



A feedback model applied in a ubiquitous microlearning environment using SECA rules

Marta S. Tabares¹ · Paola Vallejo¹ · Alex Montoya¹ · Daniel Correa¹

Accepted: 9 December 2021 / Published online: 8 February 2022
© The Author(s) 2022

Abstract

Understanding learners' behavior is the key to the success of any learning process. The more we know about them, the more likely we can personalize learning experiences and provide successful feedback. This paper presents a feedback model implemented in a ubiquitous microlearning environment based on contextual and behavioral information and evaluation results. The model uses SECA rules where the Scenario (S) represents the ubiquitous context variables reflecting the learner behavior during the learning process. The Event (E) identifies the probability that a learner fails or passes its evaluation. Condition (C) evaluates the results of the events. Moreover, Action (A) provides feedback to the learner. The proposal is developed through a controlled experiment whereby a microlearning environment can collect data from a ubiquitous context. The feedback model applies an analytics process to find the best context and behavior variables through different classification models. Those models predict whether a learner could fail, determine evaluation results' causes, and provide feedback. The Random Forest was the model with the best performance. Thus, 94% accuracy, a 97% Recall, a 93% Precision, an F1 score of 95%, and a Jaccard of 91%. Hence, each scenario is defined from a branch of every tree obtained from the Random Forest model personalizing feedback actions applying clustering techniques. Finally, we presented an exemplified set of feedback rules, providing automatic recommendations and improving learner experiences. Thus, the experiment allows analyzing the learner behavior in a ubiquitous microlearning context from a feedback perspective.

Keywords Ubiquitous microlearning environment · Ubiquitous context · Feedback rules model · Learning process · ECA rules

✉ Marta S. Tabares
mtabares@eafit.edu.co

Extended author information available on the last page of the article

Introduction

Nowadays, ubiquitous learning (u-learning) guides novelty models, rules, and strategies to grow virtual education. The technological development of mobile devices enables the users' context capture. The student population's massive use of mobile devices allows analyzing aspects influencing the mobile learning (m-learning) process, especially in microlearning environments that deliver recommendations and feedback to learners. According to Lin et al. (2019), microlearning is "the service that generates and provides users personalized small chunks of learning materials". Microlearning refers to small learning units, short-term activities, and flexible technologies that can enable learners to access them more easily at specific moments and conditions Hug (2006). Therefore, understanding learners' behavior in microlearning environments is essential to successful learning, specifically when a feedback process is included. For example, Lin et al. (2019) presented a microlearning framework that provides personalized content to improve learning. Mohammed et al. (2018) presented microlearning as an alternative to support the students' memorization process. The authors used a manual feedback form to evaluate the motivation for learning.

Even so, feedback is still a challenging activity in the learning process. According to Sedrakyan et al. (2020), proactive measures can be seen from feedback with two main streams of knowledge. The first stream of knowledge is based on explanations aiming at improving the cognitive dimensions of knowledge (*e.g.*, understanding). The second stream of knowledge focuses on guidance to influence student behavior (*e.g.*, participating in a specific activity related to a successful learning path). We are interested in the second feedback stream of knowledge, which assumes that the students have the theoretical knowledge and seeks to orient them about the practices to obtain better results. According to Lim et al. (2019), "a challenge for contemporary educators is how to provide feedback to all students in large courses in a personalized, timely and instructive manner."

This challenge inspires us to improve the learning feedback strategies taking advantage of ubiquitous technologies. We start with the question: why does a student fail an exam? It is complex to answer, even more in current times when most students are not face-to-face in a controlled environment. Instead, they attend classes using ubiquitous environments with digital technologies, *e.g.*, smartphones, tablets, laptops, wearables, and handheld devices. Therefore, that requires understanding those environments' behavior or scenarios whereby students consume learning content and evaluate their performance through tests anytime, anywhere, and with any device. The data captured from these ubiquitous environments can achieve effective and more accurate feedback (*e.g.*, personalized feedback). Specifically, it refers to testing events that identify fail results and trigger actions by applying analytical and intelligence models. Those actions help the participants (learner-teacher) improve the learning process.

The key contributions of this paper are the following:

- We propose a feedback model that is based on SECA rules to support ubiquitous microlearning. The model is designed to conduct the evaluation and give feedback about the result applying SECA rules. Scenario (S) represents the ubiquitous context variables reflecting the learner's behavior during the learning process. The Event (E) identifies the probability that a learner fails or passes its evaluation. Condition (C) evaluates the results of the events. Moreover, Action (A) provides feedback to the learner.
- We implemented a microlearning environment in a PWA (Progressive Web App) to collect data from a ubiquitous context during the learning process. Thus, we took advantage of microlearning characteristics, recreating a short course.
- We developed the proposal using the methodologies of Design Science in Information Systems Research (ISR) Hevner et al. (2004) and ASUM-DM adapted by Angée et al. (2018). Thus, we conducted a controlled experiment whereby a microlearning environment was designed to understand how the learners could achieve the learning goals through a learning path.
- We propose that the model uses the random forest tree trained with 16 different variables selected using an adapted genetic algorithm. Experimental tests were set up, choosing the decision trees with the most representative samples. It allowed us to identify patterns through the tree's branches with the probability that a learner fails. Thus, the model could diagnose weaknesses in the user context and take feedback actions that characterize the area of improvement and the context that could influence the evaluation results.

Considering the above, we proposed a feedback model to extend the work presented in Tabares et al. (2021). This model applies SECA rules where the Scenario (S) represents the ubiquitous context variables reflecting the learner behavior during the learning process. The Event (E) identifies the probability that a learner fails or passes its evaluation. Condition (C) evaluates the results of the events. Moreover, Action (A) provides feedback to the learner. We developed the proposal using the methodologies of the Design Science in Information Systems Research (ISR) Hevner et al. (2004) and the ASUM-DM adapted by Angée et al. (2018). Thus, we conducted a controlled experiment whereby a microlearning environment was designed to understand how the learners could achieve the learning goals through a learning path. Furthermore, the microlearning environment was implemented in a PWA (Progressive Web App) to collect data from a ubiquitous context during the learning process. Hence, the data could be available to the feedback model and used through an analytics process to: predict if a learner fails or passes, to find the best context and behavior variables applying different classification models, to determine evaluation results' causes, and to provide feedback.

This paper is structured as follows. “[Literature review](#)” section presents the literature review. “[Methodology](#)” section illustrates the methodology that guides the paper's development. “[Microlearning environment setup](#)” section defines a microlearning environment setup. “[Tool development](#)” section describes the tool used to capture students' data. “[Experiment](#)” section presents an experiment with students and corresponding data analysis. “[Feedback Model](#)” section presents the feedback

rules definition. Finally, “**Conclusion**” section summarizes the contributions and suggests future research directions.

Literature review

Mobile devices provide a powerful platform for learning content anytime and anywhere, ideal for microlearning activities. Ubiquitous computing provides context variables that help understand elements that directly influence individuals’ learning. Advances in mobile devices and networks have enabled m-learning and u-learning, supported by context information captured from sensors integrated into mobile devices. In this way, it is possible to capture ambient noise, internet network speed, ambient light, battery level, network speed, and more information about the learner’s learning context. This context is used by recommender systems, Machine Learning models, and analytical systems to improve learner’s performance. These education applications seek to provide students with feedback based on the context information and their activities. For example, Sakamura and Koshizuka (2005) used the ubiquitous learning concept in a ubiquitous digital museum to learn in a daily living environment instead of classrooms or textbooks. Marinagi et al. (2013) proposed an integrated system (IS) based on blended learning, which allows m-learning (mobile learning),

e-learning (electronic learning), u-learning, and classroom learning. In this way, virtuality is combined with face-to-face learning. The IS system and its different components provide adaptability to students based on the contents shown. It also has a Help Desk, which allows mutual help and feedback to the students. Wu et al. (2013) proposed a context-aware expert system based on u-learning that enables students to observe and compare rocks to identify characteristics as type, texture, color, and hardness. The expert system evaluates whether students correctly identify rocks and presents them with hints or guides that act as feedback when they do not correctly identify rocks.

