A MULTI-LAYER ARCHITECTURE FOR AN E-LEARNING HYBRID RECOMMENDER SYSTEM

Lisa Roux, Pantxika Dagorret, Patrick Etcheverry,
Thierry Nodenot, Christophe Marquesuzaa and Philippe Lopisteguy
Université de Pau et des Pays de l'Adour/E2S UPPA,
Laboratoire d'Informatique de l'Université de Pau et des Pays de l'Adour, EA3000, 64000 Pau, France

ABSTRACT

Distance computer-assisted learning is increasingly common, owing largely to the expansion and development of e-technology. Nevertheless, the available tools of the learning platforms have demonstrated their limits during the pandemic context, since many students, who were used to "face-to-face" education, got discouraged and dropped out of school. In this context, a main issue is to conceive tools that would allow the teachers to supervise their students at a distance, by monitoring their progress and ensuring follow-up action as required. Another issue is to equip the learning platforms with intelligent systems able to guide the students involved in pedagogic activities.
In this work, we propose a novel architecture of recommender system for vocational higher education that provides the students with personalized advice and the teacher with suitable information, in order to make the task of monitoring easier and involving them in the machine learning. Our system is supposed to act in a hybrid environment, and, for this purpose, has to explain its predictions in an interpretable and faithful manner, both to the students and the teachers, so that the former can determine the relevance of what is suggested and the latest can act on the future analyses and recommendations. This is a multi-layer architecture, so that each step of the recommendation process is meaningful, thus explicable to the users.
The design of this architecture is a preliminary stage of a recommender system. It is designed on top of a learning digital infrastructure exploited since 2018 by the 1000 students of Bayonne Institute of Technology.

KEYWORDS

Recommender Systems in Vocational Higher Education, Supervised Classification, Explainable AI, Hybrid Learning Context

1. INTRODUCTION

This paper relies on the assumption that the future of vocational higher education will be hybrid. If there was still a need for evidence, one year and a half of COVID pandemy in our university has demonstrated that the learning interactions offered by available Learning Management Systems (LMS) to students (e.g. Moodle and its available plugins) are useful, especially when lecturers and teachers are accompanied by pedagogical engineers ; but used as a whole, these LMS are considered quite inappropriate by teachers to foster and support active learning in and out of the classroom. In the works presented in this paper, we use the Moodle Platform as an organised container providing the students with learning opportunities (description of available exercises) and with resources in relation with the competences addressed by these exercises: course contents, wikis, forums, tutorials, quizzes… Nevertheless, in our institute, most of the available exercises are realized outside of the Moodle environment, face to face for half of a group (8 from 16 or 15 from 30 in the COVID context) and also at a distance for the second half of a group and thus, thanks to dedicated virtual environments offering the required professional tools. For example in the context of a computer-science curriculum in our institute, students are called to practice C++ exercises with a code editor including a debugger, some test scenarios. Those at a distance can reach these programming tools thanks to a single internet connection, then also can cooperate in a synchronous way through screen-sharing and showcase facilities; and all the students (those in the campus and the others at a distance) can also take advantage from the Moodle resources also put at their disposal. This stuff can be later used by students to go deeper by themselves into the learning material. From a pedagogical viewpoint, this hybrid approach for practising on authentic exercises and on teaching through practice has demonstrated all its strengths all this year-long but also the following important weakness: teachers have encountered difficulty to succeed in eliciting which students to tutor in priority, which students should be considered as a single group to advice at a given stage of a session, ... They asked for a system that could assist them in managing the learning process followed by their students and the most motivated teachers agreed to spend time and provide us with their pedagogical expertise in a an applied research focusing on a recommender system for learning from authentic learning situations.

