

A time-driven FCA-based approach for identifying students' dropout in MOOCs

Carlo Blundo  | Giuseppe Fenza  | Graziano Fuccio  |
Vincenzo Loia  | Francesco Orciuoli 

Department of Business Sciences-
Management & Innovation Systems,
University of Salerno, Fisciano, Italy

Correspondence

Vincenzo Loia, Department of Business
Sciences-Management & Innovation
Systems, University of Salerno, Via
Giovanni Paolo II-132, 84084 Fisciano,
SA, Italy.
Email: loia@unisa.it

Funding information

CORISA (Centro di Ricerca Sistemi ad
Agenti)

Abstract

In online learning, the dropout phenomenon is a relevant issue to address with practical solutions. Several data sets stimulate original, and resolute data analysis approaches, demonstrating the importance of the dropout phenomenon. This study proposes a novel approach to predicting massive online open course (MOOC) students at risk of dropout stressing the need to consider the temporal dimension in the data log. The proposal aims to build a data-driven decision support system able to identify students at risk of dropout based on the conceptualization of such students' behavior and its evolution along the time dimension. The primary theoretical model behind the proposed method is the formal concept analysis, and its temporal extension (i.e., temporal concept analysis) for analyzing time-stamped data and carrying out a timed lattice. The main result of the paper is a method to extract behavioral patterns of MOOC students at risk of dropout. Such patterns are defined as Time-based Behavior Rules extracted from the aforementioned timed lattice obtained through the preprocessing of MOOC platform log files. The resulting rule set can be easily integrated for implementing educational DSS, as shown in the last part of the paper. The conducted experiments reveal promising results in terms of F-score and students' monitoring time.

KEYWORDS

association rule mining, dropout analysis, educational data mining, formal concept analysis, massive online open course, temporal concept analysis

1 | INTRODUCTION AND MOTIVATION

In recent years, massive online open courses (MOOCs) have spread successfully and rapidly, offering the possibility to enroll and attend high-quality courses without space and time constraints.

One of the most critical problems of online classes is the high dropout rate (students that have abandoned a course).¹ Indeed, it is challenging to have continuous growth in class size while maintaining high course quality due to several limitations in communication and information sharing caused by the way students connect to lectures.

In Reference [2], authors discuss the persistence of learners related to what makes students constant in attending the course. They state that learners' persistence is a multifactors indicator of emotions, motivation, cognition, and behavior. Their results shed light on three crucial aspects that impact on the student's engagement and persistence. The first aspect concerns the self-efficacy that influences the engagement; thus, the authors suggest simplifying tools provided by the platform encouraging the students' self-efficacy to keep them active and involved. The second factor regards the teacher's presence as an essential variable influencing the student's engagement and persistence. So, the teacher should create situations where students can interact with each other and with the teacher, for example, with a forum. The third aspect concerns the perceived usefulness during the use of the MOOC platform that impacts the student's engagement. Students appear more engaged if they perceive the online learning support system as helpful and rewarding. Consequently, they increase their involvement in participating in various courses and not to abandon them. However, learners often complain that MOOCs fail in providing enough practical experience.³ Furthermore, some students can fail to self-organize their learning path and complete learning activities.⁴ Thus, despite their popularity, the problem of low completion rates in MOOC courses is persistent, and the causes are multiple. The relevance of the problem is also testified by numerous contests based on specific data sets (e.g., KDD Cup*).

A considerable number of studies facing the dropout problem have been published. In literature, many studies use support vector machine (SVM), logistic regression (LR), and decision trees (DT) to predict dropout students. For example, in Reference [5], the authors start with the preprocessing of the log file, employ principal component analysis (PCA), and then train an SVM model. In Reference [6], the authors quantify the student's activities by analyzing log file and click-stream. After that, using weeks as temporal slices, they define two-timing phases of their predictions: *Lead* to represent how many weeks in advance they can attempt to predict the dropout and *Lag* to denote how many weeks of historical data they need to make a prediction. The model used is the *Randomized Logistic Regression* which provides 70% accuracy with a lag of 2–3 weeks.

The research work proposed in this paper aims to build a data-driven decision support system (DSS) to identify students at risk of dropout. It allows the educational staff to provide all suitable actions to mitigate such risk (e.g., help teaching, motivational feedback, and learning path personalization).