Cheng et al. (2020) used microlearning principles to emit digital badges (micro-credential) to certify what has been learned. Aldosemani (2019) presented the effectiveness of microlearning from learners’ perspective regarding its major supports and challenges. Lee et al. (2021) explored the impact of mobile microcourses in learning efficacy.

Hung et al. (2010) presented a proposal based on ubiquitous computing and the integration of an e-library. In the proposal, evaluations are generated for students on ecological topics. From the results of these evaluations, feedback is developed to improve and accelerate the learning process. The underlying idea is “if it is known how students think during learning, then teachers will help their students overcome conceptual issues to improve their learning.” Coulby et al. (2011) conducted a study. They used a context-aware m-learning system to provide feedback, which allowed students to improve their skills due to the quick feedback delivered from their results. Chen et al. (2013) proposed a learning system that performs feedback to students from a URL reading or a QR code related to the contents displayed in the system. The system provides two types of feedback,

one that indicates the contents that have not been visited in the learning process. The other provides feedback from the results in the assessment. Chu et al. (2019) proposed an AR-based learning system for an architecture course. The system is based on formative assessment, “which guides students to find answers on their own by giving hints when they fail to correctly answer questions.” The study investigated the formative assessment strategies usage in mobile learning activities. According to Chu et al. (2019), several studies consider that formative assessment is crucial to improve learners’ learning performance and assist professors in providing timely feedback to learners. It is also critical to engage learners in a self-reflection process to develop more effective learning. The study considers that a successful setting is measured by learning achievements, increased motivation, and reduced cognitive load. Shang (2017) suggested using asynchronous peer feedback (APF) and synchronous corrective feedback (SCF). He also mentioned that students prefer asynchronous feedback. Li et al. (2020) conducted a quasi-experimental study to examine the usage of feedback delivery methods (text only, video only, or both). The findings revealed that students who received just video or text feedback were most motivated than those who received both video and text feedback. Wang and Lehman (2021) designed a personalized motivational feedback based on students’ achievement goals.

We selected the following related work to compare our proposal, highlighting characteristics such as research aim, feedback proposal, and Analytics, ML, or AI applied techniques. Nasim et al. (2017) presented a combination of ML and lexicon-based approaches for sentiment analysis of student feedback. The feedback is from sentiments expressed by students. Those are collected in a knowledge base in which TF-IDF and lexicon-based features techniques are applied to provide valuable insights to improve the overall teaching quality and methodology. Kochmar et al. (2020) presented automatically generated personalized feedback. The proposal uses the Korbit learning platform from which automated and personalized feedback is generated considering students’ individual needs; thus, the feedback is without expert mediation or predefined rules. The proposal uses Machine Learning, NLP (Natural Language Processing) techniques, and Decision Trees to provide students with personalized suggestions, Wikipedia-based explanations, and mathematical suggestions. Yu and Wu (2020) started from learning processes activated by student-generated feedback corresponding to potential answers given to student-generated questions (SGQ). These were explained, and their learning effects were examined applying covariance (ANCOVA). The proposal presents a non-equivalent pretest–posttest quasi-experimental research study that lasted nine weeks with 109 students. The content analysis of students-generated feedback provided the finding where significantly more benefits were gained from students engaging in feedback-generation for SGQ. Edalati et al. (2021) proposed and conducted a study to evaluate various ML models for aspect-based opinion mining to address the students’ feedback assessment effectively. The proposed approach is trained and validated on a large-scale dataset of manually labeled students’ comments collected from the Coursera online platform. Thus, it is possible to identify teaching-related aspects and predict opinions/attitudes of students using ML techniques as Random Forest, Support Vector Machine, and Decision Tree.

As it is possible to see in the previous review, most of the proposals define the feedback from comments in text previously provided by the learners from different types of courses. Thus, large volumes of data are analyzed using ML and AI techniques to find new ways to improve teaching–learning. On the other hand, our proposal presents a ubiquitous microlearning environment. It provides the learners short courses that are monitored in a context-aware to provide feedback in real-time and quasi-real-time. Data captured through the learning path let us understand how the learner use content and takes the evaluations. During that process, we analyze in real-time the progress and send forward feedback messages. Besides, among a set of context variables, we detect the best predictors applying machine learning techniques like Random Forest, SVM (Support Vector Machine), Logistic Regression, etc. Thus, we identify behavior patterns from the random forest trees to provide personalized feedback in quasi-real-time related to the context, which could influence the evaluation results.

Methodology

Figure 1 shows the methodologies used to achieve the proposal development. We used phases of both the Design Science in Information Systems Research (ISR) Hevner et al. (2004) and ASUM-DM adapted by Angée et al. (2018) to support some tasks between them. IRS methodology provides a “conceptual framework for understanding, executing, and evaluating IS research combining behavioral-science and design-science paradigms.” ISR is carried out in three phases: Environment, IS Research, and Knowledge Base.

The Environment phase guides identifying “goals, tasks, problems, and opportunities that define business needs as people perceive them within the organization.” We considered the Target Group and the Project Team. The Target Group consisted of individuals playing the learner role. They were identified from the business’ needs and classified through the working progress. The Project Team was responsible for creating and evaluating the microlearning application and make the data analytics process.

The team consisted of four subgroups:

- *Creatives* They are individuals responsible for understanding the students’ education needs and define the project’s basis. To recognize the environment, they established what the problem was. Higher education researchers and startups provided ideas from a research project to achieve an innovative solution. They created the following Researchers, Engineers, and Data Scientists subgroups to be part of the strategic plan, execute the defined activities and achieve the main objective.
- *Researchers* Individuals with research capabilities to create new knowledge and scientific outcomes. Senior researchers, postgraduate students, and undergraduate students are part of this group. They were responsible for carrying out reviews (e.g., a Systematic Mapping Review) and identifying the concepts to be applied and developed in the project.

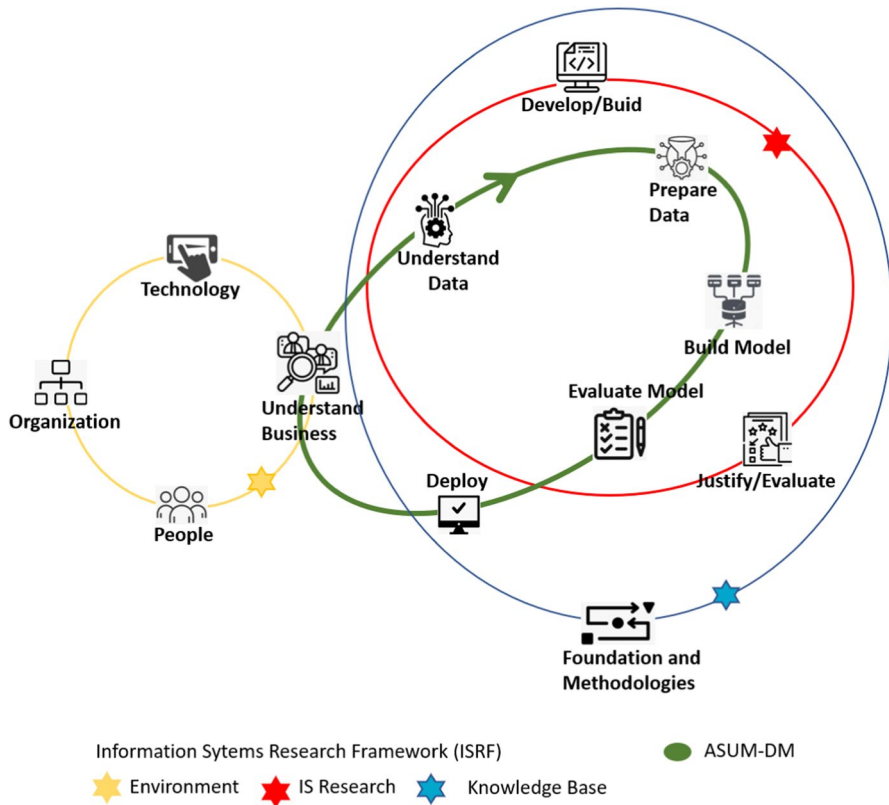


Fig. 1 Information system research framework and ASUM-DM methodologies

- *Engineers* Individuals with skills in the design and development of information technology products. They perform tasks of primary research when new technologies are required to create complex architectures. In addition, they have experience working with agile methodologies.
- *Data Scientists* Individuals with skills in data mining, Machine Learning, deep learning, and visualization. They worked with data analytics and ASUM-DM.

