International Journal on Advanced Science Engineering Information Technology

Vol.12 (2022) No. 6
ISSN: 2088-5334

Incremental Knowledge-based System for Recommending Content Adaptation in Dynamic Learning Environment

Alva Hendi Muhammad [a,*], Dhani Ariatmanto [a], Yuhefizar [b]

[a] *Magister of Informatics Engineering, University of Amikom Yogyakarta, Yogyakarta, 55128, Indonesia*
[b] *Department of Information Technology, Politeknik Negeri Padang, Padang, 25166, Indonesia*
*Corresponding author: *alva@amikom.ac.id*

***Abstract*—Dynamic learning environment (DLE) provides an opportunity for students with a remarkable learning experience in a limited time and specific situations. In this condition, adaptation and personalization have been key issues to accommodate differences between students. Both paradigms emphasize tailoring learning activities to students' understanding and interest through learning objectives, instructional approaches, and learning pathways. In addition, the students will learn optimized instructional activities at their own pace. This paper presents an incremental knowledge-based system to facilitate learning content adaptation in DLE. To be specific, the knowledge base contains a set of rules incrementally constructed using Ripple Down Rules (RDR) after evaluating a series of test cases. The test cases are generated automatically by analyzing the attributes that reflect the learning situation. Since it is impossible to perform thorough testing involving all input parameters, the selection criteria using pairwise testing are applied to minimize the refinement. Therefore, the evaluation of the theoretical concept is then carried out on a real case. The selected case study for the analysis is the subject of Computer Networking for an undergraduate course. Several adaptive scenarios are presented based on some criteria. An education expert is involved in recommending suitable content for adaptation during the evaluation phase. However, the knowledge base development is automatically constructed from the incremental knowledge acquisition process. As the evaluation progresses, the knowledge base is validated for its accuracy in predicting learning content recommendations.**

***Keywords*— Incremental knowledge-based system; adaptive learning system; learning content recommendation; dynamic learning environment; ripple down rule.**

Manuscript received 16 Mar. 2021; revised 23 Nov. 2021; accepted 19 Aug. 2022. Date of publication 31 Dec. 2022.

I. INTRODUCTION

Adaptation and personalization are increasingly important areas in a dynamic learning environment (DLE). The learning characteristics of DLE that constantly change over time have transformed the way students learn and progress. Unlike traditional e-learning system where provides similar content to all student (*one-size-fits-all*), learning in DLE offer benefits where the system will decide the best content for the student during the time of learning [1], [2]. It also outlines that anytime students interact with e-learning, the stored knowledge in students' minds will be recalled, and e-learning responsible for recommending the next activity based on the current situation assessment. E-learning systems that adapt the dynamic learning approach are concerned about providing the student's learning context and recommending the learning resources suitable in that context [3], [4]. The fundamental of adaptation in DLE is that the e-learning system must have the capability to understand the students' situation or background (context) and immediately suggest a new activity for learning [5], [6]. As the context changes every time, the student learning progress must be stored for continuous learning improvement.

Understanding the complexity of modeling the students in dynamic learning is a key point that leads to learning recommendations. The student can be modeled as a user profile that interacts with e-learning. Thus, the student produces many data that the system can utilize by adapting the content that represents the user. The data collected from a student includes anything from prior knowledge to personal traits and preferences [2]. The learner or student modeling literature has highlighted several recommendations of learner's attributes that accomplish each other [7]–[9]. However, approaching student modeling has always been an issue to change along with a new pedagogical and technological opportunities. During the initial appearance of adaptive learning, several user models had proposed to interact with the environment. The initial adaptive learning system delivers customized services by tailoring the user's knowledge, interest, goals and tasks, background, individual traits, and context of work [10], [11]. The trend has been a shift towards mobile and ubiquitous learning. Context acquisition and representation have been an issue with devices and environments [5], [12]. Furthermore, the MOOC's era has stepped further in user modeling by taking into account user attributes correlated with system interaction, such as clickstream data, forum posts, learning activities, and all data related to learners' persistence and engagement [8], [13], [14].

Although adaptive mechanisms can be achieved easily through identifying a learner's knowledge and characteristics, the approaches show subtle changes over time. Traditionally, the adaptation model uses a straightforward '*diagnose and prescript*' to recommend possible learning content [15]. This model sees a learner as unchanged attributes through course interaction, and adaptation occurs only once during learning. Thus, the adaptation engine commonly applied a set of rules to determine learner attributes recommendations [4], [16]. The second model extends the previous model by capturing the learning context, analyzing them, and presenting the learning recommendation [15]. The model considers course interaction as a dynamic condition that constantly changes during time. The adaptation in this condition takes into consideration both the static (background) and dynamic learner's attributes (situation) [5]. The approaches in this model often utilized a machine learning or artificial intelligence algorithm to closely monitor student interaction, gather and analyze data, and discover patterns to deliver adaptive content to the learner [11], [17]. Each model has its advantages and drawbacks. Each model has its advantages and drawbacks. Implementing the first model has shown an advantage when an absence or missing data exists in the system. In contrast, the trade-off is when the student's data is always available, the second model demonstrated its ability to analyze and provide content recommendations flawlessly.

This paper addresses the challenge of how to present the content adaptation in a short and dynamic learning environment by modeling the learning situation in a series of test cases. The dynamic environment that regularly changes is the main obstacle when presenting adaptive learning. This paper promotes the use of knowledge acquisition ripple-down rules to assess the student's context during learning. The experiment emphasizes the test case generation and construction of the rule to perform content adaptation. This approach enables time to develop a knowledge base by reusing knowledge from the available test cases. The simple cases are presented in the system, and then the expert will decide the content that tailors those cases. The decisions from the expert are stored in the knowledge base as the heart of the rule-based system. The construction of the rules is administered automatically by the ripple-down rules algorithm. This knowledge acquisition approach has been successful in maintaining large-scale rule-based systems, such as in a massive corpus [18], fraud analysis bank dataset [19], and data-centric for IoT devices [20]. Our designed tool can store and share the knowledge base from the presented case among similar systems.

The remainder of this paper is structured as follows: The material and methodology are presented in the following section, the results and conclusions are shown in the third section, followed by a discussion of potential future research.

II. Material and Method

We implemented a design science research methodology that involved a systematic investigation in identifying the problem and producing knowledge for the design of artifacts [21], [22]. Artifacts have different levels of abstraction, from instantiations (level 1) to design theory (level 3) [23]. This paper focuses on the midrange level 2, which involves the process of knowledge as design principles. As illustrated in Figure 1, our research process will focus on the following three aspects:

- Stage 1: Gather student behavior information in a dynamic learning system. The student and domain models' learning contexts were formed into an adaptation model.
- Stage 2: Design a knowledge base through iterative case refinement in a real subject of study.
- Stage 3: Evaluate the number of rules created and system accuracy results.

