Journal of Universal Computer Science, vol. 27, no. 10 (2021), 1001-1025
submitted: 2/4/2021, accepted: 10/9/2021, appeared: 28/10/2021 *CC BY-ND 4.0*

A new approach to identify dropout learners based on their performance-based behavior

Rochdi Boudjehem
(LabSTIC Laboratory, 8 May 1945 University, Guelma, Algeria,
https://orcid.org/0000-0002-9998-4833, boudjehem.rochdi@univ-guelma.dz)

Yacine Lafifi
(Computer Science Department, 8 May 1945 University, Guelma, Algeria,
https://orcid.org/0000-0001-8232-4196, lafifi.yacine@univ-guelma.dz)

Abstract: Distance learning environments are increasingly offering more comfort to both learners and teachers, allowing them to carry out their academic tasks remotely, especially in critical times where it is difficult, or even dangerous, to bring these actors together in one physical place. Nevertheless, These same environments are complaining about the massive dropout numbers among their learners. Therefore, designing new intelligent systems capable of reducing these numbers becomes imperative. This paper proposes a new approach capable of identifying and assisting endangered learners experiencing difficulties by monitoring and analyzing their behavior inside the e-learning environment. By building dynamic models to follow the learners' current situation, the proposed approach could intervene autonomously to save learners identified as struggling. Relying on distributed artificial intelligence instead of humans to closely monitor learners within distance learning environments can be very effective when identifying struggling learners. Furthermore, targeting these learners with early enough and carefully designed interventions can reduce the number of dropouts.

Keywords: learning difficulties, dropout learners, learner traces, difficulties detection, early warning system, intelligent agent
Categories: I.2, J.4, K.3, L.0, M.4
DOI: 10.3897/jucs.74280

1 Introduction

In traditional classrooms, the teacher is supposed to know if one or more of his learners are not receiving their instruction well and could drop out. Sometimes, it takes only a few minutes of conversation to save some of them. However, with the expanding use of distance learning, the physical interaction between teachers and learners is becoming less frequent. Decreased physical interaction between the learner and his peers and teachers can have severe consequences on him, especially when he runs into problems. In this case, no one is around to notice his condition and step up to help him.

Fortunately, new intelligent tools are continuously being integrated into modern learning systems, allowing them to automatically detect and even predict such scenarios. Furthermore, new learning theories evolve every day, focusing on the learner as the main actor and transforming the learning process into a more adaptive, personalized, and custom-made operation. Moreover, having early enough feedback from learners can help understand their situation and predict their potential fall, giving instructors enough time to act sooner, preventing them from falling. In addition, allowing learners to be aware of their current skill level and warning them about likely future threats can have a decisive effect on preventing them from dropping out. Systems with such capabilities are often called: Early Warning Systems (EWS)[Sandoval et al., 2018, Howard et al., 2018, Waddington et al., 2016].

Through the use of means like Learning Analytics (LA) algorithms, it is possible to analyze the traces left by the learners to identify among them who are most likely to drop out. Consequently, let the instructors choose between various strategies to communicate with these learners to help them improve their performance and thus increase learner retention.

At this point, we may ask some questions:

- In distance learning environments, where we have no prior knowledge about the learner, can we rely mainly on his performance-based behavior to detect if this learner is facing difficulties or not?
- If so, can an early intervention save those learners from dropping out?
- Moreover, How efficient this intervention can be to reduce the number of dropouts inside those environments.
- Finally, what is the best technique to adopt to achieve that goal?

We try to answer those questions by proposing a novel agent-based approach capable of detecting and assisting at-risk learners based on their detected behavior. The proposed method comprises autonomous subsystems in constant collaboration and holds intelligent agents charged to perform specific tasks and have multiple triggers.

This paper is organized as follows:

Section. 2 presents a literature review on learners' dropout detection. The proposed approach is described in Section. 3. In Section. 4, the conducted experiment is presented alongside results and discussion. Finally, the general conclusion, work limitations, and future works are highlighted in Section. 5.

2 Related works

In recent years, there has been extensive use and considerable interest in using e-learning systems, especially in crisis times like the Coronavirus pandemic, even though these systems have some drawbacks like the significant number of learners' dropouts which turned out to be a severe problem. Furthermore, even though this problem is well known in traditional learning, the reasons behind it can be utterly different in distance learning. For example, issues of isolation, disconnectedness, and lack of technical mastery may be factors that can increase the ratio of dropout among remote learning students and drive them to leave online courses [Willging and Johnson, 2019]. It was Richardson-Koehler and his co-authors who first apply the social constructivist theory to study at-riskiness in education [Richardson-Koehler et al., 1989]. After that, many works have been conducted to identify or predict at-risk learners who may face difficulties during their journey to acquire knowledge.

It is essential to mention here that we did not include "Massive Open Online Courses" (MOOCs) in our Literature Review search, as it is beyond the interest of this work that focuses on distance and hybrid education. The majority of the works that we have reviewed used Learning Analytics (LA) to identify or predict at-risk learners or foresee their outcomes.

Recently, LA has attracted increasing attention thanks to its analytical abilities to provide real-time reports and summaries about learners' performance and behavior. This information can be used to identify and focus on under-performing students, allowing educational institutions to increase their learners' retention [Dietz-Uhler and Hurn, 2013].

We found a considerable number of developed e-learning systems that are based on the use of LA. For example, "iMoodle" is an EWS that can predict at-risk students then intervene to save them [Denden et al., 2019]. "INSPIREus" is another system capable of analyzing and visualizing some specific indicators in the form of detailed views within an adaptive educational hypermedia system [Papanikolaou, 2015]. In that direction, [Azcona and Casey, 2015] have designed a Virtual Classroom Environment (VLE) for learning the Assembly programming language. The proposed VLE is a real-time dashboard that can provide teachers with instant feedback on the successful and unsuccessful compilations of their learners. Finally,"Course Signals" is a web-based system that provides real-time feedback to students while measuring their engagement based on their interactions with the Purdue Learning Management System called "Blackboard Vista" [Jayaprakash et al., 2014, Pistilli and Arnold, 2010].

Each one of the reviewed works has its purpose and uses a different set of indicators. To understand the relationship between dropout-related works, we have grouped them according to their respective "Purpose of Research" objectives and the "Used Indicators".

2.1 Purpose of Research Works

In Table. 1, we have used the "Purpose of Research" as a criterion for grouping the found scientific works covering the topic of at-risk learners. For each work, we have listed which indicators have been used. Results show that 36% of the works focused on identifying learners who are or will be "at-risk," while 29% of the works tried to predict the outcomes of the learners rather than simply identifying the at-risk ones themselves. In addition, 15% of the selected works aim to analyze and visualize the learners' current status in real-time to both learners and instructors. Systems offering that kind of functionalities are sometimes called "dashboards."

2.2 Used Indicators

Even though scientists tried to achieve many goals, they almost relied on similar predictors. We analyzed, then grouped research works that focused on detecting or predicting struggling learners to see which ones were the most significant indicators or factors used to identify "at-risk" or learners. We grouped the works into two categories: performance-based and behavioral-based indicators.

In the first group, we found the work of Sandoval and his co-authors, who stated that the "students' grade point average (GPA)" was the most relevant indicator among 36 other indicators, followed by "the school in which the students were enrolled" as a moderately relevant indicator [Sandoval et al., 2018]. In the work of [Howard et al., 2018], the authors affirm that "continuous assessment" is the best indicator among three categories: "students' background information," "students' engagement," and "continuous assessment results."