The knowledge base phase was made by the Project team, mainly from Creatives and Researchers. This phase guarantees that the results are research contributions and are the product of applying appropriated theories and methods. The team developed a Systematic Mapping Review (SMR) to trace and categorize the existing literature, including mobile learning, ubiquitous learning, context-aware, learning processes, and analytical. Thus, the data capture selection defined in 4.3 was based on a collection of variables treated by several authors, which can constitute a ubiquitous context. These variables are categorized by Vallejo-Correa et al. (2021) as internal, external, and activity.

IS Research phase puts the proposal in action. First, the Project Team defined activities and tasks to develop the ideas with the Target Group. The two main activities carried out at this stage are:

- *Develop/Build* End-user motivations and business needs are considered to identify ideas generated by making brainstorming meets. The gathered ideas will be transformed into different artifacts described in “Microlearning environment setup, Tool development, Experiment, [Feedback Model](#)” sections. Here, an agile plan defines sprints which include tasks of prototyping, experimentation. Besides, we applied the ASUM-DM to guide the data analytics process. In ASUM-DM, data understanding allows initial data collection and identification of data quality issues. Then, in the data preparation step, data cleaning is performed. It covers all activities to construct the final dataset from the initial raw data. Then, the modeling step allows building models using data mining tools. Next, the evaluation step determines if the results meet the project objectives and identify issues that require an early arrangement. Finally, the deployment step allows putting the resulting models into practice.
- *Justify/Evaluate*. At the end of each sprint, the Project Team reviewed and made a retrospective of the achieved product and learned how it must be improved from the focus group. Several sprints were required because the product was defined from the research process.

Microlearning environment setup

Students reflect a need to maintain regular and periodic learning rather than training marathons, which can be achieved through microlearning. According to Lin et al. (2019), the microlearning concept “aims to effectively utilize learners’ fragmented time to carry out personalized learning activities through online education resources.” For example, Correa et al. (2018) used microlearning to accelerate the learning experience of novice software web developers. Redondo et al. (2021) used micro-learning content in formal distance learning environments. Microlearning’s main characteristics are:

- Learning experiences are short and focused, allowing students to access them when and where convenient.
- Learning resource formats are varied, dynamic, and valuable.
- Each learning resource acts as an independent, self-contained resource but can also be part of a learning program.
- A learning resource focuses on a single learning outcome that allows learning a new skill or knowledge.
- Data collected on learners’ accuracy, speed, and confidence in responding to questions about the learning content tailor the content to the learners.

We established a microlearning environment through a short course, taking into account the advantage of microlearning characteristics. In our prior work, we had

developed a microlearning structure based on Topics of knowledge and Contents. In this paper, we extended the structure by adding the top layer named Theme. This new layer facilitates the decomposition of a theme into more specific knowledge segments. Besides, we made explicit the relationship between the learners, their devices, and the used channels.

Figure 2 shows an overview of the microlearning environment designed to achieve the learning goals through a learning path. The learning path represents learner interactions and the diverse and dynamic conditions of the context in which learning occurs. In that figure, the green area represents the learner interactions (see “Learner interactions” section); the blue area represents the contents’ structure (see “Course content structure” section); the gray area represents the captured context data (see “Data to be captured” section). Finally, the red area represents the feedback rules application (see “Feedback Model” section).

Learner interactions

The microlearning environment starts with a learner interaction. First, the learner uses a device (for example, a laptop or a mobile phone) to access the learning system (Progressive Web Application—PWA, available at <https://omnilearning.dis>.

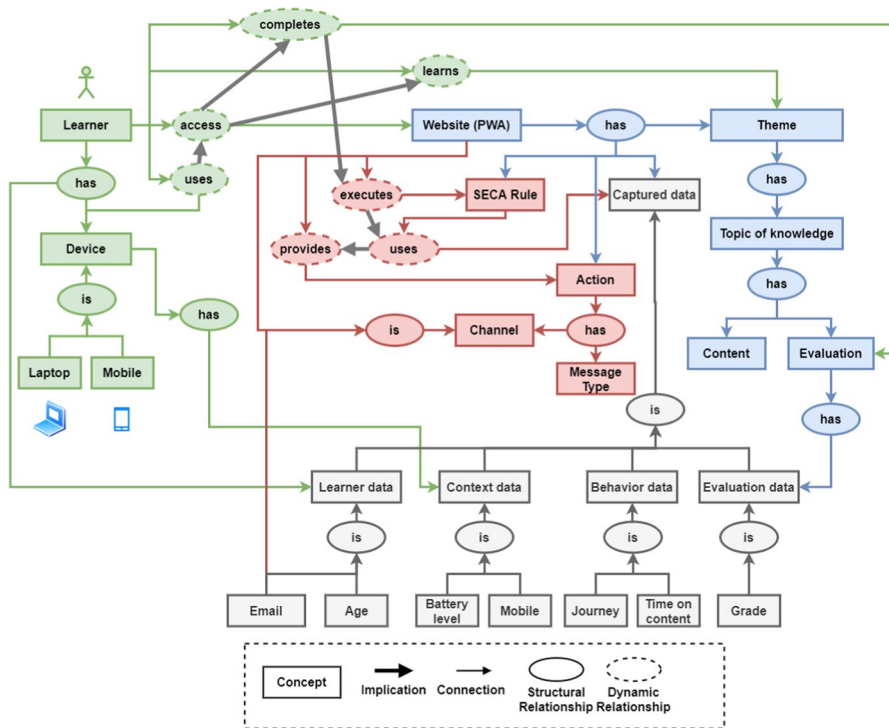


Fig. 2 Microlearning environment

eafit.edu.co/). The learner registers the profile in the application. Then the learner uses the microlearning environment, which has been previously prepared under a pedagogical model guided by questions to be solved through different sessions. The learner can follow the learning path suggested for each Topic of Knowledge through its content or directly to its evaluation. The learning system offers various formats that can be rated according to the learner satisfaction level.

In each session, the learner will be able to carry out an evaluation that facilitates the measurement of the cognitive causality relationship between the training action and the impact of the contents in the process. For validating the learning level, the learner can play luck into the evaluation or access the contents to study the concepts and subsequently perform the evaluation.

Course content structure

The proposed microlearning environment contains a set of Themes compound of Topics of Knowledge (ToK) which internally have Contents and Evaluations. A Theme is a subject that refers to the various units that make up the microlearning environment. A ToK represents each one of the units that make up a theme. Contents are a set of documents and videos that explain the ToK. Those are educational contents detached in small chunks that usually last no longer than a few minutes.

Each content belongs to one of the following categories: example, definition, motivation, or code. Evaluations are a set of questions for formative evaluations. That means the learner makes the evaluation and is informed or given feedback about the evaluation's context and result. This approach contributes to diagnosing weaknesses from content use and identifying the context that could influence the evaluation results.

The evaluation is based on multiple-choice questions. Each question Q is associated with a topic ToK and classified by a type T . Each T has a weight W that determines the difficulty degree within the evaluation: Conceptual ($W=0.64$), Example ($W=1.54$), and Code ($W=1.54$). Thus, for each ToK, a learner can take the evaluation up to three times. In each attempt, the questions are randomly taken from the evaluation question bank. Furthermore, each question is related to one or more contents ($Q_{(T)R} C_{1..i}$).

Besides, the context is activated with variables; then, the variables are processed to analyze their effect on the evaluation. Thus, feedback is provided to the learners to improve the results and the contexts in each session.

Data to be captured

To obtain representative information about learners' behavior and their context, we established to capture the following data from the microlearning environment:

- Learner data. It represents data characterizing the learner, *i.e.*, Age.

- Content data. It represents contents characteristics, *i.e.*, Topic of knowledge, Content type (video, document), Content category (example, definition, motivational, code), and Content rating.
- Context data. It represents context conditions when the learner interacts with content or evaluations, *i.e.*, Ambient Noise, Accelerometer, Device Type, Device Battery, Location, and Internet network speed.
- Learner behavior data. It represents the learner behavior performed during the interactions with contents and evaluations. It includes Time on content, Journey, Day of the week, Entry hour, and Focus time on content.
- Evaluation data. It represents data associated with the evaluation, *i.e.*, Evaluation grade.

