Journal of Computing, Science &Technology
https://focjournal.unidel.edu.ng/
editorjcst@unidel.edu.ng
ISSN: 3043-6818 Print
https://focjournal.unidel.edu.ng

Reinforcement Learning-Based Model for Adaptive Personalised e-Learning Course Sequencing System

[1]Daramola C. Y. and [2]Adewale O. S.
[1]Department of Computer Science, Federal University, Oye-Ekiti, Nigeria
[2]Department of Computer Science, Federal University of Technology, Akure, Nigeria
[1]daramola.bola@gmail.com; [2]adewale@futa.edu.ng

Corresponding Author's Email: daramola.bola@gmail.com

Article Info

Date Received: 22-08-2024
Date Accepted: 25-10-2024

Keywords:
Reinforcement learning, course sequencing, adaptive personalized learning, transition diagram.

ABSTRACT

In an era where personalised education is increasingly pivotal, the integration of adaptive learning technologies has emerged as a transformative force in the realm of e-learning. Traditional educational approaches often fail to cater to the diverse needs of individual learners, resulting in a one-size-fits-all model that leaves many underserved. Recognising these limitations, the development of a q-learning-based model introduces a sophisticated mechanism to tailor course content to each students unique learning style, preferences, and pace. By leveraging reinforcement learning techniques, this model dynamically adjusts the sequence of instructional material, enhancing engagement and optimising knowledge retention. The new system leverages reinforcement learning techniques to autonomously adapt to user behaviour, delivering tailored content aimed at fulfilling learning objectives based on the feedback received, whether affirmative or negative. Functioning as an intelligent agent, the system scrutinises user interactions and selects the most suitable responses to enhance the overall learning experience. The primary goal of this research is to create a dynamically adaptive e-learning system utilising reinforcement learning methodologies. The reinforcement learning algorithms entail making targeted decisions that yield varying rewards, with each knowledge component associated with a specific reward based on its relevance. These algorithms are grounded in the principles of Markov decision processes, which encompass a set of actions and the probabilities of transitioning between different states. Within this Markov decision process framework, both a reward function and a transition function are defined. The core function of the proposed system is to recommend learning pathways by concurrently considering sequential behaviour, learning styles, activities, materials, difficulty levels, feedback, preferences, competencies, and knowledge levels, employing the q-learning algorithm. The optimal path for the active learner in the course used for the implementation is $s_0 \rightarrow s_1 \rightarrow s_6 \rightarrow s_9$. The proposed system identifies the study trajectory favoured by learners for a particular course. The results demonstrated that after 200 iterations, the performance of the q-learning algorithm exceeded that recorded after 100 iterations. The success rate is 60.86% and 70.82& for 100 and 200 iterations respectively while the optimal course selection path training time is 10 and 8 for 100 and 200 iterations respectively.

1. INTRODUCTION

The proliferation of online courses has complicated the decision-making process for learners seeking to identify the most suitable options for their educational needs, which has, in turn, negatively impacted their learning outcomes. Recently, the development of personalised course recommendations has emerged as a significant area of research aimed at mitigating the challenges posed by information overload. The growing focus on customised and flexible learning experiences by educational institutions is expected to facilitate an increased integration of artificial intelligence (AI) within remote learning environments. AI-driven technologies will enable students to access and enrol in programs or courses from any location globally.

Various methodologies are employed to assess the e-learning behaviours of students, with particular attention to the adaptability of recommender systems. By incorporating time series data into the adaptive framework, the recommendation process can be improved by aligning the learning behaviours of the target learner with the academic performance and study patterns of comparable learners. The vast array of available resources has rendered the selection of appropriate educational materials from numerous academic tools increasingly challenging. One viable approach to address this issue is the implementation of a personalised recommender system based on reinforcement learning (RL). Such systems can alleviate the problem of information overload by delivering engaging content tailored to the user's preferences. Typically, recommendation algorithms utilise multiple data sources to suggest potential items to users. In real-world applications, these systems generate recommendations based on the history of user-item interactions and incorporate user feedback to refine their suggestions. Alternatively, the recommender system seeks to discern users' interests through their interactions and propose products that may resonate with them. The initial studies on recommendations primarily focus on developing content-based and collaborative filtering techniques to achieve this objective.

Historically, recommendation systems have utilised collaborative-based filtering techniques to derive implicit feedback that reflects a learner's preferences [4,7].