Purpose of Research	Works	Used Indicators	Nb.	Pcg.
Identify or predict at-risk learners	[Von Hippel and Hofflinger, 2021]	Performance	9	36%
	[Mubarak et al., 2020]	Engagement, Time Series, Persistence		
	[Sarra et al., 2019]	Performance, Motivation and Resilience		
	[Casey and Azcona, 2017]	Concept reuse		
	[Kuzilek et al., 2015]	Demographic, Traces		
	[Abu-Oda and El-Halees, 2015]	Traces		
	[Jayaprakash et al., 2014]	Demographic, Performance, Assessment		
	[Yukselturk et al., 2014]	Demographic, Self-Efficacy, Readiness, Locus of Control, Previous experience		
	[Lykourentzou et al., 2009]	Demographic, Performance, Assessment		
Predict learner outcomes	[Aggarwal et al., 2021]	Academic History, Demographics, Financial	7	28%
	[Baneres et al., 2019]	Performance		
	[Denden et al., 2019]	Performance-based Behavior		
	[Howard et al., 2018]	Demographic Information, Performance, Assessment		
	[Sandoval et al., 2018]	Demographic, Performance, Assessment		
	[Waddington et al., 2016]	Performance, Academic History		
	[AL-Malaise et al., 2014]	Performance, Academic History		
Analyze and visualize learners' behavior	[Kokoç and Altun, 2021]	Interaction with Dashboard, Performance	3	13%
	[Azcona and Casey, 2015]	Traces		
	[Papanikolaou, 2015]	Cognitive and Social indicators		
Describe dropout learners' behavior	[Lakhal and Khechine, 2021]	Technological, Demographic , Social, Psychological	2	8%
	[Zhou et al., 2020]	Academic History, Interaction Traces, Time Series, Financial		
Compare predictive models	[Maldonado et al., 2021]	Profit Metrics	2	8%
	[Marbouti et al., 2016]	Traces		
Measure learner engagement	[Toti et al., 2021]	Performance, Behavioral	2	8%
	[Hussain et al., 2018]	Traces		
		Total	25	100%

Table 1: Dropout-related works grouped by "Purpose of Research"

In the other group, we found the work of Aggarwal and his co-authors who found that demographic indicators such as "age", "gender", "location",or "family income" are more significant in predicting the student's outcome at an early stage..[Aggarwal et al., 2021]. Furthermore, [Tarimo et al., 2016] affirm that "engagement" and "learning speed" are the best indicators for the final course grades . You and his co-authors have identified the "regular study" as the most pertinent indicator of the learners' performance, same as to "submission delay" and "proof of reading" of the course [You, 2016].

Both performance-based and behavior-based indicators are good indicators. Each one of them provides a significant amount of valuable information that can be used to report on the learner's situation. Therefore, we have chosen to combine them both. We intend to closely monitor the learners' behavior while they are performing their online activities. This behavior Analysis can provide a better perception of the learners' status, such as their punctuality, persistence, or even readiness.

Unfortunately, we only found one work [AL-Malaise et al., 2014], that combined Machine Learning with Multi-Agent System technology to predict the outcomes of the learners. Even though the nature of the detecting or predicting operation requires to monitor all the actors' interactions individually require the use of distributed and autonomous intelligence to obtain reliable results in a brief time, the only kind of artificial intelligence presented In most of the studied works, is centralized artificial intelligence like Machine Learning algorithms or some personalized intelligent algorithms.

3 Agent-based At-risk Learner's Detection Approach

The main objective of our work is to be able to detect and assist learners who are experiencing difficulties or are most likely to drop out. The proposed detection process may rely mainly on the Learner's performance-based behavior while studying inside the remote learning environment with no prior knowledge about his previous outcomes. The proposed approach should meet the following objectives:

1. Build for each learner a model that reflects his performance-based behavior inside the learning environment.
2. calculate a set of indicators using the available information within the built model.
3. Assess and rate each learner's situation using a color-coded base (Green, Yellow, Orange, Red).
4. Display the learner's situation to both learner and teacher, so these two main actors will be aware of the actual situation of the learner.
5. In case of detected difficulties, the proposed approach may act autonomously, without human intervention.

To do so, the "Learner Model Update Subsystem" tracks and collects every learner's interaction within the "Conventional e-Learning System." Then, it builds a contextual model for each learner and keeps updating it, so it reflects his actual condition in real-time. Finally, the learners' models are stored inside the "Learner Model Database" and then used by the other subsystems.

The "Difficulties Detection Subsystem" will calculate and store a set of indicators for each Learner used to assess his condition in real-time. If difficulties are detected, it sends a message to the "Intervention Subsystem." This latter will intervene on the Learner and takes the appropriate actions (Figure.1).

3.1 The Learner Model Update Subsystem

This subsystem is responsible for creating and updating a contextual *Model* for each *Learner* present in the "Conventional eLearning System." The other subsystems will use

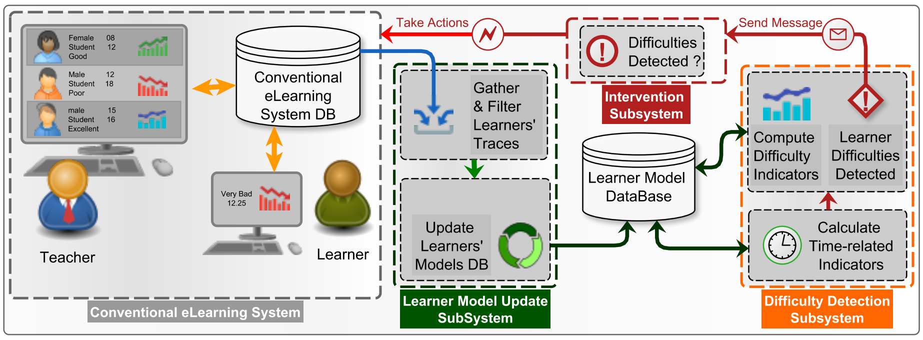


Figure 1: General Description of the Proposed Approach

it to detect and assist struggling learners. For example, when the learner enrolls in his first course, the "Learner Model Update Subsystem" creates a model for this learner and store it the "Learner Model Database." The model is continuously updated by all the subsystems together. *Course C* in our research comprises a set of *LearningObjects LOs* linked between them with a "prerequisite" relationship and having each an evaluation test passed before entering the next *LO*. As a result we have:

- A set of "*m*" *Learners* $\{L_1, L_2, ..., L_m\}$,
- A set of "*p*" *Courses* $\{C_1, C_2, C_3, ..., C_p\}$,
- A a set of "*n*" *LOs* for each *Course* C_j=$\{LO_{j1}, LO_{j2}, LO_{j3}, ..., LO_{jn}\}$.

The *Model* M of the *learner* L_i is defined as $ML_i(S_i, D_i)$, with:

- S_i is the static part of the *Model*, containing static information like his biographic and registration information.
- D_i is the dynamic part of the *Model* containing all cognitive and behavioral data about the learner L_i for each enrolled *Course* C_j (Figure. 2).

$$D_i = \bigcup_{j=1}^{p} C_{ij} \tag{1}$$

Formula. 1 defines D_i, where C_{ij} is the *Course* C_j enrolled by the *Learner* L_i and p is the number of the enrolled courses.
Furthermore, $C_{ij} = \{CP_{ij}, CDS_{ij}, CDL_{ij}, CP_{ij}, CED_{ij}, CDM_{ij}\}$, where:

CDS_{ij} : *Course* C_j Difficulty Status of the *Learner* L_i.

CDL_{ij} : *Course* C_j Difficulty Level.

CP_{ij} : *Course* C_j Progress.

CED_{ij} : *Course* C_j Enrollment Date.

CDM_{ij} : *Course* C_j Difficulties Matrix.

$$CDM_{ij} = \bigcup_{k=1}^{n} LO_{ijk} \tag{2}$$

Formula. 2 defines the *CourseDifficultiesMatrix* CDM_{ij}, which contains all calculated variables for the *Learner* L_i towards the *Course* $C_j s$, and n is the number of the started *LOs*.
$LO_{ijk} = \{LP_{ijk}, LAD_{ijk}, LBS_{ijk}, LNA_{ijk}, LSH_{ijk}, LPI_{ijk}, LSI_{ijk}, LODL_{ijk}\}$, where:

LOP_{ijk} : Progress of the *Learner* L_i on of the *LO* LO_k of the *Course* C_j.

$LOAD_{ijk}$: LO_k Access Dates.

$LOBS_{ijk}$: LO_k Best Score.

$LONA_{ijk}$: LO_k Number of Attempts to pass the evaluation test of LO_k.

$LOSH_{ijk}$: LO_k Scores History.

$LOPI_{ijk}$: LO_k Primary Indicators values (see Subsec 3.2).

$LOSI_{ijk}$: LO_k Secondary Indicators values (see Subsec 3.2).

$LODL_{ijk}$: LO_k Difficulty Level.