Tool development

To establish the microlearning environment, we developed a microlearning Progressive Web Application (PWA) named Omnilearning. This application provides an environment that includes administering the themes, topics of knowledge, and contents, capturing learner data, evaluation management, and feedback services.

Figure 3 shows the application architecture. The main application architectural components are:

- *Client applications* They represent the devices and elements that the learners use to access or communicate with the Omnilearning application, such as mobile phones, desktops, and emails.
- *Front End* It is a front-end application developed in the Angular framework. This application shows the user interfaces to the learners and platform admins. Through this application, learners can access the themes, topics of knowledge, contents, and evaluations. Therefore, this application capture learner data.

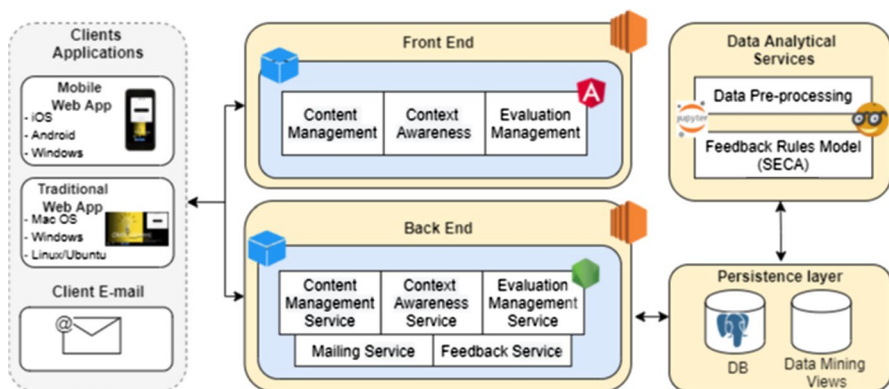


Fig. 3 Omnilearning application architecture

- *Back End* It is a backend application developed in node.js. This application provides some services to the front-end application, such as the management service, the evaluation service, the mailing service, and the feedback service.
- *Persistence layer* It captures, processes, and stores data. We used a PostgreSQL database to store the application data. Therefore, we created some data mining views.
- *Data analytical services* This component provides some functionalities to pre-process data and process the SECA rules (see “[Feedback Model](#)” section).

The application is accessible using a web browser, either from a mobile device or from a computer. The learner must begin the interaction through a registration process. First, primary data (such as name, age, and email) is requested and captured. In addition to this, the student fills out a cognitive style registration form, which is intended to obtain preliminary information about their preferences.

Omnilearning application supports the following features: (i) Login: allows the registration and login of users. (ii) Learning path: presents a list of contents about Software Design Patterns. (iii) Evaluation: through questionnaires, evaluates the concepts studied. (iv) Voting: allows ranking the contents. (v) Feedback: informs the learner about the evaluation results and context.

Experiment

We designed an experiment to capture learners’ data, which was used to define the feedback rules model. To develop this experiment, we performed the following main ASUM-DM activities: (i) Business and Data Understanding: participant selection and Omnilearning content population; (ii) Data Preparation; and Modeling. The Evaluation stage is presented in the “[Feedback Model](#)” section.

Participant selection

We contacted 62 students (mainly Systems Engineering students) from the Universidad EAFIT in Medellín, Colombia, interested in learning about Software Design Patterns. The students were asked to interact with the Omnilearning application freely, access the available contents, and make the evaluations. In the end, 43 students agreed to participate in this experiment. The 43 students interacted at least with one content, and 31 completed at least one evaluation. The experiment participants were informed that their participation was voluntary and could withdraw from the study at any time.

Omnilearning content population

We populated the Omnilearning application with content related to the Software Design Patterns theme. The application provided short learning experiences in multiple content formats (videos and PDF documents) and categories (example,

definition, motivation, and code). We defined four ToK, 37 contents, and four evaluations. Each evaluation is scored on 5 points and is composed of five questions organized as follows: three conceptual ($W=0.64$), one example ($W=1.54$), and one code ($W=1.54$). Table 1 presents an example of the course content structure.

Data preparation

Omnilearning was used for four weeks by 43 students, which interacted at least with one content; 31 of the students made at least one evaluation. After the data collection, we obtained the initial dataset of 17 variables (defined in “Data to be captured” section), the most appropriate for analyzing changes in a learner’s context concerning their evaluation outcome. Afterward, we made cleaning tasks by applying different estimators like mean, median, mode and calculate percentiles and quartiles. Thus, we guaranteed good properties such as unbiased, efficiency, convergence, and robustness for each selected variable. We defined that raw dataset under a hierarchical structure of three dimensions: information on the students’ behavior when they accessed the content, the information provided by the content, and the context information captured during the learning process. However, it was necessary to reduce the number of input variables to two dimensions to develop the predictive model. Thus, we ensured the performance of the classification models when evaluating the relationship between each input variable and the target variable. After that, we filtered to select only the learners who made both learning activities (access contents and make evaluations) at the microlearning. Thus, we obtained 551 content entries contents with corresponding context interactions. The entries were stored every 10 s to get a final dataset with 9619 records defined by 16 predictors and “*evaluation_grade*” as the dependent variable (values between 0 and 5). We transformed it into two classes: 0 (i.e., *evaluation_grade* < 3) representing a deficient result in the learner’s evaluation, and 1 (i.e., *evaluation_grade* ≥ 3) representing a right result in the learner’s evaluation. Thus, we got a balanced dataset with 232 entries related to class 0 and 319 entries related to class 1.

The reduction task was by applying Robust Principal Component Analysis (RPCA) Sapra (2010), which uses the eigenvalues, the eigenvectors, and the covariance matrix of the data to calculate the principal components of the representation

Table 1 Software design patterns microlearning—example

ToK: creational software design patterns

Evaluation			Contents	
Question	W	Type	Name	Type
What is a creational pattern?	0.64	Motivation	Creational patterns	Definition (Document)
		Conceptual	Creational patterns	Example (Document)
Can several instances of class X be created?	1.54	Code	Singleton	Code (Document)
Class Y is the concrete factory?	1.54	Example	Factory	Example (Video)

of all variables in the data set. Therefore, we could work well with outliers because RPCA is based on the matrix decomposition $M = L_0 + S_0$ where M corresponds to the dataset, L_0 to low-rank decomposition (low matrix of the data), and S_0 to the scattered matrix Wang et al., (2020.). In this research, we used the sparse matrix to identify the outliers for values far from 0. We complement it with the numerical measurement of the Z score to identify the outliers where $Z > 3$. It means that the students obtained satisfactory results in the evaluation; after all, they did not need to see the content because they had prior knowledge or good luck. On the other hand, the students who obtained unfavorable results could be because the contents were not attractive.

At this point, we want to find the highest characteristics that will be part of the training and evaluation datasets. Therefore, we used a genetic algorithm implemented based on the proposals of Hussein et al. (2001) and Amini and Hu (2021). This algorithm improved the prediction by evolutionarily identifying characteristics, that is, for each classification model (Random Forest, Support Vector Machine (SVM), Logistic Regression, Naive Bayes, Neuronal Network, Decision Tree, and Gradient Boosting Tree) to maximize a metric and detect the best predictors of that model. The validation metrics were F1, precision, and ROC, among others. This algorithm works iteratively in the following way: in each iteration, possible solutions are generated, in other words, a set of characteristics of the original data set; the algorithm iterates until it detects that the validation metric reaches a limit, that is, the best result or does not change, or until it reaches a limit of iterations or generations. Figure 4 shows how the genetic algorithm converged from a given model to find the maximization of the metric through the generations (iterations) of the algorithm step. For example, during the training activity, the Gradient Boosting Tree provided an accuracy of 85%, and the genetic algorithm results in six variables: age, subject of knowledge, type of content, ambient noise, day of the week, type of device. For Random Forest, the genetic algorithm results in nine