However, recent advancements in neural recommendation algorithms leveraging deep learning have outperformed these traditional methods [6,9]. One notable model is the neural attentive session-based RS, which emulates users' sequential behaviours and deduces their primary objectives from learning patterns [8]. Furthermore, the foundational recommendation system, designed to reduce irrelevant courses, is based on an attention network and a profile reviser, both developed simultaneously through a hierarchical RL approach [13]. Nonetheless, the effectiveness of course recommendations can be enhanced when students are enrolled in multiple courses, as hierarchical RL often overlooks the explicit needs and implicit preferences of students, potentially resulting in inadequate recommendation outcomes. While these techniques can provide course recommendations to a degree, they commonly fail to account for the evolving preferences of users throughout their sequential learning experiences. Additionally, they may not accurately reflect a user's preferences for specific content, particularly when these preferences shift over time across various courses. Consequently, these methods struggle to deliver the necessary adaptability in recommendation systems, especially in monitoring the dynamic changes in users' preferences. Traditional recommendation systems also contend with the issue of data sparsity in practical applications, where only a limited selection of course materials appears in a user's list of highly-rated or studied courses. To effectively retrieve learning materials that align with learners' interests and preferences, it is essential to explore all potential candidate courses. Sequential recommendations seek to predict users' future choices based on their historical interaction data. Markov chains serve as an effective tool for modelling sequential behaviours. A specific variant of the Markov chain, known as the Markov Decision Process (MDP), offers a mathematical structure for modelling decision-making situations.

This paper introduces a tailored adaptive and sequential path recommendation model for e-learning, utilising RL in conjunction with MDP techniques to tackle the previously mentioned challenges. A significant obstacle in the educational process involves the need to modify various components, including reading materials, listening activities, quizzes, assignments, entertainment, and gaming, to reflect potential shifts in learners' states and preferences, while also considering their prior educational experiences. RL is frequently employed to create recommender systems in contexts where user behaviour is subject to change. Consequently, the implementation of an RL agent proves advantageous in these situations, as it continuously adapts through its interactions with the learning environment. These agents are designed to adjust information dynamically in response to user preferences and temporal variations, while also regularly updating online course recommendations. Thus, this paper proposes a method that dynamically assembles adaptive online learning courses through the q-learning algorithm, a specific reinforcement learning technique. This algorithm is informed by learner behaviour and delivers course content based on both positive and negative feedback from learners, aiming to fulfil the established learning objectives. The implementation of q-learning in adaptive course sequencing hinges on identifying optimal paths for learners based on their unique interactions with the content. This process involves analysing various data points, such as prior knowledge, learning pace, and engagement levels, to formulate a state-action value function that can predict the success of specific learning sequences. Each learners' progress informs the algorithm, allowing for dynamic adjustments that cater to individual strengths and weaknesses, thus enhancing the overall learning experience. By employing reward systems that reflect the achievement of learning milestones, the algorithm continually refines its strategies, optimizing course delivery over time while encouraging deeper learning engagement. As a result of these adaptive mechanisms, the educational path becomes increasingly personalised, fostering a supportive environment that can effectively respond to diverse learner needs and preferences [10]. Hence, the potential of q-learning lies in its ability to create a tailored educational journey that evolves alongside the learner.

2. Q-learning

Q-learning is a prominent reinforcement learning algorithm that is model-free and off-policy. It is widely utilised in various studies related to reinforcement learning. The foundation of q-learning is rooted in the Bellman equation, and it is represented as follows.

$$V_\pi(s) = \sum_a \pi(s,a) \sum_{s'} p(s'|\mathrm{s},\mathrm{a})\left(W_{s\to s'|\mathrm{a}} + \gamma V_\pi(s')\right) \quad 1$$

$$\varphi_\pi(s,a) = \sum_{s'} p(s'|\mathrm{s},\mathrm{a})\left(\&W_{s\to s'|\mathrm{a}} + \gamma \sum_{s'} \pi(s',a')\varphi_\pi(s',a')\right) \quad 2$$

The Bellman equation is a fundamental concept in dynamic programming which is as follows

$$W_{s\to s'|\mathrm{a}} = \mathrm{E}[r_{t+1}|s_t = s, a_t = a, s_{t+1} = s'] \quad 3$$

The function $V_\pi(s)$ denotes the state value, while $\varphi_\pi(s,a)$ represents the action value. Since the transition from state *s* to the next state *s+1* is uncertain, the equation requires the inclusion of the expectation *E*, with s denoting the state and *r* denoting the reward. The q-learning algorithm selects the policy based on the Q-table organised as *S*A*, with *S* representing the state and A representing the action. The Q-table facilitates the identification of the subsequent action by evaluating the present state of the environment. After determining the action, the agent executes it, and upon its completion, obtains a reward from the environment. After each action is taken, the Q-table is updated within the environment, and the modification of the Q-table is executed in accordance with equation 4.

$$\varphi(s_t,a_t) \leftarrow \varphi(s_t,a_t) + \alpha\left[r + \gamma \max_{a_t} \varphi(s_{t+1},a_t) - \varphi(s_t,a_t)\right] \quad 4$$

The formula consists of variables where *s* denotes the state, a denotes the action, r denotes the reward, α denotes the

learning rate, and γ denotes the discount factor. Both α and γ have values ranging from 0 to 1.