Different directions have been proposed to design such recommender systems. Some studies have shown an initial relationship between personality profiles and meaningful learning profiles (e.g. [arruda2019]), so that it is important to provide the students with student-centered environments and propose them stimulating activities in which they take an active part. Moreover, many AI works in e-learning aim at conceiving auto-adaptive platforms, where different environments are available, so that one of them can be assigned to a student depending on them profile, obtained from tests or questionnaires (e.g. [klavsnja2011, schiaffino2008, wolf2003]: on the basis of standard learning profiles, such as those defined by Honey and Mumford [honey1989] or Kozma [kozma1991] for example, several potential interfaces or sequences of pedagogical activities are designed, each of them being associated to a defined profile.

Nevertheless, these applications address total automation, while numerous researches aim at teaming AI with people in order to improve joint performance and efficiency. In particular, in the context of e-learning, hybridization ensures that the teachers can follow each student's progress despite the distance, which deprives the teachers of direct view on their work: without AI, this would be a very onerous and time-consuming task. This is why an intelligent system is required to detect the students who need particular attention and increased monitoring (e.g. students at risk of dropping out of school, or experiencing significant difficulties). But we assume that the principal responsibility for helping and guiding students should remain with teachers and not placed on the AI authority: the teaching practices, the pedagogical support, the human capacity to tailor their actions to the individual situation cannot be replaced with machine learning and algorithms, since many factors, such as motivation, are hardly automatically measurable. The very issue of learning profiles appears questionable [claxton2013, rohrer2012, stahl2002], since assigning a student to one single profile and, therefore, one type of interface or sequence, means reducing them to one way of learning, when students can have various learning behaviors, strategies, and needs depending on situations. Yet, one of the main challenging issues in the conception of dynamic recommender systems is to take into account the possible conceptual shifts (e.g. user preferences, item popularity) in order to adapt the system to the user's evolving behavior and expectation [rana2015]: this is a key factor of the recommendation improvement [vinagre2015].

Thus the purpose of our system is not to propose a predefined and static model of pedagogical activities to a student according to their assessed type, but recommend them suitable activities at a given time, depending on the difficulties they meet and what they have done so far. Indeed, while the distance learning enables the students to work when they want (i.e. gives them the ability to manage their own time) and gives them access to a multitude of activities and resources to practice and learn, this makes the real-time and exhaustive monitoring hardly possible and requires the assistance of a AI.

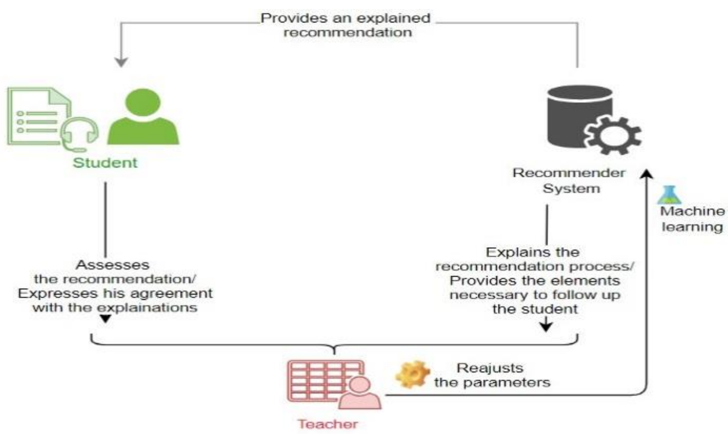

Figure 1. System operation

For these purposes, it operates at two levels, which are interwoven. A first level, which is based on an improvement detection system, aims at guiding the students who are engaged in a task and proposing them suitable activities according to the encountered problems and the situation of the student. A second level update a teacher's configurable scoreboard that indicates the recommendations that the system has done, the students' behaviors and feedbacks, and highlights those who need particular attention. According to the feedbacks and their own expert analysis of what is reported, the teacher can decide to change the recommendations or rectify the class allocation (i.e. choose which category the student should have been placed in).

