Machine Learning Approach for an Advanced Agent-based Intelligent Tutoring System

Roya Aminikia

A Thesis

in

The Department

of

Concordia Institute for Information Systems Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Quality System Engineering) at

Concordia University

Montréal, Québec, Canada

May 2018

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Roya Aminikia**

Entitled: **Machine Learning Approach for an Advanced Agent-based Intelligent Tutoring System**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Quality System Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

______________________________ Chair
Dr. Rachida Dssouli

______________________________ External Examiner
Dr. M. Zahangir Kabir

______________________________ Examiner
Dr. Babak Khosravifar

______________________________ Supervisor
Dr. Jamal Bentahar

Approved by ______________________________
Rachida Dssouli, Chair
Department of Concordia Institute for Information Systems Engineering

______________ 2018

____
Amir Asif, Dean
Faculty of Engineering and Computer Science

Abstract

Machine Learning Approach for an Advanced Agent-based Intelligent Tutoring System

Roya Aminikia

Learning Management Systems (LMSs) are digital frameworks that provide curriculum, training materials, and corresponding assessments to guarantee an effective learning process. Although these systems are capable of distributing the learning content, they do not support dynamic learning processes and do not have the capability to communicate with human learners who are required to interact in a dynamic environment during the learning process. To create this process and support the interaction feature, LMSs are equipped with Intelligent Tutoring Systems (ITSs). The main objective of an ITS is to facilitate students' movement towards their learning goals through virtual tutoring. When equipped with ITSs, LMSs operate as dynamic systems to provide students with access to a tutor who is available anytime during the learning session. The crucial issues we address in this thesis are how to set up a dynamic LMS, and how to design the logical structure behind an ITS. Artificial intelligence, multi-agent technology and machine learning provide powerful theories and foundations that we leverage to tackle these issues.

We designed and implemented the new concept of Pedagogical Agent (PA) as the main part of our ITS. This agent uses an evaluation procedure to compare each particular student, in terms of performance, with their peers to develop a worthwhile guidance. The agent captures global knowledge of students' feature measurements during students' guiding process. Therefore, the PA retains an updated status, called image, of each specific student at any moment. The agent uses this image for the purpose of diagnosing students' skills to implement a specific correct instruction. To develop the infrastructure of the agent decision making algorithm, we laid out a protocol (decision tree) to select the best individual direction. The significant capability of the agent is the ability to update its functionality by looking at a student's image at run time. We also applied two supervised machine learning methods to improve the decision making protocol performance in order to maximize the effect of the collaborating mechanism between students and the ITS. Through these methods, we made the necessary modifications to the decision making structure to promote students' performance by offering prompts during the learning sessions. The conducted experiments showed that the proposed system is able to efficiently classify students into learners with high versus low performance. Deployment of such a model enabled the PA to use different decision trees while interacting with students of different learning skills. The performance of the system has been shown by ROC curves and details regarding combination of different attributes used in the two machine learning algorithms are discussed, along with the correlation of key attributes that contribute to the accuracy and performance of the decision maker components.

Acknowledgments

First and foremost, I would like to express my gratitude to my supervisor, Prof. Jamal Bentahar, from the Concordia Institute of Information Systems Engineering at Concordia University, for the patient guidance, useful comments and advice he has provided during my master program. I have been very lucky to have a supervisor who cared so much about my work and his office was always open whenever I had a question about my research.

I would like to thank my husband, who has supported me throughout entire process. I have continually enjoyed his willingness to proof read countless pages of my thesis.

I want to thank Concordia University not only for giving me the opportunity to do my master, but also for providing the wonderful and knowledgeable time I have spent in university. Last but not least, I would like to thank God for giving me the strength, knowledge, and opportunity to undertake this research study and to complete it successfully.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the past years, technology has shaped the education from the earliest stages of primary school to the advanced stages of graduate school. Thanks to the considerable advances of online learning management systems, noticeably students, teachers, employees, managers, and others showed an unprecedented desire to enhance their skills and promote their knowledge for future professional advancement using modern styles rather than the traditional education system.

For the growth of the eLearning era, Learning Management Systems (LMSs) [3, 53, 17] have been enhanced to serve electronic coursework for distance learners. A LMS is a system framework for designing, organizing, classifying and presenting courses accessible through a distributed system. Primary LMSs have been used only in academic organizations to afford lessons and training programs in schools. However, during recent years, it has become fashionable among many corporations to deliver training to internal clerks and clients using LMSs.

A typical LMS yields the following capabilities to the teacher: (a) uploading the contents using different appropriate tools; (b) setting up and categorizing the materials as lessons; (c) designing well-structured self assessments; (d) determining due dates to complete assignments; and (e) cooperating with students to maintain high quality lesson engagement during the learning period. The system also generates a wide-range of charts corresponding to students' grades and performance by evaluating their progress in a learning activity. Likewise, a classic LMS provides the following abilities to students: (a) accessing the lectures; (b) performing quizzes and assignments; and (c) communicating with their teachers throughout the semester.

Nowadays, there is a diverse range of LMSs available with different features to accommodate users' different requirements. For example, Canvas [12] is one of those LMSs, which is used by educational and non-educational organizations. Moodle [28, 33] is another well-known LMS used world-wide [48]. Dokeos [12] is a popular LMS in French and Spanish-speaking countries. Sakai [12, 2] is an academic LMS that was built by universities of Massachusetts Institute of Technology (MIT), Stanford and Berkeley for handling students' group projects. Lately, some modern LMSs [4, 5] have been developed based on cloud, which provides modern features such as mobile accessibility and analysis.

Intelligent Tutoring Systems (ITSs) [61, 9, 10, 41, 58, 59, 69] are software systems that have been developed to improve the operation of LMSs. The main intention of a typical ITS is to give the perception to students that a virtual tutor is available during the whole learning process. For instance, during a quiz, if the student solves some questions incorrectly, the ITS steps in and elaborates on why the answers are wrong and provides guidance towards the correct solutions. Therefore, the ITS has been constructed to play a passive tutor role which facilitates student monitoring in order to set up convenient directions by generating customized and personalized advice to every individual student. Because ITSs have originally been developed to serve as intelligent virtual tutors, feedback from such systems depend on students' learning skills. Similarly, it was expected that a classic ITS could modify its functionality when interacting with students of various abilities such as knowledge and performance. However, this objective is better achieved using agent-based ITSs, where agent is an intelligent component that is able to make a decision about what action it needs to take in a given moment. In agent-based systems, agents are generally organized specifically to complete the assigned goal by considering the current situation and previous experiences [22].

The implementation of a powerful ITS component was an ambitious goal for many years, and conducted efforts led to the creation of a static complex structure. ITSs were operating without any intelligent decision making procedure. In other words, legacy ITSs are composed of different algorithms that execute simultaneously and do not generate the dynamic feature for a decision-making mechanism. To address this problem of original ITSs, an additional module to operate as a dynamic component that is able to make decisions was needed.

To overcome the lack of intelligence in ITSs, agent technology has been used in the structure of the system to apply artificial intelligence techniques and improve the feedback mechanism. Improving such a mechanism is conducted by modulating the decision-making algorithms and applying changes caused by the external environment, in addition to tracking the collected information whilst interacting with students. The purpose of using agent-based ITSs is to simulate a human tutor. In other words, an agent-based ITS operates such as a human tutor in guiding students. It can also distinguish students in terms of a variety of features. By considering the information collected from a student monitoring system, an agent-based ITS modulates an individualized guidance and prepares the new topics to provide a comprehensive learning process.

A key feature of agent-based ITSs is their ability to self-adapt and change functionality depending on students' needs, ability that is lacking in the original ITSs. Consequently, a basic ITS lacks the capability of modifying its decisions according to the external factors captured from the environment. It then only follows a static and regid algorithm and applies it whilst interacting with students regardless of their skills in an identical manner. Thus, attention to external factors is a core element when developing modern ITSs.

1.1 Problem Statement

The use of eLearning education brings time and location flexibility, which attracts several types of learners. However, in spite of its flexibility, many learners face new challenges for the lack of human guidance, which prevents them from successfully completing their courses. In fact, one of the main issues in online education is students' engagement [2, 3]. Students often start online courses with enthusiasm. However, this deteriorates over time, as they begin to feel isolated and loose motivation. In fact, when starting distance learning courses, all students are passionate to learn new materials and are dedicated to achieving high scores. However, at the mid-point of the course, students fall behind schedule, have difficulty with time management, may become demotivated and encounter difficulties during learning, finding themselves incapable of overcoming these issues. Therefore, self-motivation and time management are two essential requirements, which a distance learner should consider to benefit the most from eLearning systems.

Thus, the critical challenge for an eLearning system is to build an interactive presence to avoid loss of concentration. As a solution, there should be a virtual tutor that is perceived as an actual human teacher who engages in the students learning progress. Engaging a tutor in this process, providing individualized appropriate guidance, monitoring progress and performance, and essentially leading the students as an interactive environment are the first research problems to be discussed in this thesis. The aim is to design a dynamic system which assists students easily overcome learning challenges and improve their knowledge.

Machine learning provides powerful theoretical and practical foundations that, in the domain of interest for this thesis, allow the design of automatic education systems which are able to recommend new topics according to students preferences. That is, machine learning methods can be effectively used to modulate the intelligent Pedagogical Agent (PA), which forms the main part of our ITS, to act as a virtual human tutor. The idea is to empower the virtual tutor with a classifier to help predict students needs based on their updated status, called image, captured from the data collected during previous interactions. The use of machine learning improves the quality of tutors understanding in order to better guide the student towards the ultimate goal. The more information the pedagogical agent has of the users characteristics, the better it can predict their requirements and thus adapt to their reading and comprehension style.

Another challenging issue we aim to address in this thesis is to enable the PA to automatically adapt to the user's profile. Such an approach is not fully practiced or achieved by actual teachers in real classrooms as they do not have substantial resources and flexibility to modify their lessons to accommodate a large number of students with various learning skills. A machine learning-based methodology can, for this issue as well, effectively contribute in adding this crucial feature to the PA to offer personalized guidance. Having a private tutor and individualized guidance throughout the specified lessons and assessments makes the learning experience more attractive for students so that they can maintain their concentration.

1.2 Objectives and Contribution

Objectives

The main objective of this thesis is to contribute in designing an intelligent PA to guide learners. Correct recognition of students categories plays a notable role in the decisions made by the PA in order to choose the applicable direction to guide those students. Therefore, the machine learning algorithms to be used by the PA should be carefully selected. Nowadays, there are a collection of patterns in the machine learning field where different algorithms could be used to manage and assist students towards their unique learning objectives. Selecting suitable algorithms depends on many factors such as size, quality and nature of data. It is difficult to predict which methods are most appropriate without trying them. Hence, different methods should be tested by considering the data characteristics to choose the most suitable for students prediction and classification process. The aim is to have a PA that can select the desirable teaching technique according to the student' group, which reflects his needs and skills. Technically speaking, each teaching method which is selected by the PA is an individual decision tree, making the learning process attractive for students to maximize their learning outcome.

Contribution

This thesis is mainly about introducing a dynamic decision maker component that works as a main part of the PA. The agents used in intelligent tutoring systems are programmable and rational. They are also empowered with intelligent modules that let them process external data and rationally react to the environment. However, this mechanism does not necessarily make them adaptive to the environmental changes, unless the predictions are made and necessary adjustments have been hard-coded in the agent's code. In this thesis, we propose an intelligent PA that can learn from the environment and foster a descent knowledge to formulate a system adapting to the external factors derived from the environment. The dynamic decision maker component helps the PA adapt to the student's learning skills by considering the uncertainty factors when engaged in an interaction with human learners. Deploying such a framework would enhance the performance of the ITS to a certain level as the system is agile to uncertain acts that a student could perform while using agent's guidance prompts. Therefore, the deployment of such a system has a dramatic influence on the accuracy of the learning skills classification and therefore, the overall performance while interacting with students. The results provided in the experimental chapter provides evidence of the fact supported by strong foundations of the framework.

1.3 Assumptions

In this research, to run a system based on the intelligent PA, several assumptions are considered. These assumptions are categorized into three groups: general assumptions, technical assumptions and domain-specific assumptions.

General assumptions: The calculation of some factors is crucial to classify students into related categories and compare these groups by pre-defined evaluations. Based on these measurements, ITS offers adequate teaching methods to various students. Therefore, to be able to measure these factors (such as the average of a class knowledge in a learning progress), it has been assumed that all students are in the same grade. By considering this assumption, we do not need to consider the differences between age groups. Another assumption in this category is applying the Blooms Taxonomy approach [1] for categorizing educational goals. This is a famous framework that has been applied by teachers and college instructors in their teaching structures.

Technical assumptions: This system is developed to run in a mid-sized classroom and cannot facilitate a countless classroom (such as five hundred students) simultaneously. Whereas in a classroom consisting of an enrollment size of two hundred or less, students can benefit from online learning services through this system at the same time. This means the software structure, network infrastructure and programming strategies of this framework are assumed to assist a limited number of learners. As another technical assumption, it is considered that machine learning will provide the appropriate support for the PA. Consequently, substantial research is conducted to find the most suitable machine learning methods for this research, according to our data specifications.

Domain-Specific assumptions: In this thesis, models have been developed to support the interaction between the ITS and students. The user model depicts students status in a learning activity, whilst the ideal model depicts the best movement on the learning activity. The best movement is computed based on the teachers idea and the self-development model, which indicates the students self-measured status whilst being occupied in a learning session. Moreover, we considered a comprehensive decision tree as a core of the ITS to guide students to become closer to the ideal model. Using such an inclusive decision tree enables the ITS to make dynamic decisions based on past experiences. As a result, the flexibility of adjusting existing decision trees supports the ITS in acting smoothly to guide the students in an accurate and dynamic manner. As final assumption, we process the collected data partly in this study. This allows us to decide in a momentum way which data will be processed. The data processing will then commence by receiving data without considering any previously stored data.

1.4 Thesis Overview

The organization of this thesis is as follows. In Chapter 2, we present the related work, which is categorized into different subsections based on the relevant topics to what we proposed in this thesis. We also explain in details the concepts needed to understand the rest of the thesis. Finally, a discussion of different learning environments ends the chapter. In Chapter 3, we introduce the proposed framework with its relevant components in details. We provide a case study to show how the teacher and student engage in an interactive learning environment. In this chapter, we provide a theoretical mechanism, which is an extension of Markov Decision Processes (MDPs) to improve the PA's decision making in order to guide learners in an effective way. In Chapter 4, we discuss an experiment for collecting data. We focus mainly on the raw data structure and data collection steps in details. Then, we propose the suitable machine learning techniques: Logistic Regression and K-Nearest Neighbor (KNN) to practice by considering the nature of the collected data. Steps for the implementation of the selected machine learning algorithms and obtained results are part of this chapter. In Chapter 5, conclusions and future work are presented and the contributions of the thesis are summarized.

Chapter 2

Background and Related Work

2.1 Background

In this section, we elaborate more in details about the preliminaries and the subjects/concepts that a reader should be aware of before reading the rest of the thesis. The preliminaries are followed by a discussion of related work that shed light on the state-of-the-art about eLearning.

Interactive Learning Management System (ILMS)

The Interactive Learning Management System (I-LMS) is a LMS, which interacts with human learner through a series of relevant advices. In other words, it is not only a schema for content sharing, but also it dynamically interacts with users to improve their learning experience. The basic LMSs [31, 2, 3] were a kind of digital frameworks that suggested curriculum, training materials, and corresponding assessments, whereas the modern version of LMSs [42] have further to offer.

Some I-LMSs use an evaluation mechanism to compare each students with their classmates to develop worthwhile guidelines to guarantee an effective learning process. For example, an ordinary I-LMS enables a feature that warns the student about the assignments' due dates and upcoming quizzes to avoid any missing deadlines. On the other hand, the more prominent feature of I-LMS is student following/monitoring feature. Through students' monitoring phase, the LMS captures a global knowledge of students' feature measurements for diagnosing their skills to implement specific correct instructions.

