Deep reinforcement learning for personalized diagnostic decision pathways using electronic health records: A comparative study on anemia and systemic lupus erythematosus

Lillian Muyama, Antoine Neuraz, Adrien Coulet

Artificial Intelligence In Medicine 157 (2024) 102994

Contents lists available at ScienceDirect

Artificial Intelligence In Medicine

journal homepage: www.elsevier.com/locate/artmed

Research paper

Deep Reinforcement Learning for personalized diagnostic decision pathways using Electronic Health Records: A comparative study on anemia and Systemic Lupus Erythematosus

Lillian Muyama [a,b,*], Antoine Neuraz [a,b,c], Adrien Coulet [a,b]

[a] Inria Paris, Paris, 75012, France
[b] Centre de Recherche des Cordeliers, Inserm, Université Paris Cité, Sorbonne Université, Paris, 75006, France
[c] Hôpital Necker, Assistance Publique - Hôpitaux de Paris, Paris, 75015, France

ARTICLE INFO

Keywords:
Diagnosis
Decision support
Deep reinforcement learning
Decision pathways
Anemia
Lupus

ABSTRACT

Background: Clinical diagnoses are typically made by following a series of steps recommended by guidelines that are authored by colleges of experts. Accordingly, guidelines play a crucial role in rationalizing clinical decisions. However, they suffer from limitations, as they are designed to cover the majority of the population and often fail to account for patients with uncommon conditions. Moreover, their updates are long and expensive, making them unsuitable for emerging diseases and new medical practices.

Methods: Inspired by guidelines, we formulate the task of diagnosis as a sequential decision-making problem and study the use of Deep Reinforcement Learning (DRL) algorithms to learn the optimal sequence of actions to perform in order to obtain a correct diagnosis from Electronic Health Records (EHRs), which we name a diagnostic decision pathway. We apply DRL to synthetic yet realistic EHRs and develop two clinical use cases: Anemia diagnosis, where the decision pathways follow a decision tree schema, and Systemic Lupus Erythematosus (SLE) diagnosis, which follows a weighted criteria score. We particularly evaluate the robustness of our approaches to noise and missing data, as these frequently occur in EHRs.

Results: In both use cases, even with imperfect data, our best DRL algorithms exhibit competitive performance compared to traditional classifiers, with the added advantage of progressively generating a pathway to the suggested diagnosis, which can both guide and explain the decision-making process.

Conclusion: DRL offers the opportunity to learn personalized decision pathways for diagnosis. Our two use cases illustrate the advantages of this approach: they generate step-by-step pathways that are explainable, and their performance is competitive when compared to state-of-the-art methods.

1. Introduction

For complex diagnosis decisions, clinicians customarily refer to diagnostic guidelines, which define the successive steps required to reach a diagnosis. These documents, typically established by a panel of experts based on the best available evidence [1], aim to rationalize and standardize clinical decisions. The recommended steps may include information collection, physical observations, laboratory test orders, and other examinations. However, clinical guidelines have several drawbacks. Firstly, they are designed to cover the majority of the population and hence may not be effective for uncommon patients, such as the elderly or those with multiple morbidities. Secondly, their development process is long and expensive, with their updates often occurring after several years [2]. This makes them unsuitable for rapidly evolving practices, such as those associated with new laboratory tests or emerging diseases such as the recent COVID-19 pandemic. Moreover, due to the high cost and time investment required to produce guidelines, their development does not scale to cover the full spectrum of diseases. More versatile and scalable methods are therefore needed to provide insights when clinical guidelines are incomplete or unavailable.

We believe that machine learning approaches, trained on clinical data, can complement diagnostic guidelines. Specifically, we aim to develop approaches that can guide the decision-making process in a step-by-step manner, as described in [3]. This sequence of clinical activities performed leading up to a diagnosis is what we refer to as a diagnostic decision pathway. Such an approach could reduce the number of irrelevant tests, thereby optimizing healthcare costs, but more importantly, could propose more personalized and accurate diagnoses, particularly for patients with uncommon conditions.

* Corresponding author.
E-mail address: lillian.muyama@inria.fr (A. Coulet).

https://doi.org/10.1016/j.artmed.2024.102994
Received 26 March 2024; Received in revised form 19 September 2024; Accepted 26 September 2024
Available online 2 October 2024

The collection of patient-level data in Electronic Health Records (EHRs) offers significant opportunities to improve our understanding of actual clinical practice [4]. EHRs encompass structured, semi-structured, and unstructured data about patients' health such as medications, laboratory test orders and results, diagnoses, and demographic information. Previous studies used EHRs to train machine learning (ML) methods to automatically suggest diagnoses for patients [5–7]. However, these studies typically employ supervised ML methods to predict a unique endpoint, *i.e.*, the diagnosis represented as a class label. We believe that for data-driven approaches to be adopted in clinical practice, it is important for a diagnosis not to be limited to an endpoint, but to be represented as a pathway that follows steps of medical reasoning and decision-making. This approach aligns more closely with how clinicians think and can provide a more comprehensive and practical tool for aiding diagnosis.

In this work, we propose to leverage EHR data to train deep reinforcement learning (DRL) models that can build explainable pathways to support the diagnosis of complex conditions. Our approach focuses on two primary use cases: anemia and Systemic Lupus Erythematosus (SLE), commonly referred to as Lupus. Anemia is a clinical condition characterized by a lower-than-normal amount of healthy red blood cells in the body, which we chose for three reasons: its diagnosis is mainly based on a series of laboratory tests that are available in most EHRs; it is a common diagnosis, implying the availability of a sufficient amount of data to train RL models; and the differential diagnosis of anemia can be complex, making its guidance particularly useful for clinicians. SLE is a chronic autoimmune disease in which the immune system attacks the body's healthy tissues. We selected SLE because its diagnosis is inherently complex and follows a different diagnostic schema compared to anemia. While anemia diagnosis typically follows a decision tree schema [8], SLE diagnosis relies on a weighted criteria score, representing an alternative decision-making approach.

We chose to use reinforcement learning (RL) because it builds models that operate sequentially, moving through various steps (*actions* and *states*) to reach a final objective state, which aligns with our diagnostic goals. In this work, we propose adapting the RL framework to construct individualized pathways in a step-by-step manner, aimed at suggesting diagnostic decisions. For instance, in the anemia use case, a pathway is a sequence of laboratory test requests (actions), whose results are then obtained (states). The process continues by either requesting an additional test or terminating with a diagnosis. We hypothesize that the constructed pathways can complement clinical guidelines, thus aiding practitioners in the decision-making process during diagnosis.

Our main contributions are: *(i)* an adaptation of the RL framework to progressively construct optimal sequences of actions to perform in order to reach a diagnosis and *(ii)* an empirical analysis using synthetic EHRs that illustrates the validity of our approach, identifies the most suitable algorithms, and evaluates their robustness in the presence of varying levels of noise and missing data.

The rest of the paper is structured as follows. Section 2 presents related works. Section 3 details the methods we deploy in this paper, with Section 3.5 explaining the construction of our synthetic datasets. Section 4 presents the results of our empirical analysis, evaluating these methods in various scenarios. Section 5 discusses our findings and their limitations. Finally, Section 6 concludes the study and provides directions for future work.

2. Related work

Previous studies have employed various process mining, machine learning, and statistical methods to extract clinical pathways from medical data. Zhang et al. proposed using Markov Chains to identify clinical pathways from patient visits data [9]. Similarly, Najjar et al. combined clustering with process mining to identify clinical pathways using hospital administrative data [10]. Huang et al. adopted a process mining approach to extract data-driven clinical pathways [11], while Baker et al. used a Markov model to extract meaningful clinical events from patient data [12]. Others have adopted probabilistic models to build treatment pathways [13–15]. Machine learning methods such as Long Short-Term Memory (LSTM) [16], transformer-based neural networks [17] and RL [18–20] have also been used to construct pathways for optimal treatment in several studies. However, while these studies focus on building pathways for *treatment* decisions, our aim is to develop pathways for *diagnosis*.

Likewise, numerous studies have leveraged ML methods for diagnosis. Given the longitudinal nature of EHRs, Recurrent Neural Networks (RNNs) have been a prevalent choice [5,7]. RNNs have also been used to predict future patient outcomes such as in [21]. Furthermore, several studies have made novel efforts to employ ML approaches that not only provide clinical diagnoses but also include explainable elements that enable the interpretation and contextualization of the model's results [22–24]. However, our aim extends beyond diagnosing the patient with the correct condition. We also seek to identify the optimal sequence of features to observe to reach this diagnosis for each patient. These sequences can be seen as elements of the explanation for the decision made. In other words, we aim to construct personalized diagnosis pathways that delineate the steps leading to the diagnosis, thereby explaining the diagnostic process.

Previous works have used RL methods for costly feature acquisition in classification tasks [25,26]. However, these works did not primarily focus on diagnosis. Yu et al. aimed to optimize the financial cost of several clinical processes, including the prediction of Acute Kidney Injury, using RL [27]. While their results demonstrate the success of RL methods in achieving lower costs for laboratory tests compared to other models, the sequence of actions leading to these cost reductions and subsequent diagnoses are not discussed or shown. This omission neglects the relevance and explainable dimension of pathways. Other studies have used RL to diagnose patients by inquiring about the presence of disease-related symptoms from users [28–30]. While our approach similarly formulates the diagnosis process as a sequential decision-making problem and applies DRL techniques, these studies use a symptom self-checking approach. In contrast, we aim to use EHRs because we believe that they offer an interesting alternative as they encompass data routinely collected in clinical practice, including objective and standardized measurements such as laboratory results. This comprehensive data provides a valuable resource for training ML models and obtaining data-driven insights to aid clinical decision-making.

3. Methods

3.1. Decision problem

We consider the clinical diagnosis process as a sequential decision-making problem and formulate it as a Markov Decision Process (MDP) [31], following the RL [32] paradigm. Accordingly, we define an *agent* that interacts with an *environment* to maximize a cumulative reward signal. At each time step t, the agent receives an observation o of the *state* of the environment, takes an *action*, and obtains a *reward*. The goal is to learn a policy (*i.e.*, a function that maps states to actions) that maximizes the cumulative reward. In this study, our agents use observations from synthetic EHRs in a Clinical Data Warehouse (CDW) to reach a diagnosis, which constitutes the final action.

Let D denote a dataset of size $n \times (m+1)$ consisting of n patients, m features, and one diagnosis. Let F represent the set of m feature names, and C the set of possible diagnosis values. An instance D^i in D is a pair (X^i, Y^i) with $X^i = \{x^i_1, \ldots, x^i_j, \ldots, x^i_m\}$ where x^i_j is the value of the feature j for the patient i, m is the total number of features and $Y^i \in C$ is the anemia diagnosis of patient i. Accordingly, our MDP is defined by the quadruple (S, A, T, R) as follows:

- S is the set of states. At each time step, the agent receives an observation o of the state s_t, which is a vector of fixed size m comprising the values of the features that the agent has already queried at time t. Features that have not been queried yet are associated with the value -1. Eq. (1) defines the jth element of o, where F' denotes the set of features that the agent has already queried. A simulation of a step in the anemia environment is presented in Algorithm 1, and an illustration of how the state is updated is shown in Table A.1, both in Appendix A.

$$o_j = \begin{cases} x_j, & \text{if } f_j \in F' \\ -1, & \text{otherwise.} \end{cases} \tag{1}$$

- A is the set of possible actions, which is the union of the set of *feature value acquisition actions*, A_f (or *feature actions* for short), and the set of *diagnostic actions*, A_d. At each time step, the agent takes an action $a_t \in A$. Actions from A_f involve selecting values from the set of features F. Accordingly, a specific value $f_j \in F$ will trigger the action of querying the CDW for the value of this feature. Actions from A_d involve selecting values from the set of possible diagnoses C. At any time step t, if $a_t \in A_d$, the episode is terminated. Also, for *anemia*, if an episode reaches the specified maximum number of steps without reaching a diagnosis, it is terminated.
- T denotes the transition function that specifies the probability of transitioning from a state s_t to a state s_{t+1} given an agent action $a_t \in A$.
- R is the reward function. r_{t+1}, which can be expressed as $R(s_t, a_t)$, is the immediate reward received when an agent in state s_t takes action a_t. For a diagnostic action $a_t \in A_d$, if the diagnosis is correct, the reward is $+1$; otherwise, it is -1.

$$\text{if } a_t \in A_d, R(s_t, a_t) = \begin{cases} 1, & \text{if } a_t = Y^i \\ -1, & \text{otherwise} \end{cases} \tag{2}$$

In the **anemia** use case, for a feature action $a_t \in A_f$, if the feature has already been queried, the agent incurs a penalty of -1, and the episode is terminated. If the feature has not been queried before, the agent receives a reward of 0. Thus, the reward function for the feature actions is formalized as follows:

$$\text{if } a_t \in A_f, R(s_t, a_t) = \begin{cases} -1, & \text{if } a_t \in F' \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

In the **lupus** use case, for a feature action $a_t \in A_f$, if the feature has already been queried, the agent receives a penalty of -1, and the episode is terminated. Otherwise, the agent incurs a penalty determined by the penalty weight associated with the action (See Section 3.5).
The reward function is:

$$\text{if } a_t \in A_f, R(s_t, a_t) = \begin{cases} -1, & \text{if } a_t \in F' \\ \frac{-1}{\lambda c}, & \text{otherwise,} \end{cases} \tag{4}$$

where c is the penalty weight of the feature action and $\lambda \in \mathbb{R}$ is a scaling factor. We decided to include a step penalty in the lupus use case due to its more complex diagnostic process, and the wider range of features that vary significantly in financial cost, level of invasiveness, turnaround time, and other factors. This diversity influenced our reward function design to account for the desirability of selecting different features.

3.2. DQN and its extensions

Q-learning [33] is an RL algorithm that determines the best action to take in a given state based on the expected future reward of taking that action in that particular state. This expected future reward is referred to as the Q-value of that state–action pair, denoted as $Q(s, a)$. At each time step, the agent selects an action according to a policy π, and the goal is to find the optimal policy π^* that maximizes the cumulative reward. During model training, the Q-values are updated using the Bellman Equation as described in Eq. (B.1).

In our study, given the large and continuous state space, we propose using a **Deep Q-Network (DQN)** [34], which employs a neural network to approximate the Q-value function. To improve the stability and performance of the DQN algorithm, several extensions have been developed. Particularly, we explore **Double DQN (DDQN)**, **Dueling DQN** and **Prioritized Experience Replay (PER)** in the subsequent sections of this paper and provide their brief descriptions in Appendix B. These techniques enhance stability, accelerate the learning process, and reduce overestimation bias, thereby improving overall performance.

3.3. State-of-the-Art (SOTA) classifiers

As part of the study, we compared the performance of the DRL models with five traditional supervised learning algorithms that are commonly used for classification tasks, namely Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), Extreme Gradient Boosting (XGBoost), and Feed-Forward Neural Network (FFNN). The DT holds a particular significance in our study for two reasons. First, in the anemia use case, the labels of the synthetic dataset were assigned according to a decision tree, making it likely that the DT approach would perform very well on this data. Second, DT is the only classifier among those considered that is explainable, as the path taken in the DT to classify an instance constitutes a pathway to the diagnosis. The hyperparameters used for these models are described in Appendix C.3 and Appendix D.4 respectively.

3.4. Evaluation approach and implementation

80% of the dataset was used to train the model while the remaining 20% served as the test set. Additionally, 10% of the training dataset (*i.e.*, 8% of the dataset) was used for validation. One aim of this study is to learn diagnosis pathways for anemia and lupus, and to evaluate the robustness of the methods to noise and missing data. Therefore, the validation and test sets used in all the experiments are constant (*i.e.*, with no added noise or missing data).

Ten runs were conducted for the first experiment comparing DRL approaches and five runs for the subsequent experiments on the models' robustness. Each run was performed using the same datasets but with different seeds, which influenced initialization and the models' interactions with the environment.

The metrics used to measure the performance of the models are as follows:

- **Accuracy**: The ratio of episodes that terminated with a correct diagnosis.
- **Mean episode length (MEL)**: The average number of actions performed by the agent per episode. This metric does not apply to the SOTA classifiers that consider all the features.
- **F1 score**: The harmonic mean of the precision and recall scores. We report the one-vs-rest and macro-averaging F1 score.
- **ROC-AUC (Receiver Operating Characteristic - Area Under Curve) score**: It measures the model's ability to distinguish between the different classes. We also used a one-vs-rest approach and macro-averaging.