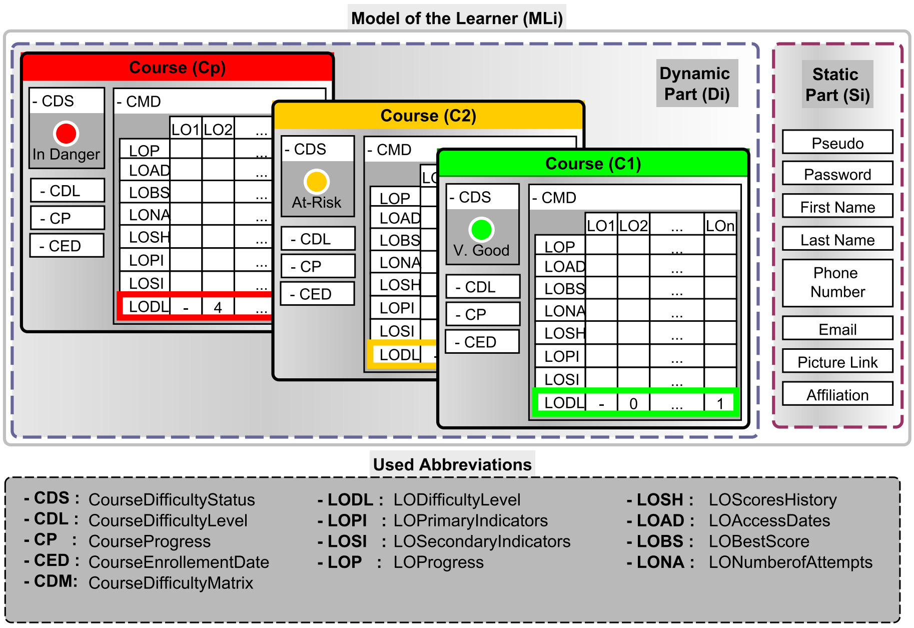


Figure 2: Contextual model of the Learner L_i

3.2 The Difficulty Detection Subsystem

This subsystem is responsible for assessing the condition of each learner towards all his enrolled courses. For that, it uses two main variables: *CourseDifficultyLevel* (CDL) and *CourseDifficultyStatus* (CDS).

CDL is a percentage value that reflects how many difficulties this *Learner* is facing during this *Course*. On the other hand, CDS is a scalar value used to classify the *Learner* in four states.

CDL and CDS are global values calculated for each course. For example, a CDL value of 100% means the *Learner* is "in Danger," while 0% means that he is "Very Good." Four Learner states are used and stored into the CDS, by comparing CDL to a predefined set of static thresholds. CDS value is defined as one of four states, each one of them is identified visually by a color:

- **Green**: The learner's situation is "Very Good." He is not facing any difficulties does not need any assistance.
- **Yellow**: The learner's situation is "Good." He is not facing any worrying problems, and he is most likely to succeed the course without any help.
- **Orange**: The learner's situation is "At-Risk." He is struggling, and there is a good chance that he will drop out if he did not get serious attention.
- **Red**: The learner's situation is "In Danger." His condition is critical, and he needs immediate assistance.

The *Course* CDL is the mean value of the LODLs of all available *LO*s that compose this course. So, to calculate the CDL of the Learner L_i in the Course C_j CDL_{ij} we used all stored $LODL_{ijk}$ (Formula. 3)

$$CDL_{ij} = \frac{\sum_{k=1}^{n} LODL_{ijk}}{n} \tag{3}$$

The *LODifficultyLevel* (LODL) is calculated using a set of indicators. It is essential to highlight here that even though the LODLs are calculated for each *LO* separately, some indicators may need information about previous *LOs*. Therefore, we grouped used indicators by importance as *Primary* and *Secondary*:

A. *Primary* indicators are critical. They can easily influence the value of LODL:
   1. Maximum Number of evaluation's Attempts Reached (MNAR) (Subsection. 3.2.1).
   2. The Slope of the Last three Attempts' scores of the Current *LO*'s evaluation (SLACLO)(Subsection. 3.2.2).
   3. The Slope of regression of the regression the Past three passed *LOs*' evaluations Best Scores (SPLOBS)(Subsection. 3.2.3).

B. *Secondary* indicators have a marginal influence on the value of LODL:
   1. The number of Delay Days past *LO* deadline (NDDLO) (Subsection. 3.2.4).
   2. The Slope of the Number of Attempts regression to pass the of Attempts to pass the Past three *LOs*' evaluation tests (SNAPLO)(Subsection. 3.2.5).

Each time a Learner access a new *LO* or an *LO* deadline is due, a new record is created the contextual model of this Learner with a value of LODL equals zero. Whenever a significant change in one of the *Primary* or *Secondary* indicators is detected, the LODL is increased by one of two values: *HighWeight* (HW) for *Primary* indicators and *LowWeight* (LW) for *Secondary* ones (Figure. 3). For example, if the third primary SPLOBS indicator is lower than -0.25, $HW_3 = 1$.

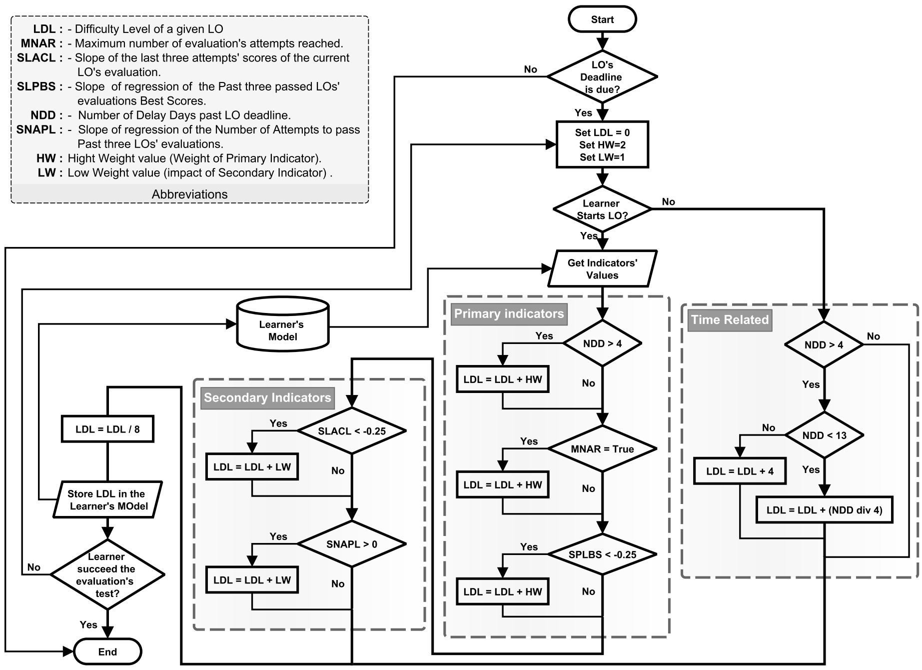


Figure 3: Flow Diagram of the LODifficultyLevel (LODL) calculation algorithm, based on Primary and Secondary indicators

HW and LW are assigned values 2 and 1 by default in the system and adjusted by the *Course Teacher* (HW has to be greater or equal than LW). As a result, the maximum value for LODL is eight. When all indicators are present, the LODL will be $LODL = 3 * HW + 2 * LW = 3 * 2 + 2 * 1 = 8$. Nevertheless, we want the value of LODL as a percentage, so we divide the LODL value over 8 (Formula 4)).

$$LODL = \frac{HW_1 * 2 + HW_2 * 2 + HW_3 * 2 + LW_1 * 1 + LW_2 * 1}{8} \tag{4}$$

At this point, we provide a thorough explanation of the previously mentioned indicators.

3.2.1 Maximum Number of evaluation's Attempts Reached (MNAR)

The Learner is allowed to repeat the same evaluation test several times until he succeeds or reaches the maximum number of repetitions. Whenever a learner attempts an evaluation test, the number of attempts is stored in his contextual Model alongside each attempt score. Whenever the number of attempts exceeds a specific threshold, this may hint that the learner could not pass the test however he tries. Thus, we consider this indicator to be the most important one that proves the presence of a learner's critical condition. The value attributed to this indicator is "1," if the threshold is reached; otherwise, its value is "0."