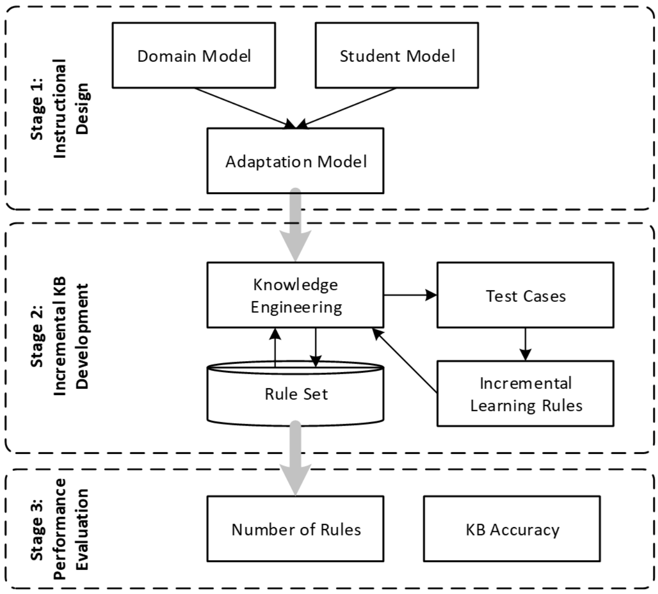

Fig. 1 Overview of Research Design Process

A. Adaptive Learning Model

The abstraction of the model is conducted through a case study in AES. The subject selected for the case study is the Computer Networking Fundamental for undergraduate students in Computer Science-related courses. The domain model is organized into three abstraction levels: learning goal, concept, and content. As seen in Figure 2, the subject consists of 10 modules. Each module contains several chapters and lessons. The smallest unit of learning that interacts with the learner is the lesson. It has several activities, from a set of theory, tutorial, simulation, and assignment. The lesson also has various presentations, such as the text, recorded audio or video, an interactive animation, and PDF documents.

The teaching strategy outlined in this paper is designed so that users can complete the learning without accessing all of the lessons. The instructional approach in the case study is constructed such that students can finish the subject without taking all available lessons. For this purpose, a new adaptation strategy is required. The module classification must be organized depending on the theme of the topics. Regardless of difficulty level, some modules have been placed similarly according to their topic. As shown in Figure 3, the ten modules are divided into five categories: Concepts, Application, Network Access, Network Operation, and Transmission. If students follow the sequential learning path, they should finish modules 8 and 9 before taking module 10. According to the themes, module 9 has no prerequisite and may be taken instantly between activities. However, if we compare the case with module 8, which is the last component of the Network Operations cluster. Then, the student must finish previous module 7 and module 6, since it is the requirement for module 7.

Furthermore, the proposed learning plan allows students to take any lesson to complete the subject. However, they need to complete at least two chapters within a module and read the summary to ensure that a module is considered finished. We searched for a learning pathway that would work in this case study that was comparable to the adaptation scenario. The learners will receive recommendations for learning content based on objectives, interests, and past lessons. If a student has completed a module, the fundamental principle for selecting the optimal learning route is to verify the last lesson access and goal of the study. Moreover, we identified three input variables as the input space that should be covered, i.e., Domain Concept, Goal, and Interest. After reviewing the case study with an educational expert, we identified 34 potential learning paths that students may use to finish learning.

B. Test Cases Generation

The knowledge base construction requires some test cases, that is a collection of test data combinations that consist of inputs, criteria for an execution, and expected results. In reality, creating test cases is time-consuming and expensive. It is impossible to conduct exhaustive testing during the evaluation to evaluate all input parameters. Therefore, it is vital to establish the selection criteria in order to limit the refinement rigorously.

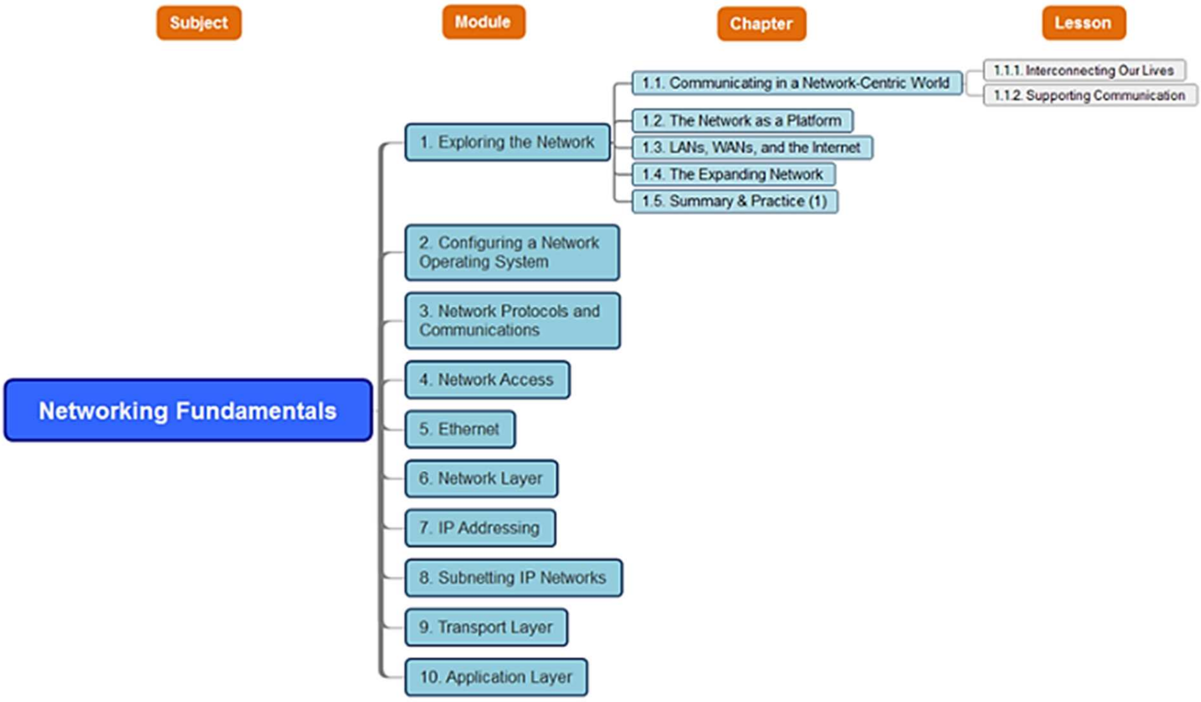

Fig. 2 Structure of the Domain Model (Case: Computer Networking Fundamentals subject)