Regarding the AI designed to operate in a hybrid context, accuracy is far from being the only preoccupation [grosz1996]. In spite of the accuracy decreasing that such a multi-layered architecture can imply, it allows a better understanding of how AI works, thus better control. If accuracy has been regarded as the major development for years and is still a major research issue, much recent concern has been focused on trust in AI, especially for the conception of successfully creating human-AI teams. Indeed, recent research has demonstrated that an increasing AI accuracy does not always translate to a better general team performance [yin2019, lai2019] because it is closely linked to the quality of the relationship between the human and the AI (i.e. trust, knowledge of the limits and the potentials of the AI, understanding of system operation, etc.). That is why it is important to find a compromise between accuracy and comprehensibility - and that is what why try to reach in this study.

2. OVERVIEW OF THE RECOMMENDER SYSTEM

2.1 The Recommendation Process

Our recommender system is designed to help students in choosing relevant activities depending on their situation. Its main purpose is to recommend useful and interesting resources or tasks to e-learners based on their different learning and gesture behavior, and other meaningful attributes. It is integrated in the Moodle environment, because this is free Open Source software package, widely used at our University and others. Moreover, we can use the Moodle logs to track student activity, thus have relevant information about their attendance, attempts, preferences, etc. When the systems used in e-commerce, vod platforms, social networks, etc. mainly implement content-based, knowledge-based or collaborative filtering techniques to recommend items, these methods cannot be directly applied to the e-learning area. Indeed, in e-learning, recommendations must take into account that the learning courses, activities, and materials need to be proposed in order to ensure that the prerequisites of some courses are met and that they match their particular in-time needs. Hybrid recommender systems (i.e. recommender systems combining different techniques) are more popular for e-learning than single recommendation method-based systems for avoiding the drawbacks of individual recommendation methods. Since our system has to adaptively model and respond to the change of the learner profile and propose them adequate activities, it has to find similarities between users to determine which kind of activities are interesting and relevant for the target user at a given time, taking into account the strong influence of the context, but also consider the complex pedagogical relationship between items in e-learning. We consider that students can experience difficulties in some subjects but feel very comfortable with others, and this may change over time, depending on the addressed notions, their personal lives, the course organization, etc. It is therefore necessary to take into account all aspects, either external and internal, that can dynamically contribute to the emergence of the student's specific need, and that have to be identified to detect this need. That is why our approach needs to both recommend activities that have been proven to be relevant for learners with close profiles in similar contexts and integrate knowledge-based techniques. Moreover, since the learner cannot be described by one single class but may adopt various cognitive and metacognitive strategies and approaches, have different levels of cognitive engagement and motivation over time, for example depending on courses and tasks, we will use classification methods based on fuzzy-logic [isaza2006, roux2015]. Moreover, we will use supervised classifications firstly because the class design can rely on a precise knowledge that teachers have about their students and that they therefore can put at our disposal. Moreover, the formalization of this knowledge will be based on classifications recognized in the field of educational sciences.

Despite this system is designed as a general recommender system for different types of courses, the first completely implemented and tested version will be for an introductory C++ programming course, since the reflection needed to be contained within the framework of one particular pedagogical situation, so that, as the teachers involved in this class, we can study how we recommend activities to students during the in-person classes, thus how the AI could do it in our place. Therefore, some tools used for the improvements detection in a C++ program are not suitable for any other type of task, but the architecture is thought to be generic. As we can see on the figure 2, a first step is to detect the possible improvements to the student's work. This can be done in an automatic way or through a questionnaire. The questionnaire is designed by the teachers, who select the relevant questions that enable the system to measure or assess the achievement of task objectives. The questions can be tailored for every single exercise or automatically assigned through a labelling system: a set of questions can be initially written by the teacher for one label, so that he only has to tag a task with this label to associate the questions to it. For example, in the case of a C++ programming task, the questions could be: "Do your variable names describe what your variables are for?", "What kind of structure do you use to implement your sorting function?" "Try to assign 0 to this variable. Does it work?". There are checkboxes for each question on the questionnaire, and the answers allow direct detection of improvements. Questions can be added later, if the teacher notices that some important aspects to assess are missing. This tool has two main objectives. Firstly, as we have just mentioned, it is a detection mechanism. Secondly, it allows the active implication of the student in the understanding of the possible improvements of what they have done. By answering the questions, they operate a structured review of their own work, assess their practices by confronting it to the explicit objectives of the task, check for missing exception cases processes, etc.