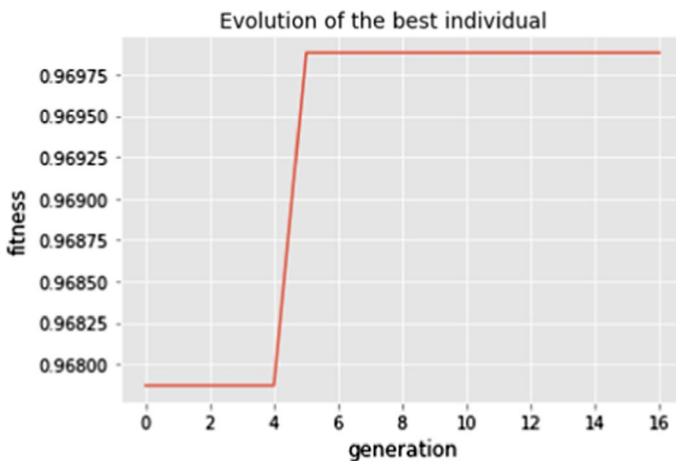


Fig. 4 Fitness evolution

variables: age, knowledge topic, content type, day of the week, time of entry, device type, device battery, travel, and ambient noise. With these characteristics, the models were retrained, increasing their precision; for example, the Gradient Boosting Tree increased its accuracy to 91%; similar behavior occurred in the other models. Therefore, the predictors are 0: age, 1: topic_of_knowledge, 2: content_type, 5: day_of_the_week, 6: input_time, 7: device_battery, 11: device_type, 12: travel, and 13: ambient_noise. Figure 5 shows the individuals that appeared more frequently in the best predictors used to test the models.

Modeling

Our goal of identifying whether a learner will get a right or deficient result in the evaluations made in its learning path. Therefore, we used the 551 pre-processed records for these models' training, which represent the inputs to contents. These records were divided into two sets, one representing the training dataset with 70% of the records (369 records), and the other representing the test dataset with 30% of the records (182 records). The validation of the models' performance is usually done only with the test dataset; however, as in our case, we have few records, we use the complete dataset to validate the models. The validation was done in two stages, the first using cross-validation with the training dataset. The second was to apply the ROC AUC metric to the test dataset. When using cross-validation, the training dataset is randomly separated into k subsets of approximately the same size, in our case $k = 10$ (when testing with different k -values and getting similar results, we decided to use the default k -value that cross-validation handles). $k - 1$ subsets are used to train the model, and one is used as a test. This process is repeated k times using a different test subset in each iteration. Finally, the result is the average of the results of each iteration.

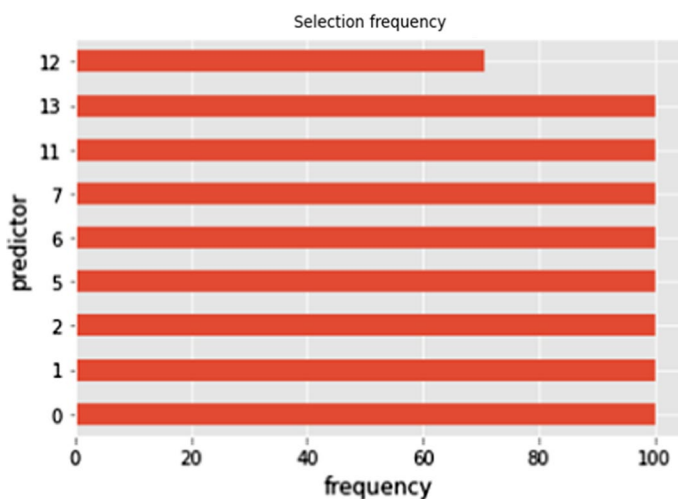


Fig. 5 Best predictors frequency

Table 2 Result of prediction models using different metrics (%)

	Random forest	Gradient boosting tree	Decision tree	SVM	Logistic regression	Naïve bayes	Neuronal network
F1	95.2	92.4	91.5	89.0	68.5	58.6	78.4
Recall	97.3	92.9	91.2	94.4	77.9	53.3	95.3
Precision	93.3	92.0	91.2	84.4	62.0	66.6	67.0
Jaccard	91.0	86.5	83.4	80.3	52.2	41.6	64.8

Table 3 Variables with their measurement ranges

Variable	Range	Capture instrument
Age	15 to 25–26 to 35–36 to 45–46 to 55	Registration form
Topic of knowledge	ToK 1–ToK 2–ToK 3–ToK 4	Learning system
Content type	Video–Document	Learning system
Ambient Noise	0 to 39 decibels	Device's microphone
Device Type	Laptop–Mobile	Device's OS
Device Battery	2% to 100%	Device's OS
Journey	Morning–Afternoon–Evening	Device's clock
Day of the week	Monday to Sunday	Device's calendar
Entry hour	0:00 to 2:00–6:00 to 0:00	Device's clock

The equation $\hat{\epsilon}_{cv} = \frac{1}{n} \sum_{i=1}^n \mathcal{L}(y_i, \hat{f}_{-k}(x_i))$ represents the cross-validation, where \hat{f}_{-k} corresponds to the models trained with the $k - 1$ subsets. $\hat{f}_{-k}(x_i)$ represents the prediction obtained by the given model x_i corresponding to the training data. In our case, x_i predicts one of the two classes (*evaluation_grade*: 0 or 1) from the context information, contents, and the interaction with contents. y_i represents the real value to be predicted; in our case, it is the real class. \mathcal{L} is the validation metric that has as input the predicted values and the real values to measure the models' performance. We used Accuracy, F1 score, Recall, Precision, and Jaccard metrics. The result of cross-validation is $\hat{\epsilon}_{cv}$, which corresponds to the average of the results delivered by the metrics used. At the same time, the genetic algorithm had iterated long enough to reach the best result of prediction by every model using every metric (see Table 2).

Consequently, the genetic algorithm outcomes nine variables: Age, Topic of knowledge, Content type, Ambient Noise, Device Type, Device Battery, Journey, Day of the week, and Entry hour. Table 3 presents these variables, their measurement ranges, and the capture instruments used. Variables such as the Topic of knowledge and Content type are part of the Omnilearning application, captured through our learning system. The Age is provided directly by the learner through the registration form. The Omnilearning application automatically captures variables such as Device Type and Entry hour without asking the learner's permission since they do not represent private data. These variables are captured through the

device's operating system and other applications. Finally, the Ambient Noise is captured through the learner's device's microphone. Learners are asked to provide access to their microphones when using the application; they decide to accept or reject this requirement. The Ambient Noise takes the value "No microphone" (when the learner does not provide access to the microphone) or a value between 0 to 39 decibels (when the learner grants access to the microphone).

Feedback model

We define a feedback model to support the formative assessment as an alternative form of evaluation in microlearning environments. It should help shape teaching and learning and be especially useful when available before or during learning. However, the delivering feedback system must know the context and scenarios that lead to a student's success or failure in the assessment processes. Furthermore, it represents a paradigm shift since traditional feedback is reactive, *i.e.*, if performance is low, the system will recommend actions to improve future attempts. In contrast, proactive feedback will seek to influence student behavior during the learning process to enhance their current assessment processes.

In microlearning, contents are presented in atomic units; therefore, the feedback should fulfill that same property. A microlearning system in a ubiquitous context can identify scenarios that increase the probability that a learner fails or passes an evaluation and inform the learner about the characteristics of this scenario. As mentioned earlier in the formative evaluation, feedback shapes the learning process by providing helpful information during the process.

The feedback model uses SECA rules, provides the Analytics Scenario to determine the context basis, and then personalizes SECA actions to provide the feedback regarding the learner's behavior.

SECA rule definition

The feedback rule definition in microlearning environments is the basis of our proposal. It is achieved by unifying ideas presented throughout the document regarding timely feedback in ubiquitous environments.