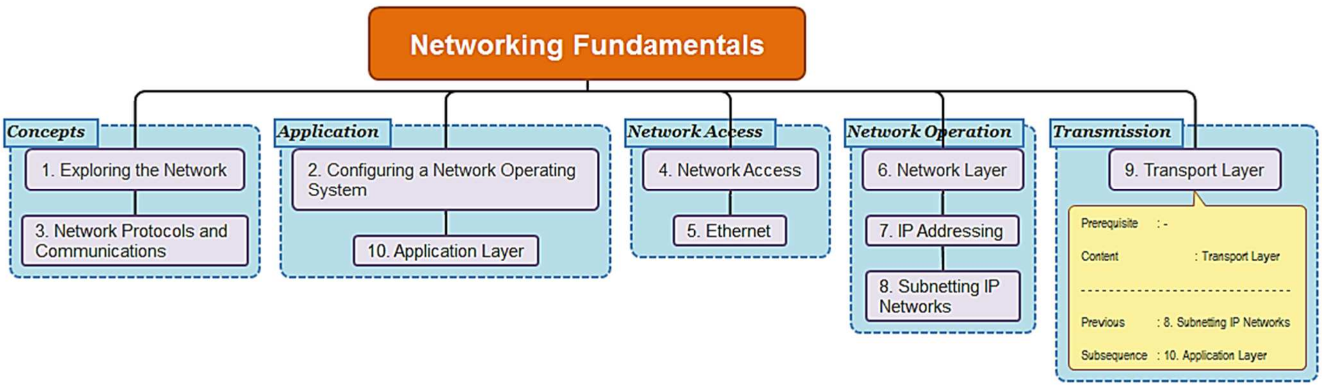

Fig. 3 Structure of the Domain Model

The first step in formalizing test case generation is by extracting and presenting information from domain knowledge. The following five variables make up the test case format in DE:

1) Test Case ID: a number that increases by one with each test case, starting at 1.
2) Input Variables: the test case's input parameters and values.
3) Expected Output: The expert's definition of the expected outcome for each test case.
4) System Output: The results of running the test case as the system produced them.
5) Description: An analysis or summary of the test scenario

To create test cases, a variety of techniques are available. However, rather than analyzing a predetermined number of test instances, our system operates by keeping track of the knowledge acquisition process. As a result, it needs ideal data that accurately simulates the test scenario and actual domain knowledge coverage. The criteria for the test cases must be reasonable and contain the required amount and coverage of test cases.

Furthermore, paired testing is performed to balance between criteria. Combinatorial testing problems are tackled with paired testing [24], [25]. This method determines the parameters that compose the test scenario space. Then, test cases are rigorously selected to cover every pairwise relationship between parameters and values. During pairwise testing, at least one test case will cover all possible parameter value combinations [26]. Pairwise testing aims to determine the smallest possible collection of test cases with substantial test coverage, but the number of the outputs is still acceptable for testing.

TABLE I
PARAMETERS AND VALUE HIERARCHY

No	Parameters	Value Hierarchy
1	Background	Programming, Computer Networking, Human and Computer Interaction, Artificial Intelligence, Web Technology, Database, Operating System
2	Competency	IT Certification, non-IT Certification | No certification
3	Course	Information Technology
4	Experience	Work experience, Internship | Placements, No experience
5	Qualification	Certificates | Diploma, Bachelor | Graduate Certificate/Diploma, Masters | Doctoral, Others
6	Bandwidth	Low, High
7	Device	Smartphone | Tablet, PC
8	Emotion	Anger, Sadness, Happiness, Fear, Pride, Elation
9	Learning Motivation	Unmotivated, Average, Motivated
10	Location	Home | Library, Workplace | University, Outdoor
11	Time	15 m, 30 m | 45 m, 1 h | >1 h
12	Goal	New Start, Complete a Module, Complete a Lesson | Continue Last Study, Perform Assignment
13	Instructional Plan	Assessment, Lecture, Tutorial | Simulation, Case Study | Problem Statement | Project
14	Learning Activity	Study, Practice, Review
15	Objective	Understanding the characteristics and functions of each layer of the OSI model, Describe the networking processes within and between networking hardware, Manage basic components of a Cisco router and a Cisco switch, Apply VLSM Addressing to an Internet Protocol v4, Designing and implementing a hierarchical IPv4 addressing scheme, Recognise and make suitable choices of physical networking equipment for a small network
16	Skill	Analytical | Critical Thinking, Logical Thinking | Mathematical Skills, Problem Solving
17	Cognitive Style	Field Dependence, Field Independence
18	Learning Style	Diverging, Assimilating, Converging, Accommodating
19	Personality	Conscientiousness, Neuroticism, Extroversion
20	Interactivity	Low, Medium, High
21	Interest	Basic, Intermediate, Advanced
22	Preference	Cognitive capabilities, Practical capabilities
23	Presentation	Video, Application, Text
24	Difficulty Level	Easy, Medium, Difficult
25	Domain Concept	1. Exploring the Network, 2. Configuring a Network Operating System, 3. Network Protocols and Communications, 4. Network Access, 5. Ethernet, 6. Network Layer, 7. IP Addressing, 8. Subnetting IP Networks, 9. Transport Layer, 10. Application Layer
26	Knowledge Level	Novice, Intermediate, Expert
27	Learning Performance	Low, Medium, High
28	Prior Knowledge	1.1. Communicating in a Network-Centric World | 1.2. The Network as a Platform | 1.3. LANs WANs and the Internet | 1.4. The Expanding Network, 2.1. IOS Bootcamp | 2.2. Getting Basic | 2.3. Address Schemes, 3.1. Network Protocols and Standards | 3.2. Using Requests for Comments | 3.3. Moving Data in the Network, 4.1. Data Link Layer | 4.2. Media Access Control | 4.3. Physical Layer | 4.4. Network Media, 5.1. Ethernet Protocol | 5.2. Address Resolution Protocol | 5.3. LAN Switches, 6.1. Network Layer Protocols | 6.2. Routing | 6.3. Routers | 6.4. Configuring a Cisco Router, 7.1. IPv4 Network Addresses | 7.2. Connectivity Verification, 8.1. Subnetting an IPv4 Network | 8.2. Addressing Schemes, 9.1. Transport Layer Protocols | 9.2. TCP and UDP, 10.1. Application Layer Protocols | 10.2. Well-Known Application Layer Protocols and Services

In this method, each pair interaction will arise in at least one but potentially do so in several test instances. The paired process represents the system's parameters in a tabular format. Each possible value for every parameter should be represented by at least one covered, uncovered, or excluded test case. For a test case to be covered, it must satisfy a particular combination and vice versa for an uncovered test case. When slots are excluded, they are removed from the combination to be covered. It uses a greedy heuristic as its generating algorithm. If no additional covered slots are discovered, it builds one test case and ends. As a result of utilizing this method, the final result will have more coverage with lower test cases. In addition, it reduces the combinatorial explosion that may occur when assessing a system with several input options, as in this study. Pairwise testing produces a reduced test case compared to exhaustive test data sets. Therefore, paired testing is an appropriate strategy when processing involves parameter interaction [25], [26].