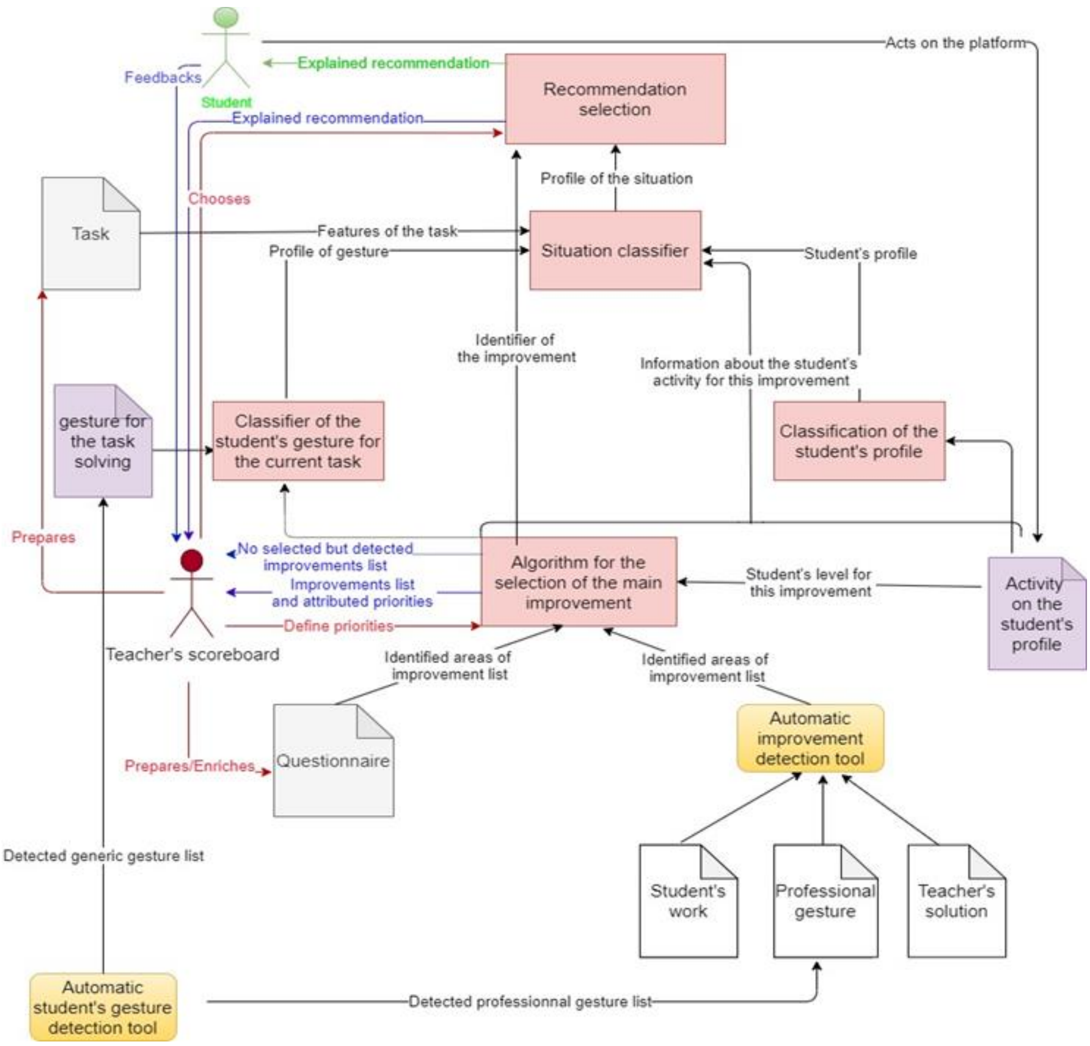

Figure 2. Architecture of the system

There is another detection tool, which is completely automatic, based on three elements. The first element is the student's work, that can be, to some extent, assessed by an automatic tool designed for what we want to check (e.g. spell and syntactic checker, anti-plagiarism system, parser generator tool that builds a concrete syntax tree of a program), sometimes by comparing it to a second element, which is the teacher's solution. The third element is the professional gesture, which permits to detect whether the student has adopted the good practice (i.e. the professional practice, as taught in their courses) during the task. For example, in a programming task, the examined issues can be: Has the code been progressively written, by validating each step with a compilation operation? Has the student used the debugger to analyse the execution errors? Is the code structured and developed in an incremental way? The information about the professional gesture is got from a tool that we will describe more precisely in section IV.

Both improvement detection tools are complementary. Indeed, some questions are hardly detectable in an automatic way, thus the questionnaire appears to propose a more complete check. Nevertheless, the student may have not checked the right box (i.e. the box associated to their situation) for several reasons, such as they have not understood the question or improvement that could be done, or also because they do not want the improvement to be detected. With the automatic detection tool, some improvements will be detected despite this, and a conflict situation will be raised so that the teacher is informed.

The student's situation, whose description is necessary to recommend them a suitable activity, depends on three main elements: how they feel during the task (i.e. does he seem confident, hesitating, etc.?), the quality of their learning process, and their historical relationship to the selected improvement. Thus, the system integrates four distinct processes, which are, on a first level, the selection of the main improvement, the student's quality of learning, and the student's gesture analysis. This student's gesture (i.e. hesitating, confident, etc.) is different from the professional gesture that we have mentioned above, since the former is dedicated to describe how confident or hesitating a student feels during a task, and the latter describes how much the student has assimilated the taught professional practice. Once the detection is completed, the list of improvements is processed by a selection algorithm, which chooses one improvement regarded as a priority issue. The teacher will be able to know every improvement detected if they are interested in; moreover, the