A SECA rule defines the required elements to provide feedback to learners during their learning process. A rule is composed of four elements. A Scenario *S* whereby different ubiquitous context variables reflect the learner's behavior during the learning process. An Event *E* indicating whether the current scenario could affect the evaluation result. A Condition *C* evaluates the event value and determines feedback actions. Furthermore, the feedback Action *A* (or set of actions) guides the learning process intervention using messages intended to influence the learners' behavior and improve their performance in the evaluation.

$$SECA = Scenario + Event + Condition + Actions$$

Scenario

It reflects the ubiquitous context's behavior during the learner's learning process. It is defined as the union of two sets of variables. LCV (Learner-controlled variables), which gathers variables provided or managed by the learner, and LPEV (Learning Process and Environmental variables), which gathers the learning process and environment learner's variables.

$$Scenario = \left(\bigcup_{i=1}^n LCV_i \right) \cup \left(\bigcup_{j=1}^m LPEV_j \right), LCV \neq \emptyset, LPEV \neq \emptyset$$

$$LCV = \{Age, DeviceType, Journey, DayOfTheWeek, EntryHour\}$$

$$LPEV = \{ToK, ContentType, AmbientNoise, DeviceBattery\}$$

At the ubiquitous learning process deployed through microlearning (see “[Micro-learning environment setup](#)” section), the teacher defines the pedagogical environment with variables as the Topic of knowledge and Content type. Moreover, variables as Ambient Noise and Device Battery are monitored to know the learner's environment. On the other hand, Age is provided by the learner, who chooses the day, time, and device to start or continue the learning process. Thus, a scenario requires a minimum of one LCV and one LPEV since these two types of variables allow identifying a context in a learning process. For example, the student is located at a quiet place and uses a laptop with a battery level below 50% to access the learning contents.

Event

It detects changes in a scenario indicating whether the current context behavior affects the evaluation results. In other words, the event identifies the probability that a learner fails ($Class = 0$) or passes ($Class = 1$) an evaluation.

$$Event \Rightarrow Class = 0 \vee Class = 1$$

Condition

It evaluates an event value which indicates a *Result* of *Fail* (i.e., $Event.Class = 0$) or *Pass* (i.e., $Event.Class = 1$). Besides, it proceeds to determine the required feedback actions. A learning system provides the required actions to allow learners to change their learning style or improve how the ubiquitous context may be successful in the evaluation.

$$if(Result == Fail) then$$

Action.message.type = Improvement

else Action.message.type = Motivation

Actions

A Feedback Action is a message focused on making the learner realize the impact of its context on the learning outcomes. The message can be distributed through different channels (*i.e.*, email and PWA for our proposal).

$Action \Rightarrow Message \wedge \{Channel(email) \vee Channel(PWA)\}$

At microlearning, we manage two types of messages. (i) Improvement influences learners to change their behavior in unfavorable contexts to improve their evaluation processes. (ii) Motivation rewards the learner for continuing or completing the learning process to achieve a much better score.

SECA analytics

Scenarios ground the SECA analytic as the feedback source. Hence, considering the following premises, we related each Scenario to a branch of every tree obtained from the Random Forest model (see “[Data preparation](#)” section).

- *Premise 1* The trees selected for the analysis have a root node with a representative sample, *i.e.*, like the original sample or not less than 50% of it.
- *Premise 2* A tree’s branch would be a scenario if at least one of its internal nodes contained the variable of “Topic of Knowledge.” Thus, any leaf node and its respective event will be associated with that variable to guaranteeing analytical consistency at the decision-making.
- *Premise 3* The selected branches must have internal nodes with the most representative sample of their level. This premise links with identifying content characteristics that might affect the evaluation and the learners’ context’s behavior.
- *Premise 4* From every selected tree, two types of branches are identified. One of them, whereby the leaf node linked the “fail” prediction event ($class=0$), and the other whereby the leaf node linked the “pass” prediction event ($class=1$).

The Random Forest Model’s evaluation is considered for preparing feedback based on context-aware variables (select as predictors). Therefore, we chose the decision trees with the most representative samples, which allowed us to identify patterns through the tree’s branches with the probability that a learner fails. Figures 6 and 7 show two of the most representative trees of the test set. We applied each premise to select both trees as an example among so many trees. They have an initial sample of 369 records. In their first levels, branches are focused on the Ambient Noise variable. The second level corresponds to Device Battery, etc., until they reach the leaf nodes with their respective predictions. Thus, we identified which

variables were more influential in the prediction process. For example, in the first tree (Fig. 6), the scenario analysis flow through the branch with the most significant sample, *i.e.*, Ambient Noise less than 14 decibels (sample=77). At the same time, Device Type and Topic of Knowledge variables will be essential for prediction based on Learning Process and Environmental Variables—LPEV mainly.

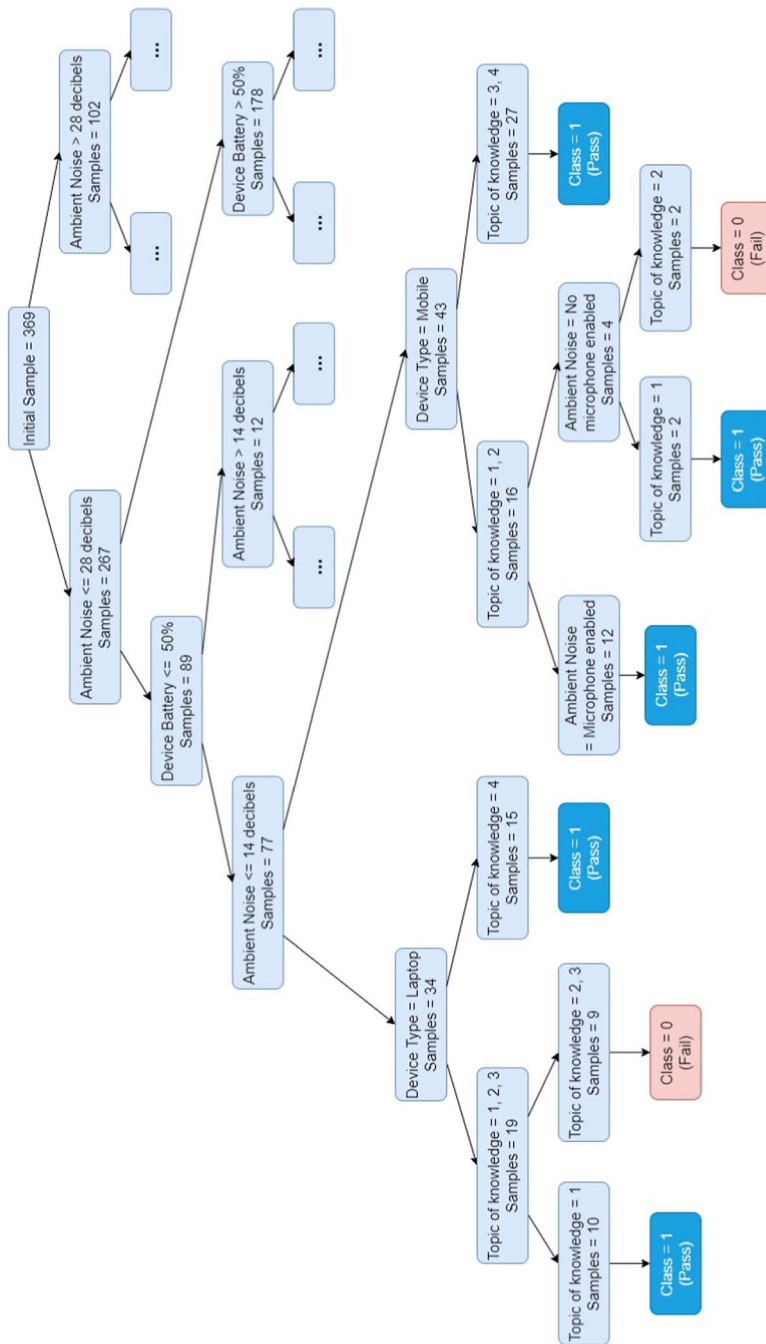
The second tree (see Fig. 7) considers guiding a complimentary scenario of the first tree. The Learner-Controlled Variables influence the Topic of Knowledge variable—LCV to determine the prediction. Hence, different leaf nodes were related to a “Class” variable as part of the prediction result (*i.e.*, an Event). In other words, it predicts that given a Scenario, the student has a high probability of “fail” or “pass” its evaluation. Thus, we proved that trees help provide feedback from the SECA Rules with the previous example.

Based on premise 4, we proceeded to create two new test datasets. One contains data from branches whose leaf node is linked to “fail” events and base the “Improvement” feedback actions. The other includes data from branches whose leaf node is connected to “pass” events and will establish the “Motivation” feedback actions. Table 4 presents a resume of some data. We primarily worked with the ‘fails’ dataset because this occupies the most significant interest in our proposal to achieve the learning objectives. In the table, Ambient Noise is measured in dB (decibels). “No Mic.” represents that the learner did not authorize the use of the microphone. Class 0 is associated with the “fail” event, and Class 1 is associated with the “pass” event. Message Type can be M (Motivation) or I (improvement). Channel can be E (Email), PWA, or both.

