AI Prediction and Teaching Strategies for a Two-Phase Engine in a Smart Learning Platform

Alberto Real-Fernández[1], Javier García-Sigüenza[1,2], Faraón Llorens-Largo[1], Rafael Molina-Carmona[1] *

[1] Smart Learning Research Group, University of Alicante (Spain)
[2] ValgrAI - Valencian Graduate School and Research Network for Artificial Intelligence (Spain)

* Corresponding author: alberto.real@ua.es (A. Real-Fernández), javierg.siguenza@ua.es (J. García-Sigüenza), rmolina@ua.es (R. Molina-Carmona), faraon.llorens@ua.es (F. Llorens-Largo).

Received 5 March 2024 | Accepted 4 March 2025 | Published 29 January 2026

Abstract

The impact and progress of Information Technologies has led to a process of change in most environments of our society, specially education. Even more with the current rise of Artificial Intelligence, what has led to the creation of different new tools aiming to improve the learning experience. This fact has contributed to the creation of systems that aim to adapt the learning process to each individual learner and offer them a personalised experience. The problem of letting automated systems manage the whole learning process is the lack of human factor, but learning objectives and teacher criteria are crucial. That is why this research proposes a solution that combines the potential of AI without neglecting the teacher decision. Concretely, the proposal is an AI model that selects the most suitable activity to each learner. To do so, this proposed model is structured in two phases. The first is the prediction phase, in which the model predicts the score a learner will obtain and the time they will spend to complete an activity. Then, in the second phase, the selection of a single activity is done by means of instructional strategies. These strategies are based on the previously obtained metrics and establish the criteria to follow for selecting activities. The selected strategy is always set by the teacher, who will guide the learners through the process. With this model, this research proposes a combination of AI techniques with human decision-making. Instead of relying the learning process to an automated engine, it includes the teacher as the one to guide the AI by making the last decision.

Keywords

Artificial Intelligence, Explainable AI, Instructional Strategies, Smart Learning.

DOI: 10.9781/ijimai.2026.6348

I. Introduction

In recent years, the impact and growing progress of Information Technologies (IT) has led to a process of change in most environments of our society. Even more with the current rise of Artificial Intelligence (AI). Specifically, education is one of the most affected, where many new tools and technologies have emerged. With the aim of improving the learning experience, this technology has turned education into a process to fulfill the new characteristics and needs of our current environment [1]–[3]. It is a digital transformation towards a continuous, dynamic and lifelong learning, different from the one we knew [4], [5].

Since in the learning process we can observe that each learner has different styles, needs and preferences, it is important to focus this transformation process on a learning model that can adapt to each individual and offer them a personalised experience [6], [7]. It is a concept known as smart learning, and it has been enhanced by the power of IT, particularly AI, which has contributed to the creation of algorithms and methodologies that aim to adapt the learning process to each student and their own way of learning.

In general terms, these are algorithms that are able to assign the most appropriate activity to a learner at any given time. This choice is made according to the learner's own characteristics and aiming to keep them motivated. But it is crucial to consider the learning objectives set by the teacher when using this type of technology in the learning process.

It is for these types of problems that Explainable Artificial Intelligence (XAI) is emerging as a field of research and development of modern AI systems, addressing the transparency, interpretability and trustworthiness of these systems [8]. As artificial intelligence permeates our lives, the ability to understand and trust AI-generated decisions becomes paramount.

In addition, the fact that those automated and autonomous learning systems assume all the learning process by themselves can lead to a change in the socialization and interaction of the learners [9]. Thus, educational technology should be developed by those principles that

Please cite this article as: A. Real-Fernández, J. García-Sigüenza, R. Molina-Carmona, F. Llorens-Largo. AI prediction and Teaching Strategies for a Two-Phase Engine in a Smart Learning Platform, International Journal of Interactive Multimedia and Artificial Intelligence, vol. 9, no. 6, pp. 76-85, 2026, http://doi.org/10.9781/ijimai.2026.6348

come from human decisions, avoiding to delegate all the pedagogical tasks in a automated system, so it is important to include a social factor. Keeping this concept in mind, the potential of IT and AI can lead to a real and needed evolution of the socio-educational environment [10].

To this end, we propose in this research an AI model created for the smart learning platform Khipulearn[1], whose main feature is that the selection of activities given to the learners is managed according to their individual learning progress. This selection process is usually made by an AI engine, but to achieve the before mentioned purpose of a social factor, we propose a two-phase model: a prediction phase made by the AI itself to estimate the results to be obtained; and a selection phase that is managed by instructional strategies created by the teacher. In this way, we aim to introduce a social factor by letting the teacher be the one who guides the AI selection engine. The detailed structure of this proposal is displayed in Fig. 1, where the whole process of activity selection in Khipulearn is represented.

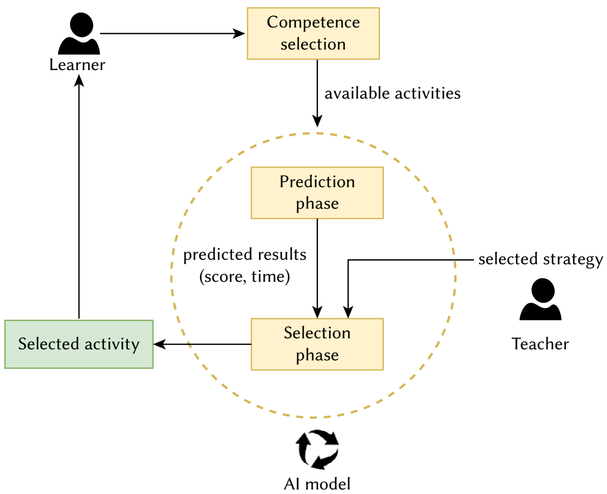


Fig. 1. The two-phase model proposed for activity selection engine in a smart learning platform.

Thus, firstly the background of the research is exposed in section II, that explains the main concepts of Khipulearn platform and its activity selection process, carried out by an AI model. Then, the two phases proposed in this research for this model are explained, in section III (prediction phase) and section IV (selection phase), detailing what they consist on and how they have been implemented. After the explanation of the proposed model, a practical application is given in V, showing results obtained in some pilot experiences and exposing an example of the model's decision process. Finally, section VI analyses the conclusions drawn from the research.