information about which improvement is selected and which are not is stored in the database to feed further processing. The algorithm will be described more precisely in section III. In the same time, a classification allows to allocate the student's gesture to a class, and another permits to identify to what class belongs the student's quality of learning. Both classifications will be described in section IV. Those three processes will allow to get the student's situation class through a third classification process. Then, the system is able to recommend to the student an appropriate activity thanks to the predefined teacher's recommendations: this last step requires that the teacher has precised, for every possible situation (i.e. every class provided by the situation classification), which activities would be relevant. Those are generic activities that refer to available Moodle activities (e.g. reading a lesson, asking to a classmate, doing an exercise, posing a question on a forum); among them, the recommander system is able to choose which are available for this specific improvement (i.e. which resources and tasks have a label related to the improvement one) and relevant for the particular situation of the student (i.e. what the student has already done or not).

2.2 Explainable Recommendations

Once the recommendation is done, the student can choose to follow it or not, and is invited to assess its quality: does it meet their needs? Are they happy with their gesture assignation? Do they agree with the improvement? These feedbacks allow the teacher to improve the system, in particular by improving the questionnaire, modifying the weights they have associated to the possible improvements, and assigning the student's gesture or the student's learning quality to the suitable classes if the previous classifications appear to be inaccurate. In this way, the teacher is actively involved in the student monitoring: they can choose what recommendations are made depending on situations, know which difficulties are met by every student, and reajust the classification processes from their own analysis and observations. Thus AI explains its decisions to the teacher by providing them with every information it has used to take them, and, this doing, ensures that they can keep control on the recommendation process, and tailor them to their pedagogical practices. Similarly, the student can understand why this recommendation, for which improvement, etc, to help them become more aware of them and learn how to step back and assess their own work more effectively - in order to make them be gradually more autonomous. This allows to increase the user's trust in the system and keep them being actors of the student monitoring for the former, and their learning for the latter.