3. Related work

Ronald Howard is recognized for his foundational contributions to the concept of instructional sequencing, which involves the strategic organization of various educational activities through RL to enhance student outcomes. Researchers in this field have extensively examined the impact of instructional order on learning effectiveness. For instance, Atkinson (1972) utilized reinforcement learning to create a mathematical model aimed at optimizing instructional sequences. In a related vein, Atkinson [2] proposed an educational framework comprising four critical elements: modelling the learning process, specifying acceptable behaviours, establishing objectives, and creating a measurement system to assign values to actions and rewards associated with achieving these objectives. These components can be linked to the Markov Decision Process (MDP) within the context of instructional theory. Atkinson [2] further clarified that the transition function corresponds with the learning process model in instructional settings, where the states of students are analogous to the states in an MDP. Instructional activities are viewed as actions that can be tailored according to the cognitive states of students, which may include tools such as flashcards, problem-solving tasks, worked examples, exercises, and levels in educational games. Additionally, each instructional action can be linked to a specific cost, which should be incorporated into the reward function.

In the field of instructional sequencing within intelligent tutoring systems, two predominant strategies can be identified: task loop adaptivity and step loop adaptivity [11,12]. Task loop adaptivity pertains to the RL agent's ability to select various instructional activities, whereas step loop adaptivity involves the RL agent's decision-making regarding the particulars of each step within a predetermined instructional activity. An example of step loop adaptivity is the choice between revealing the solution to the next step or prompting the student to solve it independently, as noted by Chi *et al*. [5]. Moreover, the implementation of adaptive learning in online education not only has the potential to enhance student outcomes but also alleviates the workload for educators, course designers, and learners. A pioneering study by Bassen *et al*. [3] introduced the first RL model designed to dynamically organise learning activities for a large-scale online course through active learning methodologies. This model minimises the number of tasks assigned while optimising the course activities sequence to enhance student performance. A thorough investigation was carried out with more than a thousand participants to assess the effects of this scheduling policy on student feedback, dropout rates, and learning outcomes. The results revealed that the reinforcement learning model produced results similar to those of a self-directed learning approach, but with a reduced number of activities and lower dropout rates. Furthermore, it demonstrated superior learning gains compared to a traditional linear assignment framework.

4. Personalised adaptive e-learning and sequential learning path model

4.1 The model

The personalised adaptive e-learning and sequential learning path system is depicted in Figure 1. The personalised adaptive sequential learning leverages RL to recommend the optimal sequence of tasks for each learner, thereby enhancing educational outcomes and reducing feelings of dissatisfaction and disengagement.

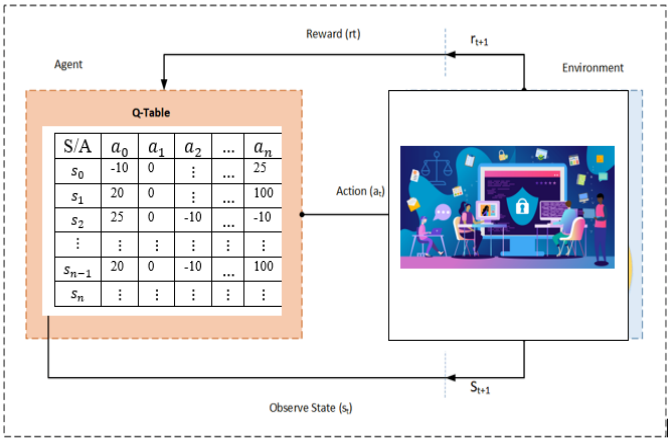

Figure 1: Q-learning for the proposed adaptive learning sequence

From Figure 1, the learner is represented as the Agent. The student engages with the system throughout various processes, thereby aligning the student with the agent in the context of reinforcement learning algorithms. The agent's role is to select the subsequent content to be presented from an e-learning repository. The Environment denotes the learning context. Actions (A) encompass the recommendation of new tutorials, reading materials, assignments, quizzes, exams, or advertisements. States refer to the interaction characteristics of the learner, with the state-value serving as an indicator of the quality of the current state, such as the learner's progress. The state (S) represents the condition to which the environment returns following the agent's action. Consequently, the state reflects the learner's learning condition, illustrating the extent of learner's acquisition of knowledge. Data is organised in a vector format, with all state values ranging from 0 to 1. A state value of 1 for a student indicates complete mastery and comprehension of the material, whereas a value of 0 signifies a lack of mastery. Rewards serve as feedback mechanisms from the environment to the agent following an action, signifying whether the agent's decision was advantageous within that specific context. Positive incentives are associated with behaviours such as viewing class videos or completing exams, while negative rewards may arise from actions like exiting the platform, engaging in gaming, or exhibiting signs of disinterest. The reinforcement learning relies on the framework of MDPs, which is defined by a set of actions and state transition probabilities. The MDP model includes a reward function (R) and a transition function (T) as follows.