II. Personalised and Adaptive Learning Platform

The platform used in this research is called Khipulearn, which is based on CALM (Customized Adaptive Learning Model) [11]. CALM is a theoretical model for a learning system that aims to meet the needs of the digital society while still achieving the intentional objectives of teaching. It is described as a personalised and adaptive model that enhances learner motivation through interaction, reward and progression. It provides learning results in real-time and a continuous learning cycle, giving learners autonomy in their learning process [12]. The model offers a wide range of activities in terms of quantity and variety, and serves as a complete teaching tool, providing different ways to view and manage the process of each learner.

[1] https://khipulearn.com/

Khipulearn platform was created as part of the AdaptLearn project, which was funded by the Ministry of Universities as part of the UniDigital action of the Recovery, Transformation and Resilience Plan of the Government of Spain. The University of Alicante leads the development of this project, which aims to create an open, collaborative, flexible, and easily scalable system in conjunction with four other Spanish universities.

Let us see below the main concepts and features of Khipulearn platform.

A. Khipulearn

Khipulearn structures its learning content into competences, which are defined as the specific knowledge and skills that learners should acquire during a course [13]. These competences are connected in a directed graph known as a competence map. In this map, competences are the nodes and the edges represent the dependency relation between them. This map outlines the possible paths learners could follow in their learning process, allowing them to progress by completing and unlocking competences.

At the start of the course, learners have all competences locked, except for those designated as initial ones. They must unlock competences as they progress through the course.

Each competence has an associated acquisition value for each learner, representing their level of fulfillment, known as competence strength. Several concepts are linked to each competence based on this value:

- Minimum threshold: the required value to consider the competence as overcome.
- Maximum threshold: the highest value a learner can achieve in a competence, considering it as completed.
- Connection threshold: the minimum value required to unlock a dependency.

For example, Fig. 2 shows the competence map of a learner, where two competences are completed, one overcome, one started but not overcome and the rest of them locked.

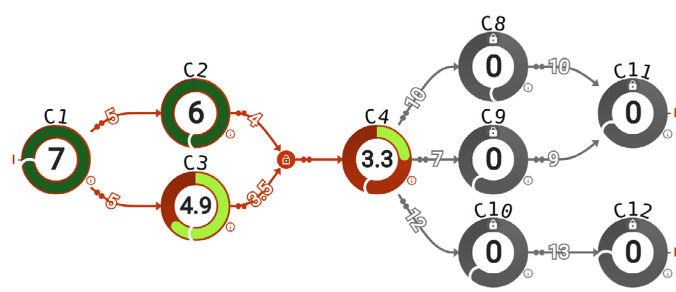


Fig. 2. Instance of a competence map for a learner, with two competences completed (C1 and C2), one overcome (C3), one unlocked and started (C4), and the rest of them that remain locked.

Learners will develop each competence by completing activities, which are tasks associated with one or more competences, and through which the learning content is presented. Besides, an activity in the platform, known as CALM activity, has specific characteristics:

- It is atomic, containing the minimum necessary information.
- It is relevant for the associated competences.
- It provides immediate feedback to the learner regarding their result.
- It requires interaction with the learner.

An example of such an activity is shown in Fig. 3, a multiple-choice question with answers presented in markdown style as programming code sections.

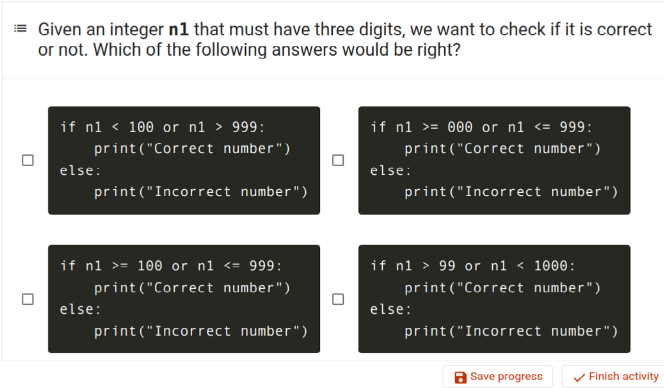


Fig. 3. Example of an activity given to a learner, in this case a multiple choice answer with markdown style.

The activities are assigned by the system when a learner selects a competence to develop. Learners can select any unlocked competence, which provides them with the mentioned autonomy in their learning process. They also have the option to choose from different paths available in the competence map. In this moment, the selection of an activity is made by the AI engine proposed in this research, further explained in the following sections.

The learning content is designed and managed by the teacher. They are responsible for creating the competences, designing the map, establishing connections between them and their associated threshold. And they will also create the activities related to the competences for the learners to complete.

On the other hand, Khipulearn is implemented in Angular, for its front-end application, that is connected to a Laravel API, using MySQL as database management system, concretely MariaDB. Then, the AI engine is implemented in Python 3, connected to a Python Flask API, and it communicates with the Laravel API using RabbitMQ.

B. Activity Selection Engine

As previously stated, learners develop and complete competences by performing activities, which are automatically assigned when they select a competence. This process embraces the entire logic of Khipulearn's adaptive learning, facilitated by the so-called selection engine. It is responsible for choosing the most suitable activity for each learner at any given time. For this purpose, an AI model is used to select the best activity according to a prediction based on the results obtained by the student.

Among the different existing techniques in the field of AI, the activity selection engine has been implemented using deep learning [14]. This technology has revolutionized the field of AI, enabling the development of new models with an accuracy never seen before. The capabilities of these new models allow accurate predictions to be made and AI to be applied to new fields, however, the development of deep learning brings with it the development of black boxes. This term refers to the fact that the developed models have a behavior that is too complex to be understood by humans [15], so that the result obtained cannot be explained.

In the field of education, being able to explain the different decisions that guide student learning is especially relevant, as each student has unique characteristics that affect the development of learning. Therefore, it is necessary to understand the decision making of the model to understand if the learning is being appropriate and adapted to the learner. There are different techniques to try to provide explainability to the predictions of deep learning models, which fall under the field of explainable artificial intelligence (XAI) [8]. However, these techniques are limited in the sense that they allow understanding the behavior of the model, but do not allow influencing its behavior, preventing the teacher from adjusting the behavior of the model if necessary.

In the platform, when a learner selects a specific competence to train, the system assigns them the activity that is deemed most suitable for them. This decision is based on the learner's current state, their characteristics and those of the available activities. In this way, the system adapts to the learner's performance and preferences as they progress through learning process.

To achieve this, the system stores a set of variables that represent both the activity and the learner. Each activity in the platform has static features that describe it individually, while dynamic information is stored for the learner, such as their skills, their progress and behavioral data. In Khipulearn this information is stored in a data structure called feature vector, one per each activity and one per each learner [16], [17]. Thus, the system can adapt to each learner individually by processing both feature vectors with the AI engine to choose the most suitable activity for the learner at any time.