The approach consists of using historical data (divided into time slots) to extract a set of patterns under the form of rules (a rule shows the shape $B' \Rightarrow B''$). Such rules characterize the behavior (on a MOOC platform) of students who made dropout and, in particular, how the behavior of such students changes along the time dimension. More in detail, if the rule is $B' \Rightarrow B''$, then the behavior change is observed from time slot i to the next time slot $i + 1$, where B' represents the conceptualization of the behavior in time slot i and B'' represents the conceptualization of the behavior in time slot $i + 1$. After constructing the aforementioned set of rules, when a new course starts, some new data concerning the MOOC-based learning activities of students arrive. If such data satisfy one or more of the extracted patterns (rules), the system will alert the teacher that a specific student is assuming a dropout-conducting behavior. Hence, the teacher can react to this situation to face the risk of dropout. Some experiments have been conducted to build a model based on the extracted rules from students' behavior data. The built model was validated on different course periods of 5 days each, attempting to anticipate the dropout-conducting behavior. The best result obtained, in terms of identifying dropout-conducting behavior, exhibits a precision of 78.97% in the early 10 days of the course schedule.

In a nutshell, the main research contributions of this study are:

- Definition of a temporal data mining approach to extract *time-enriched* association rules describing interesting behavioral patterns in a meaningful and understandable way.
- Capability to apply the proposed approach for implementing educational DSS to promptly recognize students who are likely following a dropout pattern to allow the educational staff to select a suitable course of actions (e.g., replanning the course, giving custom feedback, and so on).
- Evaluation of the the proposed approach performance by experimenting with the data set included in *KDD Cup 2015*.⁷

The remaining part of the paper is organized as follows: Section 2 reports related works dealing with MOOC dropout prediction approaches and provides a comparison between such approaches and the proposed one in this study; Section 3 describes the theoretical background behind this study; Section 4 introduces additional definitions dealing with time-based rules representing behavioral patterns of students at dropout risk. Section 5 describes the proposed approach in this study work by providing details on the steps included in the main workflow. Furthermore, Section 6 describes the experimentation activities and the obtained results. Lastly, Section 7 provides insights on the use of the proposed approach in the context of a whole Educational DSS, and Section 8 provides final remarks and anticipates possible future works.

2 | RELATED WORKS

The dropout phenomenon has been addressed using different computational approaches, mainly focused on predicting students who are going to leave an online course before its normal conclusion. This section proposes the main ideas underlying the aforementioned approaches to draw differences with the method proposed in the present paper.

The specialized literature mostly proposes approaches based on binary classification of dropouts using SVM or LR that seem to be widely used due to their ease of use and easy to understand. For instance, the authors of References [8] introduce a predictor, based on students' daily activities, able to identify students at risk of dropout. In particular, the SVM model

is applied to the *KDD Cup 2015*⁷ data set getting an area under the curve (AUC) of 89%. Moreover, in the work⁹ the authors use a data set (whose data are gathered from a real MOOC platform) to build the passing and failing model by using SVM, random forest, ada boost, and LR to evaluate classification performance upon the built models and obtaining 96% of accuracy taking into account the 25% of the students' learning paths. A more sophisticated approach¹⁰ adopts a workflow including: (i) a feature generation phase, based on counting the different event types in the clickstream data, and (ii) a feature selection phase realized by means of three different methods: Mutual information, random forest, and recursive feature elimination. Each method assigns individual scores to all features and uses such values to calculate an average score for each feature. A descendent feature rank is obtained. The feature selection phase is aimed to carry out a feature search step, basing on a LR model. The authors of Reference [11] define a composite model that uses machine learning algorithms, applied on web-logs, to predict the students' dropout. They use a set of autoencoders to extract the most relevant features. In particular, the first autoencoder takes as input a student's activity data and produces a partial result becoming the input of the second autoencoder, and so on, along the computational chain. This structure is called *Stacked Sparse Autoencoder* (SSAE). In this way, a more abstract result is provided as the processing advances along the chain. In the end, the *Softmax Regression* method is applied to execute the classification task that achieves an accuracy of 89%. In Reference [12] the authors propose a context-aware model, namely context-aware feature interactive network, in which they use features related to both learning activities and users' profiles to execute a classification task. After the evaluation, the resulting model obtained a final average accuracy of 90. Moreover, some studies propose the use of decision trees. In particular, the authors of Reference [13] propose a two-modules framework. The first module is based on a feature extraction step where features are selected from students' behavior records. The second module is represented by a decision layer that implements the feature selection step using a *Decision Tree*. The stop criterion for the decision tree is the *Information Gain Ratio*, an approach coming from information theory. Once the tree is constructed, it is mapped on the *Extreme Learning Model* (ELM). In particular, each node in the decision tree becomes an artificial neuron in ELM. Nodes are mapped on neurons in the hidden layer of the artificial network and classes of the decision tree become neurons in the output layer of the aforementioned network. The two-models framework is called *DT-ELM* and is able to get 89.28% of accuracy. Furthermore, the authors of Reference [14] employ a feature engineering process (starting from an available data set) able to generate more than 30 features representing event counts, such as course item coverage, daily, weekly, and hourly students' activities. Then, a *Gradient Boost Decision Tree* algorithm is applied on the above features to achieve the 87% of AUC. A different approach is proposed by the authors of Reference [15]. In particular, they define an evaluation algorithm to extract more complex features able to better characterize students' behavior along the days of an online course and, then, they use such behavior representation, showing it as a radar graph helping the student to analyze her/his learning progress and to understand how to improve it. The radar graphs can also be used to predict possible dropout behaviors.