$$T: SxAxS \rightarrow [0,1]$$
$$R: SXAXS \rightarrow \mathbb{R}$$

In this model, states are represented as S ∈ {0, 1, 2, 3, …, 10}, with each state reflecting the learner's current position within the educational process. Rewards are assigned to each state-action pair according to the particular problem and learning environment. The Markov property dictates that the agent's focus is solely on the current state of the process, disregarding the entire history. This property is mathematically expressed in Equation 1.

$$P(s_{t+1}|s_0, s_1, \dots, s_t, a_t) = P(s_{t+1}|s_t, a_t) \quad 5$$

where *P* represents the probability of a state transition, s denotes the state, *a* signifies the action, and *t* indicates time. During each epoch, the agent takes an action that alters its environment and results in a reward. Value functions and the optimal policy are proposed as additional computational techniques for determining the reward value. This approach offers a mathematical foundation for simulating decision-making processes in scenarios where an individual's decisions and random variables interact to affect the outcome. In accordance with the Markov property, only the present state has an impact on future states, while past states have no influence. The Markov chain is a probabilistic approach where future states are conditionally independent of past states and are solely dependent on the current state by the current state solely determined by the transition probability in the process of moving from one state to another. A group of states exhibiting the Markov property, denoted as $S_1, S_2, \dots, S_n$, is referred to as a process within this proposed model. The transition function P, representing the probability of transitioning from one state to another, along with the state S, are the two key parameters used for its definition. The accumulation of rewards in a Markov process is defined as a Markov reward process, formulated with state *S*, transition function *P*, reward *R*, and discount factor *γ*. The discount factor elucidates how rewards in the future are valued when considering a reward in the present. A *γ* value of 0 indicates that the agent only considers the immediate reward, while a *γ* value of 1 signifies that the agent takes into account all potential future rewards. In the context of Markov decision processes, a state *S*, transition function *P*, reward *R*, discount factor *γ*, and a set of actions *a* collectively form its representation. An MDP plays a dynamic interaction between an agent and its environment. The environment responds to the agent's specific actions by providing rewards and altering its state. Only the preceding state and action influence the subsequent state and reward.

The state of the system is represented by the learner's interaction features, with the state-value v(S) serving as a metric to assess the quality of the current state. The transition probability quantifies the likelihood of the agent transitioning between states. This probability can be mathematically expressed as shown in Equation 6.

$$P(S_{t+1}|S_{t0}) = P(S_{t+1}|S_1, S_2, S_t) \quad 6$$

The current condition of the agent is denoted by S_t, while the subsequent condition is represented by S_{t+1}. As per this equation, the transition from state S_t to S_{t+1} is entirely unaffected by the preceding state.

The symbol *P* is used to denote transition probability. Consequently, if the model exhibits the Markov property, the right side of the equation holds the same significance as the left side. It can be deduced logically that the present state retains information about past states. The probability of state transition from a Markov State at S_t to S_{t+1}, or any other subsequent state, is shown in Equation 7.

$$\rho_{ss'} = P[S_{t+1} = s'|S_t = s] \quad 7$$

The probabilities of transitioning between states are illustrated using a matrix known as the state transition probability matrix is presented as follows:

$$P = \begin{bmatrix} p_{11} & p_{12} \cdots & p_{1n} \\ p_{21} & p_{22} \cdots & p_{2n} \\ \vdots & \vdots & \vdots \\ p_{n1} & p_{n2} & p_{nn} \end{bmatrix} \quad 8$$

It is worth mentioning that $p_{ij} \geq 0$, and this holds true for all i, as indicated in Equation 9.

$$\sum_{k=1}^{r} p_{ik} = \sum_{k=1}^{r} P(S_{m+1} = k|S_m = i) \quad 9$$

$$\sum_{k=1}^{r} p_{ik} = 1 \quad 10$$

The matrix displays the likelihood of moving from the starting state *i* to any other state *k*. The sum of each row in the transition matrix is denoted as P. The total of each row amounts to 1.