Furthermore, in Khipulearn this selection process also depends on the instructional strategy. Due to the mentioned importance of involving the teachers in the decision-making process, they are able to monitor and manage both the overall state of the course and the individual learner's progress. They can also modify the strategy for the course or for a learner, based on any necessary changes in the learning process or if deemed appropriate. A strategy defines how the content is presented and assigned to the learners, it is a set of techniques and approaches used to optimize the learning process and to help encourage learner's motivation and attention, and understanding the knowledge to be acquired [18].

III. Prediction Phase

In order to provide decision-making capabilities in the use of AI, we propose a two-phase model: a first phase, in which an AI model obtains two metrics (score and time) that determine a prediction of the performance of a student for each possible activity; and a second phase which, based on the two metrics obtained, allows the application of one of the different instructional strategies available through algorithms, allowing the teacher to guide and personalize the student's learning at each moment.

In this section, to explain how the model works in its first phase, we define the structure established for its creation, that is based on the characteristics of the problem, being the data of the learning process a combination of sequential and spatial data. The problem presents both dimensions since the student presents an evolution in his knowledge based on the exercises performed previously, as well as structured, since the activities are related to each other, having common knowledge and being the result of some activities more relevant than others depending on the context. Therefore, it is necessary to look for an architecture that is able to work efficiently with both dimensions.

A. Model Architecture

As indicated above, the problem is composed of both temporal and spatial component data, so in order to obtain a model that allows using both types of data, the Graph Convolutional Network (GCN) [19] architecture, which allows processing spatial information, and the Gated Recurrent Unit (GRU) [20], which allows processing temporal information, have been combined.

Gated Recurrent Unit (GRU) is a type of Recurrent Neural Network (RNN) architecture designed to address certain limitations of traditional RNNs, such as difficulties in learning and retaining long-range dependencies on sequential data. The distinguishing feature of GRUs lies in their gating mechanisms, which control the flow of information within the network. Computation in GRUs involves a trade-off between forgetting and updating information, which allows the network to adaptively capture dependencies in sequential data. The simplified architecture of GRUs, compared to long short-term memory (LSTM) networks, makes them more computationally efficient and easier to train with fewer parameters.

A Graph Convolutional Network (GCN) is a type of neural network designed to process and analyse graph-structured data. Unlike traditional neural networks, which operate on grid-structured data such as images or sequences, GCNs are ideally suited for tasks involving relationships and dependencies between interconnected nodes in a graph. The fundamental operation of a GCN is graph convolution, inspired by convolutional operations in image processing. The idea is to aggregate information from neighboring nodes and update the representation of the target node accordingly.

Let $G(V, E)$ be an undirected graph, where V represents the set of n nodes, E is the set of edges and A the adjacency matrix of the graph G. Since the model is based on a convolutional graph network (GCN) whose graph convolution operation approximates a first-order polynomial Chebyshev expansion, it can be generalized to a multidimensional network as follows:

$$Z = (I_n + D^{-\frac{1}{2}}AD^{-\frac{1}{2}})X\Theta + b \tag{1}$$

with:

- $Z \in R^{n \times f} \longrightarrow$ Output of the GCN.
- $I_n \in R^{n \times n} \longrightarrow$ Identity matrix of size n×n.
- $D \in R^{n \times n} \longrightarrow$ Degree matrix.
- $A \in R^{n \times n} \longrightarrow$ Adjacency matrix of G
- $X \in R^{n \times c} \longrightarrow$ Input of the GCN layer.
- $\Theta \in R^{n \times n} \longrightarrow$ Weights of the GCN Layer.
- $b \in R^{f} \longrightarrow$ Bias of the GCN layer.
- $n \longrightarrow$ Number of nodes.
- $c \longrightarrow$ Number of input variables.
- $f \longrightarrow$ Number of output variables.

A model that makes use of GRU and GCN, and that has been shown to be accurate in problems using spatial and temporal features is Adaptive Graph Convolutional Recurrent Network (AGCRN) [21], which also makes use of the Node Adaptive Parameter Learning (NAPL) and Data Adaptive Graph Generation (DAGG) modules.

The NAPL module learns node-specific patterns by factoring traditional GCN network parameters and generating node-specific parameters from weights and biases from a shared set, since assigning a set of specific parameters $\Theta \in R^{c \times f}$ to each node generates a set of $n \times c \times f$ parameters, a number so large that it would make the model prone to overfitting. It is therefore proposed that the model learns two smaller matrices, then generates the matrix $\Theta \in R^{n \times c \times f}$ using the multiplication of both, i.e,

$$\Theta = E \cdot W \tag{2}$$

where $E \in R^{n \times d}$ is the embeddings matrix, $W \in R^{d \times c \times f}$ is the set of shared weights and d is the size of the embedding, where d is smaller than n and independent of the number of nodes. The use of both matrices allows to reduce the number of parameters to be learned by the network.

For the bias generated by b the same principle is followed, i. e.

$$b = E \cdot b_p \tag{3}$$

where b_p is a shared bias set.

In contrast, the DAGG module infersthe interdependencies between nodes from the data and generates graphs that adapt to the data so that the hidden dependencies of the data can be inferred, with the model itself calculating an interdependency matrix between the nodes. This interdependence matrix will be the Laplacian matrix of the graph $\widetilde{\mathcal{L}}$, from which the adjacency and degree matrices can be computed. Thus, the Laplacian $D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ is redefined as

$$\widetilde{\mathcal{L}} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}} = softmax(ReLU(E \cdot E^T)) \tag{4}$$

where the function $ReLU$ is used to avoid negative values and $softmax$ is used to normalize the values.

However, the AGCRN model is not capable of processing dynamic graphs. Therefore, modifications have been made to this model to generate a new version that we have called DynAGCRN, which is an adaptation of the model for problems with dynamic graphs, where the graph nodes can vary between iterations of the time sequence.

B. Dynamic Adaptative Graph Convolutional Recurrent Network (DynAGCRN)

Adding the ability to use dynamic graphs to the model requires two features: being able to calculate the embedding of any node at any time and being able to modify these embeddings based on temporal information.

To support training with nodes that can change between predictions, a layer has been used to generate a different embedding for each node and return those used in the current prediction. In this way, the graphs that represent the evolution of activity performance can use different nodes that vary according to the progression of the student, allowing the use of a dynamic structure that adapts to the graph generated for the context of the prediction.