3.2.2 The slope of the Last three Attempts' scores of the Current *LO*'s evaluation (SLACLO)

The learner is given the opportunity to repeat the same evaluation test several times until he succeeds or the maximum number of repetitions is reached. Each time the learner attempts an evaluation test, his score is stored in his model, whether he passes or fails. Tracking the variation of these scores may come in handy to understand whether he is improving or regressing. It can witness whether the Learner is learning from his mistakes or not. In our case, we calculate the slope of the Last three Attempts' scores of the Current *LO*'s evaluation. We treat each score as a geometric point, with the score value as its y coordinate and the attempt number as its x coordinate. We have to find the Least Squares Regression Line slope: $y = mx + b$ that is the closest to these points., where m is the slope value. We want to minimize the sum of the squares of the vertical distances. That is finding m and b such that ${d_1}^2 + {d_2}^2 + {d_3}^2$ is minimum. To calculate the slope m we use Formula 5.

$$m = \frac{n\sum_{i=1}^{n}(x_i y_i) - \sum_{i=1}^{n}(x_i)\sum_{i=1}^{n}(y_i)}{\sum_{i=1}^{n}(x_i)^2 - (\sum_{i=1}^{n}(x_i))^2} \tag{5}$$

A positive value of this indicator means that the Learner's scores keep getting better, while a negative value indicates that the test scores worsen over time. A zero slope indicates stagnation, so no improvement and no regression (see Example 1).

Example 1. In the following, we present three calculated slope states examples:

– **Improvement** : during his first two attempts, the learner could not pass the current *LO*'s evaluation test with scores of 30 and 45 over 100, respectively. After his third attempt, the learner obtained a score of 55, which is above 50, so he passed the test. The slope of (30, 45, 55) is 12.5, which is the value of our indicator.

– **Stagnation** : 45, 30, 45 slope is 0, so, no improvement nor regression.

– **Regression** : 40, 30, 25 slope is -7,5 (Figure. 4).

3.2.3 The slope of regression of the Past three passed *LOs*' evaluations Best Scores (SPLOBS)

As the Learner progress through the Course, he must pass the evaluation tests for the previous *LOs* before passing on to the next ones. The best score for each *LO* is stored in the Learner Model. The value of this indicator for the current *LO* is the regression

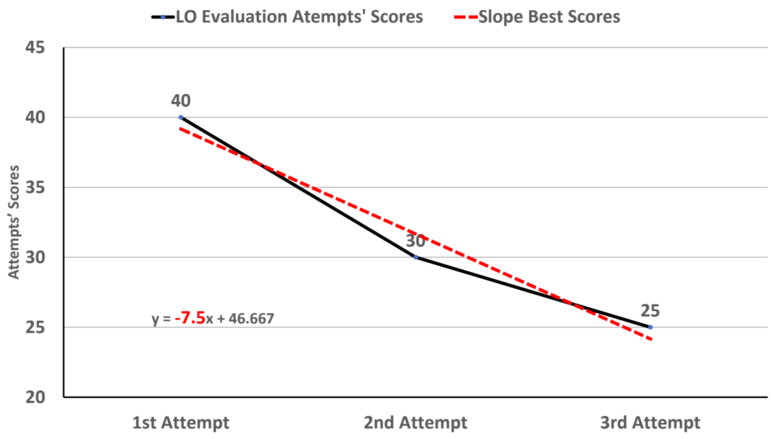


Figure 4: Example of SLACLO (Regression)

slope of the best scores of the past three *LOs*. A positive slope indicates that the Learner is finding his way through the Course and that things are getting easier for him as he progresses through the Course, while a negative slope indicates that the Course is starting to become more difficult for him. A zero slope implies stagnation, so there has been no significant change, either positive or negative (see Example 2).

Example 2. Table. 2 represents an example of a succession of the best scores of a learner during a course, where each column represents his best score obtained for that *LO*'s evaluation test. We can distinguish three states:

LO Nb.	*LO1*	*LO2*	*LO3*	*LO4*	*LO5*	*LO6*
Best score (/100)	**65**	**60**	**90**	60	50	
Slope				**12,5**	0	-20

Table 2: Example of the SPLOBS of a given learner

- **Improvement** : *LO1*(65), *LO2*(60), *LO3*(90), the slope for *LO4* is 12.5 (Figure. 5).
- **Stagnation** : *LO2*(60), *LO3*(90), *LO4*(60), the slope for *LO5* is 0.
- **Regression** : *LO3*(90), *LO4*(60), *LO5*(50), the slope for *LO6* is -20.

3.2.4 Number of Delay Days past *LO* deadline (NDDLO)

Each *LO* has a start date and a deadline. We would consider a learner as "late," if he did not access the *LO* before its due deadline. Each day passes before he access this *LO* gives more probability that he is facing problems. However, other possibilities exist, such as when a learner is lazy or does not care about his curricula. That is why we judged

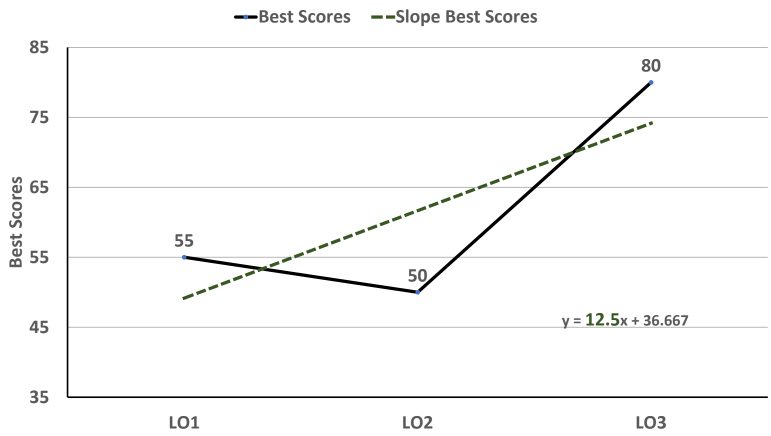


Figure 5: Example of SPLOBS (Improvement)

this indicator as non-conclusive and considered it Secondary. This indicator is updated every day for all started *LOs*. Therefore, it is increased by one every day until the learner finally accesses the concerned *LO* when its value is permanently fixed.

3.2.5 The slope of regression of the Number of Attempts to pass the Past three *LOs*' evaluation tests (SNAPLO)

Each time a learner attempts an evaluation test, the number of attempts be- fore succeeding this *LO*'s evaluation test is increased by one and stored in the learner's model. To calculate this indicator, we first fetch the stored numbers of the past three *LOs*, then we calculate the slope of regression –except for the first and second *LOs*–. The positive values of this indicator imply that the learner is finding it increasingly difficult to take the evaluation tests, causing him to attempt the tests more often than last time. In contrast, negative values are a good sign indicating that the learner is doing better. As with the previous indicator, there are many other scenarios where the learner is likely to repeat the tests more than once (see Example 3).

Example 3. In *LO1*, the learner attempted the evaluation test three times before passing on the fourth one. Then, two times in *LO2* and three times in *LO3*. To calculate this indicator for *LO4*, we use the information from *LO1*, *LO2*, and *LO3*. The calculated slope for (2, 2, 3) is 0.5. However, in this case, a positive value means that the Learner is making more and more attempts before passing the assessment test, which means that he is struggling (Figure. 6).

3.3 The Intervention Subsystem

It is responsible for taking the necessary actions to save the learners, either by contacting the Learner himself or by warning another actor that can intervene with the learner, like the teacher. It pushes system notifications to learners and the teachers about any probably faced critical situation. For the Learner, it displays his actual situation analysis. On the other hand, the teacher receives weekly reports about his learners' achievements,

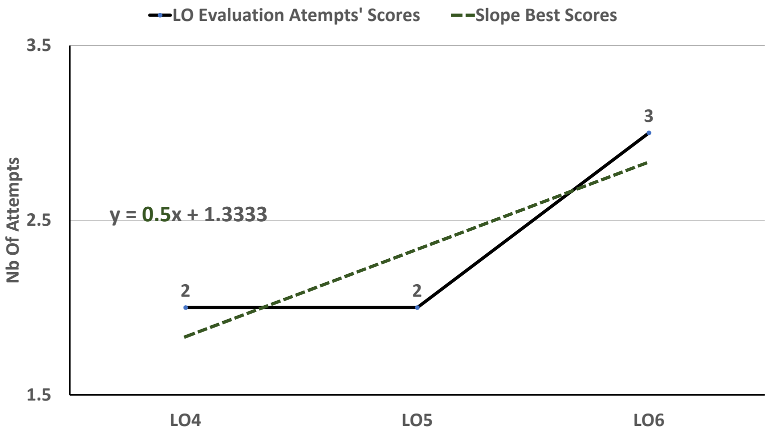


Figure 6: Example of SNAPLO.

progress, and, more importantly, about learners identified as ”at-risk.” Furthermore, it also sends emails and SMSs to learners and teachers and thus ensuring that all the actors are informed even if they do not enter the e-learning system.