Additionally, we defined two specific metrics tailored to the **lupus** use case, as described below:

- **Average pathway score (APS)**: The average score of a set of pathways. Formally, the score of a pathway p is computed using Eq. (5) below:

$$s(p) = 1 - \frac{\sum_{i=1}^{|p|-1} w_i}{\sum_{i=1}^{m} w_i} \tag{5}$$

where w_i represents the penalty weight of the ith feature of p, $|p|$ is the length of p, and m is the total number of features.
Eq. (6) computes the average pathway score for the entire model:

$$\bar{s} = \frac{1}{n}\sum_{i=1}^{n} s(p_i) \tag{6}$$

where p_i is the ith pathway generated by the model and n is the total number of pathways generated.

- **wPAHM (weighted Pathway score and Accuracy Harmonic Mean)**: We define wPAHM as the weighted harmonic mean of a model's accuracy and APS. This metric provides a unique value that reflects both the accuracy and APS, taking into account their relative importance in the context of the study. Formally,

$$\text{wPAHM} = \frac{w_a + w_{\bar{s}}}{\frac{w_a}{a} + \frac{w_{\bar{s}}}{\bar{s}}} \tag{7}$$

where w_a and $w_{\bar{s}}$ denote the weights for the accuracy and APS, respectively. In our case, we prioritize the accuracy of the diagnosis over the score of our pathways. We, therefore, assigned weights of 0.9 and 0.1 to the accuracy (w_a) and APS ($w_{\bar{s}}$), respectively.

We implemented our environment using the OpenAI Gym Python library [35]. For model training, we instantiated the environment using the training data. At the beginning of each episode, the environment was reset using a randomly selected instance from this data. To build our agents, we used the stable-baselines package [36]. The hyperparameter values, listed in Table B.1 in Appendix B, were selected based on prior knowledge, existing literature, and experimentation. For the remaining hyperparameters not listed, we used the default values specified in the Stable baselines documentation [37].

Regarding the number of time steps, in the *anemia* use case, we employed a grid-search strategy. By evaluating performance on the validation data, we identified the best time step configuration in the initial run. This number was then consistently applied across the subsequent runs for that particular experiment. In contrast, for the *lupus* use case, all the models were trained using a fixed number of time steps, and checkpoints were saved at regular intervals. The model checkpoint with the best performance, as evaluated using the validation data, was selected. The hyperparameters for the SOTA models were chosen through a grid search strategy. The source code is available at https://github.com/lilly-muyama/Deep_RL_diagnosis_pathways.

3.5. Datasets

To experiment with our method, we constructed two synthetic datasets, one for each use case (*i.e.,* anemia and lupus). Detailed information on the selection of features, their values, the labeling of diagnostic classes, and the addition of various levels of noise and missing data is provided in Appendix C.1 and Appendix D.1.

The main difference in the construction of the two datasets is how synthetic patients are labeled with their diagnoses. For anemia, labels are assigned according to the DT illustrated in Fig. 1. In contrast, the lupus dataset is labeled based on the weighted criteria score system defined in [38]. Here, the entry criterion for diagnosing lupus is the presence of Antinuclear antibodies (ANA) at a titer of $\geq$ 1:80 on HEp-2 cells. The remaining features are divided into several categories, and only the highest criteria weight within a category contributes to the patient's total weighted criteria score. A patient is diagnosed with lupus if their total criteria weight is $\geq$ 10. For example, if a patient is experiencing both delirium (criteria weight=2) and seizure (criteria weight=5), only 5 will be added to the patient's total score since both features belong to the neuropsychiatric category.

Additional details about the use cases, such as the penalty weights for lupus features, and the datasets' characteristics, are presented in Appendix C and Appendix D.

4. Results

4.1. Performance comparison of DQN and its extensions

In the first round of experiments, we trained DQN, Double DQN, Dueling DQN, and Dueling Double DQN on our datasets, and also enabled PER for each of these models. In addition, we trained a Proximal Policy Optimization (PPO) algorithm for comparison. Models with the suffix "-PER" indicate the corresponding DQN variant combined with PER. To evaluate the stability of each model, we conducted ten runs of each experiment using different seeds, and the results are presented in Tables 1 and 2. The training curves for these models are shown in Figs. E.1 and E.2.

In both use cases, Dueling DQN-PER achieved the highest accuracy, while Dueling DDQN-PER exhibited a comparable performance with slightly lower accuracy. Additionally, Dueling DDQN-PER had the lowest standard deviation (SD) in the anemia use case and the second lowest in the lupus use case. Remarkably, it also demonstrated a significantly shorter mean episode length than the Dueling DQN-PER in the lupus use case. For these reasons, we focused on these two DQN models for the remainder of the experiments. It should be noted that there are a few minor methodological differences between the two use cases, as detailed in Section 3.

4.2. Performance comparison with SOTA classifiers

The results of these experiments are shown in Tables 3 and 4. For the *anemia* use case, since the diagnosis labels of the synthetic dataset were assigned following the decision tree in Fig. 1, the tree-based agent, which follows the same tree achieved a perfect score. This tree was not applicable to the *lupus* use case. Additionally, we built a baseline agent that selects an action randomly at each time step for comparison. Since the SOTA classifiers use a constant set of features to make diagnoses, they do not have a mean episode length. The tree-based algorithms and the FFNN outperformed the DQN models, while SVM showed comparable performance in the *lupus* use case but lower performance in the *anemia* use case.

4.3. Performance comparison using different metrics

For the *lupus* use case, as accuracy is not the only desirable metric, we also selected the models with the best performance according to an alternative metric, *i.e.* the wPAHM as defined in Eq. (7). The comparison results between the models with the best accuracy and wPAHM scores are shown in Table 5.

4.4. Results with different values of λ

In the reward function for the feature value acquisition actions in the *lupus* use case (shown in Eq. (4)), λ acts as a scaling factor. We conducted experiments with various λ values, and the mean accuracy and wPAHM scores are illustrated in Figs. 2(a) and 2(b) respectively. As shown, increasing the λ value results in a higher accuracy score but a lower wPAHM score.

Additionally, Fig. 3 illustrates the trade-off between the accuracy and the average pathway score for different λ values.

4.5. Varying levels of missing data and noise

Figs. 4(a) and 5(a) depict the mean accuracy as the level of missing data in the *anemia* and *lupus* training datasets changes, respectively. In particular, for lupus, missing values were imputed before model training using the K-Nearest Neighbor (KNN) imputer [40] with the number of neighbors set to one. All models exhibited a decline in performance, with SVM demonstrating the fastest deterioration in both

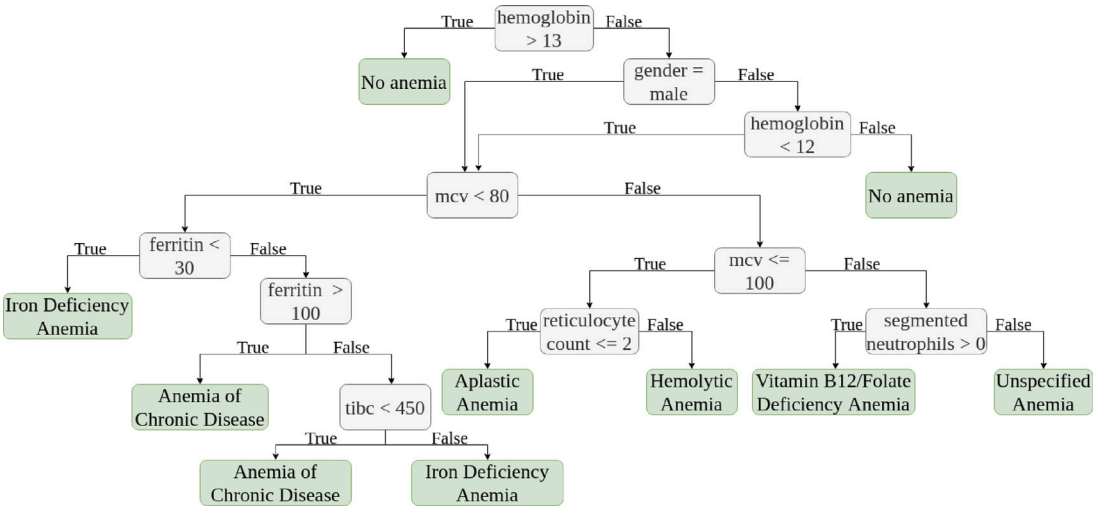

Fig. 1. The decision tree used to label the anemia dataset.
Source: Adapted from [8,39].

Table 1
Performance of RL models in the *anemia* use case with metrics presented as the average and standard deviation over 10 runs.

Model	Accuracy	MEL	F1	ROC-AUC
PPO	61.51 ± 17.11	3.34 ± 0.86	55.80 ± 20.11	78.17 ± 9.70
DQN	89.31 ± 14.68	4.27 ± 0.45	88.73 ± 15.95	93.90 ± 8.28
DDQN	92.79 ± 5.37	4.66 ± 0.31	92.18 ± 6.25	95.74 ± 3.28
Dueling DQN	92.46 ± 5.31	4.70 ± 0.24	91.72 ± 6.57	95.65 ± 3.27
Dueling DDQN	77.15 ± 31.99	4.44 ± 1.02	74.36 ± 34.98	86.84 ± 18.06
DQN-PER	93.71 ± 5.51	4.67 ± 0.29	93.62 ± 5.20	96.41 ± 3.03
DDQN-PER	95.57 ± 2.01	4.92 ± 0.36	95.34 ± 2.12	97.46 ± 1.16
Dueling DQN-PER	**96.64 ± 1.46**	**4.59 ± 0.20**	**96.50 ± 1.52**	**98.11 ± 0.82**
Dueling DDQN-PER	96.33 ± 1.32	4.82 ± 0.45	96.17 ± 1.35	97.86 ± 0.81

Table 2
Performance of RL Models in the *lupus* use case with metrics presented as the average and standard deviation over 10 runs.

Model	Accuracy	MEL	F1	ROC-AUC
PPO	88.06 ± 7.19	3.81 ± 1.63	82.20 ± 4.93	82.73 ± 4.41
DQN	88.57 ± 4.89	14.82 ± 3.17	88.60 ± 4.85	88.62 ± 4.83
DDQN	90.97 ± 1.56	11.42 ± 2.68	90.97 ± 1.56	90.97 ± 1.56
Dueling DQN	97.08 ± 0.72	22.57 ± 1.05	97.08 ± 0.72	97.08 ± 0.72
Dueling DDQN	94.20 ± 2.05	16.28 ± 3.75	94.21 ± 2.06	94.21 ± 2.06
DQN-PER	94.17 ± 5.18	20.46 ± 2.52	94.19 ± 5.16	94.19 ± 5.15
DDQN-PER	94.26 ± 2.79	13.27 ± 2.64	94.27 ± 2.78	94.27 ± 2.78
Dueling DQN-PER	**98.81 ± 0.32**	**22.09 ± 0.85**	**98.82 ± 0.32**	**98.82 ± 0.32**
Dueling DDQN-PER	98.39 ± 0.41	14.40 ± 2.49	98.39 ± 0.41	98.39 ± 0.41

Table 3
Performance comparison of the DQN and the state-of-the-art classifiers for the *anemia* use case. The tree-based agent achieves a perfect score because it selects actions following the decision tree used to label the dataset. The mean episode length is not applicable to models that do not generate pathways.

Model	Accuracy	MEL	F1	ROC-AUC
Random Agent	12.34 ± 0.33	1.53 ± 0.01	12.34 ± 0.33	50.00 ± 0.20
Tree-based Agent	**100 ± 0.00**	**3.98 ± 0.00**	**100 ± 0.00**	**100 ± 0.00**
Decision Tree	99.96 ± 0.00	N/A	99.96 ± 0.00	99.98 ± 0.00
Random Forest	99.90 ± 0.01	N/A	99.90 ± 0.01	99.95 ± 0.01
XGBoost	99.99 ± 0.00	N/A	99.99 ± 0.00	99.99 ± 0.00
FFNN	97.97 ± 0.27	N/A	97.91 ± 0.28	98.81 ± 0.16
SVM	94.89 ± 0.00	N/A	94.36 ± 0.00	96.80 ± 0.00
Dueling DQN-PER	96.64 ± 1.46	4.59 ± 0.20	96.50 ± 1.52	98.11 ± 0.82
Dueling DDQN-PER	96.33 ± 1.32	4.82 ± 0.45	96.17 ± 1.35	97.86 ± 0.81

Table 4
Performance of the DQN and state-of-the-art classifiers for the *lupus* use case. The mean episode length is not applicable to models that do not generate pathways.

Model	Accuracy	MEL	F1	ROC-AUC
Random Agent	17.69 ± 0.38	4.75 ± 0.02	26.15 ± 0.51	50.04 ± 0.26
Decision Tree	99.42 ± 0.01	N/A	99.42 ± 0.01	99.42 ± 0.01
Random Forest	99.38 ± 0.02	N/A	99.38 ± 0.02	99.37 ± 0.02
XGBoost	99.81 ± 0.00	N/A	99.81 ± 0.00	99.81 ± 0.00
FFNN	**99.92 ± 0.03**	**N/A**	**99.92 ± 0.03**	**99.92 ± 0.03**
SVM	99.54 ± 0.00	N/A	99.54 ± 0.00	99.94 ± 0.00
Dueling DQN-PER	98.81 ± 0.32	22.09 ± 0.85	98.82 ± 0.32	98.82 ± 0.32
Dueling DDQN-PER	98.39 ± 0.41	14.40 ± 2.49	98.39 ± 0.41	98.39 ± 0.41

Table 5
The mean performance of the best-performing DQN models for the *lupus* use case, based on the selected goal metric (accuracy or wPAHM), with λ=9 in the reward function.

Model	Goal metric	Acc.	MEL	F1	ROC-AUC	APS	wPAHM
Dueling DQN-PER	Accuracy	**98.81**	22.09	**98.82**	**98.82**	10.24	50.98
	wPAHM	95.83	17.30	95.82	95.83	30.06	77.93
Dueling DDQN-PER	Accuracy	98.39	14.40	98.39	98.39	43.15	86.18
	wPAHM	96.41	9.56	96.42	96.42	**63.64**	**91.59**

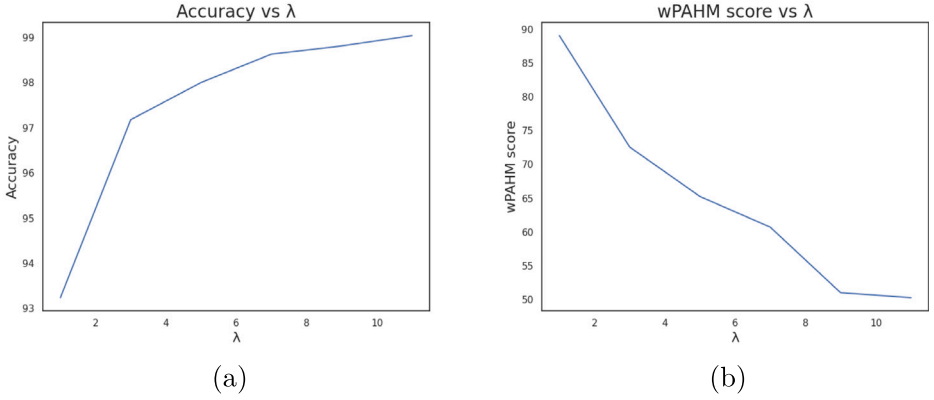

Fig. 2. Graphs showing the impact of the λ value in the reward function on (a) accuracy and (b) wPAHM scores for the dueling DQN-PER model.

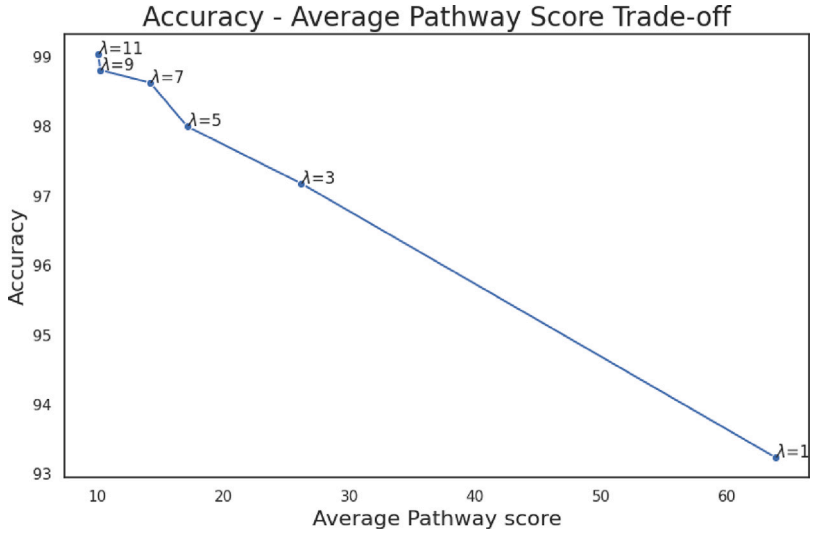

Fig. 3. The accuracy-average pathway score tradeoff for different λ values in the reward function, using results from the dueling DQN-PER model.

use cases. The DT and FFNN showed the best performance for *anemia* and *lupus*, respectively.