Differently, the approach proposed in this paper does not target the best accuracy in terms of dropout recognition. It is aimed to promptly detect students who are likely following dropout-conducting patterns through the elicitation (from log files) and use (against new incoming data) of time-based behavior rules. Such rules represent behaviors through concepts learned from data and behavior changes through the analysis of the time slots in which these concepts (behaviors) emerge. Therefore, the main idea of the proposed approach is to detect

common behavior changes which are frequently recognized for students who make dropout. Among the other advantages, it is important to underline that the result of the proposed approach provides rules which are readable and executable both for humans and machines and they can be used by didactic staff also to understand how to react to the risk of dropout for specific students. The aforementioned rules can be used for enabling a wide range of DSS architectures: from the simple analytic dashboard for the didactic staff to the fully automatized adaptive learning system. Lastly, the proposed approach achieves an F-score of around 85% (for a given confidence threshold), using 10-days data of students' interactions, when detecting students' dropout-conducting behaviors. Although such result is not the best one among comparable approaches, it is achieved with less input data than the method described in Reference [12] that seems to be the one providing the highest F-score (an average F-score of around 91.5%).

3 | THEORETICAL BACKGROUND

The main formal model behind the proposed methodology is the formal concept analysis (FCA).¹⁶ It is a theoretical framework aimed at grouping together data items that share some features. Simultaneously, it hierarchically organizes these groups of items and corresponding attributes, namely *Formal Concepts*. This data analysis theory supplies a taxonomy of concepts suitable for knowledge processing and extraction. In literature, FCA finds many applications to information retrieval, taxonomy (or ontology) learning, and so on. Furthermore, there exist some theoretical extensions of the baseline framework, like fuzzy FCA¹⁷ or temporal concept analysis (TCA)^{18,19} suitable for managing the uncertainty and the evolution of concepts along with the time dimension. The next sections introduce basic notions of FCA (in Section 3.1) and its temporal extension TCA applied in this study for extracting *Timed Lattice* (in Section 3.2), and finally Association Rules extracted browsing Lattice carried out by FCA (in Section 4).

3.1 | Formal concept analysis

FCA considers the notion of a *Formal Context* specifying objects and corresponding attributes; thus, a formal context may be viewed as a binary relation between the object set and the attribute set. Formally, a Formal Context is defined as a triple $K = (G, M, I)$, where G is a set of objects, M is a set of attributes, and I is a binary relation, s.t. $I \subseteq G \times M$. The pair $(g, m) \in I$ is read "object g has attribute m ." A Formal Context is often represented as a "cross table": the rows represent the formal objects, and the columns are formal attributes; the relations between them are represented by the crosses.

In this study, we consider binary relations between students' enrollments (i.e., objects) and their features such as completed homework, time spent on the platform, and so forth (i.e., attributes).