ISBN: 978-989-8704-33-7 © 2021

2.3 Labels and Tags

The system requires that the teacher has conceived a skill graph for their course. This is a directed graph, in which they indicate what are the skills learnt and practiced during the course, in what sequence (i.e. which ones are prerequisites necessary for others). Each skill corresponds to one label, so that every improvement, every resource, every task should be tagged by the corresponding label (i.e. the skill to which it refers). The more detailed it is, the more precise the recommendation can be; nevertheless, the skill graph can be improved and developed over the years.

3. SELECTION OF THE MOST IMPORTANT IMPROVEMENT

Among all the detected improvements, the recommendation concerns one of them, which is regarded as primary, so that the student can concentrate on every point one by one, with the computed priority order. The algorithm take five parameters into account: the situation of the improvement in the skill graph, the student's skill level, the priority level assigned by the teacher to the different improvements, how many times the improvement has appeared over the past ten recommendations, how many times the improvement has been selected over the past five recommendations.

Situation of the improvement in the skill graph: This indicates how they articulate relative to one another, depending on the number of skills that they require among the skills associated to the selected improvements.

Student's average skill level: Students have to acquire the skills taught in a course to validate it. For every skill of the course, they can further it by completing tasks tagged by the associate label. There are five skill levels, related to a value between 0 and 4: not started (0), beginner (1), workable (2), mastering (3), expert (4).

Priority level assigned by the teacher to the different improvements: If they want, the teacher can assign priority levels to the possible improvements, either generically or specifically to a task. This is an optional indicator, since the teacher could prefer letting the algorithm decide on the basis of the other information, or have no time for this. They can also assign a priority level to a handful of improvements.

Number of times the improvement has appeared over the past ten recommendations: In order to foster the selection of the repeating improvements, the system takes into account the number of times an improvement has appeared in the student's works over the past ten analyses done on them. Only the past ten recommendations are taken into account to avoid outliers.

Number of times the improvement has been selected over the past five recommendations: We assume that the subject of recommendation has to change regularly to facilitate learning and avoid fatigue: a break can be necessary, because learning is a process that can take time. When an improvement has been selected many times over the recent recommendation, the likelihood of being selected for the current one decreases. We only consider the five last recommendations in order to ensure a good turnover and avoid outliers.

4. IDENTIFICATION OF THE SITUATION AND PERSONALIZED RECOMMENDATION

4.1 Identification of the Student's Gesture

In interviewing teachers, we found out that a first factor that generally determines which kind of help a teacher would provide to a student is their ease to achieve a task: are they actively involved in the task solving? how much is the student hesitant or confident? Are they trying to solve the difficulties they meet by themselves? Do they know how to face the problems and where to look for the needed information? The study of the student's gesture aims to assess how confident and proactive the student is during the task. In order to assess the student relationship to the group, we can use an automatic detection tool that collects in XML format all their actions: clicks, written characters, open windows, etc. These tracks can then be processed and used to provide our system with useful and meaningful information. This tool allowed us to compose five indicators, related to the above questions: the rate of activity, the rate of research, the rate of research related to the selected improvement, the rate of erasure, the frequency of the work verification.

Rate of activity: This indicator assesses how much the student is active on the computer, using the keyboard or the mouse. It is calculated as time using them, divided by the total task time.

Rate of research: It shows whether the student tries actively to solve the problems he meets, by searching on the Internet or the online course material. It is calculated by dividing the research time online by the total activity time.

R*ate of research related to the selected improvement*: It indicates whether the main improvement detected by the system has been identified by the student as a problem in their work. It is calculated by dividing the research time about the improvement online, divided by the total research time.

Rate of erasure: This assesses the student's certainty about what they write. It is calculated by dividing the number of erased characters by the number of written characters.

Frequency of the work verification: This indicators shows whether the student regularly checks the validity of their work. In the case of a programming task, it is calculated by dividing the number of times the student has launched the compilation operation by the number of complete structures in their program. This is an optional indicator, since not all software programs can offer such a functionality.

Therefore, data are described by a 4 or 5-dimensional vector and can be classified to indicate how much the student encounter difficulty during the task and how much they try to solve them.