The embeddings of the activities are static, since for an activity with the same characteristics its embedding will always be the same, but in the problem of predicting results the characteristics to be extracted from an activity can change due to the context. Therefore, apart from extracting an encoding using the embedding layer described above, we have also made use of a positional encoding, changing the encoding based on when the student performed the activity, adding the temporal component to the embedding. This technique has been effectively tested in other architectures [22], showing its potential. Thus, the resulting embedding is defined as:

$$E = E_N + E_P \tag{5}$$

where E_N is the embedding calculated from the node, E_p the positional encoding and E the embedding resulting from adding both and used by the model.

Finally, once all the spatial and temporal components are integrated, the definition of the model is:

$$\begin{aligned} \widetilde{A} &= (I_n + softmax(ReLU(EE^T))), \\ z_t &= \sigma(\widetilde{A}\,[X_{:,t}, h_{t-1}]EW_z + Eb_z), \\ r_t &= \sigma(\widetilde{A}\,[X_{:,t}, h_{t-1}]EW_r + Eb_r), \\ h_t &= z_t \odot \hat{h}_t + (1 - z_t) \odot \hat{h}_{t-1}, \\ \hat{h}_t &= \tanh(\widetilde{A}\,[X_{:,t}, r \odot h_{t-1}]EW_{\hat{h}} + Eb_{\hat{h}}) \end{aligned} \tag{6}$$

where E is the resulting embedding of the node, z_t is the update gate, r_t is the reset gate, h_t is the output at step t, x_t is the input at step t, $\hat{h}_t$ is the candidate activation in step t, W, U, b are parameters to be learned by the model, $\odot$ is the Hadamart product, σ is the sigmoid function, and $[\cdot]$ is the concatenation operator.

To obtain the final model, which will perform the prediction, several layers are chained, followed by a last layer that will perform a linear transformation, obtaining as a result, a matrix of dimension $R^{n \times \tau}$, where τ is the size of the window, indicating the number of steps to be predicted.

The resulting model is capable of estimating the result obtained by the student and the time required to perform an activity using both the temporal and spatial information of the problem by making use of dynamic graphs and supporting the performance of an estimation. The resulting architecture can be seen in Fig. 4.

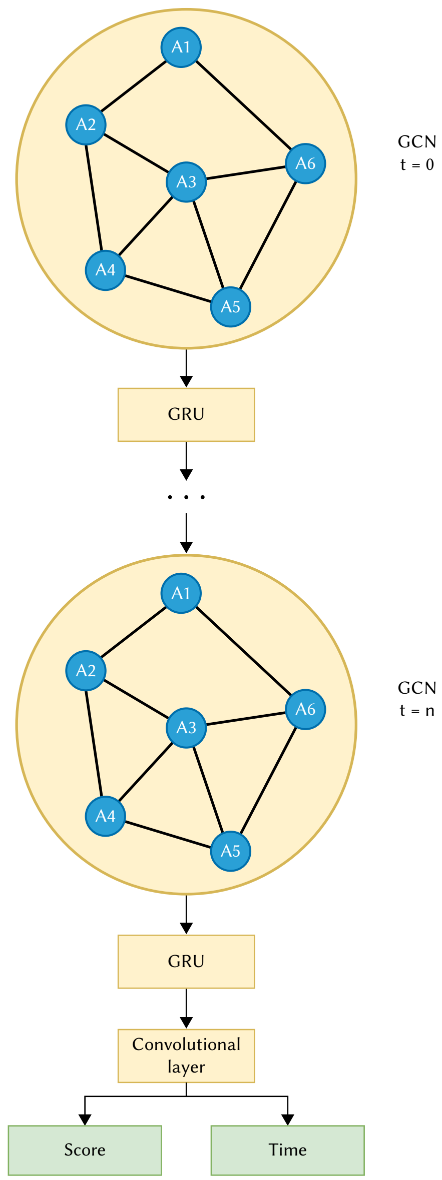


Fig. 4. Diagram of the prediction model architecture. In yellow the components of the model and in green the values obtained.

C. Model Training

To express the student's evolution, the graph is composed from the last n activities performed by the student, therefore at each value of t the graph is modified, eliminating the oldest activity and adding the new activity that could be performed, in order to predict the result for this one. In this way, the aim is to represent a graph that shows the progression of the student during the last activities and allows extracting information at a temporal and spatial level.

The model developed, once trained on the data set, allows obtaining for each of the possible activities, both an estimate of the score that the student will obtain and the time required to perform the activity. Together, both metrics allow to have an estimate of the performance of each student for each of the possible options.

The dataset used is a proprietary dataset, generated from the results obtained by different students answering true or false questions, where the score obtained in the activity could be 0 or 100. This dataset has been adjusted to have a similar distribution of answers with score 0 and 100, in order to avoid overfitting to the most common result. For training, different metrics have been tested, being the mean absolute error (MAE) the one that has provided the best result, being defined as:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{7}$$

The training was performed by splitting the data set into training, validation and test partitions. On the test set an average error of **29.18** in the score and **24.31** in the time of completion was obtained. This value shows the predictive capacity of the model, as the average error of the score with random predictions would be 50, thus demonstrating the ability to infer patterns and extract information from the data.

These metrics are used as input for the second phase of the model, which allows the application of the chosen instructional strategy.

IV. Selection Phase

After making the prediction of results as previously explained in the first phase, the AI model enters a second phase, the selection one. In this phase a single activity is selected by the application of instructional strategies. By means of these strategies, the learners can be guided through the whole process according to the criteria set by the teacher.

With this structure, our AI model represents a change of the paradigm compared to other existing models, in which the process is carried out as a black box. In this kind of models there is no chance to intervene in the learning process and there is no explanation given. Instead, our model proposes a AI-assisted learning that is guided by the teacher.

This second phase is implemented with different algorithms used to represent each instructional strategy. These algorithms make use of the variables obtained in the prediction phase, the activity history of each learner and a set of characteristics related to the activities. Through this implementation, the selected strategy can be modified at any time, both globally for all the learners in a course and individually for a concrete learner. It allows the deep learning model focus on the accuracy of the results instead of having to retrain it in case a learning objective is changed.

Therefore, let us deeply explain what instructional strategies consists on, what variables they consider, how they are applied and the ones proposed in our model.