3.4 Intelligent Agents

Tracking and monitoring each learner is tedious work for humans. The proposed approach comprises intelligent and autonomous subsystems, which collaborate to accomplish the system's main goals with minimum human intervention. Each subsystem relies on a set of cognitive agents. As shown in Figure. 7, four types of agents are used. They are distributed as follows:

A. **The Learner Model Update Subsystem:** uses one type of agents:

   1. **PersonalAgent:** This agent creates and then keeps updating the contextual Model of the Learner. One instance of this agent is associated with each Learner, and each time the Learner access the System, his PersonalAgent is activated and put to work.

B. **The Difficulties Detection Subsystem:** rely on two types of agents:

   1. **ContinuousControlAgent:** This agent performs all time-related tasks and checks the learners' persistence. One instance of this agent is associated with each course. This agent is triggered one time each day.

   2. **DifficultyDetectionAgent:** This agent calculates the CourseDifficultyLevel of every Learner in real-time. In case of any detected difficulties, it notifies the intervention agent of any noticed difficulties. One instance of this agent is associated with each course. This agent is triggered on the first day of the course and discarded on the last day.

C. **The Intervention Subsystem:** needs one type of agents:

   1. **InterventionAgent:** The role of this agent is to take action to save detected or predicted in-distress learners. One instance of this agent is associated with each course.

Appendix A gives more details about used intelligent agents.

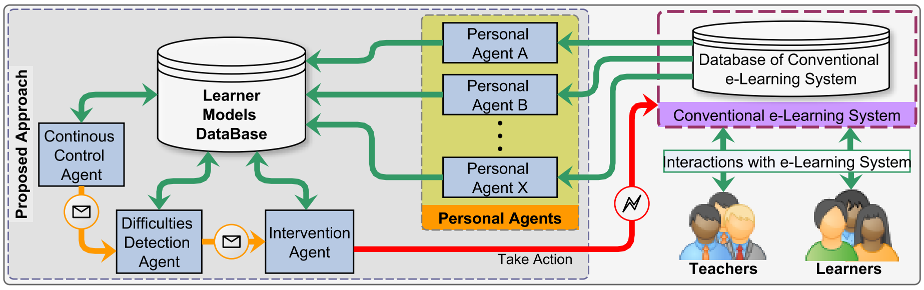


Figure 7: Intelligent Agents of the Proposed Approach.

3.4.1 Difficulty detection and intervention scenario

Figure. 8 illustrates a detection and intervention scenario undertaken by the proposed system on a *Learner A*. All the *LearnerA* actions and the system's reactions are enumerated and explained in the figure. The figure also illustrates the complete architecture of the proposed intelligent system, including all interactions between the different agents' modules and the system.

4 Experiment and Results Discussion

To validate the proposed approach, we have conducted an experiment on second year computer science students from the University of Guelma, in April 2020 and during the Coronavirus mandatory home confinement. In this test, we aim to measure the effectiveness of the proposed technique on: (1) Reducing the number of drop-out students, (2) improving the cognitive level of students. A new system based on PHP scripting language called "LearnDiP [1]" was designed, coded, and hosted separately from the university and the access was granted to students from any connected device such as PCs, Laptops, or Phones (Figure. 9). A newly created Gmail account was used to create a dedicated YouTube channel and a Facebook page. They were made available for the students to answer their questions and inquires

[1] **Learn**ing **Di**fficulties **P**revention: *https://bit.ly/learndip*

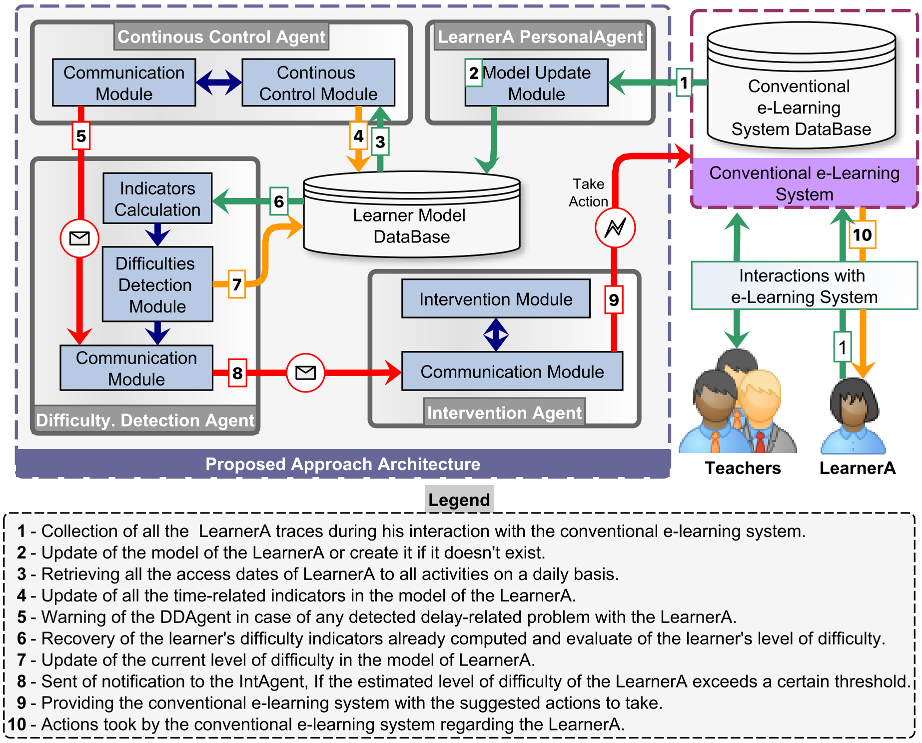


Figure 8: System Architecture.

4.1 Participants

A total of 40 students have participated in this experiment. They were randomly assigned to two groups. The first Group is the Test Group (TG), composed of 20 students who received assistance during the experiment and were well-aware of their situation through the informative dashboard. The second Group is the Control Group (CG), composed of 20 students who did not receive any assistance during the experiment and were unaware of their situation.

4.2 Methodology

Registered students can attend the ”Theory of Languages” course, composed of four LOs, each having an evaluation test consisting of a set of questions in the form of a Multiple Choice Question (MCQ). The student must pass the test before moving on to the next one. To compute The Learner Li score in the evaluation test of the LO_k, we divide the number of correct answers on the total number of questions (Formula 6).

$$Score\,(L_i)_j = \frac{Number\ of\ correct\ answers_j}{Total\ Number\ of\ questions_J} \tag{6}$$

For both groups, the same teacher instructed the course to ensure equitable conditions regarding the quality. In addition, both groups have no prior knowledge of the course,

