–159.
[5] Ryan Cobb, Sahil Puri, D. Zhe Wang, Tezcan Baslanti, and Azra Bihorac. 2013. Knowledge extraction and outcome prediction using medical notes. In *Proceedings of the ICML Workshop on Role of Machine Learning in Transforming Healthcare*.
[6] Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. 2019. A guide to deep learning in healthcare. *Nature Medicine* 25, 1 (2019), 24.
[7] Carl Benedikt Frey and Michael A. Osborne. 2017. The future of employment: How susceptible are jobs to computerisation? *Technological Forecasting and Social Change* 114 (2017), 254–280. https://doi.org/10.1016/j.techfore.2016.08.019
[8] Satoru Fujishige. 2005. *Submodular Functions and Optimization*. Elsevier.
[9] Victor Gabillon, Branislav Kveton, Zheng Wen, Brian Eriksson, and S. Muthukrishnan. 2013. Adaptive submodular maximization in bandit setting. In *Proceedings of the Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 2697–2705.
[10] Mihai Gurban and Jean-Philippe Thiran. 2009. Information theoretic feature extraction for audio-visual speech recognition. *IEEE Transactions on Signal Processing* 57, 12 (2009), 4765–4776.
[11] H. Hamid, S. J. Fodeh, A. G. Lizama, R. Czlapinski, M. J. Pugh, W. C. LaFrance Jr, and C. A. Brandt. 2013. Validating a natural language processing tool to exclude psychogenic nonepileptic seizures in electronic medical record-based epilepsy research. *Epilepsy & Behavior* 29, 3 (2013), 578–580.
[12] Zhu Han, Mingyi Hong, and Dan Wang. 2017. *Signal Processing and Networking for Big Data Applications*. Cambridge University Press.
[13] Samir Khuller, Anna Moss, and Joseph Seffi Naor. 1999. The budgeted maximum coverage problem. *Information Processing Letters* 70, 1 (1999), 39–45.
[14] Huizhen Li, Hong Li, and Liming Zhang. 2019. Quaternion-based multiscale analysis for feature extraction of hyperspectral images. *IEEE Transactions on Signal Processing* 67, 6 (2019), 1418–1430.
[15] NHS National Services Scotland. 2019. Scottish Cancer Registry. Retrieved from https://www.isdscotland.org/Health-Topics/Cancer/Scottish-Cancer-Registry/. [Online; accessed: 2020-02-06].
[16] Alvin Rajkomar, Jeffrey Dean, and Isaac Kohane. 2019. Machine learning in medicine. *New England Journal of Medicine* 380, 14 (2019), 1347–1358.
[17] Olga Russakovsky, Li-Jia Li, and Fei-Fei Li. 2015. Best of both worlds: Human-machine collaboration for object annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
[18] Aliaksei Sandryhaila and Jose M. F. Moura. 2014. Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure. *IEEE Signal Processing Magazine* 31, 5 (2014), 80–90.
[19] Isabella Seeber, Eva Bittner, Robert O. Briggs, Triparna de Vreede, Gert-Jan de Vreede, Aaron Elkins, Ronald Maier, Alexander B. Merz, Sarah Oeste-Reib, Nils Randrup, Gerhard Schwabe, and Matthias Sollner. 2019. Machines as teammates: A research agenda on AI in team collaboration. *Information & Management* 57, 2 (2019), 103174.
[20] Shan Ouyang and Zheng Bao. 2002. Fast principal component extraction by a weighted information criterion. *IEEE Transactions on Signal Processing* 50, 8 (2002), 1994–2002.
[21] Benjamin Shickel, Patrick James Tighe, Azra Bihorac, and Parisa Rashidi. 2017. Deep EHR: A survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. *IEEE Journal of Biomedical and Health Informatics* 22, 5 (2017), 1589–1604.
[22] Giovanna Tagliabue, Anna Maghini, Sabrina Fabiano, Andrea Tittarelli, Emanuela Frassoldi, Enrica Costa, Silvia Nobile, Tiziana Codazzi, Paolo Crosignani, Roberto Tessandori, and Paolo Contiero. 2006. Consistency and accuracy of diagnostic cancer codes generated by automated registration: comparison with manual registration. *Population Health Metrics* 4, 1 (2006), 10.
[23] The Royal Society. 2018. The impact of artificial intelligence on work. Retrieved from https://royalsociety.org/topics-policy/projects/ai-and-work/. Accessed: 2020-06-01.
[24] Doris Xin, Litian Ma, Jialin Liu, Stephen Macke, Shuchen Song, and Aditya Parameswaran. 2018. Accelerating human-in-the-loop machine learning: Challenges and opportunities. In *Proceedings of the 2nd Workshop on Data Management for End-To-End Machine Learning*. ACM, New York, NY, 4 pages. DOI: https://doi.org/10.1145/3209889.3209897
[25] Yiwei Yang, Eser Kandogan, Yunyao Li, Prithviraj Sen, and Walter Lasecki. 2019. A study on interaction in human-in-the-loop machine learning for text analytics. In *Joint Proceedings of the ACM IUI 2019 Workshops – Explainable Smart Systems*.
[26] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. 2015. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv:1506.03365. Retrieved from https://arxiv.org/abs/1506.03365.
[27] Fabio Massimo Zanzotto. 2019. Viewpoint: Human-in-the-loop artificial intelligence. *Journal of Artificial Intelligence Research* 64, 1 (2019), 243–252.

Received March 2021; accepted October 2021