A. Definition of Strategy

An instructional strategy, also known as teaching strategy, is a set of approaches and methods that teachers use to optimize the learning process. It establishes how the content is to be displayed and given to the learners, trying to encourage learners' attention and motivation, and help them to remember and understand learning content [18]. To successfully apply teaching strategies a learning system should include different instructional approaches, provide multiple and different learning resources, involving learners in their own learning path and letting them know their state.

These are some of the main features of Khipulearn, that includes a wide variety of activities to be done by the learners, it let them choose their own path and know their learning state at any time. And by means of the instructional strategies the teacher will be able to guide each learner to achieve the intended learning objectives. That is why both teachers and strategies play a key role in the platform, providing the selection engine with the guidelines to select the most suitable activity at any given time. Thus, while the AI model optimizes the learning process to each learner's features, it will also be aligned with the strategy to fulfill the teaching objectives.

In our proposal, the strategies depend on different types of variables: those obtained in the prediction phase of the AI model, the activities history and the characteristics related to the activities.

Firstly, we have the metrics computed on the prediction, score estimated for an activity and time spent in completing it. Those variables allow us to combine in different ways how a learner will achieve an activity, so that we can estimate it based on the expected score they will obtain or the time they are supposed to spend on completing it.

Then, the learners' history is also considered as an important variable, concretely their completed activities history. In this way, the system manage information about the last activities done by the learner, with all the data corresponding to the activity itself, the score obtained and the time spent.

On the other hand, the other set of variables used by the strategies is the one that characterize an activity and so a learner. As previously mentioned, those features are stored by the system in the so-called feature vector, being static in the case of the activity and dynamic in the case of the learner, depending on their progress.

One of these characteristics is the difficulty, which can be defined as the time and effort required to successfully complete an activity [23]. In the platform, the difficulty is set when an activity is added to a competence, aiming to have a set of activities with different difficulty levels, in order to provide a dynamic learning process, with a progressively increasing difficulty, depending on learner's state. Also the same activity can have a different difficulty in one competence than in another, since their contribution might differ depending on the content. Although in Khipulearn the difficulty is not explicitly used in the strategies, it has a direct influence in the score obtained by the learner, so it is important for the learning process.

In addition, there are other characteristics that define an activity itself regardless of the competence it is associated with, which are set by the teacher when creating an activity. In our case, the definition of these characteristics is based on LOM-ES v.1.0, an application profile created from the standard Learning Object Metadata (LOM) version 1.0, created by IEEE [24], [25]. The objective of this work is to design a reference framework to develop repositories of educational resources based on standardised digital educational objects, so we found it suitable to define the activities in Khipulearn. This standard establishes different categories to define an educational object, each one includes a set of so-called elements, a kind of subcategories that contain a value or multiple values.

From all the existing categories, we have selected those most suitable ones to our platform to define activities, which are General and Educational. The first one contains general information to globally describe an educational object, an activity in our case, between whose available elements we have selected title, description and language.

The second category, Educational, is used to define the educational and pedagogical features of an object. This category contains multiple elements to that end, among which we have selected a total of five: Interactivity Type, Interactivity Level, Learning Resource Type, Semantic Density and Cognitive Process. In this case, these elements present a set of predefined values to be selected, whose type of selection has been adapted to the platform requirements, resulting in three types:

- Single value: only one of the available values can be selected.
- Multiple value: more than one value can be selected.
- Weighted: more than one value can be selected, assigning each of them a weight between 0 and 100.

Besides, the element Learning Resource Type has been used to reference both the type of activity and the type of resources included in an activity, making a selection from all the available values options. And in Cognitive Process, that also present a large amount of options to choose, there has been also a selection of those considered as most important, based on Bloom's Taxonomy [26], [27].

To clarify this classification, Table I shows all the selected variables, their corresponding predefined values (if any), with our selection made, and the type of selection established to each value.

TABLE I. Table With the Resulting Variables Belonging to Activities' Feature Vector

Variable	Values	Type
Title Description Language	- - -	Text Text Text
Interactivity Type	active expositive combinative	Single
Interactivity Level	very low low medium high very high	Single
Learning Resource Type	Illustration Compound audio Tutorial Web Programming tool Closed exercise Questionnaire Video	Multiple
Semantic Density	very low low medium high very high	Single
Cognitive Process	understand create assess observe practise remember	Weighted

Regarding the learner, their feature vector contains their different variables referring to their history and performance in the learning process, but also contains the variables corresponding to Educational category, the same than activities, but in this case these will be dynamic. That means the values of these variables will be changing as the learner completes activities, aiming to describe a learner by their actions, instead of classifying them in a predefined way.

To do so, all the elements of the Educational category will have weighted values, so that they will represent how the learner has develop each of them. Then, the learner's feature vector (L) is updated using activity's vector (A) from the activity they have just completed, considering the obtained score, updating all the learner's characteristics one by one (L_x), from the corresponding activities' features (A_x), as follows:

$$\lambda = L_x + (\alpha * A_x * \beta) \tag{8}$$

Where α is a pass value that represents how significant the new activity is, being $\alpha \in [0, 0.1]$. And β is a multiplier that indicates if an activity adds or subtracts, depending on the obtained score (n):

$$\beta = \frac{n}{50} - 1 \tag{9}$$

Finally, to make sure the obtained result of λ is in range between 0 and 100, it is processed as follows:

$$\lambda = max(min((L_x + (\alpha * A_x * \beta)),0),100) \tag{10}$$

B. Proposed Strategies

Therefore, using all the variables mentioned, different possible combinations have been analysed to create the strategies, mainly based on the performance and state of the learner. The aim is for the teacher to be able to manage and guide their learning process, bearing in mind that it may be closely related to their motivation and attention.

This relation may seem obvious if we consider Csikszentmihalyi's Flow Theory [28], which states that a learner experiences a maximum level of motivation, known as a state of flow, when they are doing tasks that match their skills. They should simultaneously perform tasks they know they can complete with minimal difficulty and others that present a challenge. Therefore, the aim is to keep the learner in a state of flow, preventing them from falling into a state of boredom if the activities are too easy, or into anxiety if they are too difficult.

Thus, the proposed strategies are designed following the different state of the Flow Theory, managing the variables of obtained score, spent time and the history of completed activities. In total we propose five strategies, defined as follows:

- Flow. Strategy based on obtaining the highest score possible, what means the learner is doing activities according to their skills, so they are in a flow state.
- Control. It is focused on the activity in which the learner will spend as less time as possible, what would give the learner a sense of control.
- Arousal. Strategy based on obtaining the lowest score possible, keeping the learner in an arousal state, what could be interesting at some point in order to make the learner progress in their knowledge by doing more complex activities.
- Persistence. This strategy focuses on the activity in which the learner spend as much time as possible, unlike the control strategy.
- Burst the Bubble. This is the strategy based on the activity history of the learner, that changes automatically according to their results, aiming to prevent the learner from getting stuck in the activities they are best at.