Personalizing SECA actions

The SECA actions’ personalization is achieved by grouping individuals with similar context behavior in each new test dataset. We use the clustering technique (the K-means method) and apply the elbow technique to determine the best number of most representative clusters. Besides, we applied the silhouette measurement, which told us if the collections were well represented. Thus, we worked with well-defined clusters (values equal to 1 or close to 1). Then, we did an additional validation using hierarchical clustering, shown in Fig. 8. The X-axis refers to the quantity of learners’ access to content by every cluster. The Y-axis refers to the Euclidean distance that determines the cluster selection. Thus, we were able to group the most similar clusters in a new cluster. Then, we identified four predominant clusters that guided the feedback actions.

The analysis above allowed us to know the influence of LPEVs and LCVs on the sample and then find a way to bring personalized feedback. Table 5 presents some examples. First, however, we needed to standardize feedback actions. To formalize this, we applied dispersion and central tendency measures such as mean, median, minimum, maximum, and standard deviation to each cluster. These measures allow us to understand the context’s behavior for each group of learners when we compare the values obtained by each cluster concerning the ranges defined initially for each variable. Specifically, we take reference values,



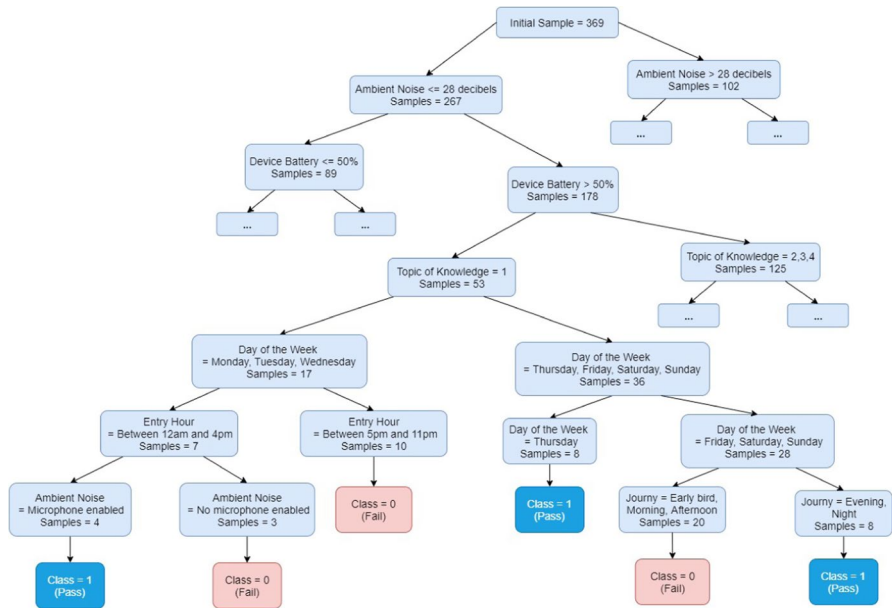


Fig. 7 Microlearning Scenarios Tree-2

Table 4 SECA rules example

SECA	Scenario					Event	Action	
Id	Device battery	Device type	Ambient noise	ToK	...	Class	Message type	Channel
1	≤50%	Laptop	≤ 14 dB	1		1	M	E
2		Laptop	≤ 14 dB	2, 3		0	I	E/PWA
3		Laptop	≤ 14 dB	4		1	M	E
4		Mobile	No Mic	1		1	M	PWA
5		Mobile	No Mic	2		0	I	E
6		Mobile	≤ 14 dB	3, 4		1	M	PWA
7		Mobile	No Mic	1, 2		1	M	PWA
...								

the median when the values are continuous or discrete, and the mode when the variables are categorical. Thus, we create three types of personalized feedbacks. (i) Feedback LCV will be sent to a group of individuals more influenced by Learning Control Variables. (ii) Feedback LPEV, which will be sent to a group of individuals more influenced by the Learning Process and Environmental Variables. (iii) Ubiquitous Feedback will be sent to a group of individuals that have influenced both LCVs + LPEVs.

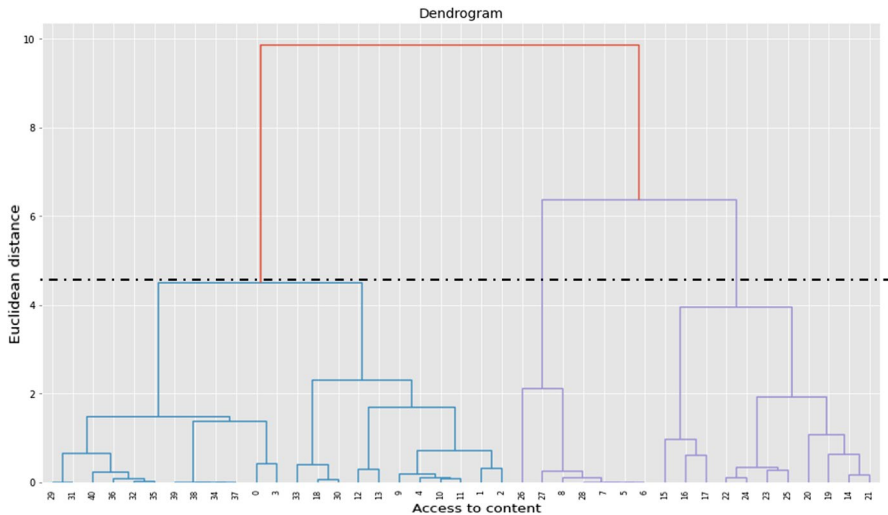


Fig. 8 Clusterization of the fail-test data

Conclusion

In a ubiquitous learning process, the learner can access different content pedagogically designed to achieve the learning objectives at any time, from any place or device. The evaluation is an essential component used for the student to self-evaluate the acquired ability each time he reaches a level of knowledge. However, this type of process almost always lacks the feedback process.

This article presented a feedback model supported by SECA rules. Thus, a ubiquitous microlearning context was deployed through a PWA (Progressive Web Application). It provided a learning process where every learner could access different Topics of Knowledge, which guided their learning path through contents and evaluations. That allowed us to do an analytic process that provided the knowledge to design a feedback model.

The microlearning environment was designed with formative evaluations to make the evaluation and give feedback about the results. Thus, this approach contributes to diagnosing weaknesses in content use and identifying the context that could influence the evaluation results. Besides, the environment allowed us to capture representative information about learners' behavior and their context.

We used decision trees from a Random Forest model as the basis of our SECA rules proposal. This model was trained with the 16 variables and re-trained with the variables delivered from classification models. The random forest model achieved the highest metrics (94.4% accuracy, a 97.3% Recall, a 93.3% Precision, an F1 score of 95.2%, and Jaccard of 91%). The relevance of the tree's different branches and how they reach specific nodes is that a scenario can be identified as deterministic for the learning outcomes. Therefore, an event is launched with a prediction of possible success or failure based on a scenario. After this, a

Table 5 Feedback actions example

SECA Id	Result	Learner	Message type	Message	Channel
2	Fail	1	I	We recommend you access this content through your mobile...	E
	Fail	2	I	You can better focus on this content if you access it through your mobile...	E

	Fail	8	I	Keep focused. More effort, less time. Good deal, no?	PWA
5	Fail	9	I	Avoid distractions. Now is the time...	PWA
	Fail	1	I	Keep low your ambient noise. It can influence your evaluation score...	E
	Fail	2	I	Leave distractions behind by grabbing your mobile and continuing over there...	E

condition validates the event and allows the execution of the necessary feedback actions that lead to improved scenarios and, thus, students' success.

Once the feedback was sent to the students, basic measurements were made regarding the use of the contents. Then, students were motivated to access content in "adequate" conditions. It motivated us to conduct a new phase for the experiment. The impact of the actions will be evaluated in several cycles (approximately three) to achieve refinement of the models proposed in this article.

We want to achieve high-level automation of adaptation or personalization from the feedback rules model in future work. It will require improving the learning path designed in the microlearning and the time and channels to send feedback messages to learners.