This study uses pairwise testing PictMaster, an excel-based test-generating program. As its name suggests, the utility overlays PICT with an Excel interface. A free program called PICT generates combination testing from command-line prompts. PictMaster is offered as open-source software that is free to use. Users need to define some parameters and value hierarchy in the first stage. The parameter has a limited range of potential values and the names of data model attribute names. The value hierarchy can be depicted as aliasing (|) and is denoted by commas (,). Aliasing is a technique for giving a single value many names. The numerous values are treated as one by this approach. Switching the value's name between the cases lowers the model's combinatorial complexity. The case study's features and value hierarchy are shown in Table 1. The case study has 142 value hierarchies and 28 factors. However, because some of the entities are aliasing, the value hierarchy only contains 102 of them.

Choosing the combination of output restrictions comes after describing the parameter and value descriptions. Users can set constraints to restrict the domain by defining undesirable parameters and value combinations. Since they might include other potential legitimate pairs, the violating test instances cannot simply be eliminated from the outcome. Pairwise testing eliminates forbidden combinations using a constraints technique rather than discarding valid pairings. As a result, the undesirable combinations were eliminated from the results, leaving only the necessary combinations. When expressing constraint conditions and targets, PictMaster uses If and Then relations. The constraint target will be generated if the constraint condition is supplied. Table 2 displays the constraint expression used for the model.

The user was required to configure the settings at the last step of generation. All the data about test configurations are available in Settings. The constraints table is selected in this model to apply constraints to the test case. PictMaster can improve the constraint expressions generated from the constraints table by selecting "Optimize constraint expression. Suppose the system notices that it will take a long time to produce test cases due to incorrect constraints. In that case, the format of the constraint will be adjusted automatically in order to finish the generation in less time. The statistics will include the generation frequency, the number of test cases created, and the time required to finish the generation.

Combination coverage percentages generated during the development of test cases are shown in Show coverage. In this case study, a 90% four-way coverage was built to ensure that the output test case would include most of the domain. In addition, the procedure will be performed five times to guarantee that the generated test cases have adequate coverage of domain knowledge.

TABLE II
LIST OF CONSTRAINTS

Constraints	
a. Domain Concept	**Goal**
1. Exploring the Network	Complete a Module, Perform Assignment
2. Configuring a Network Operating System	New Start, Complete a Module, Perform Assignment
3. Network Protocols and Communications	New Start, Complete a Module, Perform Assignment
4. Network Access	New Start, Complete a Module, Perform Assignment
5. Ethernet	New Start, Complete a Module, Perform Assignment
6. Network Layer	New Start, Complete a Module, Perform Assignment
7. IP Addressing	New Start, Complete a Module, Perform Assignment
8. Subnetting IP Networks	New Start, Complete a Module, Perform Assignment
9. Transport Layer	New Start, Complete a Module, Complete a Lesson, Perform Assignment
10. Application Layer	Complete a Module, Complete a Lesson, Perform Assignment
b. Location	**Time**
Home	1 h
Workplace	30 m
Outdoor	15 m
c. Learning Motivation	**Interactivity**
Unmotivated	Low
Motivated	Medium
Average	High
d. Interest	**Objective**
Concept	Understanding the characteristics and functions of each layer of the OSI model
Application	Recognize and make suitable choices of physical networking equipment for a small network
Network Access	Manage basic components of a Cisco router and a Cisco switch
Network Operation	Apply VLSM Addressing to an Internet Protocol v4, Designing and implementing a hierarchical IPv4 addressing scheme
Transmission	Describe the networking processes within and between networking hardware
e. Learning Activity	**Instructional Plan**
Study	Lecture, Tutorial
Practice	Case Study

C. *Knowledge Acquisition Framework*

The knowledge creation in this study is developed by assessing the available test cases. The learner model's data acquired in the earlier phase is gathered and fed into the subsequent process. Since learning is a challenging and time-consuming process, it frequently creates a bottleneck. While constructing a knowledge base, it is vital to translate the information from the given scenario into a structured model. This process transforms procedural information into in-depth domain-specific knowledge. When the educational expert recognizes the underlying idea or connection between the case that is being presented and the problem domain, they are involved in solving the problem.

For its dependability in creating a knowledge base, an incremental knowledge acquisition strategy utilizing a single classification ripple-down rule was chosen. In contrast to existing AES tools that require modeling and initial specification of all adaptation rules, our tool can build all adaptation rules by incrementally learning from the presenting test cases. When experts discover an error, all rules are constructed gradually by removing requirements from the cases that have been provided. This error-correcting procedure is beneficial for maintaining the Knowledge Base because it ensures that only valid rules are kept in the system.

The steps in the knowledge acquisition process are represented in Figure 4 as follows:

1) The initial step is to present a learning case to a knowledge base repository.

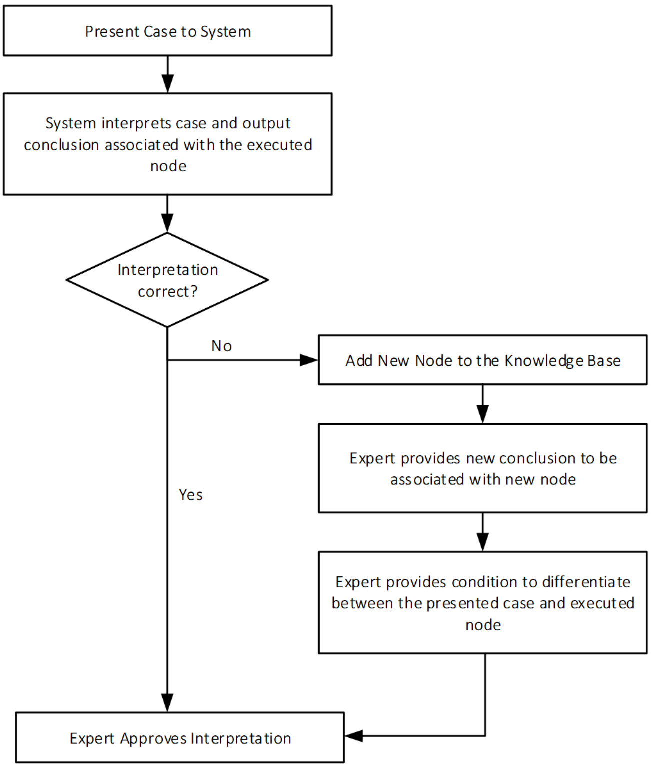

Fig. 4 Knowledge acquisition process

2) The inference process starts with the given case. The algorithm then offers a conclusion for the case that has been provided.
3) The procedure is complete if the expert concurs with the result.
4) A rule for refining the conclusion is inserted if the expert disagrees. The expert selects the expected outcome to initiate the correction method. The expert should then specify the requirements for the newly generated rule. The instances and results will be stored as a new rule. In addition to adding a rule, the cornerstone case is also kept for future reference.