As it may has been noticed, Burst the Bubble strategy presents a different performance, whose aim is to prevent the learner from continuously doing the activities they are best at. It is a complex strategy, that results from combining some of the so-called simple ones establishing some criteria. This could be done to create more strategies that combines simple ones and concrete criteria.

Concretely, in Burst the Bubble, the system assigns a default strategy among the simple ones, that is applied until the learner completes five activities correctly (considered as a score of 75 or higher) in a row. At this point, it changes to an alternative strategy. If following this strategy the learner finishes two activities in a wrong way (with a score lower than 50), it changes again to the default strategy. This is automatically managed by the AI engine, which selects the Flow strategy as the default one and the Arousal as the alternative, but they could be changed by the teacher. Fig. 5 shows the flow diagram that represents this strategy performance.

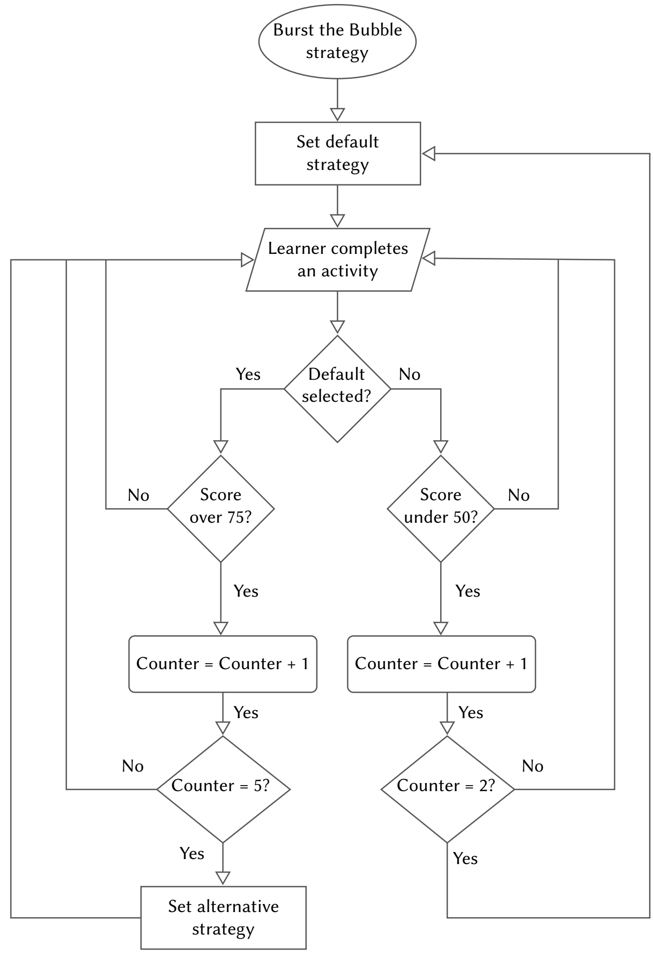


Fig. 5. Flow diagram that represents the strategy called Burst the Bubble, a custom strategy proposed to prevent the learner from getting stuck in the activities they achieve the best.

On the other hand, there are the previously mentioned variables of both learner and activity vectors. Since those belonging to Educational category are important to consider, the strategies include a filter option. This filter establishes how these features are taken into account. That means, how the selected activities will be, what it is done by three options:

- Variety. This is the default filter, that avoids to select similar activities to those recently completed.
- Reinforcement. This filter is focused on reinforcing those activities whose characteristics have the lowest weights.
- Avoid comfort zone. This filter is based on avoid those types of activities the learner is best at, that means those activities whose characteristics have the highest weights.

Therefore, in this selection phase, the second one of the proposed model, there is a two-step process. First, the teacher selects a strategy that establishes how learners will do the activities, and then they select a filter that determines how the activities will be. Both selection criteria can be applied globally for all the learners of a course or individually to a concrete learner. Fig. 6 represents the whole process of this second phase of the model.

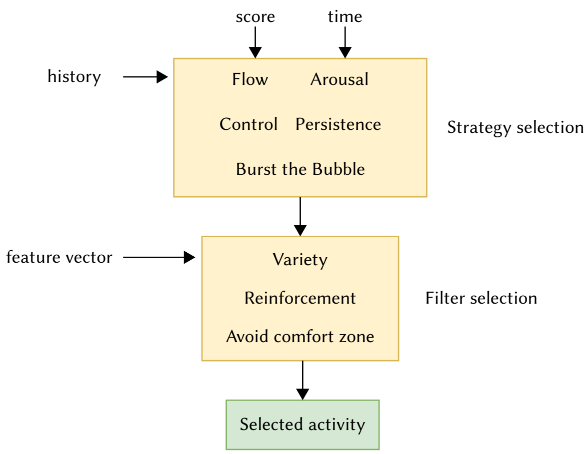


Fig. 6. Diagram of the selection phase of the model. It is made in two blocks: strategies (receiving score, time and learner history variables) and filters (receiving the feature vector variables).

V. Practical Application

As a pilot experience, Khipulearn has been used in three subjects in the University of Alicante, finished in the first semester of the current academic year. The experience of using Khipulearn has been successful for both teachers and students, and the learning results and the feedback received reflects so.

On one hand, in these results, exposed below, there are: the number of students in the course; the average score obtained in the completed activities (the score of each activity is between 0 and 100); and the percentage of students with all competences overcome (minimum threshold exceeded).

- Mathematics I (1st year of Degree in Multimedia Engineering): 107 students, 91.05 average score obtained and 90% of students with all competences overcome.
- Logic for Artificial Intelligence (1st year of Degree in Engineering in Artificial Intelligence): 62 students, 86.75 average score obtained and 91% of students with all competences overcome.
- Programming I (1st year of Degree in Engineering in Artificial Intelligence): 62 students, 86.33 average score obtained and 88.6% of students with all competences overcome.

On the other hand, the students completed a survey about the model of the platform, its main concepts and features, with positive results, summarised as follows:

- They find the competence map intuitive and they think the content divided in competences is really suitable.
- They find appropriate to be able to access the content they wish at any given time.
- They think it is suitable to set their own learning pace.
- They find useful to have multiple atomic activities in different formats.
- They find useful and suitable to know their own learning state and progress at any time.