4.2 Identification of the Student's Profile

A second factor that influences the teacher's behavior is the general quality of the student's learning process, assessed by the means of seven indicators, which are: the rate of achieved tasks, the rate of successful tasks, the rate of consulted course material, the average skill level, the average duration of task completion, the number of improvements by complete task, and the rate of started but unfinished task.

Rate of achieved tasks: The first indicator used to classify the student's profile is the number of achieved tasks by the total number of available tasks (i.e. all the open tasks among the past and current courses taken by the student). It shows whether the student is determined to get involved in the available tasks through the end.

Rate of successful tasks: It indicates how many tasks are successfully completed by the student among the total number of available tasks. We chose to consider the total number of available tasks instead of the completed tasks, in order to get a better idea of their overall comprehension and mastering of the course.

Rate of consulted course material: This indicator is meant to show whether a student seeks to understand the theoretical aspects of a course, how much they try to consult the available course material to master the content.

Average skill level: As mentioned above, there are five possible levels for each skill. This indicator refers to the mean value of the achieved levels for every skill that should have been started at this point of course.

Average duration of task completion: For every completed task, the normalized duration is calculated by dividing the amount of time spent by the student on the task by the average amount of time spent on the same task by all the students. Thus, the average duration of task completion is the mean of all these normalized lengths of time. This indicator shows whether a student generally complete quickly their tasks.

Number of improvements by completed task: This indicates how many improvements are detected for every task a student achieves, in order to assess the average quality of the final works. This corresponds to the average value.

Rate of started but unfinished tasks: This indicator expresses whether a student often gives up an ongoing work. It is measured among the total number of available tasks the total number of started tasks.

4.3 Identification of the Situation

A recommendation is made to a student according to who they are in general, but also how they are faring with the specific current task, and their historical relationship with the detected main improvement, in particular the theoretical and practical course contents related to this improvement. Thus, in order to classify the situation of the student, we take into account five indicators: the student's gesture such as obtained under the step described in IV.A, the student's profile obtained under the step described in IV.B, how long they have already spent doing the task, the number of times they have consulted the course material related to the improvement, the number of tasks related to the improvement they have already completed, the number of times the improvement has appeared over the past ten recommendations (see III).

Time spent on the current task: This indicator is normalized in relation with the average duration of the task. It indicates whether the current task takes a lot of time to the student. Indeed, in order to avoid fatigue and discouragement, the teacher often adapt their recommendation depending the time that the student has already spent doing the task. Generally, the longer is the task completion, the more steering is the provided help. For example, in some cases, the teacher could advise the student to consult the course material whether they think they can and have time to solve the problem themselves; on the contrary, a teacher could prefer providing a student with an extract from their own solution when they think that they encounter too difficult matters for the current task and spend too much time on it.

Number of accesses to the course material related to the improvement: This indicates whether the student has already consulted the related course, and how many times they have, in order to determine whether they are familiar with the theoretical aspects of the current matter they face.

Number of completed tasks related to the improvement: This indicates whether the student has already complete tasks related to the detected improvement, and how many times they have, in order to determine whether they have already faced and addressed this issue, and how much they are familiar with its practical aspects.

4.4 Assignment of a Recommendation Depending on the Student's Situation

Once the different situation classes are defined, the teacher will be able to assign them generic recommendations, i.e. specifying what kinds of activity they would suggest to a student in that situation or what kinds of resources they would provide to them. The teacher is free to set priorities to the recommendations associated to each class. In a given situation, for a specific task, some recommendations will be applicable and some other will not, for example whether there is no material course for the concerned skill. If possible, two possible activities will be recommended to the student each time, in order to let them judge which activity is most convenient for them and record their preferences. Therefore, during the recognition process, the system will chose among the different available possibilities, among those defined by the teacher, depending on different indicators, such as the assigned priorities if applicable. The system will also take into account what the student, and the students in the same situation as them, did during the previous recommendations, whether they have followed the recommendation or not, what was their assessment, depending on the type of activities or resources that was proposed to them. The recommendation will be first chosen randomly, then willing to leave the opportunity for chance in order to avoid convergence and foster the diversity of the suggested activities, this information will be used to weight this random selection.