The first inference begins with the introduction of the first rule to the KB in RDR. The default conclusion of this first rule, referred to as the "default rule," satisfies all instances. The first case is assessed against the first rule after being entered into the KB. An expert receives a provisional conclusion from the system. The following real-world example of instances and the knowledge base construction process is provided to demonstrate the inference mechanism:

1) The KB's default rule (R0) is used to start the process. As the root node, a default rule is always the case in an RDR structure. Any condition that is presented as a solution is provided by the default rule. When the default rule's conclusion is unacceptable, the expert must offer a different conclusion and modify the condition to fit the new situation. A dummy link of the initial node from the default node will be used to store the new rule. The default rule has no conditions, as shown in Figure 5. The opening chapter's "Message to the Student" section is the default conclusion (R0).

Four learning cases are shown in Figure 6 and are given to the system. The first step is to compare Case 1 to the KB, which only contains one default rule. The "*Message to the Student*" is the temporary conclusion since the default rule always results in a conclusion. The process of drawing conclusions is complete if the expert concurs with them. The system's predicted outcome, however, differs from the actual conclusion. As a result, the expert offers an updated conclusion for Case 1 and chooses the features appropriate to its setting. The conclusion is "7.2.1. ICMP" since the student only has 15 minutes to study outside and has studied concept "7.1. IPv4 Network Addresses".

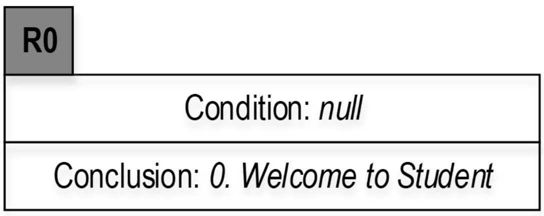

Fig. 5 The default rule of RDR Knowledge Base

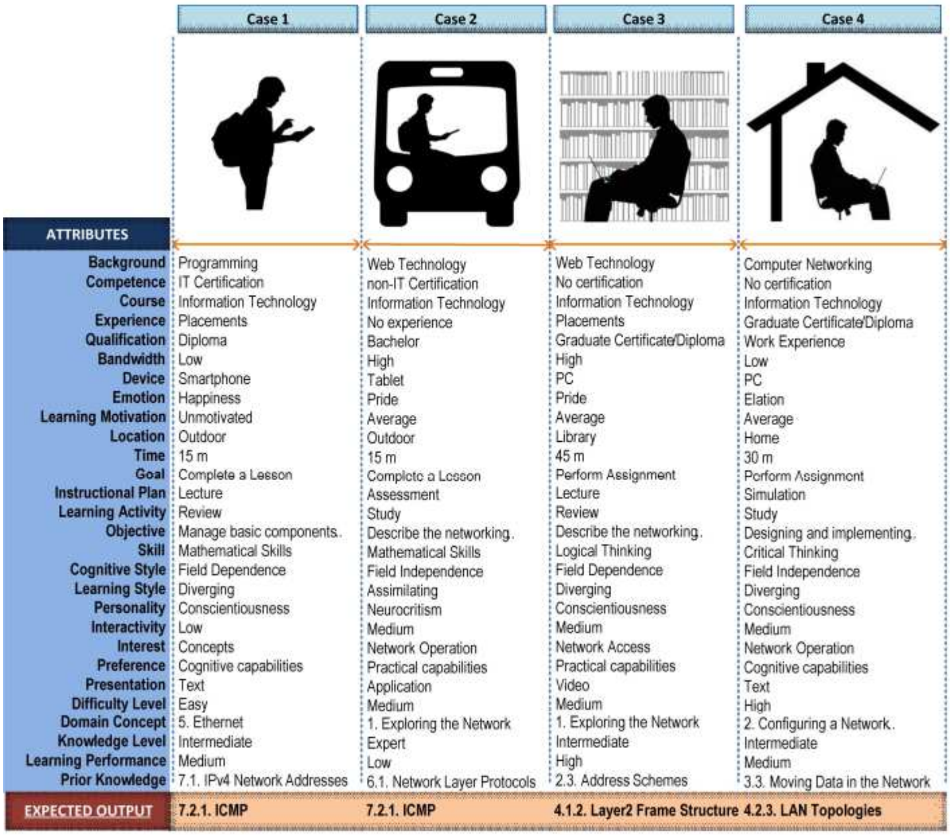

Fig. 6 Illustration of learning case scenario in the dynamic learning environment

As a child node of R0, both conditions and conclusions are stored. Consequently, the KB now contains two nodes: R0 and R1 (Figure 7). R1 is the result of Case 1 when creating the KB. Thus, all conditions in Case 1 will be considered a cornerstone case for R1. The following inference procedure will benefit from this information.

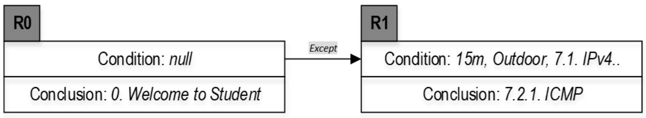

Fig. 7 KB with two nodes

2) The inference procedure used in Case 1 is used in the following case. The condition is always true when Case 2 is compared to R0. Case 2 is then evaluated against R1 via the exception path and compared to the R1 condition when the rule R1 is activated. Consequently, Case 2 and R1 imply identical conditions for "Time: 15 m" and "Location: Outdoors" This resulting "7.2.1. ICMP" as the conclusion. The process is stopped when the expert concurs with the decision.
3) Case 3 is evaluated using the same criteria as R0 and R1. Since its condition is false, a new rule is added as a false node to R1. Because of the circumstance "45m, Library, 6.1. Network Layer Protocols" in Case 3, the expert provides a new conclusion, "4.1.2. Layer 2 Frame Structures." These parameters and the result are saved as R2 (Figure 8). The core of Case 3 serves as the foundation for R2.

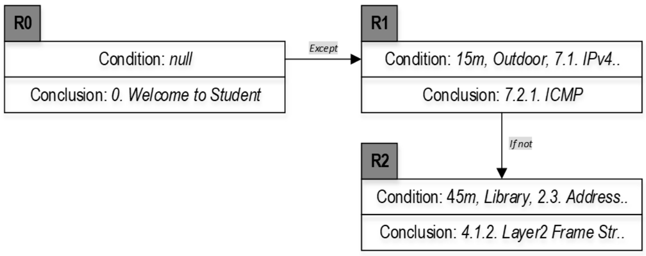

Fig. 8 KB with three nodes

4) The KB is shown in the final Case 4 (Figure 9). Comparing this case with the KB reveals that the condition is incorrect for R1 and R2. Thus, Case 4 will establish a new rule. The new insertion to R2 is a false branch. Using a similar approach allows experts to propose new results and criteria for Case 4. As a cornerstone case for R3, case 4 is also kept in storage.