The more dynamic the LMS is, the simpler students communicate with it. This is the reason for the success of I-LMS in contrast to early LMSs, which had been carried out simply for file distribution. In fact, LMS is accepted as a content and assessment delivery environment, however, the I-LMS is more advanced solution that collaborates with students through informing them of mistakes and limitations with more details and directions. Moreover, I-LMS provides immediate help to influence students by giving prompts during learning sessions.

Intelligent Tutoring System (ITS)

The ITS records students' attributes such as pre-existing knowledge level, time management and students' score in the quizzes to build a background corresponding to each student. ITS frequently updates the background information in order to offer personalized hints to guide the students. As ITS usually comes with a LMS, there is an interface for interaction between the student and system that helps the ITS to monitor students' progress and get more information as they operate. As a significant feature of ITS it can be referred to its simulation in a way that it provides the benefits of one on one learner without paying attention to distance in a large scale of students.

Another objective of using the ITS is establishing adaptive learning module. In more details ITS is able to diagnose the weaknesses of students, hints when mistakes are made, then presents the new topics to help them strengthen their weaknesses. Most ITSs start the instructional process by considering the learners' pre-existing knowledge level, which is estimated typically through a pre-assessment mechanism. The status of learner will be updated automatically based on different triggers depending on the structure of the ITS. In fact, a typical ITS compares what specific student needs to know with what is already known, then provides suitable individualized instructions that is needed to improve the learners status. Therefore, these instructions are the system's reflection to the students' current knowledge. Some of the ITSs [61, 9, 10, 41] are designed to develop and present the useful graphic reports, which is a significant source for instructors to diagnose the extent to which their teaching methods are useful.

As explained above, the ITS originally has been developed to serve as an intelligent virtual tutor. Therefore, feedback of such system depends on students' learning skills. In other words, it was expected that ITS could modify its functionality when interacting with students with various abilities such as knowledge and performance.

Agent

There are some specific factors to clarify the agents with arbitrary programs such as reaction to the environment, autonomy, goal-orientation and persistence. As explained before, an agent is a piece of a software that perceives its environment and acts upon that. Therefore, by changing the environment, the agent's behavior would accordingly change. One of the agent's essential factors is rationality. A rational agent is the one that chooses the right action which makes the agent more successful. An agent's success evaluation depends on the defined criteria that determines how successful an agent is. One other agent's attribute is autonomy that would lead to its independency. An autonomous intelligent agent is capable of choosing between different actions. In other words, an autonomous agent is the one that determines the appropriate operation by its own experience and understanding of environment rather than following of a statistic predefined structure.

Multi agent System

A multi agent system follows an intricate decision making mechanism since there is a complicated accurate link between perceptions, representations, and actions of whole available agents. A precise decision making algorithm is required for the environment that several agents exist at the same time to share common resource and communicate with each other. However, the hierarchic connection of agents is vital to synchronize them to choose the best action in order to achieve the highest success in a multi agent system. In addition, as there are multiple agents which communicate together, choosing the protocol for their communication play a principal role in decision making mechanism. Therefore a protocol which is chosen for type of messages that they can send to each other and also the syntax of these messages, is one of the challenging parts of design and development a multi agent system.

Pedagogical Agent

The potential of emotional interaction between human and computer has been interested by researchers in computer science, artificial intelligence, and education domains. The integration of intelligent tutoring systems, autonomous agents, and educational theory has been built the pedagogical agent. In other words pedagogical agent has been developed to overcome some constraints of distance learning. As explained before, the intelligent tutoring systems were used to support individualized learning to meet individual student's needs, and assist each learner in the achievement of proficiency learning. Using a pedagogical agent-based system as a part of the ITS, provide the conditions that the students perceive the learning process as a social and winsome environment. Providing a social interaction between agent and student makes PA distinctive, as an intelligent tutor that teaches the lessons to the students not only in a static course framework but also through an interactive learning environment. Moreover, PA produces encouragement, information, guidance, or any other collaboration to communicate with students.

The intelligent pedagogical agent systems are developed to infer the students' emotional through online learning when they may participate in virtual classroom from far distance. PA is capable of collecting students' data while they are reading the contents or performing any assigned assessment. It also act based on it's knowledge which is gotten from each specific student up to the moment. Moreover, it supports individualized guidance, content rephrasing, content suggestion and chart creation to aware the students of their strengths and weaknesses to improve.

2.2 Related Work

In this section, we carry on the discussion with respect to the related work that are categorized in different subsections.

2.2.1 Learning Management System (LMS)

Because of the eLearning benefits, it may be supreme revolution in modern online education frameworks. Nowadays, many educatioal institutes have started to use LMS, since they found it as one of the easiest and cheapest way to study that is available for everyone even in far distances. So in response to this demand, the LMS market is growing every day by a great leaps. Finally, these mutations has created various LMSs such as Moodle [28, 33], Canvas [12], Schoology [46, 6], Edmodo [8, 51] and Blackboard [37, 17]. These are some of the most popular LMS software that have been used by well-known schools and universities for online courses worldwide.

As there are many training platforms available, it is very important to understand the role of the learning management systems in the learning environment to select the one that provides the highest attention to concerns. However, all of these frameworks are essentially a digital learning environment to handle the students' learning process, which a human tutor usually performs in a real world. In all LMSs the instructor has the option of organizing learning time-line and assignments with deadlines. Therefore the main feature of LMS is giving the ability to deliver learning contents straight to the students as well as allowing them to create assessments for evaluating purposes. Despite all the similarities, different LMSs have various characteristics to meet specific needs.

In spite of all the LMS advantages, they have some limitations that sometimes prevent the students from continuing the learning sessions. The students often start online courses with a great excitement, however, many of them especially those with poor study habits, lose their motivation and fall behind their schedule in long run of the session. Some of them prefer traditional learning system since they want to have communication with their teacher in order to ask questions, whereas there is no social interaction in basic LMSs to connect the students to teacher and other classmates. Moreover, competition between students can be very encouraging for them in order to work harder, while there is no environment for competition, communication or discussion. One another issue in basic LMSs is that students need to be informed of their learning progress as it is necessary to know what they learned and what they are about to learn. If students are not aware of their progress, they become confused and may feel isolated and loose their passion continuing the learning process. In addition, they need someone to track them during the learning session and guide them when they made mistakes. So lack of such a valuable guideline is an considerable cause that learners are bored while doing a learning activity. By considering the aforementioned points, it is clear that the know how of the merits and demerits of the LMS improves its characteristics in order to raise students' interest in such learning framework.

2.2.2 Intelligent Tutoring System (ITS)

Originally, learning management system was developed to provide personalized training assistant for each student instead of an environment that is just used for file sharing. Alternatively, once can use many other easier ways to share files such as email and google docs. But that is not the objective. Nowadays, everyone is aware of the benefits of individualized instructions, so, the ITS emerged as add on to the LMS to produce the purposeful interaction with students during the learning session. Therefore, lack of social human communication is the crucial cause to deploy intelligent tutoring systems to support dynamic interactions while students are studying as a distance learner. In fact, the ITS plays a passive assistant role when students are reading the uploaded lessons or performing assignments through the LMS. Virtual training tutor is the most important feature of ITS that supports the benefits of one-on-one instructions. To meet the users' (students', teachers') expectations many investigations have been done and as a result a variety of ITSs have been developed that include unique characteristics. For instances Sophie [47], Meta-Tutor [55], CANVAS [12], Simself [14], Atutor [7], and SmartTutor [10, 41] are among intelligent tutors that aim to provide immediate and customized feedback to students.

Although the various ITSs consist of different components and strategies, however, all of them have a common ultimate objective that is producing individualized and immediate instructions to students in a practical format. ITS enables students to better learn by practicing their skills within highly interactive learning environment. It assess the students' actions within those interactive environments as well as builds a model of students' knowledge, performance, time management and other cognitive and non-cognitive skills. In other words, ITS monitors students in terms of skills, then records the information taken as characteristics. ITS is also able to update the collected information to perform an accurate evaluation of each student. Based on knowledge that ITS receives from students and its evaluation, ITS adjust the instructions and presents relative explanations, hints, quizzes and new topics as needed. Therefore, it is a critical feature for ITS that modify its functionality while interacting with various students and provide accurate and customized directions corresponding to each specific student automatically.

According to the previous explanations, two inseparable properties of the ITS are intelligence and tutoring. The lack of each of those features produces two conceptual problems for ITS. The first problem happens when intelligence is ignored in the architecture of the ITS. It is expected from the ITS to operate as an intelligent component that acts upon the students' instantaneous acts, then automatically adapts to better improve students' skills. The ITS is able to adapt to by analyzing students' strengths and weaknesses. Accordingly, it selects the appropriate action from other available actions for guiding students to become closer to the ultimate goal through shortest path. In the mean time, the intelligence feature is missing in many of the proposed frameworks as ITS. The notable reason for the lack of success of these kind of ITSs is that they follow a predefined structure (decision tree) to feedback upon the students' actions in a static way. Sometimes these structures developed by complex algorithms, but even with these algorithms the system is incapable of choosing the action autonomously. Some of the ITSs which are introduced above such as Simself, Meta-Tutor, Atutor are categorized in this group.

In the first category the ITSs focus on efficient interaction between students and system following the objective to act as a smooth mentor. The problem in this kind of systems is that, they act poor as an intelligent component. Whereas in the second group, the focus is on intelligence of the ITS. So, in this category the problem occurs when the concept of tutoring is missed in the architecture of the ITS. These ITSs incorporate some algorithms in order to monitor the students and try to personalize instructions on the basis of adaptation to students' skills to provide appropriate teaching strategies. In other words, this group of ITSs are able to select the relevant and customized action autonomously to interact with various students. Therefore, the second group use a powerful intelligent structure for decision making mechanism to adapt themselves with variety of learners with different skills. However, they are poor in tutoring students toward the best learning path, during the distance learning. Some of the ITSs which are introduced above such as Canvas classifies in this group. As explained before, the ITS has been developed to act as an intelligent system, as well as tutors the students smoothly such as a human tutor. In order to this emphasis, the ITSs should be built in a way that supports an artificial intelligence to operate dynamically and autonomously, as well as guides students smoothly in order to achieve their objectives.

2.2.3 Agent

An agent can be of many types such as machine, person or a software program. In computer science domain, an agent is a piece of a computer program that acts in a software environment. In more details, a software agent is a goal-oriented computer program that reacts to its environment without direct human intervention. It is important to note that a software agent possesses some necessary properties such as autonomy, interactivity, rationality, and adaptability. In this thesis, we narrow down the types to pedagogical agents that are programmed specifically for the purpose of serving as a virtual human tutor.

Prior to discussing the pedagogical in depth, it is worth to elaborate on different types of agents in further details. There are different types of agents that are categorized into collaborative agents, interface agents, mobile agents, and reactive agents. Collaborative agents are a network of intelligent autonomous agents that communicate with one another to decide about their functional operations. This operation should be selected in order to build an effective interaction between users and system for achieving a defined ultimate goal [67, 19, 15, 41]. The global companies use this type of agents to respond to rapid changes in customer needs and increase the efficiency of the whole supply chain. [19] introduces production-distribution planning system to intercede the planning gaps between two planning functions: production and distribution by using collaborative agents.

Interface agents are those agents that apply artificial intelligence techniques to the system's architecture, in order to provide a human like assistant for students [67]. It acts as a personal assistant who collaborates with learners during a learning activity. Intelligent Tutoring Systems [40, 9], Meeting Scheduling systems [13], and also News Filtering systems [67, 29] are among the interface agents. For instances, in [67], authors attempted to address the problem of managing the big amount of news feeds in media. They introduced Fido, as an interface agent in order to solve the news feeds overload problem. Fido filters news by considering user preferences and feedback to display personalized news to the users.

A Mobile Agent is an executing program that is launched by a user to migrate from one node to another autonomously in a network. In each machine, each agent interacts with other agents to continue its operation in order to accomplish its defined tasks in the destination. This kind of agent is generally attractive to use in distributed information retrieval applications [18, 21] as they are able to migrate to the location of an information resource and locally search the information. In [18], authors developed a system to manage the distributed information retrieval processing in order to search for the required technical papers from distributed database through a network. In this system, mobile agent can travel from a host machine to different destinations, perform data processing, then send the related information back to the host.

2.2.4 Pedagogical Agent

The aforementioned types were a general classification of agents used in industry. In the following, we carry on the discussion with deeper details regarding our specific type of interest: Pedagogical Agents [52, 23, 66, 44, 45]. This kind of agents are all used in intelligent tutoring systems for the purpose of interaction with human learners to provide the information that is gathered to convey a piece of information as knowledge representation. Interactive Narrative Tacit Adaptive Leader Experience (IN-TALE) [34, 35, 36] is a practice environment that authors applied the experience management framework for educating cognitive skills such as leadership, and decision-making under pressure. In this framework, learners play a leadership role in a simulation of a military exercise to achieve a specified effect. Authors also have developed narrative mediation, which is a technique where a story is defined by a linear plot progression to support related learning situations.

FearNot! [16, 50] is another pedagogical environment, which is a generative experience management framework to give students helpful roles to learn strategies for coping with bullying. In this framework, VICTEC FearNot! is a virtual learning environment to help the students to tackle the problems and allow them to explore ways of dealing with problems. They use believable synthetic characters and narrative structure to create a close relationship between characters and students. In each episode of bullying, students interact with one of the characters on each occasion by offering the appropriate advice.

In [43], authors have developed a system that is called Thespian. Their approach provides the benefits of speeding up the development of Interactive Pedagogical Dramas (IPDs) [24], supporting open-ended interaction, achieving pedagogical and dramatic objectives as well as supporting quantitative metrics for evaluating the learners' achievement. The architecture of thespian uses autonomous software agents to control each character with their personality and the defined motivations as defined for the agent as goals. The ability of goal-driven agents that can autonomously select the appropriate actions by considering the current state of the world help them to act responsively to open-ended user interactions.

In [26], Bradford W. Mottauthors have developed a narrative centered, an inquiry-based system to promote exploratory learning in the field of microbiology. This system is designed for the students who desire to discover the origin of an unidentified infectious disease at the research station. In this approach they use of U-DIRECTOR architecture, which has been implemented in a narrative planner for Crystal Island [26, 27] to cope the uncertainty enforce by student actions.

In Teatrix [25], children gain the opportunity to create their storyboards of existing fairy tales. To prepare their story, they should choose three items: (a) scenes whereas they choose their desirable scenes and connect them just with a line; (b) props whereas the characters can perform more complex actions in the story world using the props; and (c) cast as the characters that are going to play it in the story world. The pedagogical agent acts as an implicit agent that is developed as a story director, which is responsible for the narrative guidance of the story as an intelligent module in this system.

Table 2.1: Comparison of LMS, ITS, Agent-based ITS, and PA

Comparison of Learning em ironments	Learning Environment	Database	Dynamism	Interactiveness	Intelligence	Adaptability
	LMS	✓				
	ITS		✓	✓		
	Agent-based ITS			✓	✓	
	PA		✓	✓	✓	✓

Table 2.1 investigates the various learning environments that are discussed in this thesis in terms of dynamism, interactive-ness, intelligence, and adaptability. The LMS as one of the basic platforms in online learning is the learning environment to provide students' registration and learning content management through the web. As it is explained in Table 2.1, there is no dynamic relationship between students and system in LMS. However, many students need the effective communication mechanism in order to use of an active tutor in their educational path. Online tutoring via the LMS can be successful by incorporating with an ITS. In ITS there is a method for delivering customized instructions to the students based on their abilities. At first, ITS promise of a adaptive learning experience, but in spite of promising they support the students' learning process only by a number of static algorithms. Therefore, artificial intelligence has used in the architecture of ITS, to design the intelligent system. This type of ITS, not only are independent from static decision making mechanism, but also are able to act autonomously. But despite of these advantages, this type of

ITS is unable to adapt itself with different students with different characteristic. As a solution, the pedagogical agents has been developed to play the role of a learning partner or a human-like tutor in a learning environment. The feature that distinguishes a PA from other agents is its adaptive functionalities but as it is clear from Table 2.1, it also does not satisfy all the criteria of students needs. Perhaps the most important hurdle to overcome is the difficulty to develop a system with whole the features that each of mentioned learning environments provides. In this thesis, we got helped from machine learning technology to construct a system to satisfy the students' needs when they are studying online regardless of where they are located.