Figs. 4(b) and 5(b) show the mean accuracy as the level of noise in both training sets increases. For anemia, SVM showed the sharpest decline, whereas the tree-based models (DT and RF) maintained the best performance. On the other hand, for lupus, the FFNN had the best performance, with the DQN models outperforming the other models.

Figs. 4(c) and 5(c) depict the mean accuracy as the level of missing data increases, with a fixed level of noise of 0.2. Again, missing values were imputed as described earlier for the *lupus* use case. The DQN

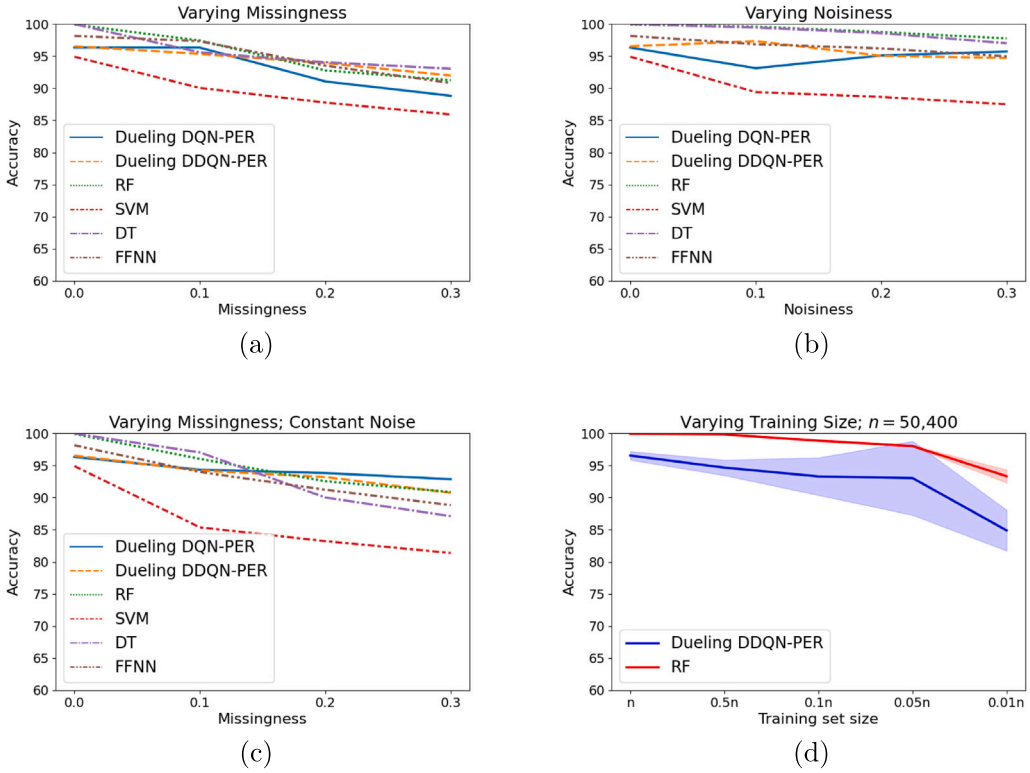

Fig. 4. Accuracy of approaches under varying conditions for the *anemia* dataset. (a) Mean accuracy of models at different missingness levels; (b) Mean accuracy of models at different noise levels; (c) Mean accuracy of models at a constant noise level (0.2) with varying missingness levels; (d) Mean accuracy and 95% confidence interval of models based on the training set size. The *y*-axis of all plots starts at 60 to highlight performance differences more clearly.

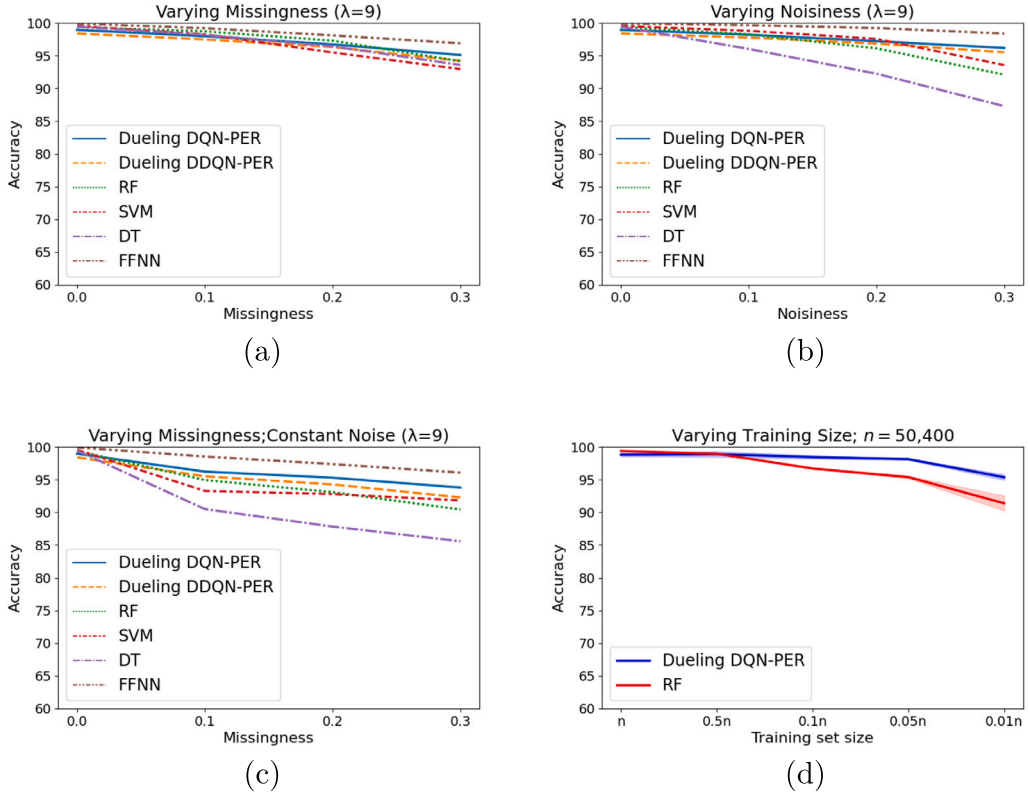

Fig. 5. Accuracy of approaches under varying conditions for the *lupus* dataset. The graphs depict the mean accuracy of the models at (a) varying missingness levels; (b) varying noise levels; (c) varying missingness levels at a constant noise level (0.2). (d) displays the mean accuracy and 95% confidence interval of models based on training set size. The *y*-axis of all plots starts at 60 to highlight performance differences more clearly.

models exhibited a more consistent performance than all the other models for anemia, while for lupus, they performed slightly worse than the FFNN but better than the other models. In the *lupus* use case, the tree-based models (DT and RF) had the worst performance, while for the *anemia* use case, SVM had the worst performance followed by the DT.

4.6. Varying the size of training sets

To determine whether this method can be applied to the diagnosis of conditions where data may be scarce, such as with rare diseases, we conducted experiments with training sets of different sizes. For each size, a fraction of the training set was randomly selected and removed, leaving the remaining data to form the new set. Each experiment was repeated five times for each training set size, with a different subset removed each time. We used the Dueling DDQN-PER model for these experiments, keeping the number of time steps consistent across runs. Figs. 4(d) and 5(d) show the mean accuracy for each training set size in the two use cases, with the shaded region representing the 95% confidence interval. For comparison, the RF classifier is also depicted in the figures due to its stability and high performance. As shown, the DQN model was able to learn the clinical pathways and diagnose the patients. However, for all training set sizes, its performance was lower than that of the RF in the *anemia* use case, but higher in the *lupus* use case. Additionally, as the training set size decreased, the model's performance became less stable.

5. Discussion

Our study illustrates that DRL models trained on EHRs are a promising solution for suggesting optimal, individualized, step-by-step pathways to diagnosis. When comparing our two best DQN models to state-of-the-art classifiers, while the latter performed extremely well on perfect data, the two DQN models performed comparably and sometimes better on imperfect data in both use cases (*i.e.*, anemia and lupus), demonstrating their robustness. Furthermore, when trained on datasets of varying sizes, the DQN Models showed their capacity to learn diagnostic decision pathways with decent performance, even with reasonably small training datasets (*e.g.*, about 600 patients).

5.1. Generated pathways

Beyond quantitative evaluation, the pathways constructed by DQN models offer two advantages. Firstly, the fact that they consist of progressive observations leading to a diagnostic decision makes them explainable. One may understand why a particular diagnosis is reached or why a new exam is ordered by looking at the sequence of selected features and their associated values, akin to clinician reasoning. We note that the DT algorithms are also intrinsically explainable, but our results indicate that they are less robust to imperfect data.

Secondly, the set of pathways generated from a test set can be aggregated to form a data structure somewhat similar to diagnostic guidelines. Fig. 6 provides an illustration of such an aggregation in the form of a Sankey diagram, showcasing the pathways learned by the model for the diagnosis of *Anemia of Chronic Disease (ACD)* and *Aplastic anemia*, colored *coral* and *blue* respectively. For instance, the most common pathway followed for patients diagnosed with Aplastic anemia is "Hemoglobin –> Gender –> RBC –> Reticulocyte count –> Aplastic anemia" depicted in blue. Additionally, Fig. C.3 presents a side-by-side illustration of some of the learned pathways, and a branch of the decision tree used to label the dataset.

For the lupus use case, Fig. 7 illustrates the possible aggregations, highlighting the combination of the three most common pathways for the *No lupus* class, each in a different color. Here, the most common pathway is "ANA –> No lupus" shown in green. These pathways were generated by one of our experiments: the dueling DDQN model chosen based on the best wPAHM score, with λ set to 9. More pathways are shown in Figures D.2, D.3 and D.4 in Appendix D.

Other pathway aggregations can be viewed and interacted with at https://lilly-muyama.github.io/. These provide visualization tools and additional information about the support of nodes and edges, as well as descriptive statistics (*e.g.*, mean, maximum, minimum values) of the various features. Algorithm 2 in Appendix A describes the process of generating pathways using the test dataset.

5.2. DQN performance

Both Dueling DQN and Double DQN improve the accuracy and stability of the standard DQN, as observed in Tables 1 and 2. Additionally, for each of the four algorithms discussed above, combining them with Prioritized Experience Replay resulted in better performance in terms of both the mean and SD of the accuracy. This is explained by the fact that in standard DQN, Double DQN, and Dueling DQN, the experiences are sampled uniformly from the replay buffer during training. However, PER prioritizes experiences so that those with a higher priority are sampled more frequently. In a task such as ours, this approach favors transitions with non-zero rewards, which include the diagnosis actions, on which accuracy is primarily based. By sampling these actions more frequently, the model learns about them more efficiently and effectively.

5.3. Quality of diagnoses

For the **anemia** use case, given that we built our dataset using a decision tree, it was expected that the tree-based classifiers would perform extremely well. Indeed, the performance of the two DQN Models was slightly lower than those of the tree-based methods and the neural network, while SVM exhibited a slightly lower performance.

Table 6 displays the classification report of one particular experiment with the Dueling DQN-PER model, with an accuracy of 97.19%, which is the closest to the mean accuracy. The report indicates that most of the anemia classes showed a decent performance. However, while the *No anemia* class had the best recall with a perfect score of 1, it also had the lowest precision at 0.92. This suggests that several instances were incorrectly diagnosed as *No anemia* despite their laboratory test results indicating otherwise. Upon further inspection, we noted that all the misdiagnosed instances had values near the threshold value. Specifically for the *No anemia* misclassifications, the hemoglobin levels were close to the 13 and 12 g/dL thresholds for men and women, respectively (see Fig. 1). In particular, men had a mean hemoglobin of 12.98 g/dL (SD=0.01), while for women, it was 11.93 g/dL (SD=0.04). However, it should be noted that from a clinical point of view, hemoglobin levels just below the normal threshold usually do not result in an anemia diagnosis suggesting that our approach may produce fewer false negatives compared to a strict decision tree guideline. Similarly, for the *Hemolytic anemia* class, which had the lowest recall score at 0.94 (along with *Aplastic anemia*), 35 of its instances were misdiagnosed as *Vitamin B12/Folate deficiency anemia*. These instances had an average Mean Corpuscular Volume (MCV) of 99.77 fL (SD=0.17), close to the threshold of 100. Another 11 instances that were misdiagnosed as *Anemia of chronic disease* had a mean MCV of 80.20 (SD=0.15), close to the threshold of 80. The remaining 30 instances received an *Inconclusive diagnosis* because their values were near the thresholds and they had missing values for other features in the pathway.

In the **lupus** use case, a positive SLE diagnosis requires the patient to have a weighted criteria score of at least 10 (See [38] for details about how this total is calculated). However, as reflected by the performance metrics, there were a few misdiagnosed episodes. For instance, the model whose pathways are presented in Figs. 7 and D.2, had an accuracy of 96.57 and a wPAHM score of 91.65. Further analysis revealed that for this particular model, 130 *Lupus* cases were misdiagnosed as *No*

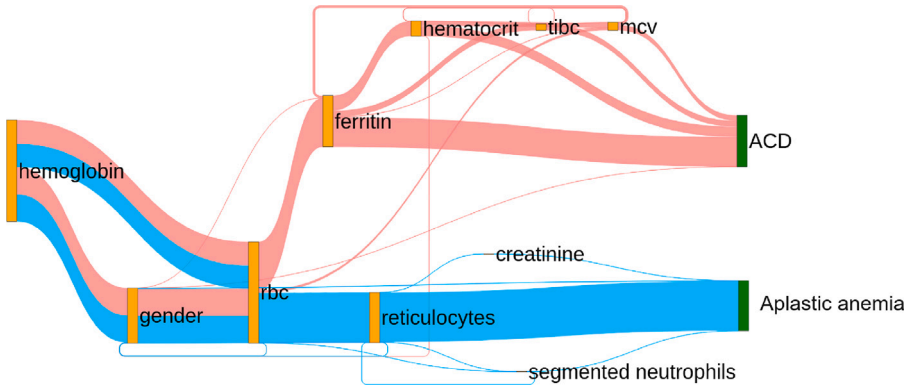

Fig. 6. Diagnostic decision pathways learned by the agent for *ACD (Anemia of chronic disease)* and *Aplastic anemia*.

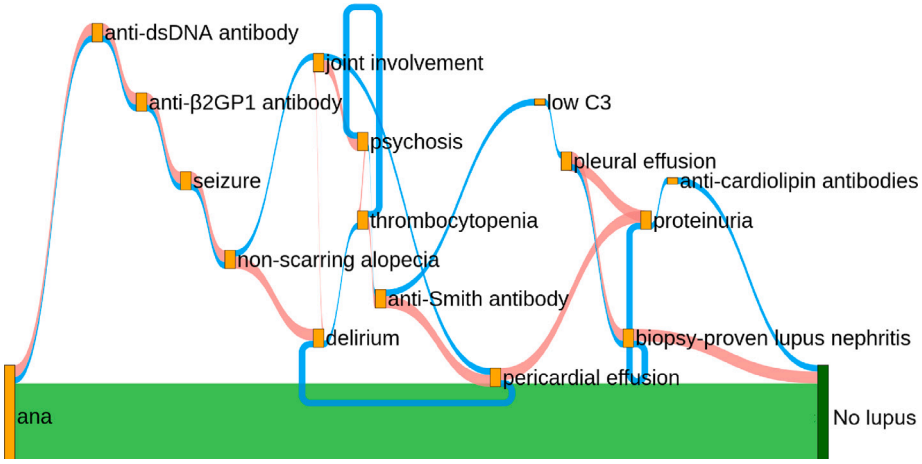

Fig. 7. The three most common diagnostic decision pathways for the *No lupus* class generated by one of the models achieving the best wPAHM score. The pathways were generated by the model with the wPAHM score closest to the mean wPAHM of the ten runs.

Table 6
Classification report detailing the performance of the Dueling DQN-PER model for each anemia class.

Class name (support)	Precision	Recall	F1-Score
Inconclusive diagnosis (1344)	0.94	0.97	0.95
Aplastic anemia (1806)	1.00	0.94	0.97
Hemolytic anemia (1805)	1.00	0.94	0.97
Iron deficiency anemia (1679)	0.98	0.98	0.98
Anemia of chronic disease (1772)	0.99	0.97	0.98
Unspecified anemia (1793)	1.00	0.98	0.99
Vitamin B12/Folate defic. anemia (1801)	0.96	0.98	0.97
No anemia (2000)	0.92	1.00	0.96

lupus because the pathways were terminated prematurely. This caused some positive features to be missed, which would have resulted in a weighted criteria score above 10 if they had been selected. As an example, the most common pathway in these misdiagnosed episodes had a length of 15. Although these features are all key (the pathway still has an accuracy of 97.73%), there are a few cases where the omitted nine features had values that would have altered the diagnosis.