Taking into account *Formal Context*,¹⁶ FCA algorithm identifies *Formal Concepts* (see Definition 3.1) and *Subconcepts* (see Definition 3.2). The notion of *Formal Concepts* relies on a derivation operator $(\cdot)'$ that, intuitively, given a nonempty set of objects retrieves corresponding shared set of attributes (dually, given a nonempty set of attributes, retrieves the corresponding set of objects sharing them). For a set of objects $A \subseteq G$, the set of common attributes can be defined as follows:

$$A' = \{m \in M \mid \forall g \in A : (g, m) \in I\},$$

while, for a set of attributes $B \subseteq M$, the set of objects sharing them can be defined as follows:

$$B' = \{g \in G \mid \forall m \in B : (g, m) \in I\}$$

Definition 3.1 (Formal Concept C). Given a context $K = (G, M, I)$, a formal concept is identified with a pair (A, B) , where $A \subseteq G$ and $B \subseteq M$, such that $A' = B$ and $B' = A$. A is called the extent and B is called the intent of the concept (A, B) .

Definition 3.2 (Subconcept \leq). Given a context $K = (G, M, I)$ and two concepts $c_1 = (A_1, B_1)$ and $c_2 = (A_2, B_2)$, we say that c_1 is a subconcept of c_2 if and only if $A_1 \subseteq A_2$. Subconcept relation is denoted by $(A_1, B_1) \leq (A_2, B_2)$.

Equivalently, we can define c_2 a *superconcept* of c_1 if and only if $B_2 \subseteq B_1$. The subsumption relation is transitive: if a concept c_1 is a subconcept of a concept c_2 and c_2 is a subconcept of a concept c_3 , then c_1 is a subconcept of c_3 , as well. Attributes belonging to a *Formal Concepts* are inherited by all its subconcepts. This corresponds to the notion of “inheritance.” The resulting hierarchy of concepts is the *Lattice* $L = (C, \leq)$ (see Definition 3.3) and it supports multiple inheritance.

Definition 3.3 (Lattice L). Given a formal context $K = (G, M, I)$, the set of all concepts C with the partial order \leq is the lattice denoted as $L = (C, \leq)$.

3.2 | Temporal concept analysis

TCA is the theory of temporal phenomena described with FCA. TCA was introduced in Reference [20] and [21], but only recently further development and application are available in the literature as in References [18] and [19]. TCA merges models of FCA with conceptual time systems.²² It is possible in two ways that are: including time slot as many-valued attribute²³ of the formal context; or indexing the timestamped objects by adding a time variable to the object of the formal context. In this study, we opt for this solution, each object consists of student identifier s_{id} , with corresponding course identifier c_{id} , at the time slot t_i .

Formally, let n be the number of observations made on a particular object over time, all the time slot ts_i , with $i \in \{1, \dots, n\}$, form a partial order:

- (g, t_i) represents object $g \in G$ at time t_i , where t is the time variable and i is an integer;
- t_i precedes t_j if $i < j$ for any two integers i and j ;
- All the time slots t_i , where $i \in \{1, \dots, n\}$ is an integer and n is the number of observations made on a particular object over time, form a partial order.

Additionally, let $C_i \in C$ be a concept belonging to the lattice L and $\circ(C_i)$ a set containing all objects (g, t_i) belonging to the extensional part of C_i , then it is possible to define:

$$\bar{\circ}(C_i) = \{g \in G \mid (g, t_i) \in \circ(C_i)\} \quad (1)$$

Starting from the early introduction of TCA in Reference [21], let us consider the formal context, where the objects are chronologically distributed along with the timeline, the resulting lattice is the *Timed Lattice* (see Definition 3.5). The time-stamped objects belonging to the *Timed Lattice* induce temporal relationships among the concepts, that are *Temporal Edges*

including temporal relationships, that are named *Temporal Edges* (see Definition 3.4). Indeed, *Temporal Edges* are derived from the temporal adjacencies of the *Formal Context* objects belonging to the concepts of *Timed Lattice*.

Definition 3.4 (Temporal edge). Given a lattice $L = (C, \leq)$, a Temporal Edge between two concepts $C_i, C_j \in C$ exists if there exist an object $g \in G$ such that $(g, t_s) \in \circ(C_i)$ and $(g, t_k) \in \circ(C_j)$ and the time t_s precedes the time t_k .

Definition 3.5 (Timed lattice L^t). A timed lattice L^t is a pair (L, E^t) , where $L = (C, \leq)$ is the lattice and E^t is the set of temporal edges which are 2-element subsets of C .

Informally, a *Temporal Edge* traces the evolution of the same object g along the timeline and, in particular, from time slot t_s to time slot t_k .