Chapter 3

MDP-based Intelligent Tutoring System

In this Chapter, we briefly explain the essential parts of an advanced eLearning system including Learning Management System (LMS), Intelligent Tutoring System (ITS), AI-based ITS and Pedagogical Agent (PA). Then we introduce the proposed framework, which includes both LMS and ITS that closely cooperate. In this approach, we consider PA deploying a decision making mechanism to develop a user model. The proposed model serves as the main part of the proposed ITS. Moreover, this chapter consists of the model explanation and system workflow demonstration. Also, we introduce the Markov Decision Processes (MDPs) as a mathematical foundation. We continue the chapter with the relative experimental section to present the extracted outcomes from the collected data. We also explain the data set and the machine learning algorithms, which are used in this experiment as well as the discussion over the findings. A detailed comparison between the outcomes of different algorithms will conclude the section.

In recent years, students, parents, and governments have become increasingly excited about the online learning by decreasing costs and improving performance. LMSs are among the most productive methods in deployment of the elearning courses. LMSs are considered as web-based frameworks with the role to integrate with the academic community and manage student-related activities like content management, course administration, course registration and reporting. Using LMSs, teachers take a practical approach to address students' needs regardless of time and place by sharing files and providing guidelines. ITS, the core part of LMS, is a computer system which enables student monitoring in order to develop appropriate instructions by generating customized and personalized guidance to every individual student. There can be an effective interaction between human learner and ITS. The guidance provided by ITS and given to the students as a prompt makes it easier for them to focus on their weak points and improve their underprivileged skills in an effective way. It can also provide real-time data to instructors and developers looking to refine teaching methods.

Because of the educational concerns worldwide, there exist many significant studies about ITSs. Some studies defined ITS as a system that is capable of diagnosing and adapting to student's knowledge and skills. According to these studies, ITS is able to provide precise feedback when mistakes are made and able to present new topics when the student is ready to learn. There have been also frameworks developed to interact between ITS and the student using some models (i.e. User model, Ideal model, etc.). Some recent investigations propose AI-based ITS. The common belief is that the overall aim of developing AI is to enable the computer to be effective and act as a knowledgeable agent in the teaching and learning process. Nowadays Agent-based systems play a fundamental role in the development of pedagogical systems. Using agents in intelligent pedagogical systems provides a more efficient system. Therefore, some studies are done in this topic. In such studies, the aim is to infer the learner's emotional state in distant learning. The main functionality of the system is to collect data and interact with the learner while the learner's data acquisition component collects learner's data. Following similar objectives, some other research works use machine learning to recognize the learner's emotional behaviors (i.e. sadness, happiness, fear, etc.).

In this research, we provide a system which establishes an effective interaction between the ITS and human learners. A typical student registers in a course and participates in a learning session through the LMS. The ITS part of the system monitors the student in terms of knowledge, performance, time and other characteristics predefined in advance and stores them in an appropriate database. The decision making mechanism (the most crucial part of the ITS) needs the data to judge the student in a specific time interval and provide an appropriate guidance as a prompt. To accomplish more effective communication, the system should distinguish the unique characteristics of each human learner, generate a customized guidance for that specific person, and suggest an appropriate topic. To attain this objective, we have developed a PA empowered by machine learning capabilities for instituting smooth and practical interactions.

Our main goal in this study is to provide a more productive learning and understanding experience for students. To achieve this goal, we have introduced a system including both LMS and ITS, closely cooperating with each other. The ITS consists of four main parts: Pedagogical Agent, Decision Making Mechanism, User Model and our Markov Decision Process (MDP)-based tool for decision making. The LMS also incorporates four main parts: GUI, Self-Developed Model, Ideal Model, and a Database to store and retrieve data. The human learner can communicate with the system through the GUI. In the ITS part, we present The MDP-based component to form the PA's decision making mechanism aiming to make the user model closer to the ideal model. We propose the reward function as an artificial intelligence concept computed in the form of score relative to the target point (the goal). So the closer user model gets to the ideal model, the higher expected reward would be. We also carried out an experiment to collect data to apply appropriate machine learning algorithms in order to predict the performance level of the learners. The logistic regression and KNN methods are selected in this experiment due to the nature of the data. Considering the results obtained from these two supervised learning algorithms, we can calculate the performance and accuracy to select the most appropriate one.

3.1 The Model

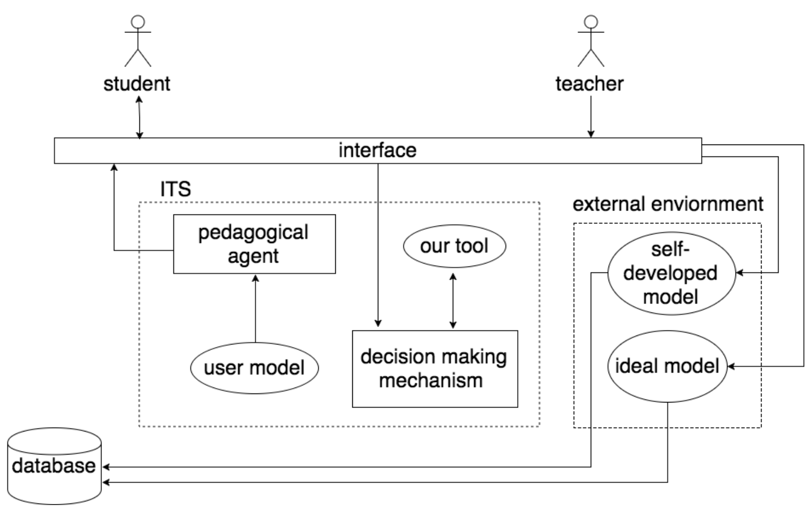

Figure 3.1: System general framework with relative components

3.1.1 Preliminaries

In this section, we introduce the proposed model and explain its components. There are different relevant pieces that are referenced in the framework illustrated in Figure 3.1 The model is an advanced LMS [3] empowered by an ITS deployed to provide real-time data to instructors and content developers looking to refine teaching methods. The guidance is provided given that the overall learning objective is already predefined. To efficiently maintain high quality guidance, there is a data processing center that closely cooperates with a decision-making mechanism to select best actions to be taken in any specific situation. The combination is therefore responsible to generate meaningful information to interact with students using an appropriate interface. The combined methods, alongside the effective content of the prompts, create an ITS that provides customized feedback to learners in order to improve their learning skills based on their input into the system. Some ITSs use intelligent agents to better perform artificial intelligence-based feedback and modulate the decision-making process to effectively apply changes and track the collected information while interacting with human learners. These agents are programmable and called Pedagogical Agents (PAs).

PAs provide practical solutions for ITS's objectives. They facilitate monitoring students, track information changes, customize and order information according to individual needs to help students reduce work overload and stress. They are helpful in teaching students to become effective and benefit from the offered content. With regard to the functionality of a PA, it receives (or seeks) input data from the environment, processes the data to find out the updates from the environment, and reacts to the environment with the appropriate action. The updates are recognized by comparing the environment status to the virtual environment (i.e. image in the form of user model) that the agent has created at the beginning of its life cycle, and by continuously updating the image to be synchronized with the outside world. In this domain of research, the input data is in the form of human-related activities, which defines the environment where the human is actively doing a work that involves the agent.

A typical PA is a rational intelligent agent that cooperates with the rest of the system components (i.e. database, interface, communication mechanism, LMS configuration system, etc.), processes the collected data, and provides guidance to human learners with respect to the predefined learning objective under which the agent has been programmed. The guidance is usually done through interactive communication with human learner. The communication is usually part of an ITS. The pedagogical agent is intended to acquire knowledge about the surrounding environment by receiving data from users in the form of students' activities while doing a learning activity. The agent then compares the latest environment status changes to update its image (virtual internal saved environment), from the outside environment. This comparison is of great importance since it indicates how good is the student's status at a specific moment with regard to time, knowledge level, order, reading status, and performance.

The core of the tutoring system is the decision-making mechanism, which is rather a widespread process. There are three sources of information that tutor's decision-making mechanism uses: (a) teacher's estimation of the ultimate goal; (b) system assessed user model that depicts student's status in learning activity; and (c) student declaration of self-assessed status in learning activity. Decision-making mechanism considers this data as an input which leverages the communication between the PA and the student. The communication is managed based on user model that the system assessed and the difference between the student's status and system-generic ideal status. Agents follow rational aptitude in dynamic decision-making, which involves algorithmic infrastructure and theoretical framework that will be explained in Section 3.2.

The theoretical foundation of this mechanism helps a pedagogical agent effectively communicate with human learners based on the processed input data via decision-making protocol as a part of ITS. Agents then effectively identify the moves that treat the best reaction towards the students with various learning characteristics. The parametric break down of user model elaborates details in deep level of granularity, which highlights the weaknesses and the approach to fix them. Agents then act confidently, however with certain level of uncertainty, and guide the students towards the predefined learning objective. This cycle is carried on till the end of the session. The session may consist of some reading materials, few quizzes and break. Finalizing each iterative interaction with student, the system-generic user model will be updated. Depending on PA decision, the learner will be either prompted according to the approach taken towards learning objective or it will tend to provide more time and not to interfere with improving students self-planning skills.

In agent's decision-making mechanism, we assume that the external environment is composed of two models: Ideal Model and Self-Developed Model. These models follow the same template and contain a list of certain parameters with defined range and assessment/update formulation. Using these parameters, the agent assesses learner's knowledge, time management, performance, and learning skills in the form of associative score. Ideal model is the imaginary student that depicts the best movement on the learning activity based on teacher's opinion. Therefore, before learning session launches, the teacher creates and updates the ideal model in order to define the target point using which learner is guided towards the learning objective. In the proposed model, the teacher influences the special user model known as ideal model. This ideal model represents the best user model, which as subject expert could experience the learning material given the predefined learning objective.

Outside the agent goal-identification mechanism, this ideal model is set with the teacher and the agent configures this special model to its internal system and becomes ready to guide students using this target point as benchmark. Therefore, all students are going to be guided using the same benchmark. As a result, our advanced LMS can compare students' performance and highlights their strengths and weaknesses in various pedagogical aspects.The teacher can also modify or customize the learning sessions according to new emerging needs before launching a new learning session following a new learning objective.

The Self-Developed Model is initially created by ITS and continuously updated by learner who indicates self-measured status of learning status while being engaged in a learning session. The Learner can always input new values for self-measured report set. However, the system prompts the student encouraging to report self-measure learning status based on some pre-defined points that the agent recognizes. The agent always compares the status of the user model with the one indicated in self-developed model. The rationale behind this comparison is the fact that the agent needs to identify substantial differences between these two models, which normally indicates student's confusion and promotes the importance of guiding prompts to be triggered. In some cases, such big different is indicative of agent's miscalculation of student-related user. In these cases, usually the agent increases its uncertainty level and the guiding mechanism acts more cautiously while interacting with a student with rare aptitude of learning.

3.1.2 The Workflow

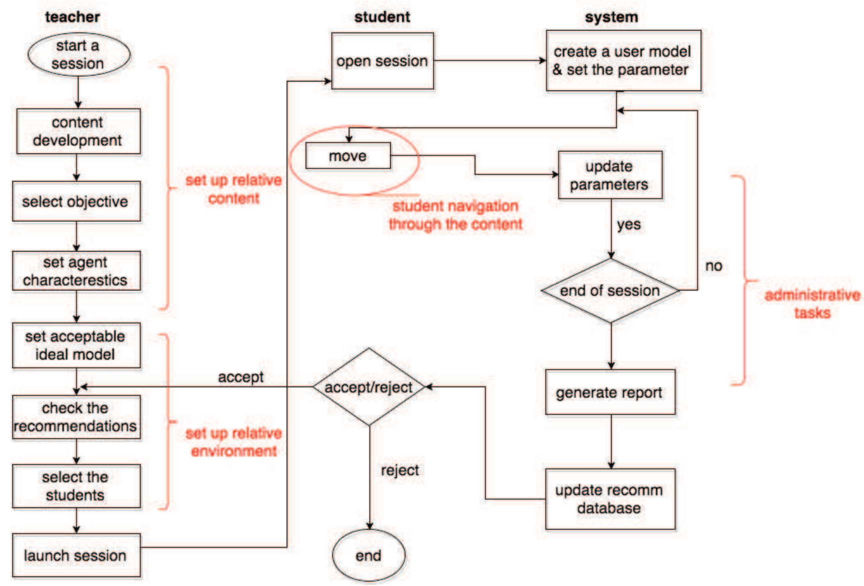

Figure 3.2: The proposed system framework

In this section, we explain a case study which demonstrates the work flow of a day-of-a-life scenario for a teacher-type user designing a learning session and engaging students into an interactive agent-based intelligent tutoring experience. The interactive learning environment is developed by deploying the rational pedagogical agents programmed and planned to follow pre-defined goals we explained in the previous section. The work flow elaborates on the relative data collected which represents the learning skills of the human learners. As shown in Figure 3.2, the teacher begins by creating the content (or modifying the previously developed content) to structure a base that is used for agent-based guidance.

To create a new learning session, the teacher selects a learning objective, which identifies the learning target the teacher is following. The learning objectives are inspired from Bloom's Taxonomy [70] and are in the form of improving students' learning, memorizing, problem-solving, and analyzing skills. In addition to these predefined and widely used learning objectives, the teacher can customize a new objective by setting new parameters or modifying existing structure of the parameter cube. Once the learning objective is set, the teacher can upload learning materials in the form of reading text blocks (that could contain a number of pages), quizzes in the form of pre-test, post-test, pop up questions, complete questions, and breaks to group learning materials into a learning modules. Obviously, there is no limit in terms of the number of text blocks or quizzes, but the teacher is aware of the overall learning activity time and fits to the learning objective already set at the beginning of the learning session design process.

By selecting each learning objective, the system associates a list of parameters that form pedagogical agents' aptitude in guiding human learners. However, the teacher can modify the parameters to promote a parameter for a specific reason. For example, in improving memorizing skill objective, the time management score is used to a certain extent when the agent guides the student. If the teacher promotes this parameter, the agent acts more cautiously regarding the time-related activities such as the time spent on reading a text block or taking a quiz. The extent to which the teacher is restrictive in any aspect (i.e. time, performance or planning) directly influences the relative parameter in a sense that the pedagogical agent can convey the guiding aptitude to students. Using the associative parameters and additive ones set by the teacher helps the agent to create the ideal model that is used as benchmark to guide the students doing learning activities at best performance. The ideal model is set once and remains static during the learning session.

The teacher can preview the session from the student standpoint and experience the pedagogical guidance the agent provides to apply necessary changes if needed. After providing appropriate corrections and necessary changes, the ideal model is shaped representing the idealistic manner, which the agent can adopt in order to guide the students towards learning objective. After launching the learning session, the student is notified through the user interface in the form of notification. The student become acquainted with the principal information about the learning session, such as the due date, session duration as the whole learning activity, number of breaks, learning objective, and the teacher note that highlights the important issues. Moreover, the student is informed about some statistics that represent the other students who are invited to take the same session. The statistics provide details regarding the percentage of the class who accomplished the session and the one who has started the learning activity but yet to be finished. The student then starts the session with complete information regarding the learning objectives, as well as the status of its classmates.

Once the student engages in the learning session, the pedagogical agent initiates a user model that depicts student's progress throughout the session. The user model is continuously updated as the system collects more information while the student is navigating through the reading text blocks and taking pre-tests, post-tests, completing quizzes as well as pop up questions. The aforementioned parameters in subsection 3.1.1 are updated according to the data collected through the user interface interacting with human learner. The updates in these parameters cause a new user model version that is recorded in the form of the numeric vector and represents the instantaneous status of the user in terms of learning progress. Holding the user model and recording continuous updates is a great resource for the agent to compare the instantaneous status of the user with the ideal model defined (or verified) by the teacher. However, in addition to what the system depicts for the student, there is another model that is generated by the student directly in the form of self-developed model. This model could be updated whenever the student inputs self-measured data that are indicative of student's progress in the learning activity. The agent does not have any control on this model expect that the student is encouraged to fill in such information on an occasional basis.