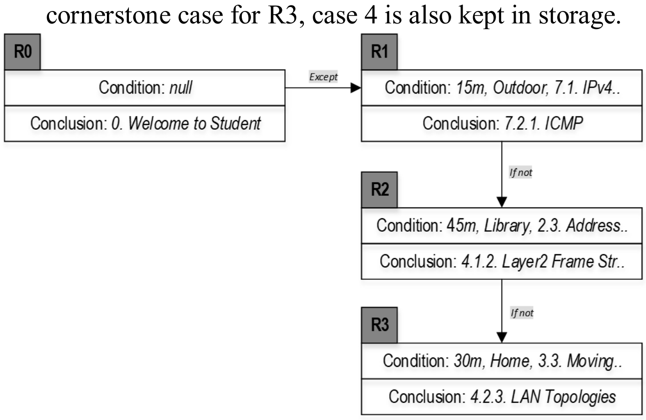

Fig. 9 Final tree with four nodes

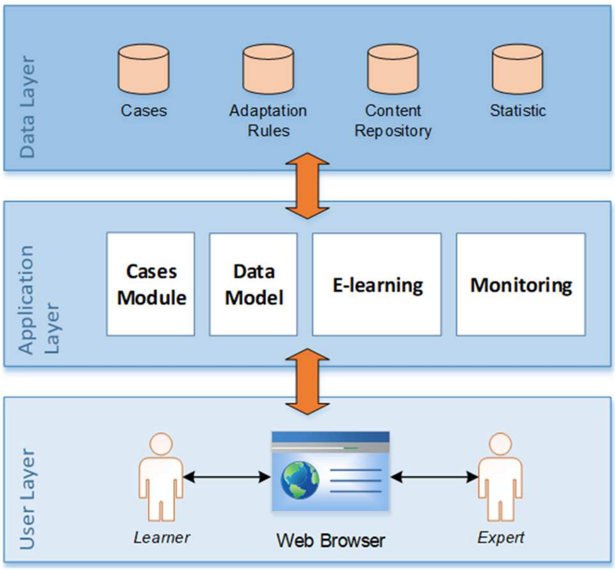

Fig. 10 The architecture of the Proposed System

The rule refinement is incredibly sophisticated despite the inference mechanism's seeming simplicity. The case is processed chronologically with the appropriate expert categorization. An expert needs to compare the case that has been presented that is being fired in order to supply the criteria. The system adds the case to the refinement path once the decision has been taken, and this procedure cannot be reversed later. This information, however, applies to all of the observed occurrences. If the KB finds a unique case that has never been provided previously, it adapts by selecting the rule that most closely matches the unique case. This strategy illustrates an adaption mechanism used in an adaptive e-learning system.

III. Results and Discussions

A. System Design

The user, application, and data layers are the three service levels that make up the RDR-based system instantiation. The initial user layer transmits the request to the application layer while facilitating interaction through the web browser. The system distinguishes between learner and expert user modes of operation. The system-presented learning materials for a specific course are the only things the learner mode can interact. The knowledge acquisition mode for establishing rules and course development is accessible to the expert mode from the application layer. The application layer receives information about the user's interactions with the interface. Processing requests and commands, coming to logical conclusions, and carrying out calculations are the functions of the application layer.

Additionally, it acts as a link that transfers data from the user layer to the storage layer. The cases module, data model, e-learning, and monitoring are the other four primary parts of the application layer. After processing the request, the user layer gets the response from the application layer and then delivers it to the user. Using adaptation rules from the data layer, the application layer obtains and recommends the most relevant learning lesson from the learner's knowledge base when presenting a learning case. Figure 10 demonstrates the system's overall design.

B. Knowledge Base Construction

The knowledge base is constructed with a default rule which offers a conclusion for every given situation. The initial test case data is inserted into the tool and gradually added knowledge to the KB. As RDR incorporates the process of knowledge acquisition and maintenance, rules are subsequently derived from test case inferences. In the RDR, the test case representation is crucial. The first test case of the case study is shown in Table 3. 28 qualities and values made up the input, and this condition's expected result was also provided. Educational professionals have already stated the intended outcome based on specific situation characteristics, namely the goal, the interest, and the domain concept.

As predicted, the conclusion of the default rule would be the inference result for the first test case. As seen in Figure 11, the system recommendation for the first test scenario is "Message to the Student." Several characteristics from the given example should be chosen as the feature to save this case as a rule. This attribute will distinguish this case from others. To achieve this, we need just consider three characteristics: goal, interest, and domain knowledge. The objective of this example is to "Complete a Module" with "6. Network Layer," serving as the domain knowledge. If these similar conditions are uncovered in a case, the expert would recommend that students take "6.3.2 Router Bootup." The following two features and a conclusion must be selected to analyze the first test case and generate a new rule: "6. Network Layer and Complete a Module" and "6.3.2 Router Bootup."

From the previous stage, the system has evaluated the first case successfully, resulting in the establishment of the first rule in the knowledge base, R1. Figure 12 illustrates the constructed knowledge base with only one rule (R1), along with its condition and conclusion. Moreover, the system added the given instance as a reference case for R1. The following stage discusses the acquisition and monitoring the performance of the acquired knowledge.

TABLE III
First Test Case of Case Study

Case No.	Input	Expected Output
1	Background: Programming, Competency: non-IT Certification, Course: Information Technology, Experience: Internship, Qualification: Bachelor, Bandwidth: High, Device: PC, Emotion: Fear, Learning Motivation: Motivated, Location: Workplace, Time: 30 m, Goal: Complete a Module, Instructional Plan: Assessment, Learning Activity: Review, Objective: Apply VLSM Addressing to an Internet Protocol v4, Skill: Analytical, Cognitive Style: Field Dependence, Learning Style: Diverging, Personality: Extroversion, Interactivity: Medium, Interest: Network Operation, Preference: Cognitive capabilities, Presentation: Text, Difficulty Level: Easy, Domain Concept: 6. Network Layer, Knowledge Level: Expert, Learning Performance: Low, Prior Knowledge: 6.1. Network Layer Protocols	6.3.2 Router Bootup