Moreover, for the patients misdiagnosed as *Lupus* instead of *No lupus*, the model made the diagnosis prematurely when the weighted criteria score was near the threshold. For example, for this particular model, the most common pathway for these misdiagnosed episodes was "ANA=1 –> Anti-dsDNA antibody=1 –> Lupus". The weighted criteria score of this pathway is less than 10 and yet a Lupus diagnosis was given. While the presence of anti-dsDNA antibody is highly weighted, it is insufficient for a lupus diagnosis on its own according to [38]. Additionally, the second most common pathway for these misdiagnosed episodes had a length of 23, missing only two features, one of which (renal biopsy-proven lupus nephritis) had the lowest penalty weight, and the other (acute pericarditis) had the lowest prevalence in the dataset.

It is important to note that the above error analysis pertains to a model based on the best wPAHM score. The model with the best accuracy score had longer pathways and fewer misdiagnosed episodes. Some of its pathways are shown in Figs. D.3 and D.4 in Appendix D.

5.4. Quality of lupus pathways

When the value of λ increases, the penalty for selecting the various actions decreases, leading to longer pathways. While this results in

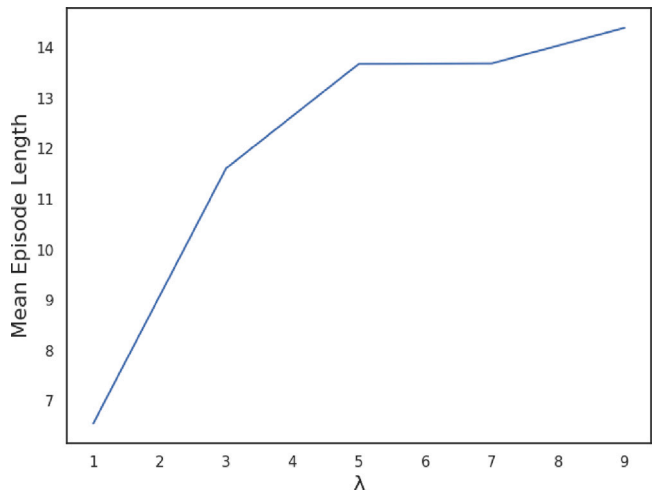

Fig. 8. Mean episode length *versus* λ for the Dueling DDQN-PER model.

higher diagnosis accuracy because the agent has more information to make a diagnosis, it also leads to a reduction in the quality of the diagnosis pathways, since unnecessary actions may be taken. Conversely, if λ is too low, pathways become shorter resulting in a diagnosis being made before all the necessary actions are completed. Fig. 8 shows the effect of the λ value on the mean episode length for the dueling DDQN-PER model.

Additionally, when comparing the performance of the Dueling DDQN-PER models based on a single goal metric, such as accuracy or wPAHM (as shown in Table 5), we observe that optimizing for accuracy results in a lower average pathway score, a lower wPAHM score, and a longer mean episode length, indicating longer pathways. In contrast, when optimizing for wPAHM, we notice a lower accuracy but a higher average pathway score and shorter pathways.

5.5. DQN model robustness

The DQN models show robustness to both noise and missing data, maintaining consistent performance across various levels of increasing noise and missingness in both use cases. This robustness can be attributed to the DQN's reliance on neural networks, which have shown robustness to noise in many scenarios [41]. This is because adding noise to the training input can act as a form of regularization by preventing overfitting and leading to better generalization of the neural network. Likewise, neural networks have demonstrated their capability to handle missing data as they can learn the underlying representation of the data even with missing values.

For instance, in the anemia use case, introducing noise and missing data led to decreased performance in the *Inconclusive diagnosis* and *Iron deficiency anemia* classes (results not shown). This is explained by the fact that some of the noisy and incomplete instances are being diagnosed with *Inconclusive diagnosis* instead of their anemia class, which is to be expected. In the real world, clinicians may need to use their judgment in such cases. Second, misclassifications in the *Iron Deficiency Anemia* class were primarily instances of *Anemia of Chronic Disease* incorrectly classified as *Iron Deficiency Anemia*. These two classes are difficult to distinguish, as they share the same branch in the decision tree (see Fig. 1).

We also note that in the *lupus* use case, the Feed-Forward neural network had the best performance across the various scenarios, followed closely by the DQN models, which outperformed all the tree-based models and SVM in the scenarios with imperfect data. This can be attributed to the fact that the anemia dataset was labeled using a decision tree hence the strong performance of the tree-based models. However, since the lupus dataset was not labeled this way, the neural network-based models performed better for that use case.

We also observe that while the DT performs well on perfect data in both use cases, its performance significantly decreases with the presence of noise and missing data, becoming lower than that of the other models. This can be explained by the decision tree's sensitivity to noise and missing data. In addition, decision trees are prone to overfitting, such that a slight change in the training dataset may lead to a different tree altogether. In contrast, the Random Forest model builds multiple decision trees during its training process and is therefore less likely to overfit than the DT, making it more robust to noise and missing data.

Interestingly, using the same hyperparameters across experiments, the DQN model maintained consistent performance as the training set size gradually decreased. Since the number of time steps was the same across experiments, a smaller dataset meant that the same experiences were sampled more frequently. However, a reduced dataset size led to a loss of information, resulting in lower accuracy and more varied results, as shown in Figs. 4(d) and 5(d). In the anemia use case, RF performed better than the DQN Model, while the DQN model performed better than RF in the lupus use case. Again, this can be attributed to the different way diagnoses are made (*i.e.*, decision tree- *vs.* weighted criteria-based approach) in the two use cases.

Experimenting with two use cases that have different diagnosis processes, *i.e.* a decision tree and a weighted criteria system, and achieving good performance with both suggests that our approach can generalize to other diagnoses that follow a similar diagnostic process to anemia or lupus.

5.6. Differences in use cases

For exploration and experimentation purposes, we handled missing data differently in the two use cases. In the lupus use case, we applied KNN imputation, whereas for anemia, this was considered unnecessary due to the presence of the *inconclusive diagnosis* class. Both methods ensured a diagnosis was given by the model even in the presence of missing data, and we achieved good results with each, thus making them both relevant to this work.

Additionally, the determination of the number of time steps was approached differently in our two use cases. In the anemia use case, lacking prior knowledge of the ideal number of time steps, we employed a grid search strategy to systematically explore various configurations. The optimal number of time steps was identified based on performance evaluation using validation data. In contrast, in the lupus use case, we use a fixed number of time steps for all the experiments, while saving checkpoints of the model at regular intervals, enabling the selection of the best model based on validation performance. This dual approach allowed us to effectively evaluate and optimize our models in both use cases.

Regarding the reward function, we chose not to include a step penalty in the anemia use case. Despite this, the model successfully learned the relevant features to select for each anemia class. However, the diagnostic process for lupus is more complex, so it was necessary to add a penalty for each step taken by the agent to avoid overly long pathways. Additionally, we applied penalty weights and additional performance metrics (average pathway score, wPAHM score) only to the lupus use case due to the greater number and diversity of lupus features. All the anemia features are laboratory test results derived from a blood sample, while lupus features include physical symptoms such as fever, invasive procedures such as a renal biopsy, and laboratory tests such as low C3 levels.

Finally, we added a maximum length constraint solely to the anemia use case. This decision was case-dependent. Anemia diagnosis follows a decision tree, so a maximum length could be interpreted as the maximum depth of the tree. Also, the anemia dataset included extra features not related to anemia diagnosis, meaning a pathway with all the

features included was never necessary to make a diagnosis. In contrast, for lupus, all the features in the dataset are relevant to the diagnosis, and in some cases, the agent may need to select all the features in the dataset before reaching a diagnosis. That said, experiments without the maximum length constraint for the anemia use case showed that the models still learned the pathways with similar lengths. Similarly, lupus experiments with the maximum length set to the number of features plus one (for the diagnosis) exhibited comparable performance to those without the maximum length constraint whose results are shown in this work.

It is therefore important to note that our approach was tailored to each use case based on empirical results to obtain optimal performance. Therefore, similar adjustments may be necessary for other use cases to ensure effectiveness.

5.7. Limitations

One significant limitation of this study is the absence of experiments on a real-world dataset to evaluate our method. While the results from the synthetic datasets are very encouraging, and we have endeavored to reproduce the imperfections of EHR data in our study, it is paramount that we test our method on real-world data. Additionally, the features that constitute the pathways need to be present in the EHRs, which is not always the case. Furthermore, the synthetic datasets used in this study are not longitudinal and therefore do not reflect the actual sequence of clinical events performed for patients. This initial study with anemia and lupus aimed to show the viability of the proposed method. Real-world EHR datasets often have a temporal dimension which will provide more insights into the progression of the pathways. With the availability of such data, we will be able to compare our approach with models capable of handling sequential data, such as Recurrent Neural Networks.

Moreover, training the DRL agents took a significantly longer time and used considerably more computing resources than the SOTA classifiers as shown in Tables C.3 and D.6 in the Appendix. Also, the DQN model has many hyperparameters that would need to be further optimized. However, it should be noted that after model training, generating a diagnosis pathway for a test instance is straightforward since the policy has already been learned by the model (see Tables C.3 and D.6in the Appendix).

Furthermore, there were slight differences in the methodology and performance evaluation strategies used in the two use cases. While we considered it important to illustrate the results with these differences for research purposes, it should be noted that they may account for some of the minor differences in performance between the two use cases. Particularly, in the anemia use case, the method used to determine the number of time steps may have introduced more variability, potentially leading to decreased performance in those specific results.

6. Conclusion

In this work, we adapted the RL framework to offer step-by-step guidance for clinical diagnosis using synthetic EHRs. We particularly demonstrated the capacity of DRL to produce accurate diagnoses in the context of imperfect data (*i.e.*, with noise and missing values) and variable dataset sizes. Additionally, we showed that DRL can generate explainable diagnoses through the creation of decision pathways.

Our approach with DQN is particularly suitable as it achieves performance comparable to the SOTA classifiers, with the added advantage of constructing personalized pathways for each patient. These pathways have the potential to guide the diagnosis of new patients or to be aggregated to summarize possible pathways for uncommon patient populations that might not be covered by existing guidelines. This approach could thus suggest enhancements to clinical guidelines, learned from practice data. In future work, we aim to evaluate our approach on real-world EHRs and extend the types of considered inputs to multimodal data. Additionally, we will explore the use of Large Language Model-based methods to further expand our work.

CRediT authorship contribution statement

Lillian Muyama: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Data curation. **Antoine Neuraz:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization. **Adrien Coulet:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the Inria CORDI-S Ph.D. program, and benefited from a government grant managed by the Agence Nationale de la Recherche under the France 2030 program, reference ANR-22-PESN-0007 ShareFAIR, and ANR-22-PESN-0008 NEUROVASC.

Appendix A. The environment

We implemented our environment using the OpenAI Gym Python library. During training, the environment is reset using a random instance from the dataset at the beginning of each episode. The state of the environment is initialized as a vector of size 17, filled with −1s, corresponding to the set of feature actions, A_f described in Section 3.1.

Algorithm 1 provides a simulation of a step by an agent in the environment for the *anemia* use case. In this algorithm, i represents the training instance used to reset the environment, s_t is the state of the environment, Y^i is the diagnosis of i, *action* is the action selected by the agent, r_t is the reward, and *done* indicates whether an episode is completed. Table A.1 illustrates how the state of the environment is updated when a *feature action* is selected by the agent.

Algorithm 1 Environment step for episode i in state s_t

function STEP(*action*)
  $epLen = epLen + 1$ ▷ Increase episode length
  if $action \in A_d$ **then**
    if $action == Y_i$ **then**
      $r_t = 1$ ▷ Reward if diagnosis is correct
    else
      $r_t = -1$ ▷ Penalty if diagnosis is wrong
    done = True ▷ Terminate the episode
  else if $epLen == maxEpLen$ **then**
    $r_t = -1$ ▷ Penalty if max length is reached
    done = True ▷ Terminate the episode
  else if $action \in F'$ **then**
    $r_t = -1$ ▷ Penalty if action has already been performed
    done = True ▷ Terminate the episode
  else
    $r_t = 0$
    done = False
    $F'.add(action)$ ▷ Add action to list of selected features
    value = $fetchValue(action)$ ▷ Fetch feature value from CDW
    $s_t[action]$ = value ▷ Update the state
  return s_t, r_t, done , F'

Algorithm 2 shows the process of generating pathways for patients in the test set using the trained model.

Table A.1
Illustration of how the state is updated when a *feature action* is selected. The state is represented as a vector of fixed size 17, and the possible feature actions range from a_f^0 to a_f^{16}.

Step	State	Action	Value	Updated State
0	[−1, −1, −1, …., −1, −1]	–	–	[−1, −1, −1, …, −1, −1]
1	[−1, −1, −1, …., −1, −1]	a_f^0	0.2	[0.2, −1, −1, …, −1, −1]
2	[0.2, −1, −1, …., −1, −1]	a_f^{15}	72	[0.2, −1, −1, …, 72, −1]

Algorithm 2 Generating pathways for test set data

Input: Test dataset D_{test}; Trained model M
Output: List of Pathways P
Instantiate empty pathway list: P
Instantiate test environment: env = Env(D_{test})
for all $i \in D_{test}$ **do**
  Instantiate empty pathway p_i
  Reset test environment using test instance: $obs = env$.reset(i)
  done = False
  while not done **do**
    $action = M$.selectAction(obs)
    obs, rew, done = env.STEP($action$)
    $p_i = p_i$.add($action$)
  $P = P$.add(p_i)

Appendix B. Deep-Q network

B.1. DQN and its extensions

Q-learning [33] is an RL algorithm that determines the best action to take in a given state based on the expected future reward for that action in that state. This expected future reward is known as the Q-value of that state–action pair, denoted as $Q(s, a)$. At each time step, the agent selects an action according to a policy π, with the goal of finding the optimal policy π^* that maximizes the cumulative reward. During model training, the Q-values are updated using the Bellman Equation as follows:

$$Q_{t+1}(s_t, a_t) \leftarrow Q_t(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q_t(s_{t+1}, a) - Q_t(s_t, a_t)] \quad \text{(B.1)}$$

where α is the learning rate, r_{t+1} is the reward received after taking action a_t in state s_t, and γ is the discount factor that determines the importance of future reward relative to immediate reward.

In our use case, due to the large state space, we propose using a **Deep Q-Network** [34] which employs a neural network to approximate the Q-value function. A DQN consists of two networks with similar architecture: the policy network, which interacts with the environment and learns the optimal policy, and the target network, which is used to define the target Q-value. At each time step, the policy network takes a state s_t as its input and outputs the Q-values for taking the different actions in that state. The DQN algorithm uses an ϵ-greedy strategy, where a random action is selected with probability ϵ (exploration), and the action with the highest estimated Q-value with probability 1-ϵ (exploitation). The weights of the target network are frozen and updated at a specified interval by copying the weights of the policy network. The DQN algorithm learns by minimizing the loss function shown in Eq. (B.2), where θ and θ^- represent the weights of the policy and target networks, respectively. Additionally, at each time step, a record of the model's interaction with the environment (known as an experience) is stored in a memory buffer in the form $(s_t, a_t, r_{t+1}, s_{t+1})$.

$$L(\theta) = \mathbb{E}[(r_{t+1} + \gamma \max_a Q(s_{t+1}, a, \theta^-) - Q(s_t, a_t, \theta))^2] \quad \text{(B.2)}$$

To improve DQN stability and performance, several extensions of the DQN algorithm have been developed and are utilized in this paper. We briefly describe these extensions below:

Double DQN [42]: In Double DQN, the action is selected using the policy network, while the target network estimates the value of that action. This is in contrast to the standard DQN where the same network is used for both tasks. The loss function is thus modified as follows:

$$L(\theta) = \mathbb{E}[(r_{t+1} + \gamma\ Q(s_{t+1}, \arg\max_a Q(s_{t+1}, a, \theta), \theta^-) - Q(s_t, a_t, \theta))^2] \quad \text{(B.3)}$$

Dueling DQN [43]: In Dueling DQN, the Q-value function is split into two parts: a value function $V(s)$ that provides the value for being in a given state, and an advantage function $A(s, a)$ that gives the advantage of taking action a in state s, as compared to other actions. These two functions are then combined to compute the Q values as shown in Eq. (B.4).

$$Q(s, a) = V(s) + (A(s, a) - \frac{1}{|\mathcal{A}|}\sum_a A(s, a)) \quad \text{(B.4)}$$

where $|\mathcal{A}|$ denotes the total number of possible actions.