At the end, the agent also has access to what the student inputs as self-measured learning progress, and the comparison of the three models in a triangular shape is the core of the agent's decision-making before triggering prompts to the student to form effective guidance service. The details of such decision-making are explained in the following section as our main contribution. The process of guiding the student throughout the learning activity is done by our intelligent tutoring system protocol, which involves a lot of data processing features used by the pedagogical agent to perform effective guidance under specific learning objective. When the student is navigating between text-blocks and quizzes or taking breaks, the user model parameters such as reading status, knowledge level, order score, performance, and time score are updated accordingly. This process is carried on till the learner reaches the end of the session.

The ITS has different parts including pedagogical agent, decision-making mechanism, and our proposed model. The aim of PA is guiding learner to get closer to the ideal model as much as possible, by considering the self-developed model. The ultimate goal is to maintain a situation where the user model is at closest possible distance to the ideal model, which indicates the successful guidance done by PA. The ITS uses our proposed model and decision making mechanism to improve guiding student in a useful and accurate manner. Our framework receives data that is collected by the system as an input, applies some predefined mathematical functions to optimize the output, and then send it to the decision-making mechanism. By receiving this input Decision-making mechanism uses a comprehensive decision tree to choose the best and shortest way to guide the human learner to get closer to the ideal, by prompting timely and accurate.

When the session reached the due date, the system classifies the collected data to generate progress reports with extensive information about students' cognitive and non-cognitive skills. This information helps the teacher to make prudent decisions and make productive adjustment in upcoming sessions. Moreover, the system generates recommendations in the form of tips to students to take into account and become better learners by improving specific skills that the system has recognized in a customized and personalized manner. Upon teacher acceptance, the recommendations are sent to students to consider a relative session following the one the recommendation was based on.

Connecting the sessions would let the agent to retrieve the data from past session to use in dynamic decision-making mechanism explained in the following session. Once the recommendation is sent to the student, there are two options that the student can choose: accepting and rejecting the recommendation. By accepting recommendation, the student's name will be added to the recommended students' list automatically, that system offers to the teacher for the next session. However, it is still optional for teacher to add/ remove this particular student to the next session. By rejecting recommendation by student, there is no update in the recommended students' list, but teacher can see which recommendation has been rejected by student. The sequence of sessions followed by recommendations that encourage the student to take another session is called adaptive learning feature that we support in our proposed platform. However, without loss of generality, we skip details of this feature to move into the dynamic decision making protocol as our main module in the proposed system. In further details, we highly concentrate on our model, which configures agent's dynamic decision making core.

3.2 Theoretical ITS Mechanism

3.2.1 Mathematical Foundation

As described in previous sections, the primary objective of the ITS is to shrink the gap between the user model and ideal model. It is also important to evaluate the self-measured model and check out how close it is to both other models. Therefore, a successful ITS is the one that minimizes the (euclidean) distance between the two described models as much as possible. To reach this goal, the ITS decision making mechanism requires a deep and complete methodology to deal with this critical issue via the accurate mathematical mechanism. The deployed mechanism serves as an appropriate pre-programmed procedure to manage the decision making part by employing the precise parameter in a pre-defined path.

In this section, we propose a natural extension of general Markov Decision Processes (MDPs) to organize pedagogical agent's decision making mechanism on guiding human learners by taking into account the ideal model compared with instantaneous user model. This mechanism is managed through the extended MDP with partial access to the environment, given that interacting with human learners would generate irrational data which could influence agent's expectation on user model supported by self-measured model. The mechanism is defined by a tuple $< I, S, A, P, R, y^i >$. As a matter of fact, the tuple contains the snapshot of students' status evolving over time. It also provides the action which each particular student selects to move to the next state. In more details, each snapshot contains each particular student's user models, list of possible states, list of reasonable actions, the probability of each selected action at any time/state, the pre-defined reward, which is associated to each action, and list of the possible updates once the action is selected and the state transmission took place. The proposed ITS decision making mechanism uses the aforementioned tuple to process the status of the human learner with respect to ideal, user, and self-measure models. Using a proper classification tool (a machine learning method), the system is able to guide the students towards the best and shortest path to achieve the ideal result during a learning session. The details of the proposed framework used by the decision making module of the ITS is described in the following.

- I denotes the set of models including ideal, user, and self-measure;

- S denotes the finite set of states identifying various situations;
- A denotes the finite set of actions to manage navigation through different status;
- P denotes the state transition probability of transitioning from state s_0 to s_1 when the action a has been taken by the agent considering all models represented in I;
- R denotes the reward function, where $R(s, a)$ is the immediate reward for being in state s taking action a;
- y^i denotes the finite set of updates for each model i in I.

Lets elaborate more on the I parameter, which represents the user model set. In a nutshell, the ITS follows a goal to minimize the difference (euclidean distance) between different models. In further details, the ITS reaches out to the stable point, where the difference between the models are at their minimum case and cannot be any closer. That way, the system generated model (user model) is close enough to the ideal model. That phenomenon denotes the fact that the user is doing a good job, which is close enough to the ideal case originally generated by the knowledge expert (teacher in this case). Likewise, if the self-measured model is also close enough to the user model, it denotes the fact that the human learner is aware of his/her status in the learning session. Therefore, one of the top priorities held by the ITS is to guide students towards choosing actions that cause standing on the closest position to the ideal model. The ITS provides the update feature whereas teachers can update the ideal model to adjust to students prior to the beginning of a session. Such an update would influence the ideal model thresholds, states and actions. But at all time during the active learning session, the ITS takes the static ideal model as the benchmark to guide students to maximize their performance by diminishing the gap between the system evaluated model and the ideal model. The point of the self-measured model is to aware the human learner of their status evaluated by the system and run a dialogue to point out the differences that the student can be aware of and perhaps could do better for the rest of the learning session.

The learning session's stages as well as different status points that a typical human learner can experience are all collected as set of states. An example of a state could be $Quiz\_Finished\_Poor$, which denotes that the finished the quiz with poor score. Another example of the state is $Reading\_$

$Summary\_Fast$, which represents the state where the learner manages the summary page reading in fast pace. It is important to note that the states represents a map of possible scenarios that a learner could navigate through and reach out to different out coming points. However, the role of ITS is to guide the learner to navigate through optimal path where the goal is reached at minimum cost (that could be time and effort to read and do exercises). Therefore, there is a path that is caused by the human learner, which is the main reason of standing on a particular state (*i.e.*, $Quiz\_Finished\_Poor$) and the ITS is the means to navigate through the optimal path that is always end up with the learning session goal (starting from any user-made state). In other words, the ITS specifies a finite set of states (along the optimal path) for students by looking at the image that it builds from student's strengths and weaknesses. The proposed path is also affected by the ideal models' factors. In general, all learners start from the initial state and navigate through different paths by acting differently, and visiting different set of states. But at all time, the ITS guides the students to reach to the final goal state through the shortest path with maximum performance. The more the student chooses the correct action to go to the next state, the higher performance recorded for the student by the system as the optimal path is not deviated. To the best of the system's knowledge, that would end up in the best result that is defined by the knowledge expert (teacher in this case) as the ideal model.

As it is described in the previous section, there is an optimal path to reach the final state by navigating through different possible (static) states. The ITS defines the optimal path by looking at students' characteristics and the ideal defined model to provide a coherent way to reach out the maximum performance by meeting the ITS's standards. This is done by following the optimal path until a deviation occurs via the learner. Then recalculation is made to shrink the gap between the user model and ideal model as much as possible. Therefore, students are guided to follow the optimal path that ends up with the final state. The ideal navigating path (sequence of states-actions) is defined by the knowledge expert (teacher). Therefore, ITS always has the best path used as benchmark, which is produced by teacher to provide an effective navigation through the states by students while active in a learning session. Therefore, there is a relevant meaningful action related to each state that takes place with relatively high probability. For instances, an action could be $Downgrade\_the\_Quiz\_Threshold$. This action could be a valuable option for the state $Quiz\_Finished\_Poor$, but definitely an irrelevant action for other states like $Reading$ state. Thus, it is concluded that if a student takes the inappropriate action to move towards the next step, they will deviate from the final state and accordingly they are detoured via a new path to reach out to the final state, however, with longer time. To this end, it is important that the human learner reasonably chooses the actions to reach out to the ultimate state through an optimal path with high performance.

The fourth parameter in the proposed mathematical model is the probability set. The set identifies the probability distribution that is used to make gesture on students' actions in any particular state. In each state, there are a number of possible actions that are defined to provide the navigation between different possible states. It is now clear that the most important issue for ITS is guiding students towards the final state. The probability set is used to estimate a measure of confidence that a particular path is going to be used to get to the destination. While the student starts from the initial stage and moves to next steps, the ITS takes into account the relevant probability of selecting the optimal action in each state transition. Selecting the reasonable action would cause a legitimate state transition, whereas the selection of irrelevant action would cause deviation from the optimal path as well as a minor/major update in the rest of probability distributions. It is important to note that the probability distribution in each state is different from one human learner to another. This is the adaptability feature of the proposed model that could provide effective learning solutions personalized to human learner's learning aptitude. Moreover, every time that an action a is taken by student, the system makes the transition from state s_i to s_j and accordingly updates future state transition probabilities.

The fifth tuple member is the reward function R that denotes the immediate reward for being in state s taking action a. This function is used to encourage/discourage the system generated user model to stay with optimal path and reach to the ultimate goal by navigating throughout the optimal states defined by the knowledge expert (teacher). In further details, there is a vector of predefined rewards relevant to each state transition process, which is specified by ITS to each particular transition. The assigned rewards value identifies the extent to which the selected action is relevant to guide students to the next step. The impose of relatively high reward value would encourage the system to positively rate the selected actions and adversely, the impose of relatively low value would cause the system to negatively rate the selected action. Accordingly, the system is always alert to make necessary changes to detour the bad path back to an optimal path that ends up reaching to the ultimate goal. In fact, this is the juice of an adaptive system that is able to process the external data and optimally react to the environment with the high efficient action selection. Therefore, it is concluded that high reward value as a result of selecting reasonable actions denotes that the student is moving in optimal and relatively precise path towards the final state.

The last parameter in the proposed model is regarding the updates that the system would receive with respect to the three underlying models. The change in a model simply influences the distributions (probability $\&$ reward) and eventually cause the system to act dynamically and adapt to the environmental changes. Considering the tuple as the mathematical framework that is used as the main brain of the system before any action, we would like to boost the performance of the learning guidance process by maximizing the gained rewards while navigating through different states in the system. The objective would be then formulated to the following equation (Equation 1).

$$argmax_{\Theta} \sum_{t=0}^{\infty} R(s_t, a_t; \theta) \tag{1}$$

In this equation, the parameter θ denotes the joint-policy among different models (ideal, user, and self-measure) and t denotes the time steps on agent's move and guidance. The reward value is associated to the action a_t that is taken at state s_t given the policy (the probability distribution indicator) at time t.

Before we define the objective function to address the maximization problem stated above, we need to clarify agent's policy in regards to model i as θ^i parameterize that via π^i, λ^i, and γ^i as $\theta^i = (\pi^i, \lambda^i, \gamma^i)$. π denotes the strategy of choosing action and we define it in the following $p(a|b;\pi)$ shortened as $\pi_{a,b}$ identifies the probability of choosing action a when in state b. Therefore, selecting a particular action depends on the policy π at any moment of acting. In further details, the system computes the probability of each action taking place by considering its current information of the best choice to achieve the defined goal. Thus, the policy π looks at only to the next step, otherwise, the policy θ has a broad view of its environment. The policy θ does not only looks at the next step, but also considers the whole upcoming steps to find the optimal path towards the ultimate goal. Consequently, the policy θ computes the probability of each action by considering its current information of the best choice of whole upcoming steps. We know that, the agent has limited information in state S so, the policy π might assigns high probability to the action a, but accordingly to the policy θ, it might not be the best action. As a result, there are two different perspective view for these two policies. In some cases the policy π might relates a higher probability to a specific action but the policy θ consider a low probability because of their different perspective. Depends on the data (external data factors), the policy θ might change the policy π elements with different levels.

By updating the model, the agent's status might change from S_t to S_{t+1}, where $\lambda_{s_t,s_{t+1},y}$ identifies the probability.

Now we formalize the update procedure from one model (p) to another (q) during which the user model changes through the pedagogical agent's guidance. In other words agent's choice on the type of guidance directs the action to be selected and the move from p to q is taken place under the adopted policy Θ and validated by the student's reaction that is formed in observation y.

$$V(p,q,s) = \sum_{a,b} \pi_{a,p}\pi_{b,q}[R(s_0,a,b) + \sum_{s_1} p(s_1(s_0,a,b)) \times \sum_{p',q'} \lambda_{p',p,y}\lambda_{q',q,z} V(p',q',s_1)] \tag{2}$$

To maximize the expected reward on the agent's joint policy on different models, we need to select the optimal values for π and λ, where V reaches to its maximum value and lead to expected reward to get maximized.

In the aforementioned equation, the joint action causes the system to undergo different states that are not desirable. In the other words, the pedagogical agent calculates the relevant probability associated with each action by looking at each individual students' characteristics to move to the best possible state. As it is clear, the student might take the action to move to the next step, which is not acceptable one from the agent's points of view. As it is pointed out, the pedagogical agent will recalculate the action's probability by model's renewing. In the current situation, when the student chooses the action b to travel to the next step which is different to the proposed one (a) by the pedagogical agent, the agent will revise the actions' probabilities in the forthcoming path to draw a fruitful direction towards the eventual objective. In this equation $\pi_{a,p}$ denotes the action a that the student chooses when she/he stands in state p and $\pi_{b,q}$ denotes the action b that the student takes when she/he lives in state q.

Solving the Problem

Recently, we introduced a tuple to set up the pedagogical agent's decision making mechanism. This tuple is built of some finite sets of dynamic arguments like the desirable states to illustrate the possible situations that might be appeared for each individual student, reasonable actions on guiding students to move optimally between different states, various predefined models by system (user, self measured, ideal), state transition probability that presents the value which is assigned by the agent for each particular action, rewards function which plays a very significant role in guiding students toward the final goal and finally whole updates that might be happen for each model which causes the system to undergo the student's various actions. The key role of the pedagogical agent is applying a mechanism to address the human learner to the eventual step through the shortest path. This mechanism employs the reward function to perform its advice more productive. In further describes, the agent allocates the unique reward to each action which is taken by students to go to each upcoming states. The most appropriate action which is taken by the human learner, the system specifies a larger value as a reward for that particular student. As a result, the greater reward value means that the student has performed a better decision on choosing actions. Therefore, the maximize rewards show that the human learner follows the straightest, and the most optimal path towards the objective and stands in the closest position to the final step. Accordingly, as we clarified earlier the important objective is addressed to the maximization of reward that has a positive effect on directing students towards the predefined steps. Although, we are aware of how this problem could be solved, but this is hard to address because of the derivative complexity. As we know, the aforementioned tuple consists of some sets of parameters to process the student's status over time. So, to solve the problem we have to maximize the n-dimensional function in order to obtain the maximum of the reward function. Solving this problem enable the agent to distinguish between the students that follow the agent's guidance with the ones that don't care about the guidance. In this section we provide a known algorithm to solve the mentioned problem.

To maximize the expected value mentioned earlier, we deploy EM algorithm (Expectation Maximization Method). EM is a popular tool to solve the statistical estimation problems which consists of incomplete data or the problems that can be located in a equivalent form. It also uses in different motion estimation platforms. Generally, it uses as maximum likelihood estimation when some of the variables in the problem are unobserved. EM tries to guess a distribution for the unobserved data then for the given data, we calculate the value of each parameter that explains that data in the best way by maximizing something that is a lower bound on the actual likelihood function.

In our problem, EM does great in a sense that the increase in value is carried on till the convergence takes place. However we might not end up with global maximum point. We address this problem in experimental analysis.

Let X denote the observable variable and Z denotes the hidden variable. θ would be our model parameter . Then we argmax the probability sum with respect to θ.

$$argmax_{\theta} \log \sum_{z} p(x, z; \theta) \tag{3}$$

This is a hard problem to solve as it might go to a non-convex optimization problem with extremely high complexity. Solving this would end up with continues coordinate ascent in the parameter space. To avoid the aforementioned problem, we first compute the lower bound of the sum function and evaluate it asθ_i. $L(\theta_l)$ is defined in the following

$$L(\theta_l) = \sum_{z} p(z|x; \theta_l) \log p(x, z; \theta) - \sum_{z} p(z|x; \theta_l) \log p(z|x; \theta_l) \tag{4}$$

Using the lower bound θ_l an initial input, we identify better parameter estimate θ_0 through an interaction. It means that θ_0 is the local optimal point started from θ_l. The same interaction could be repeated until θ_l does not change to any better parameter estimate.