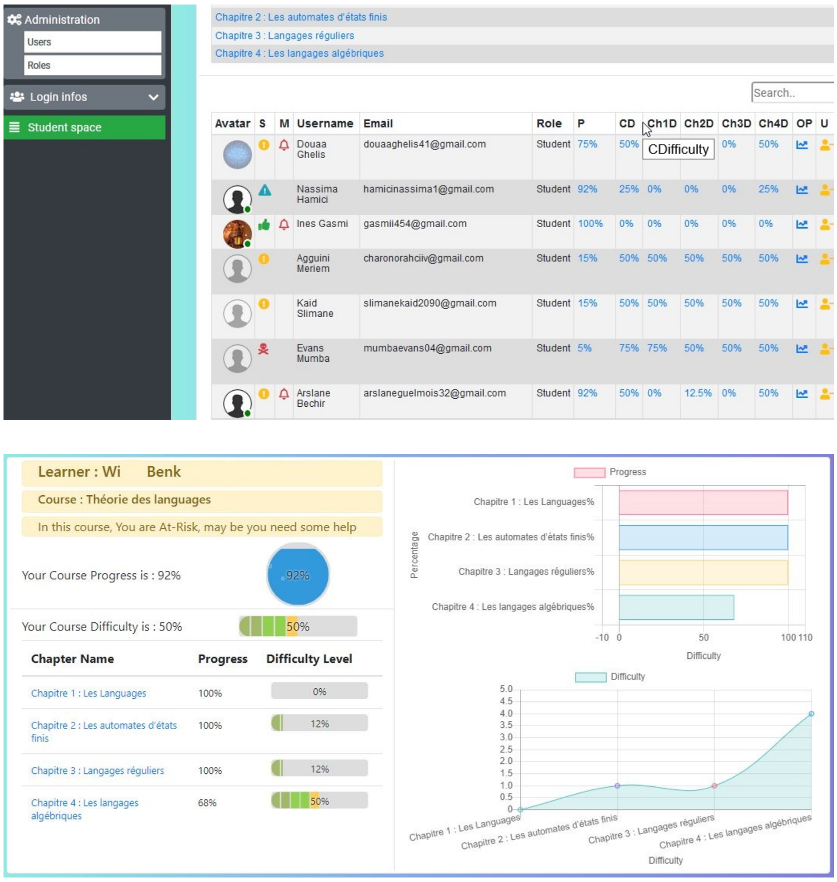


Figure 9: Screenshots of LearnDiP system

have no experience with any distance learning system before, and use the System for the first time. Our experiment would consider a learner a "dropout" if he failed to access LO_4. We conducted a "Before/After status" experiment with two randomly assigned and independent groups to see the proposed approach's effectiveness. In the first stage, we will compare the number of dropouts inside these two groups to check if the application of the proposed approach has reduced the number of dropouts. In the next stage, we calculate and compare the two groups' cognitive levels to see any improvement in the TG compared to the CG. The cognitive level in the Pretest and Posttest are given in formulas (7) and (8) respectively.

$$pretestCognitiveLevel(L_i) = \frac{Score(L_i)_1 + Score(L_i)_2}{2} \tag{7}$$

$$post\text{-}testCognitiveLevel(L_i) = Score(L_i)_3 \tag{8}$$

Among 52 subscribed students, 77% of them (40 students) have studied in the system.

Our working sample ware randomly divided into two groups Test Group (TG) and Control Group (CG), composed of 20 students each. 55% of participant students (22 students) have dropped out, while 45% (18 students) have made it to the end. The majority of droppers (73%) are from the CG, while 27% are from the TG (Table. 3).

	Passed Learners		**Dropped Learners**	
	Nb.	**Pcg.**	**Nb.**	**Pcg.**
Test Group	14	78%	6	27%
Control Group	4	22%	16	73%
Total	18	100%	22	100%

Table 3: Dropout statistics among TG and CG

Thus, the number of droppers is minimal in the TG. That means that the intelligent detection and intervention of our system LearnDiP is efficient. Nevertheless, we need to prove the efficiency of getting assistance from LearnDiP statistically, that why we need to apply statistical tests to see if there is a statically significant improvement and, if yes, how significant this improvement is.

4.3 Results and Discussion

4.3.1 Normality Test

In order to be able to choose the appropriate test, we first conducted normality tests with these hypotheses:

- H_0: The sample data are not significantly different from a normal population.
- H_1: The sample data are significantly different from a normal population.

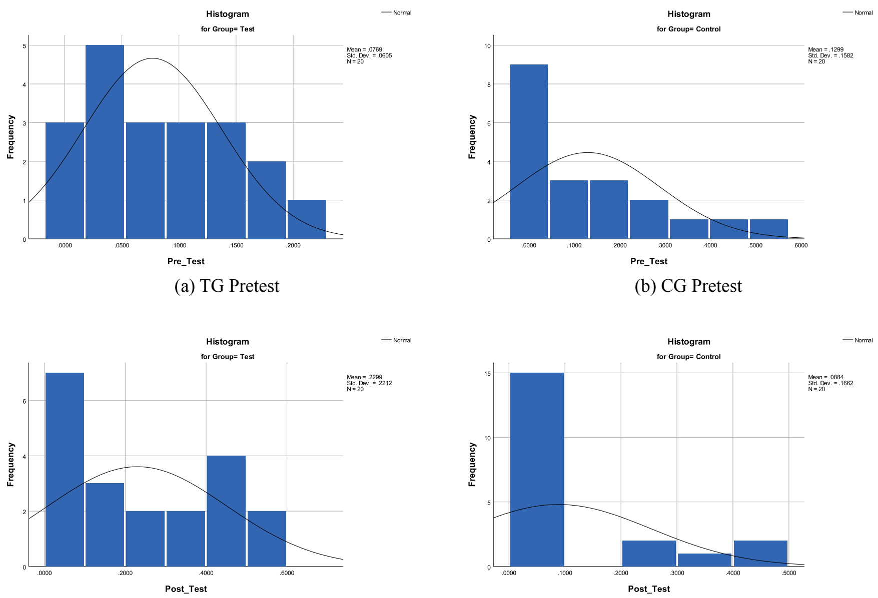
Obtained results from conducted normality tests: Kolmogorov-Smirnov and Shapiro-Wilk are shown in Table. (4). In both tests, we rejected H_0 only for TG during the Pretest and accepted it for all the other three cases, which means that most of our data-set is not normally distributed. The non-normality distribution is due to the large number of null scores caused by the dropouts, making it significantly different from normal (Figure. 10).

	Group	**Kolmogorov-Smirnov**			**Shapiro-Wilk**		
		Statistic	**df**	**Sig.**	**Statistic**	**df**	**Sig.**
Pretest	Control	.211	20	.020	.819	20	.002
	Test	.154	20	**.200**	.928	20	**.139**
Post_Test	Control	.453	20	.000	.585	20	.000
	Test	.201	20	.034	.850	20	.005

. This is a lower bound of the true significance.

a. Lilliefors Significance Correction

Table 4: Tests of Normality



(a) TG Pretest

(b) CG Pretest

(c) TG Post-test

(d) CG Post-test

Figure 10: Distribution of pretest and post-test values for TG and CG

4.3.2 Mann-Whitney U Test

Because our data-set is not normally distributed, we conducted the Mann- Whitney non-parametric U test (Wilcoxon test for independent samples) with these hypotheses:

- H_0^1: The Pretest distribution is the same across categories of Group (test, control).
- H_1^1: The distribution of the Pretest is not the same across categories of the Group.
- H_0^2: The distribution of the Posttest is the same across categories of the Group.
- H_1^2: The distribution of the Posttest is not the same across categories of the Group.

The P_{value} of H_0^1 is $P > 0.05$, so we accept H_0^1 stating that there was no significant difference between the cognitive levels of the two groups during the "Pretest" phase (before any intervention of LearnDiP on the TG). On the other hand, the P_{value} of H_0^2 is $P < 0.05$. So, we reject H_0^2 in favor of H_1^2. Therefore, we accept a significant difference between the TG and CG cognitive levels after the intervention of LearnDiP on the TG. All the details of the conducted "U Test" are presented in Table. 5.

In Figure. 11, the Mean of TG distribution during the Posttest is way higher than CG. Therefore, the difference found in the test is in favor of the TG. Moreover, Figure. 12a) shows that the distribution of CG results was slightly better than TG before the intervention of our system. On the contrary, Figure. (12b) shows that the distribution of the TG results was way better than that of CG.

In conclusion, all the results of the tests prove the efficacy of the proposed system and, more importantly, the efficiency of the proposed approach in detecting and saving at-risk learners and enhancing their cognitive level.