Prioritized Experience Replay [44]: In PER, each experience stored in the replay buffer is assigned a priority based on its estimated importance. Experiences with higher priorities are sampled more frequently during training compared to experiences with lower priorities. This approach aims to prioritize and replay experiences that contribute more significantly to learning and improving the model's performance.

B.2. Model hyperparameters

Table B.1 shows the hyperparameter values used for the DQN model in this study.

Table B.1
DQN hyperparameter values.

Hyperparameter	Value
Buffer size	1000000
Learning rate	0.0001
Target network update frequency	10 000
Learning starts	50 000
Final epsilon value	0.05
Discount factor	0.99
Train frequency	4

Appendix C. Anemia use case

C.1. The dataset

C.1.1. Feature inclusion

The first step in constructing our dataset involved defining a set of 17 features. Hemoglobin level, the primary indicator for anemia was included as a feature. Since normal hemoglobin levels differ between men and women, gender was also included. Additionally, we included features from the decision tree in [8] based on standard anemia diagnosis guidelines: MCV, ferritin, reticulocyte count, segmented neutrophils and Total Iron Binding Capacity (TIBC).

In consultation with a domain expert, we identified and included additional features not present in the decision tree, but pertinent to anemia diagnosis: hematocrit, transferrin saturation (TSAT), red blood cells (RBC), serum iron, and folate. Notably, three of these features (hematocrit, TSAT, and RBC) can be derived from other features and are correlated with features from our initial selection (*e.g.*, hemoglobin and hematocrit). Moreover, we incorporated features unrelated to anemia diagnosis to assess their potential impact on our model's behavior: creatinine, cholesterol, copper, ethanol, and glucose.

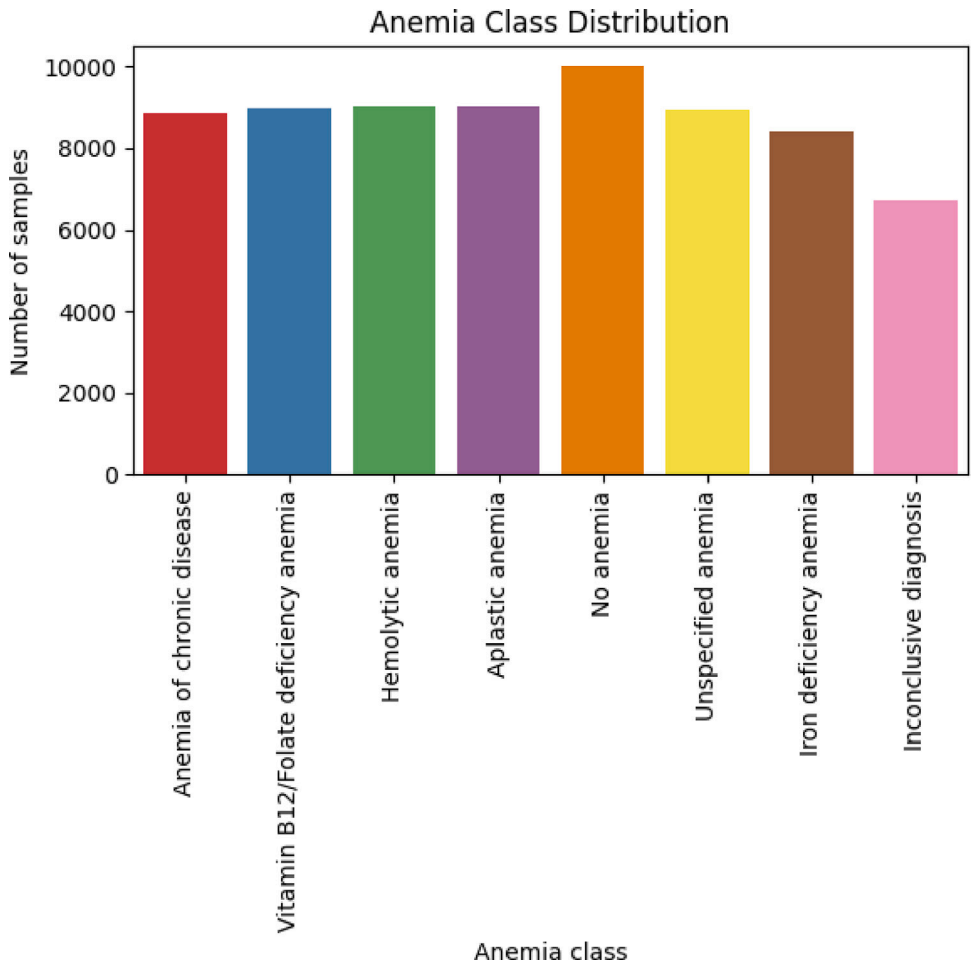

Fig. C.1. Distribution of patients across anemia classes.

C.1.2. Dataset construction

The second step in our dataset construction involved populating features with values corresponding to one of seven diagnosis classes: *No anemia*, *Vitamin B12/Folate deficiency anemia*, *Unspecified anemia*, *Anemia of chronic disease*, *Iron deficiency anemia (IDA)*, *Hemolytic anemia*, and *Aplastic anemia*. Each class dataset was developed based on the decision tree shown in Fig. 1, which was manually constructed following the guidelines from [8,39].

For each diagnostic class, feature values were generated using a uniform probability distribution, with the parameters (minimum and maximum values) determined through manual review of medical literature and the thresholds specified in the decision tree in Fig. 1. Features that are correlated with other features were derived using established equations: hematocrit [45], TSAT [46] and RBC [47].

We created 10,000 instances for each diagnostic class, combining them to form a comprehensive dataset of 70,000 instances. Additionally, an eighth diagnosis class, *Inconclusive diagnosis* was introduced for cases where the model is uncertain about the diagnosis. Instances for the *Inconclusive diagnosis* class were created from the existing 70,000 instances by randomly selecting and removing 10% of the non-missing values for each feature, except for hemoglobin, gender and MCV, which are necessary to the diagnosis of almost all the anemia classes. Fig. C.1 illustrates the class distribution; Fig. C.2 shows the ratio of observed versus missing values for each feature; Additionally, Table C.1 provides an example of an instance from the dataset; and Table C.2 presents a summarized statistical description of the features in the dataset.

Table C.1
Example instance in the *anemia* dataset.

Feature	Value
Hemoglobin	9.007012
Ferritin	–
Reticulocyte count	–
Segmented neutrophils	3.519565
TIBC	440.499323
MCV	103.442762
Serum iron	59.017997
RBC	2.612173
Gender	Male
Creatinine	0.650757
Cholesterol	114.794964
Copper	112.308159
Ethanol	25.612786
Folate	5.969710
Glucose	116.026042
Hematocrit	27.021037
TSAT	13.397977
label	Vitamin B12/Folate deficiency anemia

C.1.3. Simulating imperfect data

To compare the robustness of various approaches to imperfect data, we artificially introduced different levels of noise and missingness into our training data.

A percentage of the values of each feature, excluding hemoglobin and gender, were randomly replaced with missing values. Hemoglobin

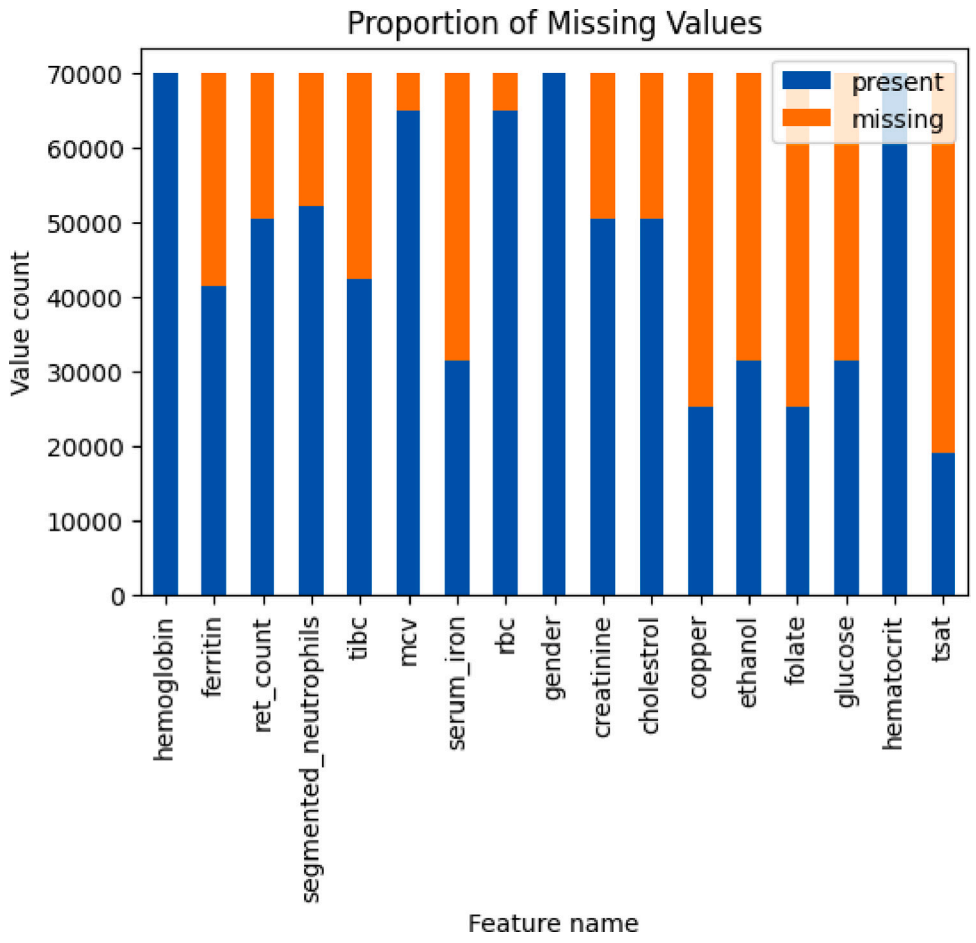

Fig. C.2. Distribution of present *versus* missing values for each feature in the *anemia* dataset.

Table C.2
Summary descriptive statistics of the *anemia* dataset, showing the mean and interquartile range (in parentheses) of the features. Gender, a binary variable, is described with the sample count and percentage.

Feature	All Classes	No anemia	Vitamin B12/Folate deficiency anemia
Hemoglobin	10.239 (8.067, 12.102)	14.570 (13.281, 15.848)	9.510 (7.771, 11.274)
Ferritin	209.968 (69.870, 343.337)	251.436 (128.145, 373.501)	251.775 (126.676, 374.486)
Reticulocyte count	2.821 (1.270, 4.342)	2.981 (1.520, 4.423)	3.024 (1.526, 4.564)
Segmented neutrophils	2.930 (0.762, 4.898)	3.532 (1.824, 5.266)	3.516 (1.791, 5.22)
TIBC	334.276 (222.108, 457.942)	310.821 (206.744, 415.958)	306.678 (202.380, 410.634)
MCV	89.998 (78.918, 101.093)	90.029 (82.631, 97.598)	102.504 (101.225, 103.794)
Serum iron	135.030 (77.793, 192.645)	134.959 (76.979, 192.764)	136.613 (80.087, 193.071)
RBC	3.348 (2.641, 3.936)	4.899 (4.379, 5.356)	2.784 (2.274, 3.298)
Gender			
Male	38268 (54.67%)	4048 (40.48%)	5108 (56.73%)
Female	31732 (45.33%)	5952 (59.52%)	3896 (43.27%)
Creatinine	1.103 (0.651, 1.552)	1.119 (0.670, 1.566)	1.102 (0.655, 1.540)
Cholesterol	74.878 (37.388, 112.244)	74.037 (36.257, 111.019)	75.020 (37.099, 112.819)
Copper	80.095 (55.182, 105.245)	79.510 (54.965, 104.816)	80.001 (55.014, 105.150)
Ethanol	39.887 (19.876, 59.749)	40.256 (19.769, 60.865)	39.445 (18.622, 59.590)
Folate	15.262 (7.832, 22.715)	15.054 (7.843, 22.174)	15.400 (8.099, 22.976)
Glucose	90.039 (65.128, 115.077)	90.918 (66.287, 115.892)	90.021 (64.290, 116.194)
Hematocrit	30.716 (24.201, 36.306)	43.709 (39.843, 47.544)	28.530 (23.313, 33.821)
TSAT	49.601 (23.103, 62.608)	52.553 (24.626, 67.235)	53.935 (25.712, 68.534)

Feature	Unspecified anemia	Anemia of chronic disease	Iron deficiency anemia
Hemoglobin	9.534 (7.769, 11.301)	9.514 (7.751, 11.254)	9.539 (7.795, 11.276)
Ferritin	250.757 (123.567, 375.763)	268.551 (152.371, 384.475)	48.654 (22.742, 74.091)
Reticulocyte count	3.006 (1.546, 4.459)	2.957 (1.458, 4.457)	2.975 (1.441, 4.502)
Segmented neutrophils	0.000 (0.000, 0.000)	3.580 (1.848, 5.284)	3.582 (1.854, 5.330)
TIBC	311.332 (208.894, 414.291)	301.558 (199.155, 402.335)	452.223 (458.074, 499.101)
MCV	102.525 (101.274, 103.774)	77.472 (76.206, 194.049)	77.527 (76.299, 78.796)

(*continued on next page*)

Table C.2 (*continued*).

Feature	Unspecified anemia	Anemia of chronic disease	Iron deficiency anemia
Serum iron	134.970 (77.499, 192.216)	135.313 (77.093, 194.049)	135.625 (77.915, 193.272)
RBC	2.790 (2.274, 3.305)	3.685 (3.008, 4.361)	3.693 (3.013, 4.371)
Gender			
Male	5175 (57.73%)	5067 (57.20%)	4765 (56.74%)
Female	3789 (42.27%)	3792 (42.8%)	3633 (43.26%)
Creatinine	1.096 (0.643, 1.554)	1.098 (0.643, 1.544)	1.103 (0.643, 1.558)
Cholesterol	75.734 (38.838, 113.730)	74.994 (37.212, 112.230)	74.700 (37.507, 112.034)
Copper	80.465 (56.729, 105.069)	80.081 (54.461, 104.850)	79.764 (54.906, 104.545)
Ethanol	39.500 (19.102, 22.784)	39.784 (19.802, 59.703)	39.501 (20.018, 58.701)
Folate	15.284 (7.919, 22.784)	15.131 (7.769, 22.650)	15.462 (7.948, 22.976)
Glucose	89.457(65.322, 113.808)	90.131 (66.091, 115.144)	90.153 (65.518, 114.407)
Hematocrit	28.602 (23.307, 33.902)	28.541 (23.253, 33.761)	28.617 (23.386, 33.828)
TSAT	52.689 (24.323, 67.385)	54.538 (24.730, 71.416)	32.675 (17.377, 42.591)

Feature	Hemolytic anemia	Aplastic anemia	Inconclusive diagnosis
Hemoglobin	9.510 (7.741, 11.262)	9.518 (7.801, 11.250)	9.486 (7.764, 11.224)
Ferritin	251.082 (123.988, 380.350)	246.611 (119.831, 374.303)	195.158 (67.362, 319.364)
Reticulocyte count	4.049 (3.079, 5.007)	1.005 (0.514, 1.500)	2.988 (1.437, 4.514)
Segmented neutrophils	3.553 (1.820, 5.271)	3.517 (1.789, 5.189)	3.478 (1.745, 5.214)
TIBC	310.197 (203.256, 418.237)	310.744 (205.529, 414.594)	338.282 (225.384, 461.215)
MCV	89.944 (84.920, 94.938)	89.985 (85.003, 95.013)	88.696 (78.100, 100.800)
Serum iron	133.068 (76.467, 191.047)	135.668 (79.215, 192.992)	133.704 (76.167, 191.807)
RBC	3.185 (2.580, 3.761)	3.186 (2.596, 3.747)	3.257 (2.628, 3.809)
Gender			
Male	5166 (57.24%)	5084 (56.31%)	3855 (57.36%)
Female	3859 (42.76%)	3945 (43.69%)	2866 (42.64%)
Creatinine	1.098 (0.656, 1.538)	1.100 (0.646, 1.551)	1.108 (0.646, 1.563)
Cholesterol	74.968 (37.216, 112.418)	74.990 (37.919, 112.314)	74.602 (37.416, 111.479)
Copper	80.348 (55.201, 105.698)	80.502 (55.372, 106.089)	80.166 (55.087, 105.649)
Ethanol	40.220 (20.430, 60.389)	40.525 (20.545, 60.523)	39.744 (21.113, 58.165)
Folate	15.396 (7.854, 22.746)	15.309 (7.731, 22.812)	15.045 (7.521, 22.613)
Glucose	90.140 (65.137, 115.555)	89.852 (64.575, 114.951)	89.348 (63.492, 113.911)
Hematocrit	28.530 (23.223, 33.785)	28.554 (23.402, 33.751)	28.458 (23.292, 33.671)
TSAT	53.800 (24.902, 69.688)	53.292 (26.181, 69.297)	48.753 (21.476, 63.018)

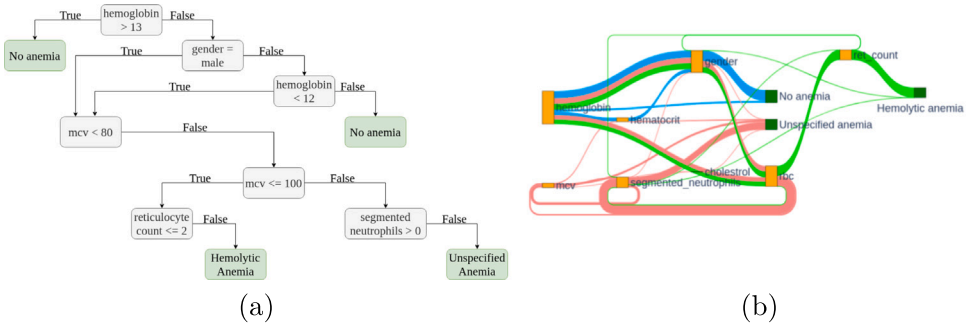

Fig. C.3. Side-by-side comparison of (*a*) a branch of the decision tree in Fig. 1 and (*b*) the corresponding pathways learned by the model for *No anemia*, *Hemolytic anemia* and *Unspecified anemia*.

and gender were excluded because hemoglobin is critical for anemia diagnosis and its normal levels vary between men and women.