Q-learning leverages its past and future actions to learn from previous experiences and make optimal decisions. In this scenario, all transitions and associated actions from one state to another are feasible. However, the learner can assign positive or negative rewards to each action using the Q-table matrix. The Q-table displays the reward of transitioning from one state to another (action). The rows of the Q-table represent states (learner's features), while the columns represent actions. These actions encompass measurable positive effects, such as increased engagement in educational activities like watching tutorials, completing assignments, taking exams, reading, writing, and so on. In such scenarios, positive actions are rewarded in proportion to the positive outcomes they produce. Conversely, actions leading to negative outcomes, such as idleness, or using social media while studying, are penalised with negative rewards, resulting in a reduction in study time. Therefore, even if an action yields positive results, if it also increases social media usage or engagement in games or entertainment, the reward may be diminished due to the accompanying negative consequences. The components utilised in q-learning are defined by Equation 11.

$$\varphi(s_t, a_t) = \varphi(s_t, a_t) + \alpha * \left[R_t + \gamma \max_a \varphi'(s_{t+1}, a_{t+1}) - \varphi(s_t, a_t)\right] \quad 11$$

α represents the learning rate ($0 \leq \alpha \leq 1$); R_t represents the observed reward, s_{t+1}represents the new state, $\gamma < 1$ represents a discounted factor applied to the future rewards that are obtained because of the selected action. The maximum reward that the system can calculate by executing some future action in the state s_{t+1} is approximated as $\varphi(s_{t+1}, a_{t+1})$. The proposed model demonstrates improved effectiveness and has the potential to improve the learning environment for sequential path recommendation within the educational framework.

4.2 Performance evaluation metrics

When evaluating reinforcement learning algorithms, a range of metrics are utilised to gain insights into different facets of the algorithm's performance. In this paper, the following metrics were used: time of training (s), reward (mean, minimum and maximum), standard deviation, action taken (optimum), success rate (average in percentage), average step range (average), number of times of training for optimal course selection path; and cumulative (reward). Each of these metrics offers a unique perspective on evaluating the performance of a reinforcement learning algorithm, with the significance of each varying based on the specific application and objectives of the task.

5. Implementation

In this paper, states are denoted as S ∈ {0, 1, 2, 3, …, 10}, with each state representing the learner's position in the learning process. Rewards are then allocated to each state-action pair based on the specific problem and learning context. The proposed states and their corresponding rewards are shown in Table 1 while the state diagram of the personalised adaptive learning with learning path sequence is shown in Figure 2.

Table 1: Proposed states, actions, and rewards

States	Actions & reward to each action
S_0 – begin	
S_1 – studying	a_1: remain on state → 10
S_2 – hypermedia lessons	a_2: last course → 20
S_3 – entertaining	a_3: supplementary content → 50
S_4 – bored or frustrated	a_4: quiz/assignment → 60
S_5 – sleeping/resting	a_5: exam →100
S_6 – writing	a_6: high-level course →100
S_7 – game playing	a_7: low-level course →70
S_8 – clicking ad on	a_8: social media -→-10
S_9 – completion of course	
S_{10} – quit study	