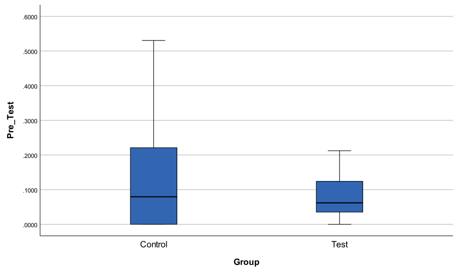
Total N	40
Mann-Whitney U	192.500
Wilcoxon W	402.500
Test Statistic	192.500
Standard Error	36.608
Standardized Test Statistic	-.205
Asymptotic Sig.(2-sided test)	**.838**
Exact Sig.(2-sided test)	.841

(a) Pretest across Group

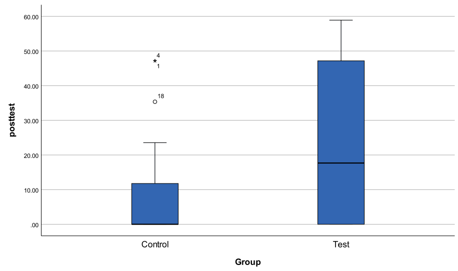
Total N	40
Mann-Whitney U	281.500
Wilcoxon W	491.500
Test Statistic	281.500
Standard Error	33.667
Standardized Test Statistic	2.421
Asymptotic Sig.(2-sided test)	**.015**
Exact Sig.(2-sided test)	.026

(b) Post-test across Group

Table 5: Independent-Samples Mann-Whitney U Test Summary Pretest and Post-test across Group

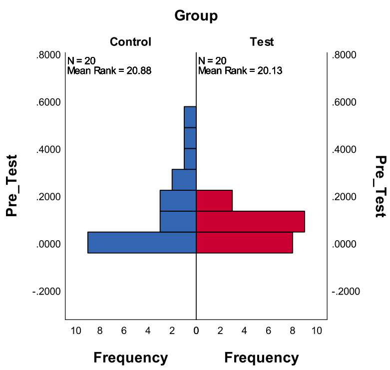


(a) Pretest

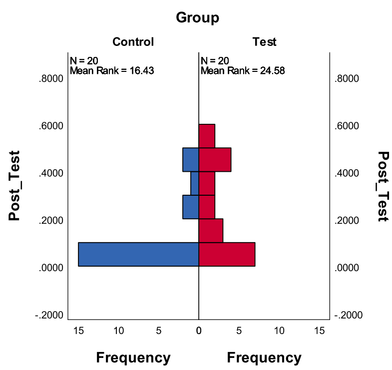


(b) Post_Test

Figure 11: Differences in means of pretest and post-test values for TG and CG



(a) Pretest



(b) Post-test

Figure 12: Independent Samples of the pretest and post-test values for TG and CG

5 Conclusion

Distance learning environments have proven to be very useful, especially in difficult times such as the Coronavirus pandemic. Unfortunately, the use of this type of environment is not without a price. Excessive numbers of dropout learners, among other problems, remain a considerable problem, as in the case of Massive Open Online Courses (MOOCs)[Capuano et al., 2020, Halawa et al., 2014]. Therefore, many researchers have been motivated to conduct more and more studies to understand the reasons behind this problem or develop new techniques that could improve the learners' cognitive level and thus reduce those numbers [Tadjer et al., 2020]. Understanding why a learner drops out can be very useful when establishing new algorithms and techniques to reduce dropouts. In addition, it is imperative to find meaningful indicators that might reflect the learner's status, as in struggling or confronting problems. The behavioral and cognitive-based indicators are the most widely used against other indicators like biographical or social indicators or even psychological indicators that seem very difficult to calculate.

In this work, we have presented a new agent-based approach capable of identifying and helping learners in critical situations, thus preventing them from dropping out. The proposed approach uses four different types of agents which collaborate to detect, help and rescue at-risk learners. The detection process starts by building a model for each learner based on gathered traces of this learner. The information stored in the learner's Model is then used to establish a difficulty level for each enrolled course by this learner. Finally, the course difficulty level is computed and continuously updated using all its available *LO*'s difficulty levels. Each *LO*'s difficulty level is computed based on two types of indicators, primary and secondary. If a learner is identified as "at-risk," the intervention process is automatically invoked to take action.

In order to validate the proposed approach, an experiment was conducted within a higher education establishment. The experiment results support the fact that it is possible to rely on behavioral-based indicators to diagnose and save at-risk learners correctly and thus reduce the number of learners who give up. Furthermore, collected data during the experiment can be used afterward to predict similar cases of learners who may encounter the same difficulties in the future.

Because the university in which the study was performed was not prepared for scenarios like the Coronavirus pandemic and had no possession of any students numeric directory containing their emails or cellphone numbers, we had a hard time reaching the students to inform them about the availability of the experiment's system, so that they could take their courses online. As a result, the number of participants was a bit small and not as we had hoped.

Moreover, and even though the experiment's system was equipped with a user-friendly help system that included numerous videos explaining how to sub- scribe and use the system and also how to choose a valid username and a secure password, a considerable number of participating students had a hard time subscribing in the system. This situation has led us to wonder about the impact of the ICT's mastery on the quality of distance learning and whether it influences the learners' decision to carry on or quit their study within these systems?

Also, and due to the time required by the proposed system to start identifying at-risk learners (even though this time is reduced to only two *LOs*), some learners decided to quit during that time. That situation has made us wonder if there is anything to do to save those learners. Therefore, we came up with the idea to treat all the learners as "at-risk" during the first two *LOs*. In that way, we could buy some time for the detection system to evaluate the status of the learners, and maybe we could keep them from withdrawing before the system is ready. Moreover, it is better if the system detects and can predict the learners who are more likely to drop out. We intend to implement and test all those ideas in our future works. Moreover, we intend to enhance our intervention strategy and make it more user-friendly by using a companion agent, for example, or transforming the look of the system to be more ”social media” similar, because we found out that the students are more familiar with the use of social media applications than those of email or learning system.

Acknowledgements

This work is being undertaken as part of the project entitled: **Predicting Learners' Difficulties within Human Learning Systems** under the number: **C00L07UN240120180001** and is supported by the General Directorate of Scientific Research and Technological Development (DGRSDT) of the Ministry of Higher Education and Scientific Research, Algeria.

References

[Abu-Oda and El-Halees, 2015] Abu-Oda, G. S. and El-Halees, A. M. (2015). Data Mining in Higher Education : University Student Dropout Case Study. *International Journal of Data Mining & Knowledge Management Process*, 5(1):15–27.

[Aggarwal et al., 2021] Aggarwal, D., Mittal, S., and Bali, V. (2021). Significance of Non-Academic Parameters for Predicting Student Performance Using Ensemble Learning Techniques. *International Journal of System Dynamics Applications*, 10(3):38–49.

[AL-Malaise et al., 2014] AL-Malaise, A., Malibari, A., and Alkhozae, M. (2014). Students Performance Prediction System Using Multi Agent Data Mining Technique. *International Journal of Data Mining & Knowledge Management Process*, 4(5):01–20.

[Azcona and Casey, 2015] Azcona, D. and Casey, K. (2015). Micro-analytics for student performance prediction leveraging fine-grained learning analytics to predict performance. *International Journal of Computer Science and Software Engineering*, 4(8)(8):218–223.

[Baneres et al., 2019] Baneres, D., Rodriguez, M. E., and Serra, M. (2019). An Early Feedback Prediction System for Learners At-Risk Within a First-Year Higher Education Course. *IEEE Transactions on Learning Technologies*, 12(2):249–263.

[Capuano et al., 2020] Capuano, N., Caballé, S., Conesa, J., and Greco, A. (2020). Attention-based hierarchical recurrent neural networks for MOOC forum posts analysis. *Journal of Ambient Intelligence and Humanized Computing*, (0123456789).

[Casey and Azcona, 2017] Casey, K. and Azcona, D. (2017). Utilizing student activity patterns to predict performance. *International Journal of Educational Technology in Higher Education*, 14(1):4.

[Denden et al., 2019] Denden, M., Tlili, A., Essalmi, F., Jemni, M., Chang, M., Kinshuk, and Huang, R. (2019). *Data Analytics Approaches in Educational Games and Gamification Systems*. Smart Computing and Intelligence. Springer Singapore, Singapore.

[Dietz-Uhler and Hurn, 2013] Dietz-Uhler, B. and Hurn, J. E. (2013). Using learning analytics to predict (and improve) student success: A faculty perspective. *Journal of Interactive Online Learning*, 12(1):17–26.

[Halawa et al., 2014] Halawa, S., Greene, D., and Mitchell, J. (2014). Dropout Prediction in MOOCs using. *eLearning Papers*, 37(March):1–10.