Since our original dataset perfectly adheres to a decision tree, we simulated noise using the following procedure: For each anemia class, except *No anemia* and *Inconclusive diagnosis*, a specified fraction of the values in each feature within its branch of the tree in Fig. 1 were replaced by values generated from a normal distribution, $N(\mu, \sigma^2)$, where the mean, μ, corresponds to the threshold in a node with that feature in the tree, and the standard deviation, σ, was defined arbitrarily.

For example, for hemolytic anemia, the features used for its diagnosis are *hemoglobin*, *MCV* and *reticulocyte count* as shown in Fig. 1. For a noise level of 0.2, we replaced 20% of the *reticulocyte count* values using a normal distribution $N(2, 0.2)$ where μ is based on the decision tree threshold. Similarly, for *MCV*, 10% of its feature values were replaced using $N(80, 2)$, and another 10% using $N(100, 2)$. We gradually increased the noise level to analyze its impact on our model's performance and the resulting pathways. Additionally, 10% of the anemic instances were randomly labeled as *No anemia* at all noise levels.

Table C.3
Computing time to train the model and generate a diagnosis (pathway) for a single *anemia* test instance.

Model	Training time	Testing time
Decision Tree	299 ms ± 9.72 ms	18.6 µs ± 453 ns
Random Forest	5.85 s ± 8.56 ms	3.27 ms ± 156 µs
SVM	25.8 s ± 437 ms	166 µs ± 8.33 µs
FFNN	2 min 36 s ± 35 s	645 µs ± 28.6 µs
Dueling DQN-PER	1 h 42 min 14 s ± 19 min 19 s	722 µs ± 35.1 µs
Dueling DDQN-PER	2 h 12 min 45 s ± 7 min 36 s	759 µs ± 56.1 µs

Furthermore, we created datasets with both noise and missing data. Starting with a training dataset with a fixed noise level of 0.2 as described above, we added missing data at various levels using the same procedure employed to create datasets with missing data.

Table D.1
Prevalence of various features in SLE-positive patients in the *lupus* dataset. The dataset utilized half the minimum percentage from the literature source to mitigate high overlap between positive ANA patients and a confirmed lupus diagnosis.

Feature name	Literature source	Percentage from literature	Percentage used in dataset creation for SLE positive patients
ANA	[38]	100%	100%
Fever	[48]	36%–86%	18%
Leukopenia	[49]	50%–60%	25%
Thrombocytopenia	[50]	20%–40%	10%
Autoimmune hemolysis	[51]	3%	1.5%
Delirium	[52]	49.33%	24.5%
Psychosis	[52]	12%	6%
Seizure	[52]	10.67%	5.5%
Non-scarring alopecia	[53]	85%	42.5%
Oral ulcers	[54]	8%–45%	4%
Subacute cutaneous lupus	[55]	10%	5%
Acute cutaneous lupus	[55]	3%	1.5%
Discoid lupus	[56]	15%–20%	7.5%
Pleural effusion	[57]	17%–60%	8.5%
Pericardial effusion	[58]	50%	25%
Acute pericarditis	[59]	1%	0.5%
Joint involvement	[60]	69%–95%	34.5%
Proteinuria	[61]	60%	30%
Renal biopsy-proven lupus nephritis	[62]	16.9%	8.5%
Anti-cardiolipin antibodies	[63]	17%–40%	8.5%
Anti-β2GP1 antibodies	[64]	10%–35%	5%
Lupus anticoagulant	[63]	11%–30%	5.5%
Low C3	[65]	45%	22.5%
Low C4	[65]	44%	22%
Anti-dsDNA antibody	[66]	70%–98%	35%
Anti-Smith antibody	[67]	9%–49%	4.5%

C.2. Sample learned pathways

Fig. C.3 provides another example of the pathways generated using our approach. It displays a side-by-side illustration of a branch of the decision tree used to label the dataset and a Sankey diagram showing the pathways learned by the Dueling DQN-PER model for diagnosing *No anemia*, *Unspecified anemia*, and *Hemolytic anemia*, which are colored *blue*, *coral* and *light green*, respectively. The feature actions are represented by the orange nodes, while the diagnosis actions are depicted as dark green nodes. In addition, the size of each flow corresponds to its support, *i.e.*, the number of patients that follow that path.

C.3. SOTA hyperparameters

We used the Keras Python library to build an FFNN comprising an input layer, a hidden layer with the ReLU activation function, and an output layer with the softmax activation function. The model was optimized using the Adam optimizer and categorical cross-entropy was employed as the loss function.

DT, RF and SVM models were defined using the Scikit-learn Python library, while XGBoost was created using the XGBoost Python library. For the SVM, data normalization was performed before training. The model utilized a polynomial kernel, a regularization parameter (C) of 100, and a one-vs-one decision function. The DT model employed the entropy criterion, while the RF and XGBoost models were implemented with the default hyperparameter values specified in their respective documentation. Any hyperparameters not explicitly mentioned were set to their default values.

The PPO algorithm was defined using the Stable-Baselines Python library. The learning rate was set to 0.0001, and the entropy coefficient was set to 0.02. The rest of the hyperparameters were set to their default values as defined in the library documentation. The source code of these models, as used in this study, is available at https://github.com/lilly-muyama/Deep_RL_diagnosis_pathways/tree/main/anemia .

C.4. Computing time

In Table C.3, the training time for each model, as well as the time taken to generate a diagnosis pathway or diagnosis (testing time) for both the DQN models and the SOTA models respectively for a single instance in the test dataset, are presented.

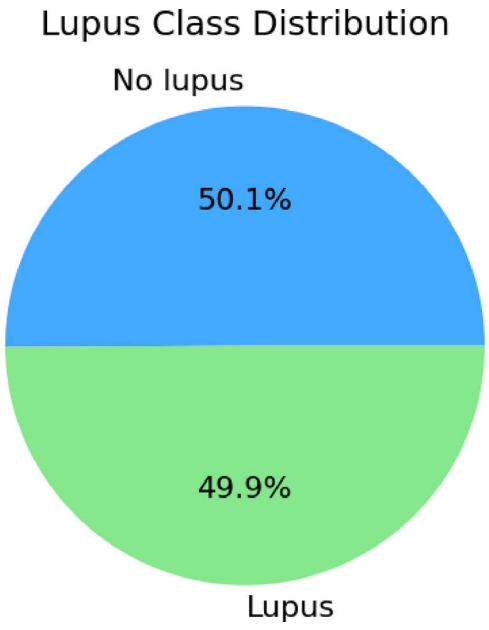

Fig. D.1. Distribution of patients across lupus classes.

Appendix D. Lupus use case

D.1. The dataset

D.1.1. Feature inclusion

To construct the dataset, we generated features based on the clinical practice guideline outlined in [38] for the classification of SLE. In adherence to this guideline, we identified a total of 24 features: ANA,

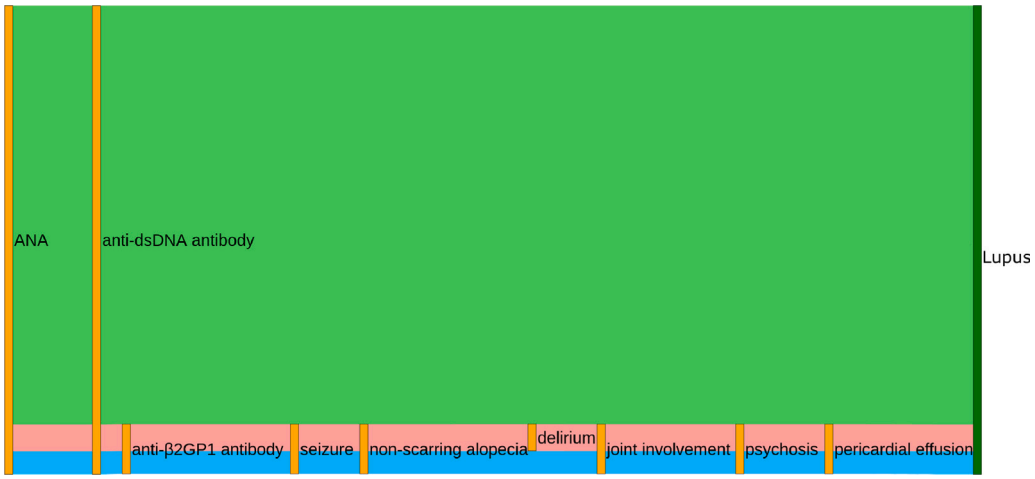

Fig. D.2. The three most common diagnostic decision pathways for the *Lupus* class generated by one of the models achieving the best wPAHM score. The pathways were generated by the model with the wPAHM score closest to the mean wPAHM of the ten runs.

fever, leukopenia, thrombocytopenia, autoimmune hemolysis, delirium, psychosis, seizure, non-scarring alopecia, oral ulcers, cutaneous lupus, pleural effusion, pericardial effusion, acute pericarditis, joint involvement, proteinuria, renal biopsy-proven lupus nephritis, anti-cardiolipin antibodies, anti-β2GP1 antibodies, lupus anticoagulant, low C3, low C4, anti-dsDNA antibody, and anti-Smith antibody.

This guideline describes a weighted scoring system using these features to determine the presence of SLE. Consequently, we applied the same criteria to label our synthetic instances, establishing two classes: *Lupus* and *No lupus*.

D.1.2. Dataset construction

With reference to [38], we considered the presence of ANA at a titer $\geq$ 1:80 on HEp-2 cells as an entry criterion (referred to as the ANA feature for clarity) for SLE diagnosis and accordingly, we constructed two datasets. In one dataset, instances were labeled as 1 (True) for the ANA feature, indicating a positive ANA result; in the other dataset, instances were labeled as 0 (False), indicating a negative ANA result. Specifically, we generated 50,000 instances with a positive ANA and 20,000 instances with a negative ANA.

In both datasets, all features are binary, indicating whether a specific feature is present (1) or absent (0) in a given instance, except for two features: cutaneous lupus and renal biopsy-proven lupus nephritis. These two features are categorical, taking one of four values (0 for False, 1 for subacute cutaneous lupus, 2 for acute cutaneous lupus, 3 for discoid lupus) for cutaneous lupus, and one of six values (0 for False, 1 for class I, 2 for class II, 3 for class III, 4 for class IV, 5 for class V) for renal biopsy-proven lupus nephritis, respectively.

In the positive ANA dataset, the prevalence of positive features was determined from literature sources (see Table D.1 in Appendix D). We opted to use half of the minimum percentage reported in the literature because using any value within the full percentage range resulted in a scenario where nearly all lupus patients could be diagnosed accurately based solely on the presence/absence of ANA. Specifically, almost every patient with a positive ANA test would meet the threshold score of ten required for a positive lupus diagnosis. To mitigate this issue, we reduced the percentage to half the minimum reported in the literature. Conversely, for the negative ANA dataset, the prevalence of features was derived considering their relative importance in lupus diagnosis, as indicated by their weight in the diagnostic criteria. Features with larger weights were considered less prevalent in patients, following the logic that their rarity contributes to their significance in the diagnosis process.

The features were populated using a weighted probability distribution, accounting for the prevalence levels or weighted criteria as outlined. Subsequently, the two datasets were labeled based on the weighted criteria specified in [38], then merged to create a unified dataset of 70,000 instances and 24 features, where 35,056 and 34,944 instances belonged to the *No lupus* and *Lupus* classes, respectively. Fig. D.1 illustrates the dataset class distribution. Table D.1 presents the prevalence of various features in patients with SLE, along with the corresponding literature sources from which they were determined. An example instance in the dataset is provided in Table D.2, while a summary statistical description of the dataset is shown in Table D.3.

Table D.2
Example instance in the *Lupus* dataset.

Feature	Value
ANA	1
Fever	0
Leukopenia	0
Thrombocytopenia	1
Autoimmune hemolysis	0
Delirium	1
Psychosis	0
Seizure	0
Non-scarring alopecia	0
Oral ulcers	0
Cutaneous lupus	0
Pleural effusion	0
Pericardial effusion	0
Acute pericarditis	0
Joint involvement	0
Proteinuria	1
Renal biopsy-proven lupus nephritis	0
Anti-cardiolipin antibodies	0
Anti-β2GP1 antibodies	0
Lupus anticoagulant	0
Low C3	0
Low C4	0
Anti-dsDNA antibody	0
Anti-Smith antibody	0
Label	Lupus

D.1.3. Simulating imperfect data

A percentage of values for each feature, excluding ANA, were randomly replaced with missing values. ANA was excluded because, as mentioned earlier, it serves as the entry criterion for SLE diagnosis;

Table D.3
Summary statistics of the *Lupus* dataset showing the sample number and percentage of each value for each feature.