4 | TIME-BASED RULES: A STEP FORWARD

The idea is to build a *Timed Lattice* from historical data, that is, the activity log file of students who made dropout. Such lattice conceptualizes the behaviors of these students. In particular, formal concepts, containing objects with time information, represents behaviors in specific discrete time slots. Therefore, a time-based exploration of the *Timed Lattice* will lead to the elicitation of changes in the behaviors of students who made dropout. The goal is to derive time-based rules representing such behavior changes. Time-based rules are similar to association rules in which antecedent and consequent are related through a temporal relation. In our case, a plausible temporal relation is that the behavior expressed in the antecedent part of the rule must precede (along with the timeline) the behavior represented in the consequent part of it. In scientific literature, association rules attracted much attention in the context of data mining and machine learning. They are widely used in market basket analysis for extracting co-occurrence implications among itemsets that are, for instance, frequently purchased together.

Various metrics are used for measuring the strength of association between co-occurring items, like support, confidence, and so on.

In literature, there are several algorithms for mining association rules, a priori, FP-growth, and so on. Despite these techniques, FCA can be applied to extract frequent itemsets through the *Formal Concepts* and, consequently, it is possible to carry out *Association Rules* by exploiting hierarchical relationships among the concepts. Association rules mining on concept lattice reduces the search space, avoid blocking unexpected discoveries so that knowledge discovery can be improved effectively.²⁴

Borrowing from the idea of association rules derived from lattices, we have defined the *Time-based Association Rules* as rules able to connect two sets of attributes because they are included in concepts appearing in two consecutive time slots.

Definition 4.1 (Time-based Association Rule). Given a Timed Lattice $L^t = (L, E^t)$ and a Temporal Edge between the concepts $C_i, C_j \in L$ induced by $(g, t_s) \in \circ(C_i)$ and $(g, t_k) \in \circ(C_j)$. Let $M(C_i)$ and $M(C_j)$ be the attributes of the concepts C_i and C_j . A Time-based Association Rule is an implication of the form $M(C_i) \Rightarrow_{s,k} M(C_j)$. The support is defined by:

$$\text{sup}(M(C_i) \xrightarrow{s,k} M(C_j)) = \frac{|(M(C_i) \cup M(C_j))'|}{|G|} \quad (2)$$

and the confidence is computed as follows:

$$\text{conf}(M(C_i) \xrightarrow{s,k} M(C_j)) = \frac{|(M(C_i) \cup M(C_j))'|}{|(M(C_i))'|}, \quad (3)$$

where $(.)'$ is the derivation operator.

The next step is to derive Time-based Behavior Rules from Time-based Association Rules. Let us start by providing two considerations. The first one is that the *Timed Lattice* previously built is useful to recognize emerging students' behaviors. Such behaviors are represented by learned formal concepts in the lattice and characterize students who made dropout, given that the whole process starts by using historical data related to these students. Therefore, it is possible to affirm that dropout-conducting behaviors are elicited. The second one is that concepts, in the *Timed Lattice*, also include time information (their extensional parts contain time-stamped objects) that can be used to understand the time slots in which the aforementioned dropout-conducting behaviors arise. Hence, it is possible to define *Time-based Behavior Rules*. More in detail, a *Time-based Behavior Rule* $M(C_i) \xrightarrow{s,k} M(C_j)$ is a Time-based Association Rule $M(C_i) \xrightarrow{s,k} M(C_j)$ such as it exists a Temporal Edge between C_i and C_j , due to the existence of at least $(g, ts_s) \in \circ(C_i)$ and $(g, t_k) \in \circ(C_j)$, of length equal to one time slot (i.e., $k = s + 1$) and the confidence of the rule is greater than or equal to a threshold τ .

Definition 4.2 (Time-based Behavior Rule). Given a Timed Lattice $L^t = (L, E^t)$ and the set $R[L^t]$ of all Time-based Association Rules derived from L^t , the set of all Time-based Behavior Rules $R^B[L^t]$ is defined as follows:

$$R^B[L^t] = \{M(C_i) \xrightarrow{s} M(C_j) \mid M(C_i) \xrightarrow{s,k} M(C_j) \in R[L^t], \quad k = s + 1, \\ \text{conf}(M(C_i) \xrightarrow{s,k} M(C_j)) \geq \tau\}.$$

Definitely, *Time-based Behavior Rules* describe the dropout-conducting patterns that are the final outcomes of the approach proposed in this study.