Acknowledgements We would like to thank Ministerio de Ciencia, Tecnología e Innovación de Colombia, Gobernación de Antioquia, and Vicerrectoría de Descubrimiento y Creación from Universidad EAFIT. We thank Sebastián Gómez (Universidad EAFIT) for his assistance in constructing the analytics models. We also thank students who participated in the Experiment.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Aldosemani, T. I. (2019). Microlearning for macro-outcomes: Students' perceptions of telegram as a micro-learning tool. *Digital turn in schools—research, policy, Practice* (pp. 189–201). Springer.
- Amini, F., & Hu, G. (2021). A two-layer feature selection method using genetic algorithm and elastic net. *Expert Systems with Applications*, 166, 114072. <https://doi.org/10.1016/j.eswa.2020.114072>
- Angée, S., Lozano-Angel, S. I., Montoya-Munera, E. N., Ospina-Arango, J. D., & Tabares-Betancur, M. S. (2018). Towards an improved asum-dm process methodology for cross-disciplinary multi-organization big data & analytics projects. *Knowledge management in organizations* (pp. 613–624). Springer.
- Chen, N. S., Wei, C. W., & Huang, Y. C. (2013). The integration of print and digital content for providing learners with constructive feedback using smartphones. *British Journal of Educational Technology*, 44(5), 837–845. <https://doi.org/10.1111/j.1467-8535.2012.01371.x>
- Cheng, Z., Richardson, J. C., & Newby, T. J. (2020). Using digital badges as goal-setting facilitators: A multiple case study. *Journal of Computing in Higher Education*, 32(2), 406–428.
- Chu, H. C., Chen, J. M., Hwang, G. J., & Chen, T. W. (2019). Effects of formative assessment in an augmented reality approach to conducting ubiquitous learning activities for architecture courses. *Universal Access in the Information Society*, 18(2), 221–230.
- Correa, D., Isaza, F. A., Mazo, R., & Giraldo, G. L. (2018). Cme—a web application framework learning technique based on concerns, micro-learning and examples. *International conference on web engineering* (pp. 17–32). Springer.

- Coulby, C., Hennessey, S., Davies, N., & Fuller, R. (2011). The use of mobile technology for work-based assessment: The student experience. *British Journal of Educational Technology*, 42(2), 251–265. <https://doi.org/10.1111/j.1467-8535.2009.01022.x>
- Edalati, M., Imran, A. S., Kastrati, Z., Daudpota, S. M. (2021). The potential of machine learning algorithms for sentiment classification of students' feedback on mooc. In *Intelligent systems and applications: Proceedings of the 2021 intelligent systems conference (IntelliSys)* (Vol. 1, No. 296, pp 11–22) in Lecture notes in networks and systems. doi:<https://doi.org/10.1007/978-3-030-82199-92>
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75–105.
- Hug, T. (2006). *Microlearning: A new pedagogical challenge (Technical report)*. University of Innsbruck.
- Hung, P. H., Lin, Y. F., & Hwang, G. J. (2010). Formative assessment design for pda integrated ecology observation. *Journal of Educational Technology & Society*, 13(3), 33–42.
- Hussein, F., Kharm, N., Ward, R. (2001). Genetic algorithms for feature selection and weighting, a review and study. In *Proceedings of sixth international conference on document analysis and recognition* (pp 1240–1244). doi:<https://doi.org/10.1109/ICDAR.2001.953980>
- Kochmar, E., Vu, D. D., Belfer, R., Gupta, V., Serban, I. V., & Pineau, J. (2020). Auto-mated personalized feedback improves learning gains in an intelligent tutor-ing system. In I. I. Bittencourt, M. Cukurova, K. Muldner, R. Luckin, & E. Millan (Eds.), *Artificial intelligence in education* (pp. 140–146). Springer.
- Lee, Y. M., Jahnke, I., & Austin, L. (2021). Mobile microlearning design and effects on learning efficacy and learner experience. *Educational Technology Research and Development*, 69(2), 885–915.
- Li, J., Wong, S. C., Yang, X., & Bell, A. (2020). Using feedback to promote student participation in online learning programs: Evidence from a quasi-experimental study. *Educational Technology Research and Development*, 68(1), 485–510.
- Lim, L. A., Gentili, S., Pardo, A., Kovanovic, V., Whitelock-Wainwright, A., Gasevic, D., & Dawson, S. (2019). What changes, and for whom? a study of the impact of learning analytics-based process feedback in a large course. *Learning and Instruction*, 72, 101202.
- Lin, J., Sun, G., Cui, T., Shen, J., Xu, D., Beydoun, G., Yu, P., Pritchard, D., Li, L., Chen, S. (2019). From ideal to reality: Segmentation, annotation, and recommendation, the vital trajectory of intelligent micro learning (pp. 1–21). World Wide Web.
- Marinagi, C., Skourlas, C., & Belsis, P. (2013). Employing ubiquitous computing devices and technologies in the higher education classroom of the future. *Procedia-Social and Behavioral Sciences*, 73, 487–494.
- Mohammed, G. S., Wakil, K., & Nawroly, S. S. (2018). The effectiveness of microlearning to improve students' learning ability. *International Journal of Educational Research Review*, 3(3), 32–38.
- Nasim, Z., Rajput, Q., Haider, S. (2017). *Sentiment analysis of student feedback using machine learning and lexicon based approaches* (pp. 1–6). <https://doi.org/10.1109/ICRIIS.2017.8002475>
- Redondo, R. P. D., Rodríguez, M. C., Escobar, J. J. L., & Vilas, A. F. (2021). Integrating micro-learning content in traditional e-learning platforms. *Multimedia Tools and Applications*, 80(2), 3121–3151.
- Sakamura, K., Koshizuka, N. (2005). Ubiquitous computing technologies for ubiquitous learning. In *IEEE international workshop on wireless and mobile technologies in education (WMTE'05)* (pp 11–20). IEEE.
- Sapra, S. K. (2010). Robust vs. classical principal component analysis in the presence of outliers. *Applied Economics Letters*, 17(6), 519–523. <https://doi.org/10.1080/13504850802046989>
- Sedrakyan, G., Malmberg, J., Verbert, K., Järvelä, S., & Kirschner, P. A. (2020). Linking learning behavior analytics and learning science concepts: Designing a learning analytics dashboard for feedback to support learning regulation. *Computers in Human Behavior*, 107, 105512.
- Shang, H. F. (2017). An exploration of asynchronous and synchronous feedback modes in EFL writing. *Journal of Computing in Higher Education*, 29(3), 496–513.
- Tabares, M. S., Vallejo, P., Montoya, A., Sanchez, J., & Correa, D. (2021). Seca: A feed-back rules model in a ubiquitous microlearning context. *International Conference on Data Science, E-Learning and Information Systems*, 2021, 136–142.
- Vallejo-Correa, P., Monsalve-Pulido, J., & Tabares-Betancur, M. (2021). A systematic mapping review of context-aware analysis and its approach to mobile learning and ubiquitous learning processes. *Computer Science Review*, 39, 100335.
- Wang, H., & Lehman, J. D. (2021). Using achievement goal-based personalized motivational feedback to enhance online learning. *Educational Technology Research and Development*, 69(2), 553–581.
- Wang, Q., Gao, Q., Sun, G., & Ding, C. (2020). Double robust principal component analysis. *Neurocomputing*, 391, 119–128.

- Wu, P. H., Hwang, G. J., & Tsai, W. H. (2013). An expert system-based context-aware ubiquitous learning approach for conducting science learning activities. *Journal of Educational Technology & Society*, 16(4), 217–230.
- Yu, F. Y., & Wu, W. S. (2020). Effects of student-generated feedback corresponding to answers to online student-generated questions on learning: What, why, and how? *Computers Education*, 145, 103723. <https://doi.org/10.1016/j.compedu.2019.103723>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Marta S. Tabares¹  · Paola Vallejo¹ · Alex Montoya¹ · Daniel Correa¹

Paola Vallejo
pvallej3@eafit.edu.co

Alex Montoya
amonto69@eafit.edu.co

Daniel Correa
dcorreab@eafit.edu.co

¹ Universidad EAFIT, Medellín, Colombia