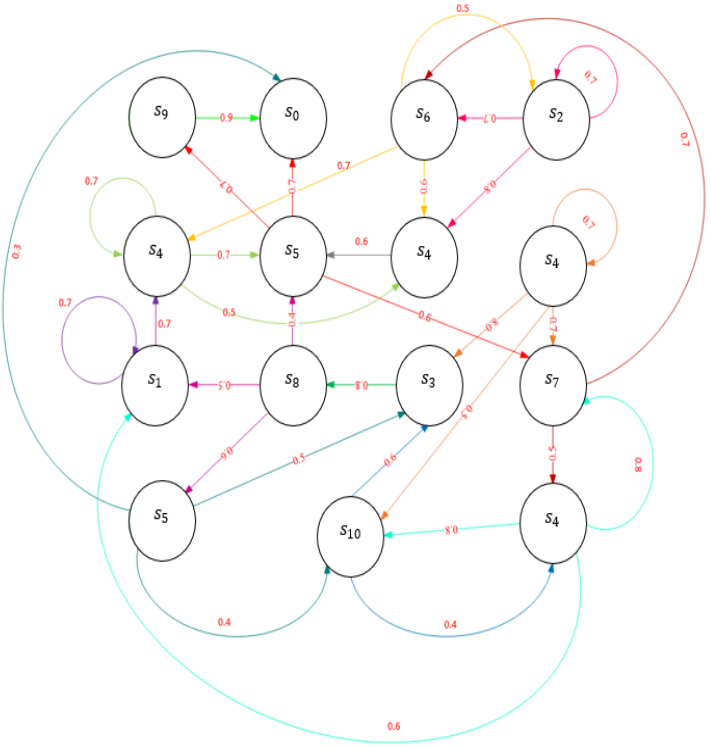

Figure 2: A personalised adaptive learning with learning path sequence

Higher rewards serve to incentivise the agent to prioritise specific actions or states that hold greater value and appeal. On the other hand, average rewards are employed to provide moderate incentives for actions that are generally positive but not essential. By offering medium rewards, the agent can explore and develop a balanced policy without showing excessive bias towards or neglecting certain actions. Negative or lower rewards, on the other hand, can aid the agent in learning to disregard actions that lead to unfavourable outcomes or steer learning in the wrong direction. The issue at hand is addressed through the utilisation of one-shot policy recommendations for modelling learning path sequence recommendation and personalised learning. The actions in the proposed system include suggesting the next course, video, game, or advertisement, proposing future influences for personalised learning, enhancing recommendations for suitable content, influencing future learning decisions, and striving to maximise learner satisfaction while minimising interactions to facilitate learning based on the learner's performance characteristics, ultimately designing personalized adaptive pathways to reduce negative experiences. The reward system is established with a cap set at 100. If the learning process continues without any breaks throughout all cycles, the maximum reward achievable is 100 as shown in the output matrix *P*. The initial probabilities are based on this matrix. The columns of the matrix correspond to different actions, while the rows represent various states. By setting the feedback value at 0.75 and conducting 100 and 200 iterations, the matrix outlining the optimal learning trajectory was obtained and the necessary rewards. *P* displays the reward distribution for the assumed state-action pairings in equation 12

$$P = \begin{array}{c} \\ S_0 \\ S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \\ S_9 \\ S_{10} \end{array} \begin{array}{c} \begin{array}{cccccccc} A_1 & A_2 & A_3 & A_4 & A_5 & A_6 & A_7 & A_8 \end{array} \\ \begin{pmatrix} 10 & 0 & 50 & 0 & 0 & 0 & 70 & -10 \\ 10 & 0 & 50 & 60 & 100 & 0 & 0 & 0 \\ 10 & 0 & 50 & 0 & 0 & 0 & 0 & -10 \\ 0 & 20 & 0 & 60 & 100 & 0 & 0 & -10 \\ 0 & 20 & 50 & 60 & 100 & 100 & 70 & 0 \\ 0 & 20 & 50 & 60 & 0 & 0 & 0 & -10 \\ 10 & 0 & 50 & 0 & 0 & 0 & 0 & 0 \\ 10 & 0 & 50 & 60 & 100 & 100 & 70 & 0 \\ 0 & 20 & 0 & 60 & 0 & 0 & 0 & -10 \\ 0 & 20 & 0 & 60 & 0 & 0 & 70 & 0 \\ 10 & 20 & 50 & 60 & 100 & 100 & 70 & -10 \end{pmatrix} \end{array} \quad 12$$

The q-learning algorithm undergoes training for 100 and 200 iterations. The parameters' formulation is as shown in Table 2. The training process of the q-learning algorithm involves the utilisation of the Belman approach. Epsilon was adjusted during training to find a suitable equilibrium between exploration and exploitation. To gradually shift from complete exploration to exploitation, the initial epsilon value was set at 1 (representing pure exploration) and decreased to 0.8 with each episode as it moves from exploration towards exploitation.

Table 2: parameters' settings

Parameter	Value
Learning_rate: α	0.5
Discount_factor: γ	0.8
States: S	11
Actions: a	8
Total_episodes	100/200
Minimum_iteration	100

The implementation was carried out using the Python programming language. The resulting simulations were subsequently utilised to evaluate established policies and juxtapose them against possible alternatives. Through simulations, an optimal sequence of actions was obtained that initiates from state s_0 and progresses through states s_1, s_5, and s_6 (Figure 2). By examining the definitions linked to these states' labels, we can grasp the significance of the outcome. Essentially, the transition was made from viewing video lessons (s_2) to completing the course (s_9), passing through states s_1, s_4, and s_6. This trajectory reflects an emphasis on study policy in e-learning initially, followed by a proactive approach to prevent study abandonment. It is conceivable that the recommended decision-making strategy has been altered to expedite problem resolution. This scenario mirrors real-life situations where initial expectations of learning challenges resolving naturally may not materialise, prompting the implementation of optimal actions to mitigate negative experiences and disengagement from learning.

200 students were considered for simulation with a learning rate of 20 content pieces. The system was randomly generated with 20 content pieces and 200 students distributed randomly across them. The rewards value over 100 iterations was reported in Table 3. The optimal reward in q-learning signifies the highest attainable reward within a given environment, reflecting the reward that an agent would obtain by consistently choosing the best action in every state. The best action is defined as the one that maximises the Q-value associated with a specific state. A Q-table in Table 4 was trained across 100 iterations, illustrating the value associated with each state-action pair. Furthermore, the results of the q-learning simulations are detailed in Table 5 Q-table (trained) and Table 6 (rewards (generated)) over a span of 200 iterations. The maximum reward obtained during this period was 77.63.