5 | EXTRACTING DROPOUT-CONDUCTING PATTERNS: THE PROPOSED APPROACH

The proposed approach aims to build a model, consisting of a set of Time-based Behavior Rules, allowing the definition of a DSS for teachers (and, in general, for didactic staff) to identify students at risk of dropout and, consequently, enable them to provide all suitable actions to mitigate such risk (e.g., help teaching, motivational feedback, personalizing learning path, and so on).

The approach consists of three phases: (i) feature selection, (ii) formal context building, and (iii) rule mining. This workflow starts by taking as input the available historical students' data. It proceeds by preparing the formal context in which the attributes are the selected subset of available features, and the objects are time-stamped traces of students' activities. It continues applying the TCA to create a formal lattice in which temporal edges are explicit (timed lattice).

Finally, the workflow is concluded by extracting rules from such lattice. All details are described in the next sections.

5.1 | Features selection

Recalling the main goal of the proposed approach, it is needed a way to model the temporal evolution of the behavior of students at risk of dropout. The specialized literature provides us with several works, like, References [13,14] and [25], which suggest us to divide the whole course period, for each student of a MOOC, in sequential time slots of a fixed length. Moreover, in each time slot, it is required to model the behavior of students at risk of dropout to obtain a snapshot for each student in that time slot. Thus, in such a way, it is possible to create a temporal sequence of snapshots representing individual students' behavior. Typically, in the analyzed works, the authors aggregate specific events in each time slot to obtain a value representing a specific part of the student's behavior in such range of time.

On the other hand, the analysis concerns how to choose useful features to model the student's behavior (related to the risk of dropout) in each slot. With respect to the above purpose, the authors of Reference [26] state that *motivation* is a factor influencing dropout, and they measure it using three factors: (i) engagement of the student, (ii) engagement source, and (iii) the energization, defined as the speed of activity together with high quality of performance since an increase of velocity in making exercises and other activities is an indicator of the student's perceived knowledge. Thus, motivation seems to be an important aspect to be measured. Such measurement requires a set of features and necessarily comes from aggregating events coming from students' interactions on the MOOC platform. This is also sustainable, since all MOOC platforms offer a logging capability. In Appendix A, there is a list of the typically used features.

In general, only those features considered applicable to data coming from a generic MOOC platform will be chosen. Table 3 in Appendix A reports, as well, the subset of features used in this study for the experimentation described in Section 6.

5.2 | Formal context building

In this activity of the workflow, *Formal Context* will be prepared for executing FCA (more precisely, TCA). The inputs of this activity are the selected features and the log of the MOOC platform. This log includes a time-stamped set of entries describing for each course the interaction of a student with the educational material. For example, student's access to a course, student's visualization of a video lesson, student's answers to a problem/exercise, and so on.

To build-up the *Formal Context* (i.e., the input of the FCA), it is needed to group the events traced by the MOOC (and included in the log file) within each considered time slot (with a fixed length). To build the *Formal Context* it is required to address the following preliminary tasks:

- Temporal segmentation of the data, the time length T of the course c , namely T_c , is divided into n time slots t_i , $i = 1 \dots n$, of fixed length w (as shown in Figure 1). The value of n is empirically decided in compliance with the average duration of the courses.
- Counting and scaling features, for each of the selected features, representing an interesting event to measure for classifying the student's behavior, since it is needed to aggregate and scale its value. Specifically, within the same time slot t_i (where $1 \leq i \leq n$), the system measures the

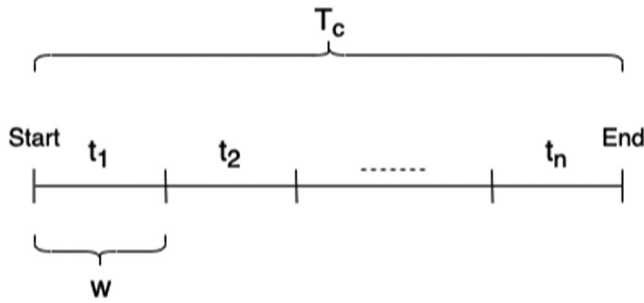


FIGURE 1 A course divided in n time slots [Color figure can be viewed at wileyonlinelibrary.com]

events by grouping the activities of each student by course and aggregating them by using function like `sum`, `mean`, in this study we adopted `count` operator. Then a scaling function is adopted for binning the value in different labeled intervals (e.g., `LOW`, `MEDIUM`, etc.).