3.3 Relavant Related Work

This research is inspired by the some earlier proposals. Because of the importance of the students' educational system, many researchers studied to find the most effective factors on students' learning. Du Boulay & Luckin [56] focused on the modality and restriction of the teaching and learning approaches that is existed in 1980s. They also focused on the different perspective of learning approaches in the past and now. The authors believes the ITSs were less focused on attaining reflection and representing the student's meta-cognition in the past but Nowadays, that perspective has changed and they are focused on implementing diverse strategies that attain educational interaction and a more learner-centred approach, which encourage meta-cognition skills such as self-evaluation and self-explanation. Arroyo et al. [57] developed an ITS for the mathematics section of the Scholastic Aptitude Test (SAT) which has several distinctive features; help with multimedia animations and sound, problems embedded in narrative and fantasy contexts, alternative teaching strategies for students of different mental rotation abilities and memory retrieval speeds. The evaluations proved that students learn with the tutor, but learning depends on the interaction of teaching strategies and cognitive abilities. Analyzing the students' behavior and representing a unique guidance is one of the important part of our work. In connection with this subject Canfield [58] defined an intelligent tutoring system (ITS) as a system that is able to diagnose and adapt to student's knowledge and skills. According to this study ITS is able to provide precise feedbacks when mistakes are made and able to present new topics when the student is ready to learn. He believes that the intelligent tutoring systems are part of a new breed of instructional computer programs. As we explained before we used of some models (User model, Ideal model, ...) for development an effective interaction between the ITS and the student. Duchastel [59] defined the Tutorial Model and the Student Model which the Tutorial Model contains information about effective tutorial practices and monitors the status of the Student Model. It recommends appropriate dialogue between the ITS and the student. Both the Tutorial Model and the Student Model cooperate with each other to work effectively. Tutorial Model controls the learning process, applies different educational strategies according to the cognitive and essential personal features of the student, together with his/her educational progress. Recent proposed ITSs is classified into two types: AI-based models and Agent-based models. Undoubtedly in AI-based models some artificial intelligence is used and its value depends on the system. [Patterson (1990)]Patterson introduced the use of AI in interactive learning environment. Also He presents AI as a system that can understand a natural language and able to perform other types of feats that require human types of intelligence. He stress the importance of human based interactive learning environment because it involves students in active learning. Heffernan [69] focused on the importance of AI in the field of educational computing and asserts that has also undergone changes in educational system. He also stressed that the overall aim of developing AI is to enable the computer to be effective and act as a knowledgeable agent in the teaching and learning process. His major research has been the design of the so-called Intelligent Tutoring Systems (ITS) which require knowledge representations to provide models of the subject domain, the learner capabilities and the tutorial pedagogy. [61] To make the students' learning and understanding efficient, Freedman coupled ITSs with virtual laboratories or educational games. To achieve this aim he also introduces four main modules of ITSs: the domain model, the student model, the teaching model and the graphical user interface (GUI) which these modules can contain AI techniques, which use representations of the domain knowledge to understand the student's behavior and provide an intelligent response. One of the main ITSs' tasks is communication between pedagogical agent and student.

The challenge here is determining how messages are received and understood and how answers have to be formulated. To attain more effective communication it is necessary to look at the semantic meaning beyond the words, since the students input is still not completely understood. In the past ITSs had an irregular progress in areas such as provision and implementation of pedagogical actions and strategies, development of communication skills and implementation of theories of motivation and affect. Du Boulay & Luckin [56] used AI techniques to incorporate into computer tutoring the skills and abilities of human tutors required to overcome these issues. Nowadays Agent and Multi-Agents-based systems play a fundamental role in development of pedagogical systems. Using agents in intelligent pedagogical systems make the system more efficient and smoth. [62] For instance Neji & Ben Ammar introduced Emotional Multi-Agents System for Peer to peer E-learning (EMASPEL). EMASPEL recognises the students affective state using a web-cam and the analysis of facial features. This system communicates an effective response through an emotional Embodied Conversational Agent. Another study is done in this topic. Chalfoun et al. [63] implemented the Emotional Response Predictor Agent (ERPA) that aims to infer the learners emotional state in distant learning. The system is collecting data and interacts with the learner through the GUI while the Learners data acquisition component collects data of the learner. In this system some quizzes is embedded which through these quizzes the data related to the cognitive state is collected and it is compared to the expected and passing scores and the quiz start time. The ID3 decision tree algorithm calculates the data gain and selects the one with the highest gain. This data is used by the Rules Extraction component to generate the rules and to store them in the Rule Base. They considered a component as emotional reaction prediction which will use these rules to predict the students emotional state. Some studies are using multiple pedagogical agents [64] They introduce MOCAS architecture which has multiple pedagogical agents with diverse roles and attitudes. The student uses the Learner's agent to navigate and interact with Pedagogical agents in the 3D environment. The Communication interface is used by the student to communicate to other students and the learning content interface. The learning content interface adapts the domain knowledge to the student's cultural traits, is modular and enables the display of video files, questions and answers interfaces. Multiple Pedagogical Agents cooperate with the student's agent to provide pedagogical responses according to the teaching strategies and domain knowledge. Multiple Learner's Agents manage each students interaction with the client interface, which can be modified by the learners agent to adapt it to the student's cultural context. The World Agent monitors all the activity and determines the strategies of action followed by pedagogical agents. To build an intelligent tutoring system we can apply Machine Learning methods or we can use other not Machine Learning applications. We're going to mention some of the studies that used machine learning algorithms. Chaffar & Frasson [65] implemented a machine learning technique through rules in the Optimal Emotion Extractor module was used to identify the optimal affective state according the students personality. They defined a component as Emotion Inducer which was used to elicit the optimal state and Different GUIs with guided imagery, vignettes, music and images were used to induce the affective states of joy, anger, fear and sadness. After, they applied a pre-test and post-test before and after inducing the learners optimal state, which was identified through directly enquiring 137 students. Results showed that more than 28% of the students whose personality is Extraversion selected joy, 36% of the students whose personality is Lie selected confident, 29% of the students whose personality is Neuroticism selected pride and 50% of students whose personality is Psychoticism selected joy. Finally to select the optimal affective state identified by the students they used a Nave Bayes Classifier. [66] PrimeClimb is an educational game for teaching maths to students in an age range between 10 and 12. In PrimeClimb, the student's emotions are recognized using a student model implemented with Dynamic Bayesian Networks (DBN). This cognitive theory of emotion defines types of emotion according to their origin. They also defined specific intensity variables and a threshold value which each type of emotion is influenced by them. A DBN involving the causes and effects of emotion during learning, was defined too. This approach is focused on identifying immediate emotions, which evolve during time until they become long-term effective states or moods. In the following study they used an architecture including two models and a prototype-based machine learning to implements the ITS[S. Gross, B. Mokbel, B. Hammer, and N. Pinkwart]The authors defined the term ”solution space” including both a technical dimension on how to analyze such spaces of learner solutions using machine learning techniques, and a pedagogical dimension on how to give useful feedback to a learner's solution within such a space. To apply and test these feedback provision strategies in multiple domains they designed a middle-ware architecture that provides typical components of ITSs and prototype-based machine learning via standard interfaces. It implements the typical components used in ITSs: a student model stores information about learners and their actions; a pedagogical module contains pedagogical strategies implementing the above mentioned feedback principles.

In this research we introduced new technologies that allow students and teachers to interact better in the classroom as well as outside the classroom. The LMS is used by teacher to provide the students' needs regardless of time and place And ITSs is used to monitor the students and diagnose their important features to generate personalized guidance. We developed a pedagogical agent conforming AI methods to provide a smooth interaction for making the students education productive.

The raw data is collected by system while students were doing the learning session online through the system. There is a vector corresponding to each student which is included nine columns as nine features. The output also is calculated from the input by applying formula. By considering the nature of collected data which has a known result we utilized two supervised machine learning methods which are KNN and Logistic Regression. We used these algorithms to learn data for prediction label of the future input. We also classify the data into high and low performer to make some appropriate reports for teacher.

As we explained, there is a pedagogical agent for making decision in the ITS and we used the machine learning to make this agent more accurate. In this moment we developed an agent, figured it to the system and then launch the system. The data which is gotten from this launch is used as an input for machine learning methods. The results which came from applying machine learning is used to change the agent's threshold offline. In the future we're going to applying machine learning online, it means while the agent is working, the system applying the methods time to time and change the PA threshold and settings to improve the agent's guiding system. In this research we employed two machine learning algorithms to affect the PA. From now on we'll try more algorithms to see if we can get more efficient ways to achieve our goal.

Chapter 4

Experimental Results

As mentioned earlier, the intelligent agent plays a crucial role to speed of students move towards the final step through the most optimal direction. The significant point of the agent is that it picks the instructions according to each specific students' skills. This experiment has been done in the direction of maximizing the performance of the intelligent agent. In further details, we searched for a convenient algorithm in order to improve the decision making mechanism ran by the agent. In this experiment, we considered two different machine learning methods and applied them to the data which has been collected by the intelligent system during a learning session. The preferred methods are explained in the following and the implementation approach is described in details. We will further dig into the attributes of the data set then provide more details related to the structure of the collected data.

4.1 Data Set

In this section, we explain details of data collection phase and explore the outcome which is extracted from the deployed methodologies. We will also discuss the raw data structure by details and serve the techniques those are suitable to practice by considering the nature of the data set. To simulate an eLearning environment, 650 students have been invited as participants to interact with the intelligent agent based system for data collection. In this simulation, all of students have been attended to one unique learning session.

The intelligent agent collects data and regularly stores measures while students are engaged in the learning session. These measurements form the raw data, which is collected with respect to participants' learning skills and other features that represent students' performance.

Discussed measurements are computed by accurate mathematical functions in form of eleven features: (1) Travel Time; (2) Study Time; (3) Failures; (4) Family Relation; (5) Free Time; (6) Go Out; (7) Health; (8) Absence; (9) Grade 1; (10) Grade 2; (11) Grade 3. In this data structure $traveltime$ refers to the number of travels the student has gone. $Studytime$ represents that how many hours each student spends to finish the specified learning session. $Health$ means how many hours the student was healthy during the session. $Failures$ counts the number of students' fails in the relevant assessments. In this eLearning system, three types of assessment have been identified for evaluating students. First one is displayed at the beginning of the session just before the content and is called $pre - quiz$. The second assessment is $quiz$ that is shown by the system while the student is studying the content and the third one is $Post - quiz$ which is displayed at the end of the session just after finishing the session by the student. $Grade1$, $Grade2$, and $Grade3$ respectively refers to these predefined self-assessments.

Table 4.1: Summary of data set

	Important Factors	Percentage	Numeric
Review of Data Structures	Feature	-	11
	Cross Validation	-	5
	Training Data	70%	519
	Testing Data	30%	130
	Total	100%	650

To efficiently run the machine learning algorithms, we process the input data and use relevant features extracted from the raw data. So the input represents a set of eleven numbers assigned to each student as the result of his/her participation in the shared learning session. We also have labeled the output corresponding to the input to run the supervised learning algorithms. In more details, the output is a binary set of numbers indicating the status of the student (1 for high performer and 0 for low performer). For performing cross validation, we divide the data set into five folders that each folder consists of 519 training and 130 testing data. Associating data into two groups of training and testing is different in each folder to cover all the possible situations (see Table 4.1 for detailed breakdown of the data).

For a better view of the mentioned database in this study, we illustrate different charts in the following analyzing individual aspects of the data set. The graphical presentation of the data set depicts the relation between different attributes of the collected data. The labeled y value is a measure to distinguish the high performer students from low performer ones.

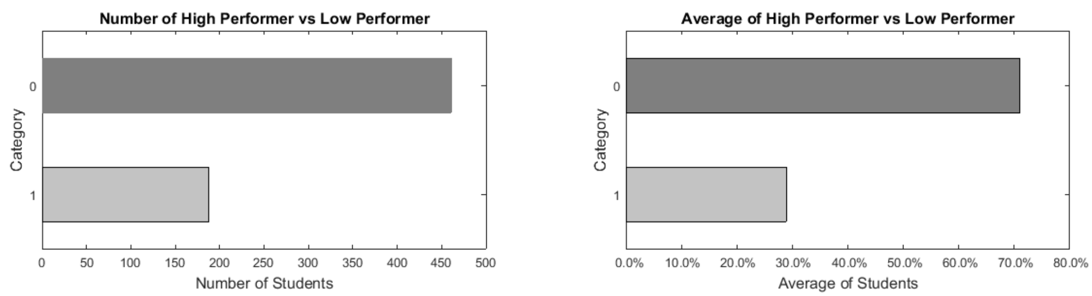

Figure 4.1: Distribution of number of high versus low performer learners as well as their average final grade

In Figure 4.1, the left plot compares the number of students who are labeled as the high performer student versus the number of students who are labeled as the low performer student. As this plot shows, 461 students have been assigned to zero and are represented as the weak group. In contrast, 188 of students are represented as the strong group holding 1 as y value. The right plot compares students' category but in terms of percentage. It is clear that around 71% of students stand in the weak category.

Plot 4.2 illustrates the average values of travel time, go out, and absences associated with different groups: labeled 0 and 1. It is clear that the absences associated with the weak group are relatively higher than the ones associated with the strong class. This is not the case for the average go out as the average for both classes are more or less the same. The travel time average is also different in both groups and is slightly higher in the weak group. It is overall the case that negative attributes are relatively valued higher in the weak group compare to the values in the string group.

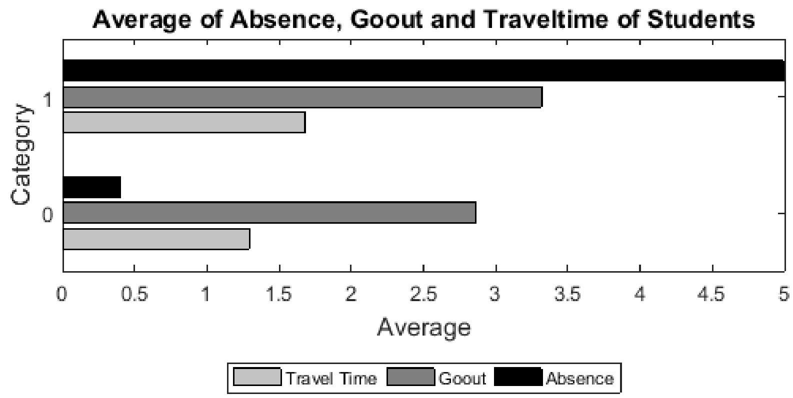

Figure 4.2: Distribution of dataset parameters to compare high performers versus low performers

The following plot (Figure 4.3) is the average health categorized into two groups. The string group relatively have a higher health level scored 3.75, whereas the average health associated with the weak groups is scored 3.45. This difference in health level score is not highly significant, but it is considered as it has the major impact on students' progress in studies.

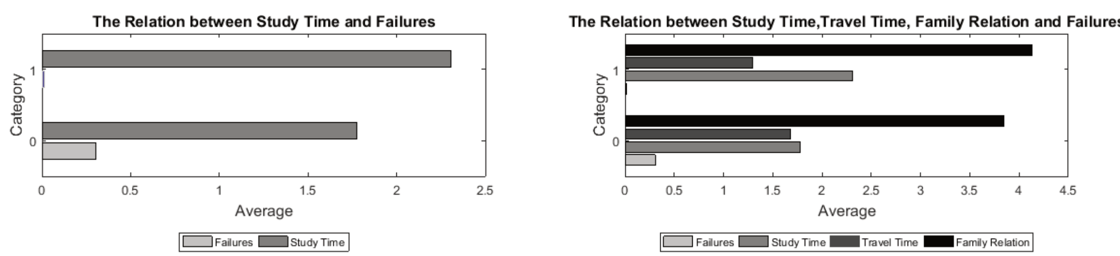

Figure 4.3: Comparison between corelation of parameters of the dataset in high performer and low performer groups

This difference is highly significant in the average of study time and the failure level. These attributes play an important role in students' classification into high performance (labeled 1) and low performance types (labeled 0). In contrast, the differences are not significantly different in the family relations, travel time and study time.