Table 3: Reward (100 iterations)

State/Action	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
s_0	0	0	0	0	58.52	0	61.35	0
s_1	0	44.59	0	4.87	0	50.98	58.31	0
s_2	0	45.35	47.55	0	0	14.31	65.92	59.12
s_3	0	43.83	0	0	39.12	38.8	6.09	54.49
s_4	0	0	32.71	0	56.99	40.19	59.83	0
s_5	36.5	0	0	0	40.64	41.85	44.69	56.08
s_6	0	0	61.25	45.11	32.15	0	30.99	72.88
s_7	0	6.09	44.89	45.11	47.86	66.21	73.53	68.2
s_8	0	26.06	0	0	47.86	0	0	68.19
s_9	0	0	61.25	0	47.86	52.47	44.69	56.13
s_{10}	0	27.59	37.99	6.39	0	0	0	74.05

Table 4: Trained Q-table of state-action pair (100 iterations)

State/Action	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
s_0	369.1	0	0	0	450.75	0	159.8	0
s_1	0	235.04	0	187.15	0	267.24	455.8	0
s_2	0	0	0	177.83	99.89	0	0	239.29
s_3	0	131.24	233.93	0	315.79	0	0	222.29
s_4	364.64	22.1	0	0	0	489.07	0	0
s_5	22.1	114.24	0	0	42.5	353.07	398.83	222.3
s_6	0	131.24	276.43	186.33	91.39	489.07	322.33	569.75
s_7	0	398.64	276.43	35.7	91.39	489.07	540.8	267.24
s_8	0	148.24	199.93	0	91.39	404.07	0	267.24
s_9	0	0	199.93	35.7	91.39	0	227.8	0
s_{10}	0	156.74	361.43	0	91.39	361.25	227.8	0

Table 5: Rewards (200 iterations)

State/Action	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
s_0	58.18	0	0	0	71.52	0	68.9707	61.38105
s_1	0	59.79	0	57.48	0	71.73	68	0
s_2	0	60.27	59.21	57.49	65.75	64.99	72.81	67.67
s_3	0	58.43	63.06	0	68.63	65.07	71.85	65.75
s_4	58.18	57.38	60.18	0	70.56	63.65	68.96	0
s_5	58.18	57.38	0	0	69.6	80.98	77.63	65.75
s_6	0	58.25	59.31	58.44	64.79	65.95	68.96	85
s_7	0	60.36	67.88	58.44	64.79	80.98	76.25	74.41
s_8	0	61.23	59.21	57.48	64.79	71.73	77.63	74.41
s_9	56.98	59.1	60.16	57.98	63.92	81.13	68.26	76.38
s_{10}	0	61.09	67.75	58.38	64.92	81.14	77.56	74.48

Table 6. Trained Q-table of state-action pair (200 iterations)

State/Action	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
s_0	518.03	0	0	0	636.91	0	621.7	0
s_1	0	530.66	0	510.96	0	636.76	604.7	0
s_2	0	534.91	526.26	510.96	585.91	577.26	647.2	602.88
s_3	0	526.38	560.26	0	611.41	568.76	638.7	585.88
s_4	518	509.41	534.76	0	628.41	721.76	613.2	0
s_5	518	509.38	0	0	619.88	585.76	689.7	585.88
s_6	0	526.41	601.06	519.46	577.41	721.72	613.2	755.88
s_7	0	543.38	602.76	519.46	577.41	721.72	689.7	662.38
s_8	543.41	526.15	510.96	577.41	636.76	689.7	662.38	0
s_9	517.92	509.25	0	0	594.36	577.24	681.12	585.94
s_{10}	0	542.9	602.13	518.83	576.9	721.13	689.32	661.9

The Q-table was obtained as a post-training output. The results of the implementation as depicted in Tables 3-6, showcasing the potential decisions the learner can make in a given situation. A value of 0 signifies that the state remains unaffected by the action taken. It is advisable to engage in the activity when the value is low rather than pursuing alternative actions. The optimal route for the engaged learner is $s_0 \rightarrow s_1 \rightarrow s_6 \rightarrow s_9$ (Figure 2). The proposed system reveals the student's preferred study sequence for a particular course. Additionally, the q-learning approach takes into account the learner's preferences, and level of knowledge when providing

recommendations. Figure 2 depicts the scenario of learning paths. The red words denote the actions the agent can take depending on the learner's condition, while the circles reflect the states in which the learner can exist. For instance, the algorithm might recommend videos or interactive visual materials to a learner who has a preference for visual learning methods. Should a learner encounter difficulty with a specific idea, the methodology may suggest supplementary materials or exercises to strengthen that particular concept.