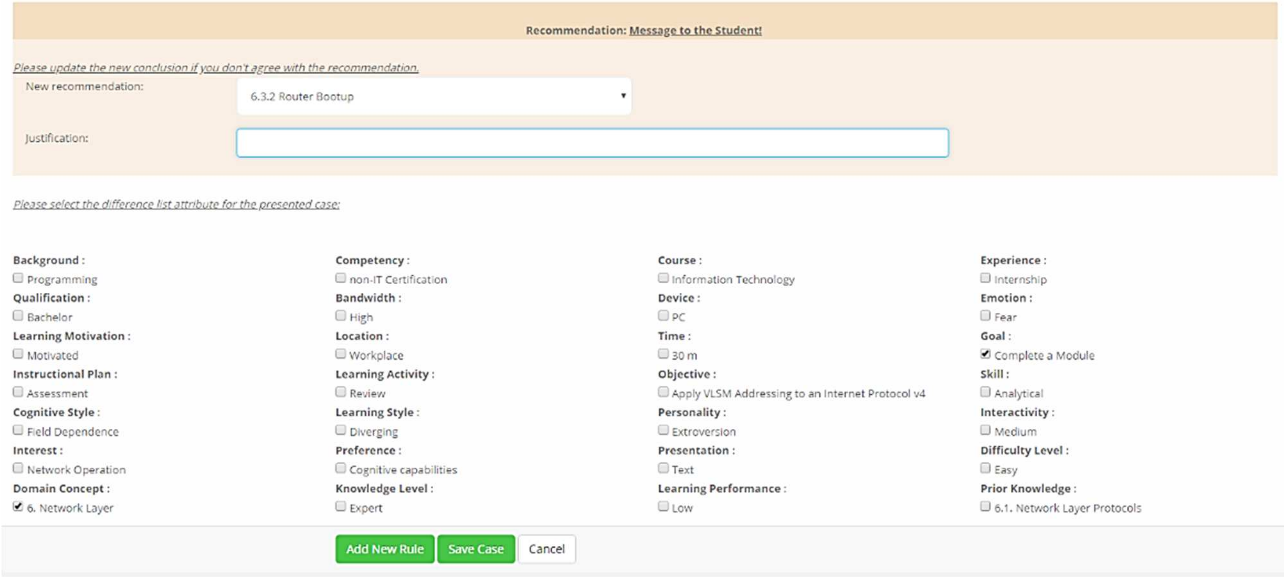

Fig. 11 Inference result for the first test case

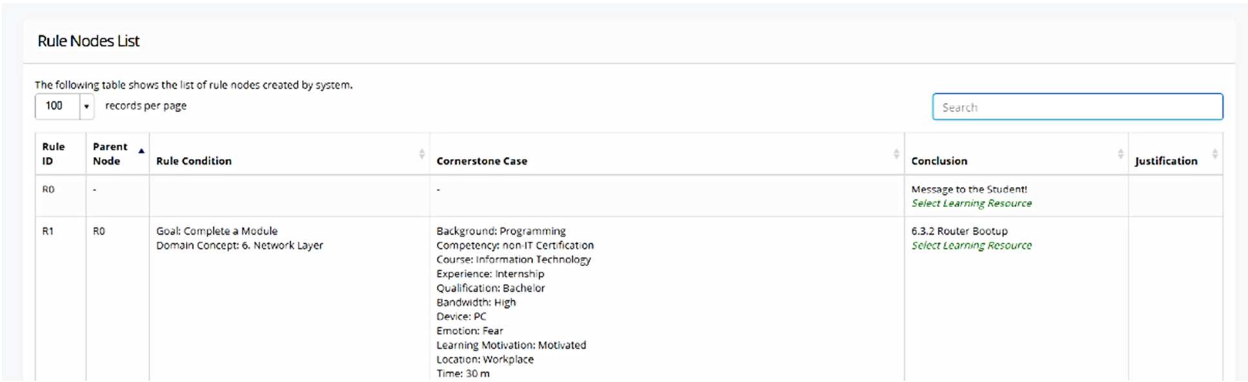

Fig. 12 First rule created in the knowledge base

TABLE IV

FIRST TEST CASE OF CASE STUDY

Case No.	Expected Output	System Output	Accepted Conclusion	Build New Rule	Rule Name	Parent
1	6.3.2 Router Bootup	Message to Student!	No	Yes	R1	R0
2	4.5.2. Class Activity - Linked In!	6.3.2 Router Bootup	No	Yes	R2	R1
3	9.1.1. Transportation of Data	4.5.2. Class Activity - Linked In!	No	Yes	R3	R2
4	3.3.1. Data Encapsulation	6.3.2 Router Bootup	No	Yes	R4	R1
5	10.2.1. Common Application Layer Protocols	9.1.1. Transportation of Data	No	Yes	R5	R3
6	3.4.2. Quiz - 3	4.5.2. Class Activity - Linked In!	No	Yes	R6	R2
7	6.1.1. Network Layer in Communication	9.1.1. Transportation of Data	No	Yes	R7	R3
8	7.2.2. Testing and Verification	3.3.1. Data Encapsulation	No	Yes	R8	R4
9	4.1.1. Layer 2 Introduction	6.1.1. Network Layer in Communication	No	Yes	R9	R7
10	5.1.1. Ethernet Operation	3.4.2. Quiz - 3	No	Yes	R10	R6
11	9.1.1. Transportation of Data	4.1.1. Layer 2 Introduction	No	Yes	R11	R9
12	8.2.1. Structured Design	7.2.2. Testing and Verification	No	Yes	R12	R8
13	3.4.2. Quiz - 3	3.4.2. Quiz - 3	Yes	No	-	-
14	5.4.2. Class Activity - MAC and Choose	5.1.1. Ethernet Operation	No	Yes	R13	R10

C. Performance Evaluation

There were 753 test cases in all that could be assessed. All ten examples in the first series had the wrong classification. Those cases were denied since the KB did not have any rules at the time, and new rules were developed each time a case was improperly categorized. Following the processing of 13 instances in the subsequent batch, the first case that was accurately categorized was identified. As shown in Table 4, no rule was produced for case number 13 since the system's output matched what was anticipated.

Moreover, several evaluation measures were required to confirm the knowledge base's performance. The process started by measuring overall accuracy and comparing the result for performance benchmarking. The overall accuracy is calculated by dividing the number of correctly identified cases (for both groups) by the total number of cases. This approach performs well for classification tasks involving balanced classes, like forecasting the contents of lessons. However, the experiment's class distribution of datasets was uneven or asymmetrical.

Due to this, we evaluated multi-class classification tasks by calculating the precision, recall, and F1 macro-averaged scores. The use of macro-averaged scoring methodologies was chosen since they gave each class the same weight and were more efficient for short courses. The confusion matrix method was used to calculate classification scores.

After the knowledge base has been developed, its performance is evaluated. The system assessed 161 test instances in this experiment and generated 48 rules. Once 48 rules have been discovered, rule generation is stopped. Hence, another experiment is conducted without altering the knowledge base. The overall accuracy of the KB was 92.55 percent, according to the results, while the precision, recall, and F1 macro averages were 93.97 percent, 91.19 percent, and 92.56 percent, respectively. Overall, the experiment's performance results demonstrate how well the knowledge base anticipated the outcomes.

IV. Conclusion

Providing adaptation and personalization in DLE is challenging considering the complexity of designing a system that dynamically changes its behavior according to the students. Adaptation provides learning content tailoring to the student's needs, interests, goals, and background. A good adaptation to support the continuous learning process could be achieved by analyzing current learners' activities. So far, most of the research in adaptive e-learning systems has used various techniques to provide adaptation. However, many of the approaches used adaptation rules that are buried within the code or the system. Therefore, the modifications of the new adaptation rules are often time-consuming, error-prone, and challenging.

This paper introduced an incremental approach using the ripple down rule to build a knowledge base that stores adaptation rules. The advantage of the proposed approach is allowing the system to manage rules routinely without having to rebuild the knowledge base from scratch. Moreover, the knowledge base is developed to facilitate learning content adaptation in dynamic learning. The learning situation is presented in the test cases generated using pairwise comparison to maintain its actual coverage. The evaluation has shown that the proposed adaptive method is performed well to recommend the learning content.