The aforementioned preprocessing tasks allow us to prepare the *Formal Context* by selecting both the attribute set M and the set of objects G and by defining the binary relation I among them. Fixed the number of time slots n , let us assume to scale each feature value, representing the counter of how many times that interaction happened, in three labeled bins: *LOW*, *MED*, and *HIGH*. Then, the set of attributes M is constructed by apply scaling with respect to these bins. For each considered feature j , the set of attributes M will contain three attributes m_j^{LOW} , m_j^{MED} , and m_j^{HIGH} (representing respectively a low, a medium, and a high value for feature j with respect to intervals boundaries). The set of objects G is a subset of the Cartesian product $S \times C \times T$, where the sets S and C contain, respectively, students and courses' identifiers, while T is the set of all possible time slots. The set G contains objects g in the form $g = (s_{id}, c_{id}, t_i)$, where s_{id} is the unique identifier of the student s , c_{id} is the unique identifier of the course c (attended by the student s), and t_i is the i th time slot whose information are associated to the object g . The combination of the student's and the course's unique identifier determines a value that is called `enrollid`. More in detail, $g = (s_1, c_1, t_1)$ and $g' = (s_1, c_1, t_2)$ represent the same student in the same course behaving in two different time slots.

Table 1 illustrates a sketch of a *Formal Context* $K = (G, M, I)$ derived by applying the characteristic function $\mathcal{I}: G \times M \rightarrow \{0, 1\}$ defined below. More formally, consider a student having `enrollid` s_{id} participating at the activities of course with identifier c_{id} during a time slot t_i and let j be the feature to be observed during all the course's time slots. We denote by $count_{(s_{id}, c_{id})}^{t_i}[j]$ the number of events, associated to feature j , that student s_{id} generated for the course c_{id} during the time slot t_i . The function \mathcal{I} can be defined in the following way.

TABLE 1 Sketch of a formal context

(G, T_{c_1})	m_1^{LOW}	m_1^{MED}	m_1^{HIGH}	...	m_k^{LOW}	m_k^{MED}	m_k^{HIGH}
(s_1, c_1, t_1)	1	0	0	...	0	0	1
(s_1, c_1, t_2)	0	1	0	...	0	1	0
(s_1, c_1, t_3)	1	0	0	...	0	0	1
...
(s_1, c_1, t_n)	1	1	0	...	1	1	1

$$\mathcal{I}\left((s_{id}, c_{id}, t_i), m_j^{LOW}\right) = \begin{cases} 1, & \text{if } count_{(s_{id}, c_{id})}^{t_{s_i}}[j] \text{ is in the LOW region} \\ 0, & \text{otherwise} \end{cases}$$

$$\mathcal{I}\left((s_{id}, c_{id}, t_i), m_j^{MED}\right) = \begin{cases} 1, & \text{if } count_{(s_{id}, c_{id})}^{t_{s_i}}[j] \text{ is in the MEDIUM region} \\ 0, & \text{otherwise} \end{cases}$$

$$\mathcal{I}\left((s_{id}, c_{id}, t_i), m_j^{HIGH}\right) = \begin{cases} 1, & \text{if } count_{(s_{id}, c_{id})}^{t_i}[j] \text{ is in the HIGH region} \\ 0, & \text{otherwise} \end{cases}$$

Table 1 shows the student $s = 1$, in the course $c = 1$. Such student changes her/his behaviour over the time. In particular, in the time slot $t = 1$ the values of the attribute m_1 lie in a low region ($m_1^{LOW} = 1$), instead in the second time slot the values for the same attribute lie in a medium region ($m_1^{MED} = 1$) and so on. Lastly, the built *Formal Context* must be processed by the TCA algorithms to obtain the *Timed Lattice* (formal lattice enriched with temporal edges). Such lattice will become the input of the next phase (the mining of time-based rules).

5.3 | Time-based Behavior Rule mining

This section describes (see Algorithm 1) a simple method to extract *Time-based Behavior Rules* from a *Timed Lattice*. The proposed method is essentially a brute force algorithm that browses all existing temporal edges among the extracted concepts in the *Timed Lattice*. It only aims to show a possible implementation of rules mining without any presumption to be efficient and optimized.