5. DISCUSSION AND CONCLUSION

The study introduces the conception of a recommender system that operates in a hybrid context, in which the teacher has total control on the involved parameters and the system is an assistant to facilitate the monitoring and provide the students with a work companion. This is meant to ensure a constant support for student within the framework of elearning. The system is thought to be enriched over time through the overhauls and refinements by the teacher, who will be able to rely on the student's feedbacks. The multi-layered architecture allows the segmentation of the computation in several meaningful thus easily describable and explainable steps.

The next step will be to carry out several experiments on an educational dataset in order to test our system first within the framework of a C++ programming course, and define meaningful classes of student's gesture, quality of work, and situation. Then it will be extended to other courses as a robustness test. A mid-term objective is therefore to compare the evolution of the classes throughout the different kind of courses.

REFERENCES

Claxton, G., 2013. *What's the point of school?: Rediscovering the heart of education*. Simon and Schuster Publishers, London, England.

Ennouamani S, Mahani Z, 2017. An overview of adaptive e-learning systems. 2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS) IEEE, pp. 342-47.

Grosz, B. J., 1996. Collaborative systems, (aaai-94 presidential address). *AI magazine*, Vol. 17, No.2, p. 67.

Haruna K. et al., 2017. Context-aware recommender system: A review of recent developmental process and future research direction. *Applied Sciences*, Vol. 7, No. 12, p. 1211

Honey P. et al, 1989. *Learning styles questionnaire*. Organization Design and Development, Incorporated, Peter Honey Publications Limited, England.

Isaza, C., et al., 2006, An optimization method for the data space partition obtained by classification techniques for the monitoring of dynamic processes. *Proceedings of the 2006 conference on Artificial Intelligence Research and Development*, pp.80-87.

Klašnja-Milićević A. et al, 2011. E-Learning personalization based on hybrid recommendation strategy and learning style identification. *Computers & education*, Vol. 56, No. 3, pp. 885-899.

Kozma, R. B., 1991. Learning with media. *Review of educational research,* Vol. 61, No.2, pp. 179–211.

Lai, V. and Tan, C., 2019. On human predictions with explanations and predictions of machine learning models: A case study on deception detection. *Proceedings of the Conference on Fairness, Accountability, and Transparency*. pp. 29–38.

Rabiu I. et al, 2020. Recommender system based on temporal models: a systematic review. *Applied Sciences*, Vol. 10, No.7, p. 2204.

Rana C. et al, 2015. A study of the dynamic features of recommender systems. *Artificial Intelligence Review,* Vol. 43, No. 1, pp. 141–153.

Rohrer D. et al, 2012. Learning styles: Where's the evidence? *Online Submission*. Vol. 46, No. 7, pp. 634–635.

Roux, E., 2015. Développement d'un outil d'aide au diagnostic pour la production de maïs permettant la réduction de la consommation en eaux d'irrigation et en traitements phytosanitaires. Doctoral Dissertation, INSA de Toulouse.

Schiaffino, S. et al, 2008, Eteacher: Providing personalized assistance to e-learning students. *Computers & Education*, Vol. 51, No. 4, pp. 1744–1754.

Stahl S. A., 2002. Different strokes for different folks. *Taking sides: Clashing on controversial issues in educational psychology*, pp. 98–107.

Vinagre J. et al, 2015. An overview on the exploitation of time in collaborative filtering. *Wiley interdisciplinary reviews: Data mining and knowledge discovery*, Vol. 5, No. 5, pp. 195–215.

Wolf, C., 2003. iWeaver: towards 'learning style'-based e-learning in computer science education. *Proceedings of the fifth Australasian conference on Computing education*, Vol. 20, pp. 273-279.

Yin M. et al, 2019, Under-standing the effect of accuracy on trust in machine learning models. *Proceedings of the 2019 chi conference on human factors in computing systems*, pp. 1–12.