References

[1] L. Tetzlaff, F. Schmiedek, and G. Brod, "Developing Personalized Education: A Dynamic Framework," *Educ. Psychol. Rev.*, vol. 33, no. 3, pp. 863–882, 2021, doi: 10.1007/s10648-020-09570-w.

[2] A. H. Muhammad and D. Ariatmanto, "Understanding the role of individual learner in adaptive and personalized e-learning system," *Bull. Electr. Eng. Informatics*, vol. 10, no. 6, 2021.

[3] A. Dhakshinamoorthy and K. Dhakshinamoorthy, "KLSAS—An adaptive dynamic learning environment based on knowledge level and learning style," *Comput. Appl. Eng. Educ.*, vol. 27, no. 2, pp. 319–331, 2019, doi: https://doi.org/10.1002/cae.22076.

[4] S. Sarwar, Z. U. Qayyum, R. García-Castro, M. Safyan, and R. F. Munir, "Ontology based E-learning framework: A personalized, adaptive and context aware model," *Multimed. Tools Appl.*, vol. 78, no. 24, pp. 34745–34771, 2019, doi: 10.1007/s11042-019-08125-8.

[5] A. Muhammad, J. Shen, G. Beydoun, and D. Xu, "SBAR: A Framework to Support Learning Path Adaptation in Mobile Learning BT - Frontier Computing," 2018, pp. 655–665.

[6] F. Cena, S. Likavec, and A. Rapp, "Real World User Model: Evolution of User Modeling Triggered by Advances in Wearable and Ubiquitous Computing," *Inf. Syst. Front.*, vol. 21, no. 5, pp. 1085–1110, 2019.

[7] H. Yago, J. Clemente, and D. Rodriguez, "Competence-based recommender systems: a systematic literature review," *Behav. Inf. Technol.*, vol. 37, no. 10–11, pp. 958–977, 2018.

[8] J. Gardner and C. Brooks, "Student Success Prediction in MOOCs," *User Model. User-adapt. Interact.*, vol. 28, no. 2, pp. 127–203, 2018.

[9] E. Mousavinasab, N. Zarifsanaiey, S. R. Niakan Kalhori, M. Rakhshan, L. Keikha, and M. Ghazi Saeedi, "Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods," *Interact. Learn. Environ.*, vol. 29, no. 1, pp. 142–163, 2021, doi: 10.1080/10494820.2018.1558257.

[10] N. Idris, S. Z. M. Hashim, R. Samsudin, and N. B. H. Ahmad, "Intelligent learning model based on significant weight of domain knowledge concept for adaptive e-learning," *Int. J. Adv. Sci. Eng. Inf. Technol*, vol. 7, no. 4–2, p. 1486, 2017.

[11] D. Shawky and A. Badawi, "Towards a Personalized Learning Experience Using Reinforcement Learning," in *Machine Learning Paradigms: Theory and Application*, A. E. Hassanien, Ed. Cham: Springer International Publishing, 2019, pp. 169–187.

[12] S. Ennouamani, Z. Mahani, and L. Akharraz, "A context-aware mobile learning system for adapting learning content and format of presentation: design, validation and evaluation," *Educ. Inf. Technol.*, vol. 25, no. 5, pp. 3919–3955, 2020, doi: 10.1007/s10639-020-10149-9.

[13] T. Duan, "A new idea for the optimization of MOOC-based teaching," *Educ. Inf. Technol.*, 2021, doi: 10.1007/s10639-021-10755-1.

[14] H. Zhang, T. Huang, Z. Lv, S. Liu, and H. Yang, "MOOCRC: A Highly Accurate Resource Recommendation Model for Use in MOOC Environments," *Mob. Networks Appl.*, vol. 24, no. 1, pp. 34–46, 2019, doi: 10.1007/s11036-018-1131-y.

[15] V. Slavuj, A. Meštrović, and B. Kovačić, "Adaptivity in educational systems for language learning: a review," *Comput. Assist. Lang. Learn.*, vol. 30, no. 1–2, pp. 64–90, 2017, doi: 10.1080/09588221.2016.1242502.

[16] S. V Kolekar, R. M. Pai, and M. P. M. M., "Rule based adaptive user interface for adaptive E-learning system," *Educ. Inf. Technol.*, vol. 24, no. 1, pp. 613–641, 2019, doi: 10.1007/s10639-018-9788-1.

[17] S. S. Khanal, P. W. C. Prasad, A. Alsadoon, and A. Maag, "A systematic review: machine learning based recommendation systems for e-learning," *Educ. Inf. Technol.*, pp. 1–30, 2019.

[18] D. Herbert and B. H. Kang, "Intelligent conversation system using multiple classification ripple down rules and conversational context," *Expert Syst. Appl.*, vol. 112, pp. 342–352, 2018, doi: https://doi.org/10.1016/j.eswa.2018.06.049.

[19] I. U. Haq, I. Gondal, P. Vamplew, S. Wen, A. Zomaya, and L. T. Yang, "Enhancing Model Performance for Fraud Detection by Feature Engineering and Compact Unified Expressions," 2020, pp. 399–409.

[20] Y. Li *et al.*, "IoT-CANE: A unified knowledge management system for data-centric Internet of Things application systems," *J. Parallel Distrib. Comput.*, vol. 131, pp. 161–172, 2019, doi: https://doi.org/10.1016/j.jpdc.2019.04.016.

[21] A. Maedche, S. Gregor, S. Morana, and J. Feine, "Conceptualization of the problem space in design science research," in *International Conference on Design Science Research in Information Systems and Technology*, 2019, pp. 18–31.

[22] R. Baskerville, A. Baiyere, S. Gregor, A. Hevner, and M. Rossi, "Design science research contributions: Finding a balance between artifact and theory," *J. Assoc. Inf. Syst.*, vol. 19, no. 5, p. 3, 2018.

[23] S. Gregor and A. R. Hevner, "Positioning and presenting design science research for maximum impact," *MIS Q.*, pp. 337–355, 2013.

[24] A. K. Alazzawi, A. A. Ba Homaid, A. A. Alomoush, and A. R. A. Alsewari, "Artificial Bee Colony algorithm for pairwise test generation," *J. Telecommun. Electron. Comput. Eng.*, vol. 9, no. 1–2, pp. 103–108, 2017.

[25] S. Sabharwal and M. Aggarwal, "Test set generation for pairwise testing using genetic algorithms," *J. Inf. Process. Syst.*, vol. 13, no. 5, pp. 1089–1102, 2017, doi: 10.3745/JIPS.04.0019.

[26] D. Gupta, A. Rana, and S. Tyagi, "Sequence generation of test case using pairwise approach methodology," *Advances in Intelligent Systems and Computing*, vol. 554. pp. 79–85, 2018, doi: 10.1007/978-981-10-3773-3_9.