From this experience, let us see an example of how the selection engine works in Khipulearn, showing the process since a student selects a competence and the system gives them an activity to complete. In this case, let us focus in a real case of a student of one of the mentioned subjects, at a given time of the process.

When this student selects a competence, concretely the one with id 13, the selection engine is called in order to select an activity. The AI model, in its first phase, makes a prediction for every available activity for the competence at this time. So that it obtains a real-time estimation of score obtained and time spent. Table II shows a list with the available activities for the mentioned competence, with the values of score and time obtained for each one.

TABLE II. Available Activities for the Competence With Id 13, With Their Predicted Values

Activity id	Time	Score
1893	24.05	87.45
1894	41.34	75.95
1895	27.89	90.00
1899	36.75	85.25
1902	51.78	68.50
1908	63.55	85.75
1909	41.16	72.25

At this point, the engine enters the second phase to select a single activity, decision that depends on the strategy selected by the teacher. To illustrate this process, Table III shows the available strategies, what they are based on and the activity that would be selected for each one, from the list in Table II.

TABLE III. Selected Activity Id From TABLE II According to the Selected Strategy

Strategy	Info	Selected activity
Flow	Highest score	1895
Control	Lowest time	1893
Arousal	Lowest score	1902
Persistence	Highest time	1908
Burst the Bubble	Based on history	1895

In the case of Burst the Bubble, let us suppose that the default strategy is Flow and the alternative is Arousal, being the first one still active. So as it can be seen, the selected activity is the same in both strategies. Let us see now what happens when the student selects another available competence, in this case the one with id 16. Table IV shows all the available activities with their predicted results.

TABLE IV. Available Activities for the Competence With Id 16, With Their Predicted Values

Activity id	Time	Score
2004	34.45	77.35
2007	52.68	85.70
2008	75.36	75.25
2012	61.56	65.25
2015	35.46	88.75
2016	28.94	72.25

And now, in Table V let us see the selected activity for each strategy in this case, from the list in Table IV.

In this case, Burst the Bubble has changed to the alternative option, so the activity selected would be the same than Arousal strategy.

TABLE V. Selected Activity Id From TABLE IV According to the Selected Strategy

Strategy	Info	Selected activity
Flow	Highest score	2015
Control	Lowest time	2004
Arousal	Lowest score	2012
Persistence	Highest time	2008
Burst the Bubble	Based on history	2012

VI. Conclusion

This research aimed to define an AI model for the smart learning platform Khipulearn. A model focused on the selection of activities to be given to learners adapting to them individually. Instead of relying this process to an automatic AI engine, it is important to involve the teacher in it.

To do so, this proposal combines the predictive capacity of AI models, specifically those using deep learning techniques, with the decision-making capability of the teacher. Since deep learning models are black boxes that prevent interpretation and intervention in the results, a two-phase AI model has been proposed.

In the first phase, a deep learning model is proposed, adapting it to process graphs as temporal series in which each node represents an activity of the learner. These activities represent the student's performance in different activities already done, so the results of other activities can be inferred from previous performance. This model has been validated on a defined dataset, demonstrating its ability to predict the performance of the existing learners.

In the second phase, an instructional strategy is applied, selected by the teacher. With this phase, the teacher is able to decide the approach to follow in the learning process of the learners. It allows the teacher to intervene in the performance of the model, using the metrics of the predicted results in the first phase.

Thus, with this proposal this research presents three main contributions:

- DynAGCRN model: adds the ability to process dynamic graphs to AGCRN model, allowing this change in the architecture of the model to use it in other cases where dynamic graphs need to be used.
- The selection phase: includes a human intervention in an automated decision-making process. This represents a solution to some of the questionable aspects of the rise of AI, allowing a human to make the last decision.
- Two-phase architecture: an AI model focused on combining the potential of AI techniques with the intervention of the teacher, and a explainable AI. It could be a great solution for smart learning platforms like Khipulearn.

Regarding the pilot experience, the learning results and the feedback obtained reflect that we are in the right direction. The aim is to continue using this platform in many subjects, repeating those already finished and new ones too. We would like to analyse their results and compare them, trying to embrace different type of subjects.

These new experiences could give us more information involving the learning process and results, but also about the platform performance. The idea for the next academic year is to use the platform in some subjects along the whole semester, what will let us compare the results of the students using the platform and without it.

Furthermore, we would like to continue the development of the platform, mainly to add more features and tools for supporting the teacher. Also, the aim is to enhance different aspects regarding performance and user experience.

One important factor we would like to improve is the teacher experience for reviewing the progress and results of the learners. They can currently check the state of each learner in the course, but there is more information to be added that could be useful. And most importantly, we consider necessary a complete dashboard with real-time information regarding all the learners and the course. With this feature, they could have all the important information at a glance.

On the other hand, we aim to improve the explainability of the AI model's decisions. That is, explore the possibilities of DynAGCRN to provide information about the score and time prediction process. This could be interesting to let the teacher know more about the activity selection and the strategy impact in the learning process.

Credit Authorship Contribution Statement

Alberto Real: conceptualization, investigation, writing – original draft, writing – review and editing.

Javier García: conceptualization, validation, writing – original draft, writing – review and editing.

Faraón Llorens: conceptualization, supervision, writing – review and editing.

Rafael Molina: conceptualization, supervision, writing – review and editing.

Data Statement

The data of this research is not publicly available due to its sensitivity and data protection policies.

Declaration of Conflicts of Interest

We have no conflict of interest to declare.

Acknowledgment

Khipulearn platform that supports this research is funded by the AdaptLearn project of the University of Alicante, within the UniDigital action of the Recovery, Transformation and Resilience Plan of the Government of Spain.

The work of Javier García-Sigüenza has been supported by ValgrAI – Valencian Graduate School and Research Network for Artificial Intelligence and the Generalitat Valenciana.

References

[1] S. Bedenlier, M. Bond, K. Buntins, O. Zawacki-Richter, M. Kerres, "Facilitating student engagement through educational technology in higher education: A systematic review in the field of arts and humanities," *Australasian Journal of Educational Technology*, vol. 36, p. 126–150, Jan. 2020, doi: https://doi.org/10.14742/ajet.5477.

[2] F. García-Peñalvo, A. Vázquez-Ingelmo, "What do we mean by GenAI? a systematic mapping of the evolution, trends, and techniques involved in generative AI," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 8, pp. 7–16, 12/2023 2023, doi: https://doi.org/10.9781/ijimai.2023.07.006.