Figure 4.4, in the left plot, we illustrate the average grades associated with each category. It is shown as not significantly different in two classes, but overall the high performer students score more top grades. This derives us to the conclusion with the right plot, where the average grades

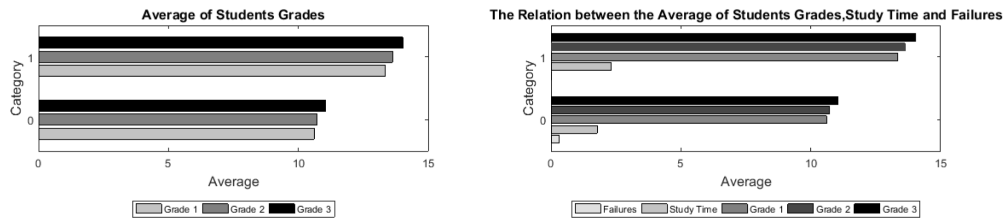

Figure 4.4: Continued comparison between corelation of parameters of the dataset in high performer and low performer groups

are compared and shown next to average study time associated with both groups as well as the average failure rate. Although there is not a significant difference between the two groups, the impact of classification should be analyzed using the two machine learning methods to theoretically distinguish the two groups from the intelligent agent's perspective. In the following, we dig into further analysis with more descriptive details about the methodology.

4.2 Experimental Protocol

This section provides details about the algorithm implementation, feature extraction, cross validation, result prediction, data labeling, the comparison of labeled to real data, measurements obtaining, performance-oriented curves plotting and the contrast of each method's measurements. The database explained in details in the previous subsection is ready to be used here. Both logistic regression and KNN use this database. We have implemented the logistic regression model to train the input data and build a model to classify the test data. We will explain the implementation of logistic regression in the following.

It is explained in details in the previous subsections that the input data contains associated feature vectors of eleven features representing students' learning skills engaging in an interactive learning session with the same content. We took 130 data points for testing and used 519 for training (see Table 4.2 for details). We also explained that we had folded five times to circulate the process over the one hundred thirty testing block. The first step is getting minimum weight vector by the $NewtonRaphson$ method, a reliable technique to solve the equations numerically by given training example $(X; Y)$ and the number of iterations. To get the best weight vector for the test part, we assume a vector of ones in the training section as an initial weight vector. The iteration affects directly the way by which we can achieve the minimum weight and stable position. That is, how many times the weights will be updated to reach the optimal weight? For the counter from one to iteration, we estimate the Y value by multiplication of weight and X. We then compute the hypothesis using a sigmoid function for each j where j is the size of Y estimated vector according to Equation 5.

$$hypothesis(j) = \frac{1}{(1 + exp(-x))} \tag{5}$$

We then use the results from the hypothesis function to generate new minimum best weights according to Equation 6, where weight is the old weight vector and hypothesis is the hypothesis result calculated in the previous step.

$$newWeight = transpose(weight) + \alpha * (transpose(Y - hypothesis) * X) \tag{6}$$

To get the hypothesis results in the testing part, we perform the same steps as in the training part, but the only difference is that the initial weight vector, while we are doing the test, is the one calculated in the training part. In other words, we did not calculate the weight vector in the testing step again. We use the one we obtained when we trained data to compute the hypothesis result in the testing part. For comparison, we labeled the hypothesis function result (which came from the multiplication of the weight vector calculated in the training part and testing X) using the threshold which can be updated anytime. In more details, to obtain the label vector, we make a comparison for each I between the hypothesis result and the specified threshold where I is the size of the hypothesis vector. If the hypothesis is bigger than the threshold, label is going to be one, otherwise is going to be zero. One point that must be considered here is choosing a suitable threshold which brings significant changes in the label vector, by changing the number of one and zero labels.

To recognize the importance of choosing an appropriate threshold, we tried an interval of numbers between zero and one that starts from zero and increases by one percent each time to realize which one affects the result reasonably and efficiently. As a result, we provide the number of zeros and ones labeled and stored in one vector called label vector. Up to this point, we used the initial

Table 4.2: Algorithm 1, Logistic Regression Data Breakdown

Algorithm 1: Classification with Logistic Regression	Important Factors	Number	Interval
	Training Data set Input	519	{0,...,1}
	Training Data set Output	519	{0,1}
	Testing Data set Input	130	{0,...,1}
	Testing Data set Output	130	{0,1}
	Iteration	4	{100,1000,10000,100000}
	Threshold	100	{0, 0.01, 0.02, ..., 0.09, 1}
	Alpha	4	{0.01, 0.1, 0.05, 0.5}

weight vector, a vector of ones for calculating the new and minimum weights and use them as initial weights in the testing part. We then predict the output by labeling the data. As described previously, the prediction is done by comparing the hypothesis, calculated by using the formula, to the considered threshold. Therefore, with this information, we can provide the error by estimating how many of our predictions are correct and how many of them are incorrect. To evaluate our prediction, we compare the real Y value of each X to the associated label. If both are ones, it means the real Y is one and we predict correctly. If both are zeros, it says the real Y is zero and our prediction is correct. If the real Y is one, but we the label is zero, or the real Y is zero, but we labeled it as one, then we conclude that the prediction is wrong. Finally, the result of correct and incorrect predictions are used to generate the performance and subsequently the accuracy. We have calculated the performance using the errors according to Equation 7:

$$performance = (2 * PR * RC)/(PR + RC) \tag{7}$$

where PR denotes the number of correct ones divided by the number of correct ones plus the number of incorrect ones and RC indicates the number of correct ones divided by the number of correct ones plus the number of incorrect zeros. We also calculated the accuracy using the errors according to Equation 8:

$$Accuracy = (correct1 + correct0)/(correct1 + correct0 + incorrect1 + incorrect0) \quad (8)$$

In the KNN methods, the data division between training and testing is the same as Logistic Regression, 519 for training and 130 for testing (see Table 4.2 for details). In the training part for each I between one to the size of the training data, we get the distance from each point to all other points, then sort and store these distances in one matrix. To get one point distance to other points, we used Equation 9:

$$resultTot = resultTot + (valueITest(k) - valueIInput(k))^2 \quad (9)$$

This equation is repeated for K times where k is between one and the size of data features (columns), to consider the whole features which are eleven in this experiment. Predicting the Y value for each trained data is the most critical phase, which is done after building the distances matrix. For prediction, we calculated the sum of distances of all the data from one to c where c is the data set size. We then divide this value by c each time and compare the result to the underlying threshold. As explained in the previous section, there is an interval of numbers between zero and one that starts from zero and increases by one percent each time we tried them to select the best threshold.

To predict the output as we described previously for each c, we considered 5 numbers including (5, 10, 20, 30) for k to analyze the results. If the result of dividing the real Y values' sum (from 1 to k) by k is bigger than the predefined threshold, the prediction will be one; otherwise, it will be zero. Then for computing the error, the labeled vector is compared to the real Y values. If both have the same value, then the error will be zero, but if they have different values, the error will be one. Excepting the cross validation which folded the data five times, there are two other assortments in the KNN method, but not in the logistic regression one. As explained above, the first assortment is since K is forty in this experiment for KNN, four number have been selected to study. In other words, these numbers represented what nearest neighbor we were working on. For instance, if K was five, it means we are calculating the errors for 5 nearest Neighbors. We performed the whole processes to achieve the error and the necessary data in the testing part to compute the performance, accuracy and plot the performance related curve.

Table 4.3: Algorithm 2, K nearest neighbors

Algorithm 2: Classification with K nearest neighbors (KNN)	Important Factors	Number	Interval
	Training Data set Input	519	{0,...,1}
	Training Data set Output	519	{0,1}
	Testing Data set Input	130	{0,...,1}
	Testing Data set Output	130	{0,1}
	Threshold	100	{0, 0.01, 0.02, ... , 0.09, 1}
	Random Numbers (second round)	519	{0.7, ... ,1.3}
	Random Numbers (third round)	519	{0.5, ... ,1.5}

4.3 Steps for Deployment of Machine Learning Methodologies

In the previous section, we introduced two machine learning algorithms as supervised learning methods. In this part, we continue in depth explaining the mechanism of supervised learning ran by these methods and how they label the outputs. The first algorithm is logistic regression and the second one is KNN. Logistic regression is one of the most popular machine learning regression algorithms and KNN is one of the instance-based algorithms. Like the other regression methods, logistic regression encompasses modeling the relationship between variables, which is continuously being updated by an error in the predictions. KNN, like all the other instance-based methods, is a decision making method based on instances or points of training data, which seems to be more important than others.

4.3.1 Logistic Regression

In logistic regression as a regression model, there are two different types of variables: dependent variable and independent variable. In this study, the independent variable is the input and the dependent variable indicates the output that can take only two Y values 0 or 1. The value 1 represents a high performer student, in contrast, 0 introduces a low performer student.

The process starts with training the data for calculating the weight measurements. To fulfill this aim, the process is using an initial weight per student equal to one in the training part. The weights (for all the students) will be updated by doing some mathematical process over the training data and using the sigmoid function. The updated weights will be used through the next step, which is the testing part. These weights are used for extracting the Y estimated value by applying the sigmoid function on the testing data. Results taken from the sigmoid function are used as a basis for decision making with regard to the labeling data. Finally, the function searches to find how many output data including 0 and 1 have been labeled correctly and how many of these labels are not correct. The performance and accuracy have been calculated by considering this information. The whole process is run for both training and testing parts separately, and we analyze the results for both.

4.3.2 K Nearest Neighbors

Although KNN is usually used for both classification and regression, in this thesis, it has only been used as a classification method. Because of being in the supervised learning category, it is necessary to use known outputs. So the input and output used in the logistic regression part to perform regression are also used by KNN, but this time for classification. Also, we considered five-fold-cross validation on the data like what we did for logistic regression.

In this approach, the goal is finding the nearest neighbors due to the value of k. For instance, if k was five, we should look for those five points which have the least distance from the others. To achieve this goal, we find out the distance between each point to all other points and sort them to extract the k closer points. By applying some mathematics on the real Y value and comparing them with intended threshold, each data point will be labeled as either zero or one. The threshold can affect the variation of correct and incorrect predictions. We repeated this process three times by multiplication of two intervals of numbers generated randomly using a predefined external function. In the first round, we just did the comparison with the k closest data points by considering the original distances without any modification. For the second and third rounds, the extracted distances are multiplied by the random numbers between 0.7 to 1.3 and 0.5 to 1.5 and then compared with the appropriate threshold. We tried divers intervals of numbers to get the wider range of distances to see how it affects the predictions in order to find out the best threshold. Finally, the function searches to find how many outputs including 0 and 1 are labeled correctly and how many of them are not. By considering this outcome, the performance and accuracy are calculated. This process is run for each training and testing part separately, and we analyze the obtained results.

The ultimate goal of this paper is improving the agent performance to guide students accurately with detailed and relevant tips. To achieve this goal, we decided to apply two different machine learning technics on the data which was collected before by the agent. Performance and accuracy of methods which are used in this study are two significant measures to evaluate and select the best algorithm to improve the decision making mechanism of the intelligent agent in terms of performance. These two measurements are calculated by the number of correct and incorrect predictions of each method, then in the next steps, the results came from each method compared together to find which one has better performance and is more accurate and apply it to the agent. Therefore two algorithms were selected with different approaches, logistic regression is used for studying regression and k Nearest Neighbor for performing classification. We did some experiments by the objective of comparing the measures achieved by the logistic method to that obtained by KNN method to apply the one with higher performance to the agent. For this reason we compare the performance of two methods in different form. We perform five-fold-cross-validation for logistic that each folder is composed of different sort of training and testing data. It is considered five hundred nineteen of inputs for training and one hundred thirty of them for testing. Each experiment is done for each folder separately and it is kept how many correct predictions of one, how many incorrect predictions of one, how many correct predictions of zero and how many incorrect predictions of zero as an information that we need to plot the obtained curves for further analysis purposes.

In KNN method we do five-fold cross-validation as we did in logistic regression. It is clear there are five folders with different sort of data for training and testing part. For KNN there is one more assortment which is done by k that it can be any number between one to five hundred nineteen because of training data set size. We applied the method to all five folders and subsequently for each k from one to five hundred nineteen. Four number as K is selected randomly which they were five, ten, twenty, thirty. The experiment is done with these numbers using k Nearest Neighbor algorithm and kept the useful information like how many correct predictions of one, how many incorrect predictions of one, how many correct predictions of zero and how many incorrect predictions of zero to plot performance curves. ”Therefore, for each experiment which is related to one of the five folders, we plot and interpret one chart that is including training curve for both logistic and KNN algorithms to simplify the comparison and perception the performance for both algorithms simultaneously.” In this way, we can choose the best method that brings us to goal.

4.4 Discussions

For the comparative trial, we followed a general procedure comparing model’s accuracy precision, and recall. In testing the accuracy of a classification rule, it is widely known that error rates tend to be biased if they are estimated from the same set of data as that used to construct the rules. Since we did not have a very large data set, we used all instances as training and test data set with 5 folded groups. The predicted and true classifications on the test data give an unbiased estimate of the error rate of the classifier. Before the comparison, we analyzed basic model assumptions, such as the homogeneity of covariances, which is relevant to classification methods such as the logistic regression. There are few measures that capture the essence of a model’s usefulness, among which, we use Receiver Operating Characteristics (ROC) curves to compare the efficiency of the two models.

4.4.1 Logistic Regression

The binary logistic regression provides good results using all possible variables (all-possible model). Because logistic regression cannot deal with non-numeric attributes these were omitted by the procedure of identifying the false positives. It is well known that linear regression models, including logistic linear regression models, become unstable when they include many predictor variables relative to the sample size. This translates into poor predictions when the model is applied to new data. Newton raphson method is an established procedure for this purpose and actually produced the best results among all classifiers. In the following, we present the results in the form of ROC curves with area underneath the curve (AUC). Typically, the higher the AUC value is, the more accurate and high performance the classifier is operating.

There are two sets of results for this classifier that present results for different groups of data as

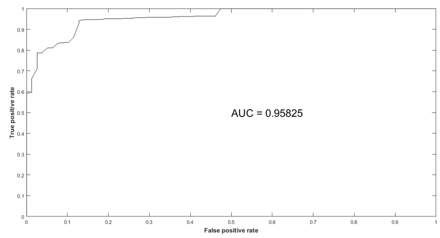

Figure 4.5: ROC Curve for $Group1$ using Logistic Regression with learning rate of 0.01 and iteration of 100 rounds

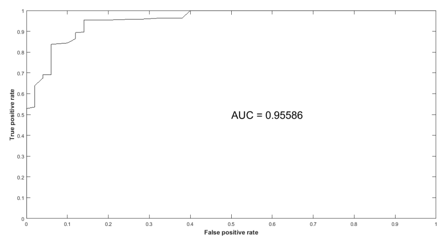

Figure 4.6: ROC Curve for $Group2$ using Logistic Regression with learning rate of 0.01 and iteration of 100 rounds

well cross different functions used for classification and labeling the y-values. The results show that logistic regression using newton raphson method performs good in maintaining relatively high area underneath the curve. The results are clear in two individual ROC curves illustrated in Figures 4.5 and 4.6 as well as the group curves separated by learning rate of 0.01 for groups 1 and 2 illustrated in Figures 4.7 and 4.8.

4.4.2 k-Nearest Neighbor Method

KNN is an instance-based learner and does not produce a model. Another disadvantage of this method is that it is computationally intensive to classify large data sets. For example, classification using our training and test data set took several hours. Nevertheless, KNN was particularly successful as a classification method in different models, and it is interesting to see how this method performs compared to logistic regression or decision tree inducers.