The statistical results of q-learning's performance in recommending learning path sequence over a 100 and 200 iteration are presented in Table 7. The Table indicates a mean reward of 4.5, showcasing some variability in rewards due to the learner's position and a decline in rewards at the start of each iteration. Consequently, the model takes longer to make recommendations with varying rewards (each move incurring a -1 point penalty). The performance metrics of q-learning improved with an increase in the number of iterations, as evidenced in Tables 3-6. For instance, q-learning excelled over 200 iterations compared to 100 iterations, with the latter showing lower performance. Despite the longer time of training, q-learning with 100 iterations consistently outperformed the 200 iterations in this aspect. Furthermore, the performance indicators of q-learning with 200 iterations surpassed those of 100 iterations, albeit with slightly longer training times than the former. The proposed method demonstrated the fewest average running times over 200 iterations and achieved a higher success rate (average) compared to q-learning over 100 iterations.

Table 7: Performance of the proposed system

Evaluation metrics	100 episodes	200 episodes
Time for training (s)	5.13	5.21
Reward (mean)	4.5	5.0
Standard deviation	2.12	2.16
Reward (Minimum)	2.0	2.0
Reward (Maximum)	7.0	7.0
Action taken (optimal)	3	3
Success rate (average) (%)	60.86	70.82
Action step range (average)	12.5	13.2
Optimal course selection path training times	10	8
Reward (Best)	9400.4	22704.01

6. Future research direction

Future initiatives will involve a diverse array of actions tailored to each state, with the flexibility to include as many states as necessary to determine the most effective approach for each learner. A significant challenge arises from the vast spectrum of possible states or action values associated with the state. In particular, the implementation of the strategy in an online environment utilising traditional reinforcement learning may lead to challenges associated with complexity and convergence. Regarding prospective research directions, several gaps have been identified: Firstly, to mitigate issues concerning complexity, convergence, and model efficiency, the application of deep Q-learning may offer a promising alternative; secondly, traditional RL techniques face various obstacles, notably the potential for algorithmic inefficiency when dealing with a large action space, as the algorithm assesses all actions simultaneously. To tackle this issue, the deep deterministic policy gradient method emerges as a viable approach. Furthermore, future research should strive to include a broader spectrum of states and actions in order to identify optimal learning pathways that correspond with the learner's adaptive sequential behaviours, learning preferences, activities, varied educational resources, customisable difficulty settings, tailored feedback, individual preferences, competencies, and levels of knowledge. This objective will be pursued through the implementation of multi-agent RL strategies to facilitate recommendations.

References

1. Atkinson, R.C (1972). Ingredients for a theory of instruction. Am. Psychol., 27, 921.
2. Atkinson, R.C. (1972a) Optimizing the learning of a second-language vocabulary. J. Exp. Psychol., 96-124.
3. Bassen, J.; Balaji, B.; Schaarschmidt, M.; Thille, C.; Painter, J.; Zimmaro, D.; Games, A.; Fast, E.; Mitchell, J. C. (2020), Reinforcement learning for the adaptive scheduling of educational activities. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, 25–30 April; pp. 1–12.
4. Bousbahi F. and Chorfi H. (2015), MOOC-Rec: A case-based recommender system for MOOCs, Proc. Social Behav. Sci., 195:1813–1822, doi: 10.1016/j.sbspro.2015.06.395
5. Chi, M.; Jordan, P. W.; Vanlehn, K.; Litman, D. J. (2009), To elicit or to tell: Does it matter? In Proceedings of the Aied, Brighton, UK, 6-10 July; pp. 197-204.
6. He X., He Z., Song J., Liu Z., Jiang Y.-G., and Chua T.-S. (2018), 'NAIS: Neural attentive item similarity model for recommendation,'' IEEE Trans. Knowl. Data Eng., 30(12):2354–2366, Dec. 2018, doi: 10.1109/TKDE.2018.2831682.
7. Jing X. and J. Tang J. (2017), Guess you like: Course recommendation in MOOCs, in Proc. Int. Conf. Web Intell., Aug. 2017, pp. 783–789.
8. Li J., Ren P., Chen Z., Ren Z., Lian T., and Ma J. (2017), Neural attentive session-based recommendation, in Proc. ACM Conf. Inf. Knowl. Manage., pp. 1419–1428.
9. Li X., Tang J., Wang T., Zhang Y., and Chen H. (2020), Improving deep item-based collaborative filtering with Bayesian personalised ranking for MOOC course recommendation, in Proc. Int. Conf. Knowl. Sci., Eng. Manage., pp. 247–258.
10. Ryoo J., Winkelmann K. (2021), "Innovative Learning Environments in STEM Higher Education", Springer Nature
11. VanLehn, K. The behavior of tutoring systems. Int. J. Artif. Intell. Educ. 2006, 16, 227–265.
12. VanLehn, K. Regulative loops, step loops and task loops. Int. J. Artif. Intell. Educ. 2016, 26, 107–112.

13. Zhang J., Hao B., Chen B., Li C., Chen H., and Sun J. (2019), Hierarchical reinforcement learning for course recommendation in MOOCs, in Proc. AAAI Conf. Artif. Intell., 33(1): 435–442.