[Howard et al., 2018] Howard, E., Meehan, M., and Parnell, A. (2018). Contrasting prediction methods for early warning systems at undergraduate level. *The Internet and Higher Education*, 37:66–75.

[Hussain et al., 2018] Hussain, M., Zhu, W., Zhang, W., and Abidi, S. M. R. (2018). Student Engagement Predictions in an e-Learning System and Their Impact on Student Course Assessment Scores. *Computational Intelligence and Neuroscience*, 2018:1–21.

[Jayaprakash et al., 2014] Jayaprakash, S. M., Moody, E. W., Lauría, E. J., Regan, J. R., and Baron, J. D. (2014). Early Alert of Academically At-Risk Students: An Open Source Analytics Initiative. *Journal of Learning Analytics*, 1(1):6–47.

[Kokoç and Altun, 2021] Kokoç, M. and Altun, A. (2021). Effects of learner interaction with learning dashboards on academic performance in an e-learning environment. *Behaviour and Information Technology*, 40(2):161–175.

[Kuzilek et al., 2015] Kuzilek, J., Hlosta, M., Herrmannova, D., Zdrahal, Z., Wolff, A., and Item, J. (2015). OU analyse : analysing at-risk students at The Open University. *Learning Analytics Review, LAK15 '1*, pages 1–16.

[Lakhal and Khechine, 2021] Lakhal, S. and Khechine, H. (2021). Technological factors of students' persistence in online courses in higher education: The moderating role of gender, age and prior online course experience. *Education and Information Technologies*, 26(3):3347–3373.

[Lykourentzou et al., 2009] Lykourentzou, I., Giannoukos, I., Nikolopoulos, V., Mpardis, G., and Loumos, V. (2009). Dropout prediction in e-learning courses through the combination of machine learning techniques. *Computers and Education*, 53(3):950–965.

[Maldonado et al., 2021] Maldonado, S., Miranda, J., Olaya, D., Vásquez, J., and Verbeke, W. (2021). Redefining profit metrics for boosting student retention in higher education. *Decision Support Systems*, 143(January).

[Marbouti et al., 2016] Marbouti, F., Diefes-Dux, H. A., and Madhavan, K. (2016). Models for early prediction of at-risk students in a course using standards-based grading. *Computers & Education*, 103:1–15.

[Mubarak et al., 2020] Mubarak, A. A., Cao, H., and Zhang, W. (2020). Prediction of students' early dropout based on their interaction logs in online learning environment. *Interactive Learning Environments*, 0(0):1–20.

[Papanikolaou, 2015] Papanikolaou, K. A. (2015). Constructing Interpretative Views of Learners' Interaction Behavior in an Open Learner Model. *IEEE Transactions on Learning Technologies*, 8(2):201–214.

[Pistilli and Arnold, 2010] Pistilli, M. D. and Arnold, K. E. (2010). In practice: Purdue Signals: Mining real-time academic data to enhance student success. *About Campus*, 15(3):22–24.

[Richardson-Koehler et al., 1989] Richardson-Koehler, V., Richardson, V., and Casanova, U. (1989). School children at-risk. *Philadelphia: The Falmer Press*.

[Sandoval et al., 2018] Sandoval, A., Gonzalez, C., Alarcon, R., Pichara, K., and Montenegro, M. (2018). Centralized student performance prediction in large courses based on low-cost variables in an institutional context. *The Internet and Higher Education*, 37:76–89.

[Sarra et al., 2019] Sarra, A., Fontanella, L., and Di Zio, S. (2019). Identifying Students at Risk of Academic Failure Within the Educational Data Mining Framework. *Social Indicators Research*, 146(1-2):41–60.

[Tadjer et al., 2020] Tadjer, H., Lafifi, Y., Seridi-Bouchelaghem, H., and Gülseçen, S. (2020). Improving soft skills based on students' traces in problem-based learning environments. *Interactive Learning Environments*, pages 1–18.

[Tarimo et al., 2016] Tarimo, W., Deeb, F., and Hickey, T. (2016). Early detection of at-risk students in CS1 using teachback/spinoza. *Journal of Computing Sciences in Colleges*, 31(6):105–111.

[Toti et al., 2021] Toti, D., Capuano, N., Campos, F., Dantas, M., Neves, F., and Caballé, S. (2021). Detection of Student Engagement in e-Learning Systems Based on Semantic Analysis and Machine Learning. *Lecture Notes in Networks and Systems*, 158 LNNS:211–223.

[Von Hippel and Hofflinger, 2021] Von Hippel, P. T. and Hofflinger, A. (2021). The data revolution comes to higher education: identifying students at risk of dropout in Chile. *Journal of Higher Education Policy and Management*, 43(1):2–23.

[Waddington et al., 2016] Waddington, R. J., Nam, S., Lonn, S., and Teasley, S. D. (2016). Improving Early Warning Systems with Categorized Course Resource Usage. *Journal of Learning Analytics*, 3(3):263–290.

[Willging and Johnson, 2019] Willging, P. A. and Johnson, S. D. (2019). Factors That Influence Students' Decision to Dropout of Online Courses. *Online Learning*, 8(4):115–127.

[You, 2016] You, J. W. (2016). Identifying significant indicators using LMS data to predict course achievement in online learning. *The Internet and Higher Education*, 29:23–30.

[Yukselturk et al., 2014] Yukselturk, E., Ozekes, S., and Türel, Y. K. (2014). Predicting Dropout Student: An Application of Data Mining Methods in an Online Education Program. *European Journal of Open, Distance and E-Learning*, 17(1):118–133.

[Zhou et al., 2020] Zhou, Y., Zhao, J., and Zhang, J. (2020). Prediction of learners' dropout in E-learning based on the unusual behaviors. *Interactive Learning Environments*, 0(0):1–25.

Appendices

Appendix A Intelligent Agents Details

Name	Tasks	Objectives	Number	Functions
1. Personal Agent	- It creates new contextual model for newly enrolled *Learners*. - It updates the Learner contextual model towards all *LOs* in all *Courses*.	Feeds the learners' contextual models with collected traces.	One (01) agent per ***Learner***	**CreateModel:** it creates a new contextual model for a learner if non-existent. **UpdateModel:** updates the learner contextual model. It runs each time there a change in the status of the learner.
2. Continuous Control Agent	- It analyzes each learner's model to evaluate if a learner is falling behind. - It updates the learner model accordingly. - It notifies the Difficulty Detection Agent if delays passed thresholds to enable this latter to calculate the *DifficultyLevels* using the other indicators.	- Follow-up of essential dates (start and end of each activity) - Follow-up of learners' attendance in academic activities.	One (01) agent per ***Course***	**Count Learning Delay:** it checks for each learner if he accessed or not the active LO. **UpdateModel:** it updates the learner's model, used later by the Difficulty Detection Agent to evaluate the learner's indicators like his state. **NotifyDDAgent:** If the number of delay days is more significant than a certain threshold, it notifies the Difficulty Detection Agent.
3. Difficulty Detection Agent	- It calculates the *DifficultyLevel* for each LO. - It analyzes each learner model to assess if the learner is struggling. - It updates the learner model accordingly. - It notifies the intervention agent to take the necessary action if any difficulties are detected	Ensures that learners can pursue their learning without problems by identifying those who have problems so they are targeted by interventions.	One (01) agent per ***Course***	**CaclulateIndicators:** it calculates the *DifficultyLevel* for each learner in each LO, used later to find the course difficulty level for each learner. **UpdateModel:** it updates the learner's model, with the learner calculated status, indicators and variables. **NotifyIntervAgent:** If a learner is identified as struggling, it sends a message to notify the Intervention Agent.

Table A.1 – Intelligent Agents tasks, objectives and functions

(Continued on the next page)

Name	Tasks	Objectives	Number	Functions
4. Intervention Agent	- It takes the necessary actions to save learners - Sends messages to both learners and teachers about critical situations.	Provides at-risk learners with necessary assistance.	One (01) agent per ***Course***	**SendMessageToLearner:** It sends a message to the learner about his performance. **SendMessageToTeacher:** Send a message to the teacher about the performance of the learner.

Table A.1 – Intelligent Agents tasks, objectives and functions
(Continued from the previous page)