A crucial parameter for KNN is k, the number of neighbors to be used for classifying a new instance. A general rule of thumb is to use the square root of n, where n is the number of training instances. We found 5-NN to perform already very well for the training data set. However, we

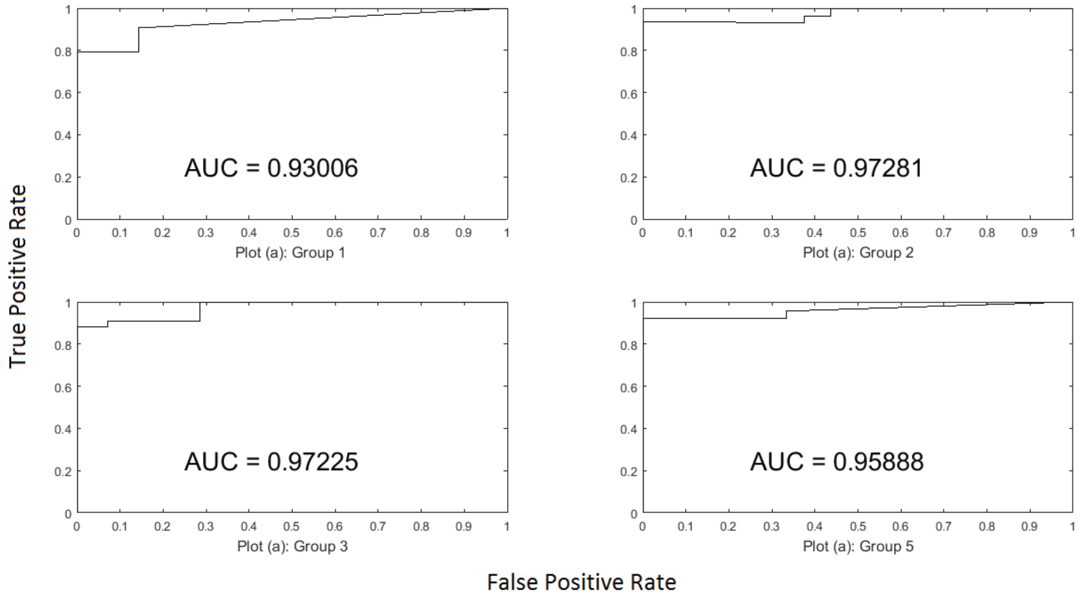

Figure 4.7: ROC Curve for using Logistic Regression with learning rate of 0.1 and iteration of 100 rounds

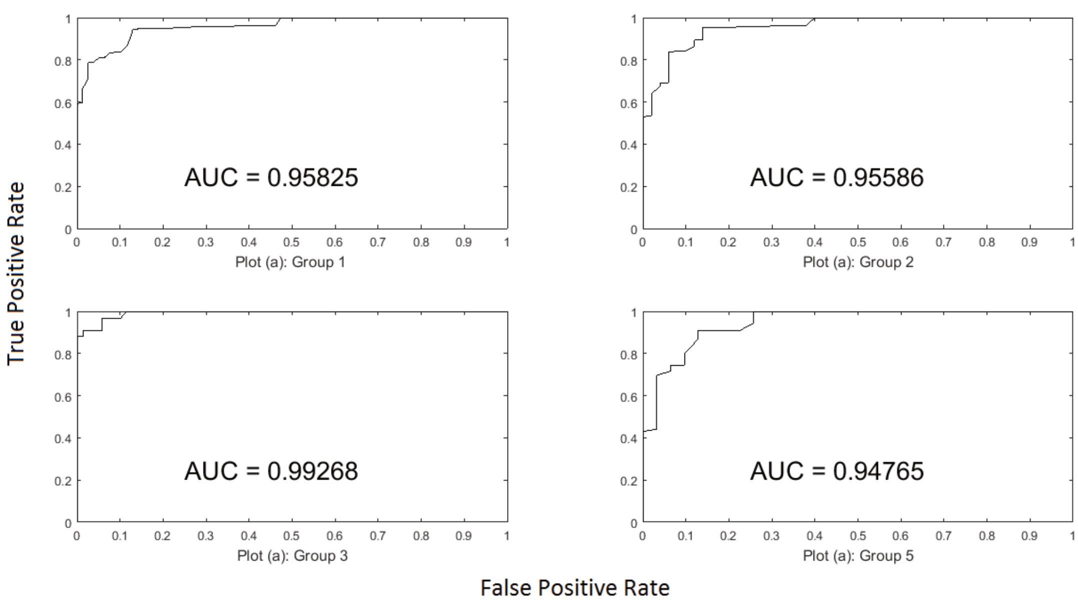

Figure 4.8: ROC Curve for using Logistic Regression with learning rate of 0.01 and iteration of 100 rounds

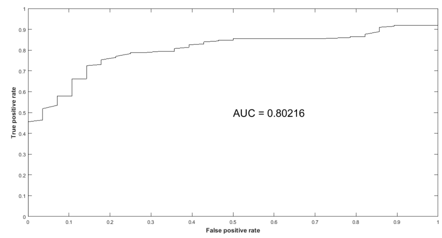

Figure 4.9: ROC Curve for $Group4$ using 5-NN

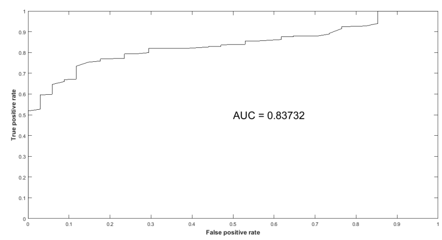

Figure 4.10: ROC Curve for $Group4$ using 10-NN

compared the results from 5-NN to 10-NN, 20-NN, and 30-NN. As shown in singular ROC curves illustrated in Figures 4.9, 4.10, 4.11, and 4.12, the high KNN presents better AUC, but not much of difference among different groups. Also as shown in group ROC curves illustrated in Figures 4.13 and 4.14, the ROC outputs are relatively same with small range of difference for AUC.

Finally in this conclusion, k for kNN classification for the testing set ranges from 5 to 30 shown in Figure 4.15. Most misclassification are relatively low and the optimal k was found to be 30. Conducting 30-NN in the training set, we achieved relatively high area underneath the curve 87%. The overall accuracy of the testing set was is high, however, with high complexity using high k. On the contrary, logistic regression performs particularly well (Figure 4.16). AUC is 0.9858 being pretty close to 1. For instance, with a learning rate of 0.1, the correct classification rate or the

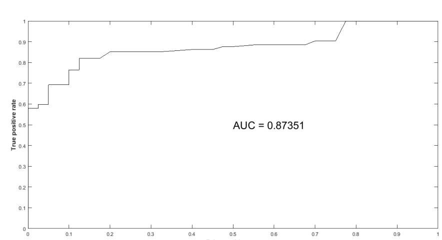

Figure 4.11: ROC Curve for $Group4$ using 20-NN

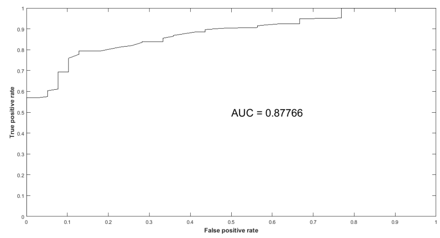

Figure 4.12: ROC Curve for $Group4$ using 30-NN

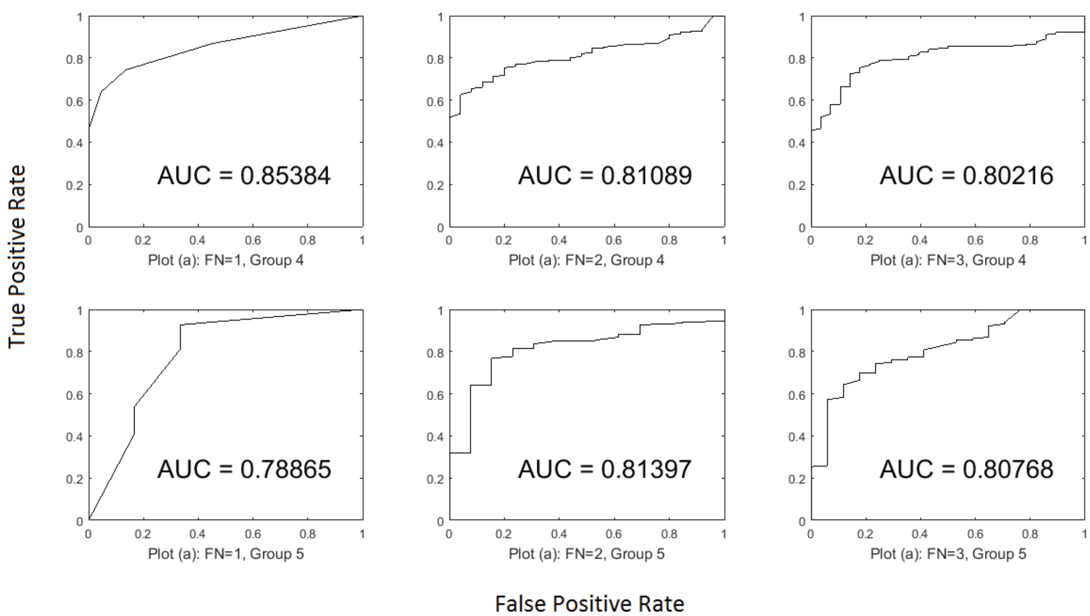

Figure 4.13: Six 5-NN ROC Curves three from $Group4$ and three from $Group5$ classified with labeling functions

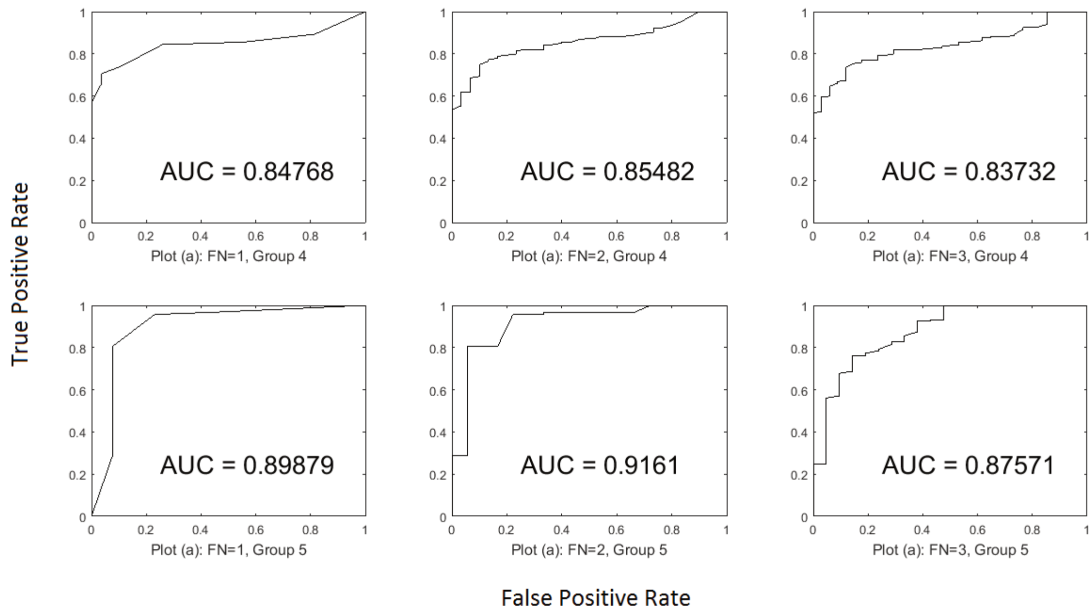

Figure 4.14: Six 10-NN ROC Curves three from $Group4$ and three from $Group5$ classified with labeling functions

accuracy is 97% at 25% error rate. That is certainly higher than the result of 30-NN.

Generally, we expected machine learning technique would outperform traditional statistical technique that should have been correct since the dataset to be analyzed was large enough to train from the sample. kNN is non-parametric while both are automatically cross-validated. However, Logistic Regression technique is still the best after data cleansing in our case. This is achieved even with less complexity as training a model is done via newton rophson method through limited number of iterations. The scalability is not a negative factor as this method can extend the training module considering more features and higher number of data points.

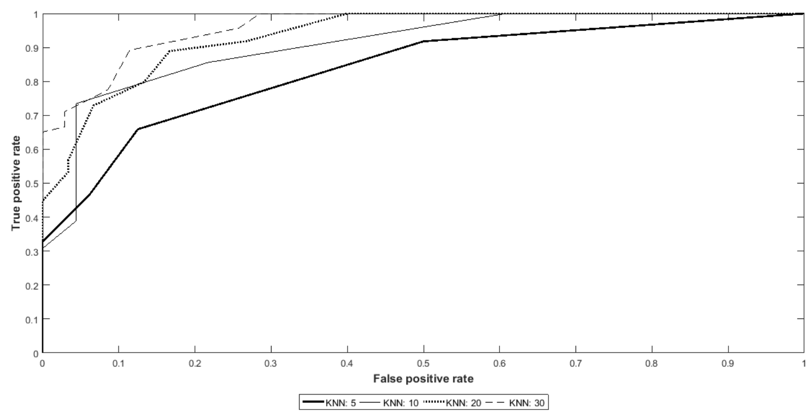

Figure 4.15: Four kNN ROC Curves associated to $Group2$

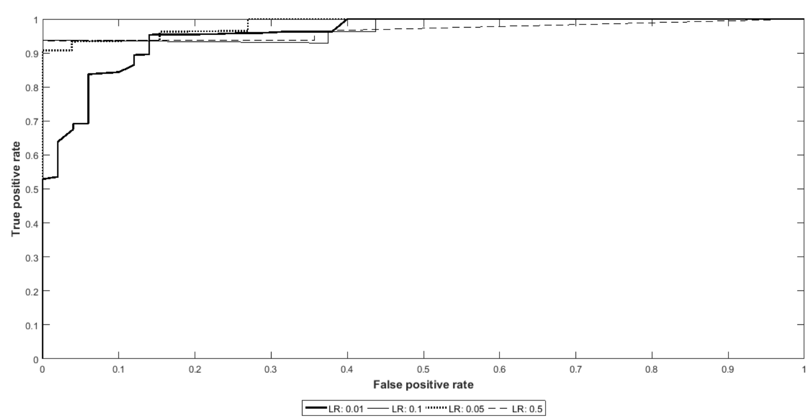

Figure 4.16: Four Logistic Regression ROC Curves associated to $Group2$

Chapter 5

Conclusion

This thesis is about studying and analyzing the ITS component in eLearning systems in order to make the learning process more productive. To achieve this aim, we proposed an eLearning system which includes both LMS and ITS. Using the LMS, students register to the system and access the learning content and quizzes. The LMS consists of four main parts: (1) GUI; (2) self-developed model; (3) ideal model; and (4) database. The ITS also includes four parts: (1) PA; (2) decision making algorithm; (3) user model; and (4) our MDP-based tool. On the other hand, the ITS provides an interactive environment to serve a purposeful communication between students and the system. The central part of the suggested ITS is a PA that captures students in terms of some predefined attributes in order to generate the relevant advice. As a rational intelligent agent, the PA benefits from AI techniques to allow students and teachers to communicate in a more efficient manner. For the implementation of the AI methods, we used MDPs to formalize the decision making process. For a realistic judgment of students' achievement, we also analyzed three kinds of models: user model, ideal model, and self-developed model. The eventual target of the ITS is mentoring students in order to minimize the distance between the ideal model and user model by looking at the self-developed model.

We later designed a robust decision making algorithm, a crucial parts of the PA. This algorithm generates the applicable precise instruction by analyzing each student's unique aspects through some predefined decision trees. Thus, the primary power of the introduced ITS is recognizing students' individual characteristics and manage them toward the eventual objective. Therefore, the database which has been deployed in the LMS infrastructure is needed to record the human learners' specific measurements over the time.

We then introduced our MDP-based model. The main power of this model is getting the input and then accordingly applying some predefined mathematical functions to obtain the optimized output. This result is employed to improve the decision making procedure of the ITS. Consequently, we modified the form of the established decision trees to improve the performance of the decision making algorithm in order to make closer the user model to the ideal model.

The input data used in our model is collected by the system while students were performing the learning session. We applied some mathematical functions to extract the appropriate output to use in our proposed framework and machine learning method as well. By considering the nature of the collected data, we selected two kinds of supervised machine learning methods: KNN and logistic regression. We applied both methods for distinguishing high performer students from low performer ones in this study. We analyzed the results of both approaches and then choose the strongest one to improve the decision making mechanism procedure in order to increase the ITS performance.