[3] F. J. García Peñalvo, F. Llorens-Largo, J. Vidal, "The new reality of education in the face of advances in generative artificial intelligence," *RIED. Revista Iberoamericana de Educación a Distancia*, vol. 27, p. 9–39, Jan. 2024, doi: https://doi.org/10.5944/ried.27.1.37716.

[4] M. Area-Moreira, J. Adell-Segura, "Tecnologías digitales y cambio educativo. una aproximación crítica.," in *REICE. Revista Iberoamericana sobre Calidad, Eficacia y Cambio en Educación*, 2021, pp. 83–96.

[5] A. Goodwill, S. Chen, "Embracing a culture of lifelong learning: the science of lifelong learning," *UNESCO Institute for Lifelong Learning*, 2021.

[6] K. Robinson, L. Aronica, *Finding your element: How to discover your talents and passions and transform your life.* Penguin, 2014.

[7] B. Fleming, "Adaptive learning technology: What it is, why it matters," 2014. Accessed: May 16, 2017. [Online]. Available: http://www.eduventures.com/2014/04/adaptiveLearning-technology-matters.

[8] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, R. Chatila, F. Herrera, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," 2019. doi: https://doi.org/10.48550/ARXIV.1910.10045.

[9] J. Adell Segura, L. Castaneda Quintero, F. M. Esteve Mon, "Towards ubersity? conflicts and contradictions of the digital university," RIED-Revista Iberoamericana de Educacion a Distancia, vol. 21, no. 2, pp. 51–68, 2018.

[10] OECD, Positive, high-achieving students? TALIS, OECD Publishing, Paris, 2021.

[11] A. Real-Fernández, *CALM: un modelo de aprendizaje adaptativo y personalizado.* Phd Thesis, Universidad de Alicante, 2022.

[12] A. Real-Fernandez, F. Llorens-Largo, R. Molina-Carmona, "Smart learning model based on competences and activities," in *Innovative Trends in Flipped Teaching and Adaptive Learning*, IGI Global, 2019, pp. 228–251.

[13] W. C. Rivenbark, W. S. Jacobson, "Three principles of competency-based learning: Mission, mission, mission," *Journal of Public Affairs Education*, vol. 20, no. 2, pp. 181–192, 2014.

[14] Y. LeCun, Y. Bengio, G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[15] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[16] A. Real-Fernández, R. Molina-Carmona, M. L. Pertegal-Felices, F. Llorens-Largo, "Definition of a feature vector to characterise learners in adaptive learning systems," in *RIIFORUM* 2019. SPC, A. Visvizi, M. Lytras Eds., Cham: Springer, 2019, pp. 75–89.

[17] A. Real-Fernández, R. Molina-Carmona, F. Llorens-Largo, "Computational characterization of activities and learners in a learning system," *Applied Sciences*, vol. 10, no. 7, 2020, doi: https://doi.org/10.3390/app10072208.

[18] A. Real-Fernández, R. Molina-Carmona, F. Llorens-Largo, "Instructional strategies for a smart learning system," in *Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancingr Multiculturality*, TEEM'19, New York, NY, USA, 2019, p. 671–679, Association for Computing Machinery.

[19] T. N. Kipf, M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[20] K. Cho, B. van Merrienboer, D. Bahdanau, Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *CoRR*, vol. abs/1409.1259, 2014.

[21] L. Bai, L. Yao, C. Li, X. Wang, C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," *Advances in neural information processing systems*, vol. 33, pp. 17804–17815, 2020.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, pp. 5998–6008, 2017.

[23] F. J. Gallego-Durán, R. Molina-Carmona, F. Llorens-Largo, "Estimating the difficulty of a learning activity from the training cost for a machine learning algorithm," in *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality - TEEM'18*, 2018, pp. 654–659, ACM Press.

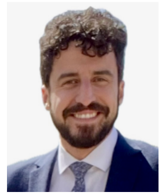
[24] "Ieee standard for learning object metadata," *IEEE Std 1484.12.1-2020*, pp. 1–50, 2020, doi: https://doi.org/10.1109/IEEESTD.2020.9262118.

[25] M. Canabal, A. Sarasa, J. C. Sacristán, "Lom-es: Un perfil de aplicación de lom," in *V Simposio Pluridisciplinar sobre Diseño y Evaluación de Contenidos Educativos Reutilizables*, 2008.

[26] B. S. Bloom, *Taxonomy of Educational Objectives: The Classification of Educational Goals.* 1956.

[27] L. W. Anderson, D. R. Krathwohl, *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives: complete edition.* Addison Wesley Longman, Inc., 2001.

[28] M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience.* Perennial Modern Classics, Harper & Row, 1990.



Alberto Real-Fernández

Associate professor at the University of Alicante, in the Department of Computer Science and Artificial Intelligence. PhD in Computer Science (University of Alicante, 2022). He is the technical director of AdaptLearn project, within the UniDigital plan, whose objective is to develop a smart learning platform based on AI, as a collaborative and open tool between different universities. Is one of the authors of CALM, the learning model this platform is based on. His main researches and publications are focused on smart learning systems, learning models, students and activities characterization and AI applied to education.

Javier García-Sigüenza

He holds a Computer Science degree from the University of Alicante since 2019, where he also obtained a M.Sc. in Data Science in 2022. Currently, he is pursuing a PhD in Computer Science, holding a predoctoral scholarship granted by ValgrAI – Valencian Graduate School and Research Network of Artificial Intelligence and the Generalitat Valenciana. His research focuses on the possible applications of artificial intelligence on graph-structured data and the integration of explainability into these models.

Faraón Llorens-Largo

Professor of Computer Science and Artificial Intelligence of the University of Alicante. PhD in Computer Science. Director of the Polytechnic School (2000-2005) and Vicerector of Technology and Educational Innovation (2005- 2012), both at the UA and Executive Secretary of the ICT Sector Commission of the CRUE (2010-2012). His work is in the fields of artificial intelligence, adaptive learning, gamification and video games, IT governance and digital transformation of educational institutions.

Rafael Molina-Carmona

Professor of Computer Science and Artificial Intelligence of the University of Alicante. PhD in Computer Science. Former Director of department of Computer Science and Artificial Intelligence (2008-2012). Current Vice Rector for Digital Transformation at University of Alicante. Coordinator of the Smart Learning Research Group. Research interests on the applications of Artificial Intelligence to different fields: computer aided design and manufacture, computer graphics, learning, creativity, information representation and IT governance.