Feature	All classes	No lupus	Lupus
ANA	0 - 20000 (28.6%) 1 - 70000(71.4%)	0 - 20000 (57.1%) 1 - 15056(42.9%)	0 - 0 (0%) 1 - 34944(100%)
Fever	0 - 60274 (86.1%) 1 - 9726(13.9%)	0 - 32363 (92.3%) 1 - 2693 (7.7%)	0 - 27911 (79.9%) 1 - 7033(20.1%)
Leukopenia	0 - 56776 (81.1%) 1 - 13224 (18.9%)	0 - 31929 (91.1%) 1 - 3127 (8.9%)	0 - 24847 (71.1%) 1 - 10097 (28.9%)
Thrombocytopenia	0 - 64388 (92.0%) 1 - 5612 (8%)	0 - 33712 (96.2%) 1 - 1344 (3.8%)	0 - 30676 (87.8%) 1 - 4268 (12.2%)
Autoimmune hemolysis	0 - 68600 (98%) 1 - 1400 (2%)	0 - 34304 (97.9%) 1 - 752 (2.1%)	0 - 34296 (98.1%) 1 - 648 (1.9%)
Delirium	0 - 56898 (81.3%) 1 - 13102 (18.7%)	0 - 31521 (89.9%) 1 - 3535 (10.1%)	0 - 25377 (72.6%) 1 - (27.4%)
Psychosis	0 - 66335 (94.8%) 1 - 3665 (5.2%)	0 - 33840 (96.5%) 1 - 1216 (3.5%)	0 - 32495 (93.0%) 1 - 2449 (7.0%)
Seizure	0 - 66762 (95.4%) 1 - 3238 (4.6%)	0 - 34277 (97.8%) 1 - 779 (2.2%)	0 - 32485 (93.0%) 1 - (7.0%)
Non-scarring alopecia	0 - 48040 (68.6%) 1 - 21960 (31.4%)	0 - 29220 (83.4%) 1 - 5836 (16.6%)	0 - 18820 (53.9%) 1 - 16124 (46.1%)
Oral ulcers	0 - 67247 (96.1%) 1 - 2753 (3.9%)	0 - 33792 (96.4%) 1 - 1264 (3.6%)	0 - 33455 (95.7%) 1 - 1489 (4.3%)
Cutaneous lupus	0 - 61370 (87.7%) 1 - 3143 (4.5%) 2 - 1111 (1.6%) 3 - 4376 (6.3%)	0 - 32408 (92.4%) 1 - 1003 (2.9%) 2 - 447 (1.3%) 3 - 1198 (3.4%)	0 - 28962 (82.9%) 1 - 2140 (6.1%) 2 - 664 (1.9%) 3 - 3178 (9.1%)
Pleural effusion	0 - 65212 (93.2%) 1 - 4788 (6.8%)	0 - 34043 (97.1%) 1 - 1013 (2.9%)	0 - 31169 (89.2%) 1 - 3775 (10.8%)
Pericardial effusion	0 - 57098 (81.6%) 1 - 12902 (18.4%)	0 - 33075 (94.3%) 1 - 1981 (5.7%)	0 - 24023 (68.7%) 1 - 10921 (31.3%)
Acute pericarditis	0 - 69328 (99.0%) 1 - 672 (1.0%)	0 - 34633 (98.8%) 1 - 423 (1.2%)	0 - 34695 (99.3%) 1 - 249 (0.7%)
Joint involvement	0 - 52195 (74.6%) 1 - 17805 (25.4%)	0 - 33248 (94.8%) 1 - 1808 (5.2%)	0 - 18947 (54.2%) 1 - 15997 (45.8%)
Proteinuria	0 - 54516 (77.9%) 1 - 15484 (22.1%)	0 - 32138 (91.7%) 1 - 2918 (8.3%)	0 - 22378 (64.0%) 1 - 12566 (36.0%)
Renal biopsy-proven lupus nephritis	0 - 64148 (91.6%) 1 - 1306 (1.9%) 2 - 954 (1.4%) 3 - 1089 (1.6%) 4 - 1829 (2.6%) 5 - 674 (1.0%)	0 - 33352 (95.1%) 1 - 1126 (3.2%) 2 - 211 (0.6%) 3 - 95 (0.3%) 4 - 83 (0.2%) 5 - 189 (0.5%)	0 - 30796 (88.1%) 1 - 180 (0.5%) 2 - 743 (2.1%) 3 - 994 (2.8%) 4 - 1746 (5.0%) 5 - 485 (1.4%)
Anti-cardiolipin antibodies	0 - 64890 (92.7%) 1 - 5110 (7.3%)	0 - 33395 (95.3%) 1 - 1661 (4.7%)	0 - 31495 (90.1%) 1 - 3449(9.9%)
Anti-β2GP1 antibodies	0 - 66581 (95.1%) 1 - 3419 (4.9%)	0 - 33703 (96.1%) 1 - 1353 (3.9%)	0 - 32878 (94.1%) 1 - 2066 (5.9%)
Lupus anticoagulant	0 - 66424 (94.9%) 1 - 3576 (5.1%)	0 - 33640 (96.0%) 1 - 1416 (4.0%)	0 - 32784 (93.8%) 1 - 2160 (6.2%)
Low C3	0 - 57962 (82.8%) 1 - 12038 (17.2%)	0 - 32112 (91.6%) 1 - 2944 (8.4%)	0 - 25850 (74.0%) 1 - 9094 (26.0%)
Low C4	0 - 58221 (83.2%) 1 - 11779 (16.8%)	0 - 32179 (91.8%) 1 - 2877 (8.2%)	0 - 26042 (74.5%) 1 - 8902 (25.5%)
Anti-dsDNA antibody	0 - 52168 (74.5%) 1 - 17832 (25.5%)	0 - 33107 (94.4%) 1 - 1949 (5.6%)	0 - 19061 (54.5%) 1 - 15883 (45.5%)
Anti-Smith antibody	0 - 67399 (96.3%) 1 - 2601 (3.7%)	0 - 34439 (98.2%) 1 - 617 (1.8%)	0 - 32960 (94.3%) 1 - 1984 (5.7%)

without this feature, a patient should not be diagnosed with lupus according to [38].

Additionally, we introduced noise into the dataset through the following method: for each level of noise, a specified fraction of instances in the dataset was randomly selected, and their labels were changed to the opposite class. For instance, at a noise level of 0.2, the diagnoses of 20% of the instances in the training set were altered; those initially diagnosed with *Lupus* were changed to *No lupus*, and vice versa. As a result, the training dataset no longer perfectly adhered to the diagnosis criteria outlined in [38].

Furthermore, we generated datasets incorporating both noise and missing data. Starting with a training dataset with a fixed noise level of 0.2, we introduced missing data at various levels using the same procedure employed to create datasets with missing values. This approach allowed us to explore the impact of both noise and missing data on the dataset.

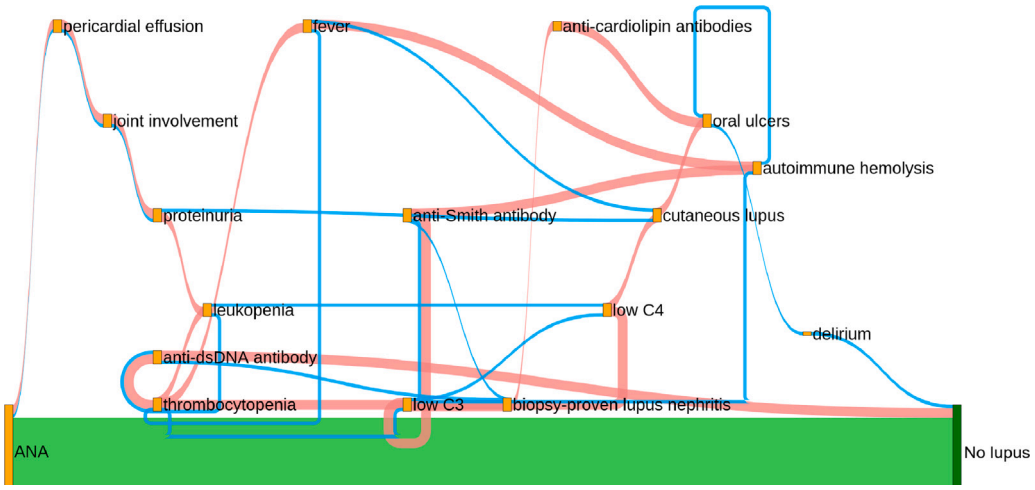

Fig. D.3. The three most common diagnostic decision pathways for the *No lupus* class generated by one of the models achieving the best accuracy score. The pathways were generated by the model with the accuracy score closest to the mean accuracy of the ten runs.

Table D.4
Penalty weights assigned to different features. The final score for each feature is calculated as a weighted sum: $2 \times$ invasiveness $+ 0.5 \times$ turnaround time $+ 0.5 \times$ financial cost. Details on how the penalty weights for each criterion were determined are provided in Table D.5.

Feature	Invasiveness	Turnaround time	Financial cost	Weighted score (c)
ANA	4	3	4	11.5
Fever	5	5	5	15
Leukopenia	4	4	4	12
Thrombocytopenia	4	4	4	12
Autoimmune hemolysis	4	4	4	12
Delirium	5	5	5	15
Psychosis	5	5	5	15
Seizure	5	5	5	15
Non-scarring alopecia	5	5	5	15
Oral ulcers	5	5	5	15
Cutaneous lupus	2	2	3	6.5
Pleural effusion	5	3	3	13
Pericardial effusion	5	3	3	13
Acute pericarditis	5	3	3	13
Joint involvement	5	3	3	13
Proteinuria	5	3	4	13.5
Renal biopsy-proven lupus nephritis	0	2	0	1
Anti-cardiolipin antibodies	4	1	3	10
Anti-β2GP1 antibodies	4	0	3	9.5
Lupus anti-coagulant	4	0	3	9.5
Low C3	4	1	4	10.5
Low C4	4	1	4	10.5
Anti-dsDNA antibody	4	3	4	11.5
Anti-Smith antibody	4	3	4	11.5

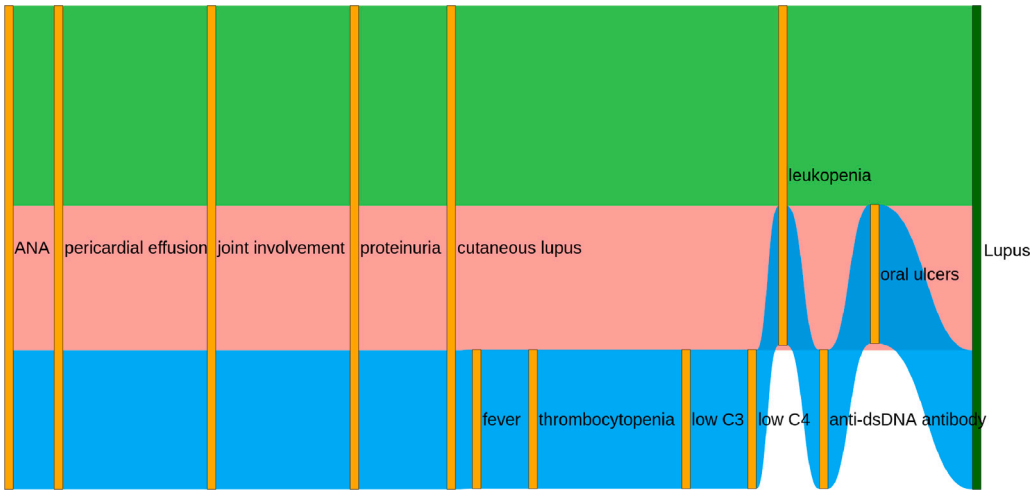

Fig. D.4. The three most common diagnostic decision pathways for the *Lupus* class generated by one of the models with the best accuracy score. The pathways were generated by the model with the accuracy score closest to the mean accuracy of the ten runs.

Table D.5
Values used to determine the penalty weights.

Rating	Invasiveness	Turnaround time	Financial cost
5	none	none	none
4	very low	within 1 day	< USD 50
3	low	1–3 days	USD 50 - 300
2	medium	3–7 days	USD 301 - 700
1	high	7–14 days	USD 701 - 1000
0	very high	> 2 weeks	> USD 1000

Table D.6
Computing time to train the model and generate a diagnosis (pathway) for a single *lupus* test instance.

Model	Training time	Testing time
Decision Tree	25.3 ms ± 991 µs	29.3 µs ± 771 ns
Random Forest	772 ms ± 46.8	3.16 ms ± 98.5 µs
SVM	3.09 s ± 35.1 ms	114 µs ± 1.66µs
FFNN	3 min 21 s ± 633 ms	597 µs ± 22.9 µs
Dueling DQN-PER	13 h 53 min 35 s ± 12 min 10 s	1.79 ms ± 62.7 µs
Dueling DDQN-PER	15 h 16 min 38 s ± 1 h 49 min 51 s	1.2 ms ± 12.6 µs

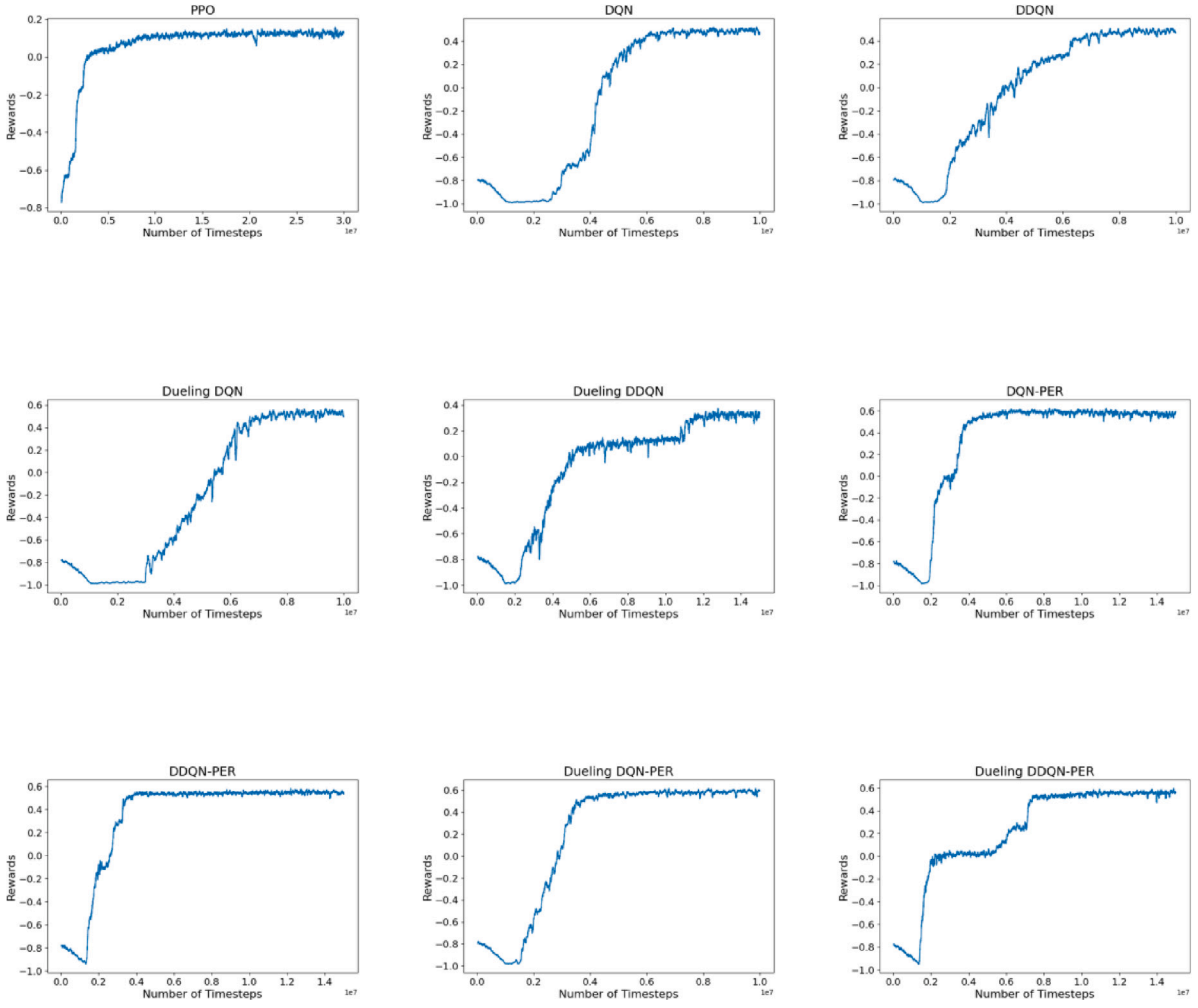

Fig. E.1. Training curves for the RL models in the *anemia* use case, with the *x*-axis representing tens of millions of time steps.

D.2. Determining penalty weights

Table D.4 shows the penalty weights assigned to different features, which were used to determine the penalty incurred by the agent when selecting that action, as specified in the reward function in Eq. (4). The scores were calculated based on the level of invasiveness of the action, the turnaround time for its observation, and the financial cost of performing that action. These values were derived from a review of existing literature and online sources. A breakdown of how the scores of each criterion were generated is provided in Table D.5.

D.3. Sample learned pathways

Fig. D.2 shows the three most common pathways for the *Lupus* class generated by one of our using the dueling DDQN model selected based on the wPAHM score.