Bibliography

[1] B. S. Bloom, M. D. Engelhart, E. J. Furst, W. H. Hill and D. R. Krathwohl. Taxonomy of educational objectives: The classification of educational goals. Handbook I: Cognitive domain. New York: David McKay Company, 1956.

[2] B. N. Abdullateef, N. F. Elias, H. Mohamed, AWS. A. Zaidan and B. B. Zaidan. Study on Open Source Learning Management Systems: A Survey, Profile, and Taxonomy. Journal of Theoretical and Applied Information Technology, 82(1): 93-105, 2015.

[3] N. Azura Adzharuddin and L. Hwei Ling. Learning Management System (LMS) among University Students: Does It Work? International Journal of e-Education, e-Business, e-Management and e-Learning, 3(3): 248-252, 2013.

[4] Aruna. R and S Prakasam. Enhancing Cloud based E-Learning using Knowledge Sharing System. International Journal of Computer Applications, 84(9): 26-30, 2013.

[5] Md. Anwar Hossain Masud and X. Huang. An E-learning System Architecture based on Cloud Computing. World Academy of Science, Engineering and Technology, 6(1): 2-21, 2012.

[6] P. Ardi. Promoting Learner Autonomy through Schoology M-Learning Platform in an EAP Class at an Indonesian University. Teaching English with Technology, 17(2): 55-76, 2017.

[7] S. Bahrami. Conceptual Graphic Design and Interaction Design of Learning Management System ATutor. Indian Journal of Science and Technology, 8(3): 263-269, 2015.

[8] S. Charoenwet and A. Christensen. The Effect of Edmodo Learning Network on Students' Perception, Self-Regulated Learning Behaviors and Learning Performance. Proceedings of the 10'th International Multi-Conference on Society, Cybernetics and Informatics (IMSCI), 297-300, Orlando, USA, 2016.

[9] H. Jin Cha, Y. Se Kim, S. Hee Park , T. Bok Yoon, Y. Mo Jung and J.H. Lee. Learning Styles Diagnosis Based on User Interface Behaviors for the Customization of Learning Interfaces in an Intelligent Tutoring System. Proceedings of 8'th International Conference on Intelligent Tutoring Systems (ITS), 513-524, Jhongli, Taiwan, 2006.

[10] B. Cheung, L. Hui, J. Zhang and S.M. Yiu. SmartTutor: An Intelligent Tutoring System in Web-based Adult Education. Journal of Systems and Software, 68(1): 11-25, 2003.

[11] C. Conati and X. Zhao. Thet Thwin. Building and Evaluating an Intelligent Pedagogical Agent to Improve the Effectiveness of an Educational Game. Proceedings of the 9'th International Conference on Intelligent User Interfaces (CADUI), 6-13, Funchal, Portugal, 2004.

[12] M. R. Elabnody. A Survey Of Top 10 Open Source Learning Management Systems. International Journal of Scientific & Technology Research, 5(9): 7-11, 2016.

[13] A.T. Ernst, H. Jiang, M. Krishnamoorthy and D. Sier. Staff Scheduling and Rostering: A Review of Applications, Methods and Models. European Journal of Operational Research, 153(1): 3-27, 2004.

[14] B. Gauch and G. Biswas. Behavior Changes Across Time and Between Populations in Open-Ended Learning Environments. International Conference on Intelligent Tutoring Systems, isbn-978-3-319-39582-1, 187-196, 2016.

[15] N. Hanna and D. Richards. A Collaborative Agent Architecture with Human-Agent Communication Model. Proceedings of the 1st International Conference on Cognitive Agents for Virtual Environments (CAVE), 70-88, Valencia, Spain, 2012.

[16] L. Hall, S. Woods and R. Aylett. FearNot! Involving Children in the Design of a Virtual Learning Environment. International Journal of Artificial Intelligence in Education, 16(4): 327-351, 2006.

[17] A. Heirdsfield, S. Walker, M. Tambyah and D. Beutel. Blackboard as an Online Learning Environment: What do Teacher Education Students and Staff Think? Australian Journal of Teacher Education (Online), 36(7): 1-16, 2011.

[18] H. Htoon and M. Mie Thet Thwin. Mobile Agent for Distributed Information Retrieval System. Proceedings of the 5'th International Conference on Electrical Engineering/Electronics, Telecommunications and Information Technology (ECTI-CON), 169-172, Krabi, Thailand, 2008.

[19] H. Jung and B. Jeong. Decentralised Production-distribution Planning System using Collaborative Agents in Supply Chain Network. The International Journal of Advanced Manufacturing Technology, 25(1): 167-173, 2005.

[20] N. R. Jennings. An Agent-based Approach for Building Complex Software Systems. Communications of the ACM, 44(4): 35-41, 2001.

[21] D. B. Lange and M. Oshima. Seven Good Reasons for Mobile Agents. Communications of the ACM, 42(3): 88-89, 1999.

[22] G. Liu. The Application of Intelligent Agents in Libraries: A Survey. Program: Electronic Library & Information Systems, 45(1), 78-97, 2011.

[23] W. Lewis Johnson and J. C. Lester. Face-to-Face Interaction with Pedagogical Agents, Twenty Years Later. International Journal of Artificial Intelligence in Education, 26(1): 25-36, 2016.

[24] S. C. Marsella, W. Lewis Johnson and C. LaBore. Interactive Pedagogical Drama. In Proceedings of the 4'th International Conference on Autonomous Agents (AGENTS), 301-308, Barcelona, Spain, 2000.

[25] I. Machado, R. Prada and A. Paiva. TEATRIX: Virtual Environment for Story Creation. In G. Gauthier, C. Frasson, K. VanLehn (eds.), Proceeding of the 5'th International Conference on Intelligent Tutoring Systems (ITS), 464-473, Montreal, Canada, 2000.

[26] B. W. Mott, S. W. McQuiggan, S. Lee, S. Y. Lee and J. C. Lester. Narrative-centered Environments for Guided Discovery Learning, Workshop on Agent-Based Systems for Human Learning in Conjunction with 5'th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), 22-28, Hakodate, Japan, 2006.

[27] B. W. Mott and J. C. Lester. U-director: A Decision Theoretic Narrative Planning Architecture for Storytelling Environments. In Proceedings of the 5'th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), 977984, Hakodate, Japan, 2006.

[28] J. Mehrabi and M. SadatAbtahi. Retracted: Teaching with Moodle in Higher Education. Journal of Social and Behavioural Sciences, 47(C): 1320-1324, 2012.

[29] K. J. Mock. Hybrid Hill-climbing and Knowledge-based Methods for Intelligent News Filtering. Proceedings of the 13'th National Conference on Artificial intelligence (AAAI), 48-53, Portland, Oregan, USA, 1996.

[30] T. T. Nguyen and G. Armitage. A Survey of Techniques for Internet Traffic Classification using Machine Learning. IEEE Communications Surveys & Tutorials, 10(4): 56-76, 2008.

[31] R. Jamili Oskouei and N. Moradi Kor. Proposing a Novel Adaptive Learning Management System: An Application of Behavior Mining & Intelligent Agents. Intelligent Automation & Soft Computing, 23(2): 199-205, 2017.

[32] C. J. Petrie. Agent-based Engineering, the Web, and Intelligence. IEEE Expert: Intelligent Systems and their Applications, 11(6): 24-29, 1996.

[33] M. D.Perez and M. Medallon. A Move to Moodle: A Perspective of Academics in the College of International Tourism and Hospitality Management. LPU-Laguna Journal of Multidisciplinary Research, 4(3): 49-71, 2015.

[34] M. O. Riedl and A. Stern. Believable Agents and Intelligent Story Adaptation for Interactive Storytelling. Proceedings of 1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE), 1-12, Darmstadt, Germany, 2006.

[35] M. O. Riedl, A. Stern, D. M. Dini and J. M. Alderman. Dynamic Experience Management in Virtual Worlds for Entertainment, Education, and Training. International Transactions on Systems Science and Applications (ITSSA), Special Issue on Agent Based Systems for Human Learning, 3(1): 23-42, 2008.

[36] M. O. Riedl, A. Stern, D. M. Dini and J. M. Alderman. Believable Agents and Intelligent Story Adaptation for Interactive Storytelling. In Proceedings of the 3rd International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE), 1-12, Darmstadt, Germany, 2006.

[37] J. Ronghua Ouyang and N. Stanley. Theories and Research in Educational Technology and Distance Learning Instruction through Blackboard. Universal Journal of Educational Research, 2(2): 161-172, 2014.

[38] J. R. Chen, S. R. Wolfe and S. D. Wragg. A Distributed Multi-Agent System for Collaborative Information Management and Sharing. Proceedings of the 9'th International Conference on Information and Knowledge Management (KMIS), 382-388, Madeira, Portugal, 2000.

[39] P. Rani, C. Liu, N. Sarkar and E. Vanman. An Empirical Study of Machine Learning Techniques for Affect Recognition in HumanRobot Interaction. Pattern Analysis and Applications, 9(1): 58-69, 2006.

[40] I. Roll, V. Aleven, B. M. McLaren and K. R. Koedinger. Improving Students' Help-seeking Skills using Metacognitive Feedback in an Intelligent Tutoring System. Learning and Instruction, 21(2): 267-280, 2011.

[41] S. Sharma, S. Ghorpade, A. Sahni and N. Saluja. Survey of Intelligent Tutoring Systems: A Review on the Development of Expert/Intelligent Tutoring Systems, various Teaching Strategies and Expert Tutoring System Design Suggestions. International Journal of Engineering Research & Technology, 3(11): 37-42, 2014.

[42] A. Scerbakov, M. Ebner and N. Scerbakov. Using Cloud Services in a Modern Learning Management System. Journal of Computing and Information Technology, 23(1): 75-86, 2015.

[43] M. Si, S. C. Marsella and D. V. Pynadath. THESPIAN: An Architecture for Interactive Pedagogical Drama. Proceedings of the 12'th Conference on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology (AIED), 595-602, Amsterdam, Netherlands, 2005.

[44] S. Maizura Sahimi, F. M. Zain, N. A. N. Kamar, N. Samar, Z. A. Rahman, O. Majid, H. Atan, F. Soon Fook and W. Su Luan. The Pedagogical Agent in Online Learning: Effects of the Degree of Realism on Achievement in Terms of Gender. Contemporary Educational Technology, 1(2): 175-185, 2010.

[45] N. L Schroeder and O. O. Adesope. A Case for the use of Pedagogical Agents in Online Learning Environments. Journal of Teaching and Learning with Technology, 1(2): 43-47, 2012.

[46] A. S. Sicat. Enhancing College Students' Proficiency in Business Writing Via Schoology. International Journal of Education and Research, 3(1): 159-178, 2015.

[47] J. SeelyBrown, R. R.Burton and A. G.Bell. SOPHIE: A Step toward Creating a Reactive Learning Environment. International Journal of Man-Machine Studies, 7(5): 675-696, 1975.

[48] D. Schneider, P. Synteta and C. Frt. Community, Content and Collaboration Management Systems in Education: A new chance for socio-constructivist scenarios?. Proceedings of the 3rd Congress on Information and Communication Technologies in Education, 26-29, Rhodes, 2002.

[49] P. Stone and M. Veloso. Multiagent Systems: A Survey from a Machine Learning Perspective. Autonomous Robots, 8(3): 345-383, 2000.

[50] M. Vala, P. Sequeira, A. Paiva and R. Aylett. FearNot! Demo: A Virtual Environment with Synthetic Characters to Help Bullying. Proceedings of the 6'th International Joint Conference

on Autonomous Agents and Multiagent Systems (AAMAS), 1381-1382, Honolulu, HI, USA, 2007.

[51] R. Warawudhi. The Evaluation of Edmodo in Business Reading Class. International Journal of Information and Education Technology, 7(2): 153-158, 2017.

[52] N. Wang, W. LewisJohnson, R. E.Mayer, P. Rizzo, E. Shaw and H. Collins. The Politeness Effect: Pedagogical Agents and Learning Outcomes. International Journal of Human-Computer Studies, 66(2): 98-112, 2008.

[53] W. R. Watson and S. Lee Watson. An Argument for Clarity: What are Learning Management Systems, What are They Not, and What Should They Become? TechTrends, Springer Verlag, 51(2): 28-34, 2007.

[54] W. Chen and T. Boen. A Personalized RSS News Filtering Agent. Proceedings in Applications and Innovations in Intelligent Systems XV, ISBN 978-1-84628-224-9, 321-326, 2008.

[55] L. Zhang, W. Burleson, M. Elena Chavez-Echeagaray, S. Girard, J. Gonzalez-Sanchez, Y. Hidalgo-Pontet and K. VanLehn. Evaluation of a Meta-tutor for Constructing Models of Dynamic Systems. Computers & Education, 75(1): 196-217, 2014.

[56] B. du Boulay, and R. Luckin, Modeling Human Teaching Tactics and Strategies for Tutoring Systems. International Journal of Artificial Intelligence in Education, 12(3): 235-256, 2001.

[57] I. Arroyo, C. Beal, T. Murray, R. Walles and B. P. Woolf. Web-Based Intelligent Multimedia Tutoring for High Stakes Achievement Tests. International Conference on Intelligent Tutoring Systems, 468-477, Macei, Alagoas, Brazil, 2004.

[58] W. Canfield. ALEKS: A Web-Based Intelligent Tutoring System [ProQuest Education Journals.].Mathematics and Computer Education, 35(2), 152, 2001.

[59] P. Duchastel. Intelligent Computer Assisted Instruction Systems: the Nature of Learner Control. Journal of Educational Computing Research, 2(3), 379-393, 1986.

[60] N. T. Heffernan. Intelligent Tutoring Systems have Forgotten the Tutor: Adding a Cognitive Model of an Experienced Human Tutor. Proceeding of CHI 98 Conference Summary on Human Factors in Computing Systems, 50-51, Los Angeles, California, USA, 1988.

[61] R. Freedman. What is an Intelligent Tutoring System? Intelligence 11(3): 15-16, 2000.

[62] M. B. Ammar, M. Neji, A. M. Alimi and G. Gouardres: The Affective Tutoring System. Expert Systems with Applications. 37(4): 3013-3023, 2010.

[63] P. Chalfoun, S. Chaffar and C. Frasson. Predicting the Emotional Reaction of the Learner with a Machine Learning Technique. Workshop on Motivational and Affective Issues in ITS, International Conference on Intelligent Tutoring System (ITS), Jhongli, Taiwan, 2006.

[64] E. Blanchard and C. Frasson. Motivation and Evolutionary Pedagogical Agents. In Workshop on Motivational and Affective Issues in ITS. 8th International Conference on ITS, 54-61, Jhongli, Taiwan, 2006.

[65] S. Chaffar and C. Frasson. Inducing Optimal Emotional State for Learning in Intelligent Tutoring Systems. In Intelligent Tutoring Systems, 7-21, Springer Berlin, Heidelberg, 2004.

[66] C. Conati and H. Maclaren. Empirically Building and Evaluating a Probabilistic Model of User Affect. User Modeling and User-Adapted Interaction, 19(3): 267-303, 2009.

[67] T. Chen, A. MdYunus, W. Ali, W. Zah and A. Bakar. Utilization of Intelligent Tutoring System (ITS) in mathematics learning. International Journal of Education and Development using Information and Communication Technology, 4(4): 50-63, 2008.

[68] W. L. Johnson, J. W. Rickel and J. C. Lester. Animated pedagogical agents: Face-to-face interaction in interactive learning environments. International Journal of Artificial Intelligence in Education, 11(1): 47-78, 2000.

[69] N. T. Heffernan and K. R. Koedinger. An intelligent tutoring system incorporating a model of an experienced human tutor. Proceedings of the 6th International Conference on Intelligent Tutoring Systems, 596-608, Springer, Berlin, Heidelberg, 2002.

[70] R. A. Ibtihal and M. S. Oqlah. Using Blooms Taxonomy to Evaluate the Cognitive Levels of Master Class Textbooks Questions. English Language Teaching, 8(5): 100-110, 2015.