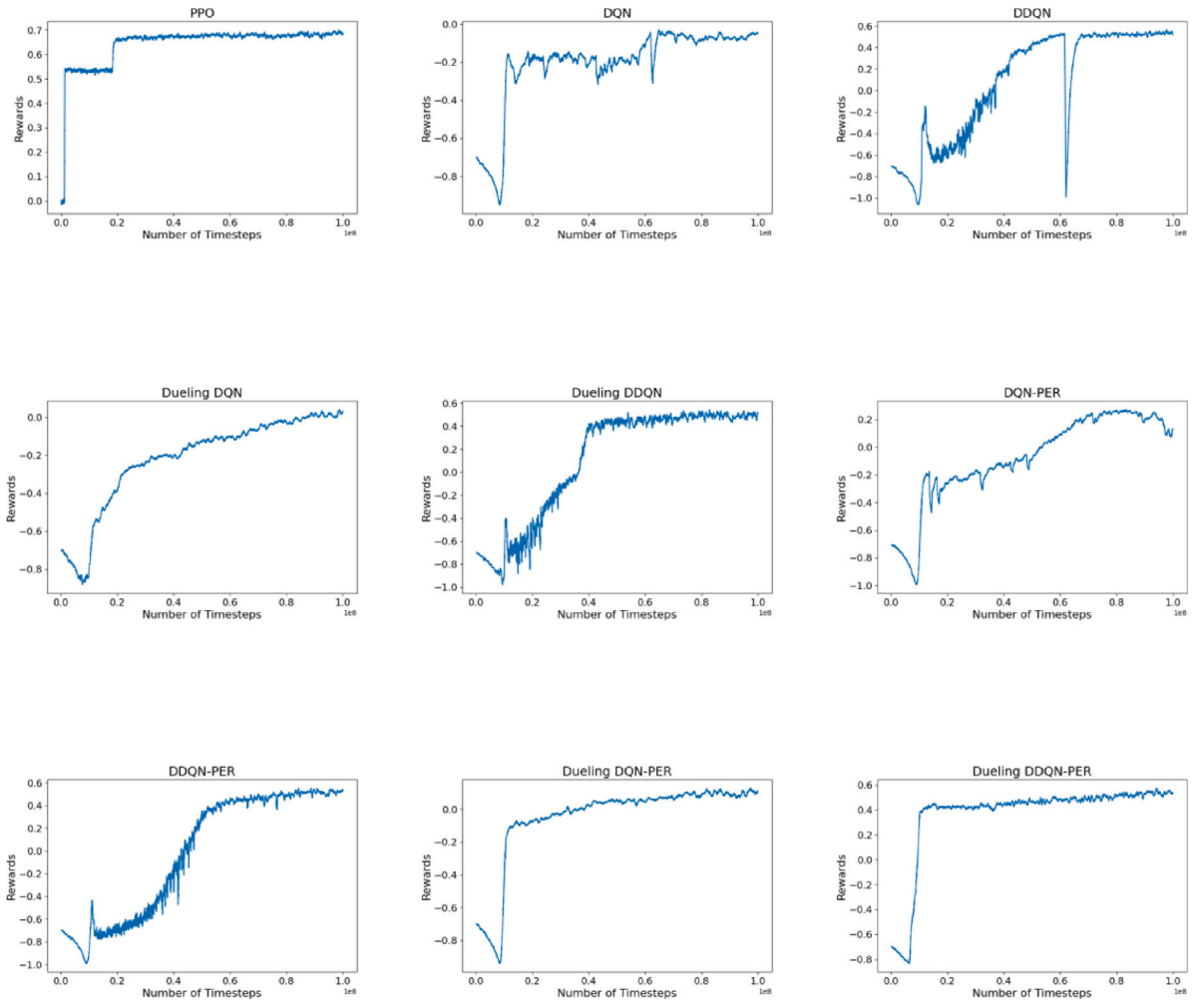

Fig. E.2. Training curves for the RL models in the *lupus* use case, with the x-axis representing hundreds of millions of time steps.

Figs. D.3 and D.4 depict the three most common pathways for the *No lupus* and *Lupus* classes, respectively, generated by another experiment using the dueling DDQN model chosen based on the accuracy score. The models used in these figures were those with a score closest to the mean metric of the ten runs, with λ set to 9.

D.4. SOTA hyperparameters

We used the Keras Python library to build an FFNN, with an input layer, a hidden layer employing the ReLU activation function, and an output layer employing the softmax activation function. The model was optimized using the Adam optimizer, and categorical cross-entropy was used as the loss function.

For the DT, RF, and SVM models, we utilized the Scikit-learn Python library. The SVM model utilized a Radial-Basis Function (RBF) kernel and a one-vs-one decision function. DT and RF were implemented with the default hyperparameter values as specified in the Scikit-learn documentation. Any parameters not explicitly mentioned were set to their default values.

XGBoost was implemented using the XGBoost Python library and its default hyperparameter values were used.

The PPO algorithm was defined using the Stable-Baselines Python library. The learning rate was set to 0.0001. The rest of the hyperparameters were set to their default values as defined in the library documentation. The source code for these models, as used in this study, is available at https://github.com/lilly-muyama/Deep_RL_diagnosis_pathways/tree/main/lupus

D.5. Computing time

In Table D.6, the training time for each model, as well as the time taken to generate a diagnosis pathway or diagnosis (testing time) for the DQN models and the SOTA models respectively for a single instance in the test dataset, are shown.

Appendix E. Training curves

Figs. E.1 and E.2 display the training curves for the RL approaches in the *anemia* and *lupus* use cases, respectively, based on a single experiment for each. Due to the large number of time steps, a moving average of the reward was used for the plots.

References

[1] Field MJ, Lohr KN, et al. Clinical practice guidelines. Dir New Program 1990;1990.
[2] Steinberg E, Greenfield S, Wolman DM, Mancher M, Graham R, et al. Clinical practice guidelines we can trust. National Academies Press; 2011.
[3] Adler-Milstein J, Chen JH, Dhaliwal G. Next-generation artificial intelligence for diagnosis: From predicting diagnostic labels to "wayfinding". JAMA 2021;326(24):2467–8.
[4] Jensen PB, Jensen LJ, Brunak Sr. Mining electronic health records: towards better research applications and clinical care. Nature Rev Genet 2012;13(6):395–405.
[5] Lipton ZC, Kale DC, Elkan C, Wetzel R. Learning to diagnose with LSTM recurrent neural networks. 2015, arXiv preprint arXiv:1511.03677.
[6] Miotto R, Li L, Kidd BA, Dudley JT. Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. Sci Rep 2016;6(1):1–10.

[7] Choi E, Bahadori MT, Schuetz A, Stewart WF, Sun J. Doctor AI: Predicting clinical events via recurrent neural networks. In: Machine learning for healthcare conference. PMLR; 2016, p. 301–18.

[8] Zaiden R. Evaluation of anemia. BMJ Best Pract. 2022. https://bestpractice.bmj.com/topics/en-us/93/diagnosis-approach, [Accessed 08 September 2022].

[9] Zhang Y, Padman R, Patel N. Paving the COWpath: Learning and visualizing clinical pathways from electronic health record data. J Biomed Inform 2015;58:186–97.

[10] Najjar A, Reinharz D, Girouard C, Gagné C. A two-step approach for mining patient treatment pathways in administrative healthcare databases. Artif Intell Med 2018;87:34–48.

[11] Huang Z, Lu X, Duan H. On mining clinical pathway patterns from medical behaviors. Artif Intell Med 2012;56(1):35–50.

[12] Baker K, Dunwoodie E, Jones RG, Newsham A, Johnson O, Price CP, et al. Process mining routinely collected electronic health records to define real-life clinical pathways during chemotherapy. Int J Med Inf 2017;103:32–41.

[13] Huang Z, Ge Z, Dong W, He K, Duan H. Probabilistic modeling personalized treatment pathways using electronic health records. J Biomed Inform 2018;86:33–48.

[14] Chiudinelli L, Dagliati A, Tibollo V, Albasini S, Geifman N, Peek N, et al. Mining post-surgical care processes in breast cancer patients. Artif Intell Med 2020;105:101855.

[15] Li W, Min X, Ye P, Xie W, Zhao D. Temporal topic model for clinical pathway mining from electronic medical records. BMC Med Inform Decis Mak 2024;24(1):20.

[16] Lin X, Li Y, Xu Y, Guo W, He W, Zhang H, et al. Personalized clinical pathway recommendation via attention based pre-training. In: 2021 IEEE international conference on bioinformatics and biomedicine. BIBM, IEEE; 2021, p. 980–7.

[17] Wilkins-Caruana A, Bandara M, Musial K, Catchpoole D, Kennedy PJ. Inferring actual treatment pathways from patient records. J Biomed Inform 2023;148:104554.

[18] Li T, Wang Z, Lu W, Zhang Q, Li D. Electronic health records based reinforcement learning for treatment optimizing. Inf Syst 2022;104:101878.

[19] Bhattarai K, Rajaganapathy S, Das T, Kim Y, Chen Y, Alzheimer's Disease Neuroimaging Initiative and Australian Imaging Biomarkers and Lifestyle Flagship Study of Ageing, et al. Using artificial intelligence to learn optimal regimen plan for alzheimer's disease. J Am Med Inform Assoc 2023;30(10):1645–56.

[20] Grolleau F, Petit F, Gaudry S, Diard É, Quenot J-P, Dreyfuss D, et al. Personalizing renal replacement therapy initiation in the intensive care unit: a reinforcement learning-based strategy with external validation on the AKIKI randomized controlled trials. J Am Med Inform Assoc 2024;31(5):1074–83.

[21] Koshimizu H, Kojima R, Kario K, Okuno Y. Prediction of blood pressure variability using deep neural networks. Int J Med Inform 2020;136:104067.

[22] Obaido G, Ogbuokiri B, Swart TG, Ayawei N, Kasongo SM, Aruleba K, et al. An interpretable machine learning approach for hepatitis b diagnosis. Appl Sci 2022;12(21):11127.

[23] Zoabi Y, Deri-Rozov S, Shomron N. Machine learning-based prediction of COVID-19 diagnosis based on symptoms. npj Digit Med 2021;4(1):3.

[24] Kavya R, Christopher J, Panda S, Lazarus YB. Machine learning and XAI approaches for allergy diagnosis. Biomed Signal Process Control 2021;69:102681.

[25] Li Y, Oliva J. Active feature acquisition with generative surrogate models. In: International conference on machine learning. PMLR; 2021, p. 6450–9.

[26] Janisch J, Pevnỳ T, Lisỳ V. Classification with costly features using deep reinforcement learning. In: Proceedings of the AAAI conference on artificial intelligence. 33, (01):2019, p. 3959–66.

[27] Yu Z, Li Y, Kim J, Huang K, Luo Y, Wang M. Deep reinforcement learning for cost-effective medical diagnosis. 2023, arXiv preprint arXiv:2302.10261.

[28] Tang K-F, Kao H-C, Chou C-N, Chang EY. Inquire and diagnose: Neural symptom checking ensemble using deep reinforcement learning. In: NIPS workshop on deep reinforcement learning. 2016.

[29] Wei Z, Liu Q, Peng B, Tou H, Chen T, Huang X-J, et al. Task-oriented dialogue system for automatic diagnosis. In: Proceedings of the 56th annual meeting of the association for computational linguistics (volume 2: short papers). 2018, p. 201–7.

[30] Kao H-C, Tang K-F, Chang E. Context-aware symptom checking for disease diagnosis using hierarchical reinforcement learning. In: Proceedings of the AAAI conference on artificial intelligence. 32, (1). 2018.

[31] Littman M. Markov decision processes. In: Smelser NJ, Baltes PB, editors. International encyclopedia of the social & behavioral sciences. Oxford: Pergamon; 2001, p. 9240–2.

[32] Sutton RS, Barto AG. Reinforcement learning: An introduction. MIT Press; 2018.

[33] Watkins CJ, Dayan P. Q-learning. Mach Learn 1992;8(3):279–92.

[34] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, et al. Human-level control through deep reinforcement learning. Nature 2015;518(7540):529–33.

[35] Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, Tang J, et al. OpenAI gym. 2016, arXiv preprint arXiv:1606.01540.

[36] Hill A, Raffin A, Ernestus M, Gleave A, Kanervisto A, Traore R, et al. Stable baselines. 2018, GitHub, https://github.com/hill-a/stable-baselines.

[37] DQN - stable baselines 2.10.3a0 documentation. 2021, https://stable-baselines.readthedocs.io/en/master/modules/dqn.html, [Accessed 08 July 2024].

[38] Aringer M, Costenbader K, Daikh D, Brinks R, Mosca M, Ramsey-Goldman R, et al. 2019 European league against rheumatism/American college of rheumatology classification criteria for systemic lupus erythematosus. Arthritis & Rheumatology 2019;71(9):1400–12.

[39] Short MW, Domagalski JE. Iron deficiency anemia: evaluation and management. Am Fam Physician 2013;87(2):98–104.

[40] Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, et al. Missing value estimation methods for DNA microarrays. Bioinformatics 2001;17(6):520–5.

[41] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT Press; 2016.

[42] Van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double Q-learning. In: Proceedings of the AAAI conference on artificial intelligence. 30, (1). 2016.

[43] Wang Z, Schaul T, Hessel M, Hasselt H, Lanctot M, Freitas N. Dueling network architectures for deep reinforcement learning. In: International conference on machine learning. PMLR; 2016, p. 1995–2003.

[44] Schaul T, Quan J, Antonoglou I, Silver D. Prioritized experience replay. 2015, arXiv preprint arXiv:1511.05952.

[45] Koperska M. Hematocrit/hemoglobin ratio calculator. 2023, https://www.omnicalculator.com/health/hct-hgb, [Accessed 22 November 2022].

[46] Nedea D. Transferrin saturation calculator. 2020, https://www.mdapp.co/transferrin-saturation-calculator-444/, [Accessed 24 November 2022].

[47] Naeim F, Nagesh Rao P, Song SX, Grody WW. 61 - disorders of red blood cells—Anemias. In: Naeim F, Nagesh Rao P, Song SX, Grody WW, editors. Atlas of hematopathology. Academic Press; 2013, p. 675–704.

[48] Timlin H, Syed A, Haque U, Adler B, Law G, Machireddy K, et al. Fevers in adult lupus patients. Cureus 2018;10(1).

[49] Fayyaz A, Igoe A, Kurien BT, Danda D, James JA, Stafford HA, et al. Haematological manifestations of lupus. Lupus Sci Med 2015;2(1):e000078.

[50] Galanopoulos N, Christoforidou A, Bezirgiannidou Z. Lupus thrombocytopenia: pathogenesis and therapeutic implications. Mediterr J Rheumatol 2017;28(1):20–6.

[51] Crawford LR, Neparidze N. Refractory autoimmune hemolytic anemia in a Systemic Lupus Erythematosus patient: A clinical case report. Clin Case Rep 2022;10(3):e05583.

[52] Nayak RB, Bhogale GS, Patil NM, Chate SS. Psychosis in patients with Systemic Lupus Erythematosus. Indian J Psychol Med 2012;34(1):90–3.

[53] Desai K, Miteva M. Recent insight on the management of Lupus Erythematosus Alopecia. Clin Cosmet Investig Dermatol 2021;333–47.

[54] Kudsi M, Nahas LD, Alsawah R, Hamsho A, Omar A. The prevalence of oral mucosal lesions and related factors in Systemic Lupus Erythematosus patients. Arthritis Res Ther 2021;23:1–5.

[55] Grönhagen CM, Nyberg F. Cutaneous lupus erythematosus: An update. Indian Dermatol Online J 2014;5(1):7.

[56] Provost TT. The relationship between discoid and Systemic Lupus Erythematosus. Arch Dermatol 1994;130(10):1308–10.

[57] Yao X, Abd Hamid M, Sundaralingam A, Evans A, Karthikappallil R, Dong T, et al. Clinical perspective and practices on pleural effusions in chronic systemic inflammatory diseases. Breathe 2020;16(4).

[58] Almousa S, Wannous H, Khedr K, Qasem H. Unusual case presentation of Systemic Lupus Erythematosus in a young woman. Rheumato 2022;2(4):93–7.

[59] Narang VK, Bowen J, Masarweh O, Burnette S, Valdez M, Moosavi L, et al. Acute pericarditis leading to a diagnosis of SLE: a case series of 3 patients. J Investig Med High Impact Case Rep 2022;10:23247096221077832.

[60] Ceccarelli F, Perricone C, Cipriano E, Massaro L, Natalucci F, Capalbo G, et al. Joint involvement in Systemic Lupus Erythematosus: from pathogenesis to clinical assessment. In: Seminars in arthritis and rheumatism. 47, (1):Elsevier; 2017, p. 53–64.

[61] Wiegley N. Low-grade proteinuria in patients with Systemic Lupus Erythematosus. Kidney News 2022;14(10/11):63.

[62] Hong W, Ren Y-L, Chang J, Luo G, Ling-Yun S. A systematic review and meta-analysis of prevalence of biopsy-proven lupus nephritis. Arch Rheumatol 2018;33(1):17.

[63] Ünlü O, Zuily S, Erkan D. The clinical significance of antiphospholipid antibodies in Systemic Lupus Erythematosus. Eur J Rheumatol 2016;3(2):75.

[64] Dema B, Charles N. Autoantibodies in SLE: specificities, isotypes and receptors. Antibodies 2016;5(1):2.

[65] Ramos-Casals M, Campoamor M, Chamorro A, Salvador G, Segura S, Botero J, et al. Hypocomplementemia in Systemic Lupus Erythematosus and primary antiphospholipid syndrome: prevalence and clinical significance in 667 patients. Lupus 2004;13(10):777–83.

[66] Fabrizio C, Fulvia C, Carlo P, Laura M, Elisa M, Francesca M, et al. Systemic Lupus Erythematosus with and without anti-dsDNA antibodies: analysis from a large monocentric cohort. Mediators Inflamm 2015;2015.

[67] Arroyo-Ávila M, Santiago-Casas Y, McGwin G, Cantor RS, Petri M, Ramsey-Goldman R, et al. Clinical associations of anti-smith antibodies in PROFILE: a multi-ethnic lupus cohort. Clin Rheumatol 2015;34:1217–23.