Algorithm 1 Time-based Behavior Rules mining over a Timed Lattice

Input: Timed Lattice $L^t = (L, E^t)$ where $L = (C, \leq)$, number of time slots n .

Output: Set of Time-based Behavior Rule *rules*.

```

1:  $H = \text{initializeDictionary}(\text{emptyList}(), n)$ 
2: for  $c$  in  $C$  do
3:   for  $(g, t)$  in  $\circ(c)$  do
4:      $H[t].\text{append}((c, t))$ 
5:   end for
6: end for
7:  $rules = \text{emptyList}()$ 
8: for  $i = 1, \dots, n - 1$  do
9:   for  $(c^{head}, i)$  in  $H[i]$  do
10:    for  $(c^{tail}, i + 1)$  in  $H[i + 1]$  do
11:      if  $\bar{\circ}(c^{head}) \cap \bar{\circ}(c^{tail}) \neq \emptyset$  and  $\text{conf}(M(c^{head}) \xrightarrow{i, i+1} M(c^{tail})) \geq \tau$  then
12:         $rules.\text{append}(M(c^{head}) \xrightarrow{i} M(c^{tail}))$ 
13:      end if
14:    end for
15:   end for
16: end for
17: Output:  $rules$ 

```

More in detail, the algorithm mainly aims at positioning the learned (from the *Timed Lattice*) *Formal Concepts* onto the time dimension.

In particular, the main idea is to enrich each concept representation by using the information about the time slot at which such a concept emerges from data. To achieve this objective, it is possible to use information hidden into object descriptions. The dictionary H is used to group the collected timed concept descriptions in several buckets, one bucket for each time slot (we assume having n time slots) (lines 16). Thus, at the end of the first part of the algorithm, we have associated concepts to the time slots in which they arise. Now, it is possible to extract all admissible rules for time slots 1–2, those for time slot 2–3, and so on (lines 716).

More in detail, the rule construction process is executed once for each couple of consecutive buckets. Thus, for $i = 1 \dots n - 1$, a cartesian product between buckets i and $i + 1$ is executed. In this way, the result is the set of all potential rules having a concept c^{head} emerged in the time slot i as the antecedent of the rule and a concept c^{tail} emerged in time slot $i + 1$ as the consequent of the rule. Only plausible rules (at least a *Temporal Edge* should exist between c^{head} and c^{tail} , in other words $\bar{O}(c^{head}) \cap \bar{O}(c^{tail}) \neq \emptyset$) with a confidence (see Definition 4.1) greater than or equal to a threshold τ are selected. The algorithm adopts Definition 4.2 to check the admissibility of a potential rule. It is important to remember that the rules are represented by using attributes belonging to the intensional parts of the considered concepts through the application of the function M . This is useful to improve also readability and understandability of such rules.

The generated rules $M(c^{head}) \stackrel{i}{\Rightarrow} M(c^{tail})$, constructed using data related to dropping-out students, represent the temporal evolution across two consecutive time slots of the aforementioned students' behavior. A sketch of the algorithm execution is depicted in Figure 2. In that figure, it is shown the concatenation of concepts (in pink) by temporal edges (the edges in red) from which it is possible to extract behavioral patterns (row 1). The example shows the pattern $A1 \Rightarrow B2 \Rightarrow C3$, representing the concatenation of the timed concepts that emerged, respectively time slot one, two, and three. This kind of pattern is transformed into rule-shaped temporal patterns, rules that are valid along a clear timeline, in this case, from time slot one to three. For each rule, it is evaluated support and confidence that represent the coverage and the strength of the rule, respectively. These measures are also used to filter out the weakest rules with confidence smaller than a threshold of τ .

6 | CASE STUDY

The proposed workflow has been instantiated on the data set *KDD Cup 2015*.⁷ An experiment has been conducted, and its results are presented and discussed in the following sections. The case study has been realized by using Python 3.6, NumPy,[†] Pandas,[‡] and Jupyter (i.e., a Python development environment specialized for Data Analysis).

6.1 | Data set description

The data set *KDD Cup 2015*⁷ was extracted from the MOOC platform XuetangX.[§] Such a data set contains information about students enrolled in one or more courses provided by the platform. In particular, the core part of the data set consists of a log file containing traces for students' interactions on the system.

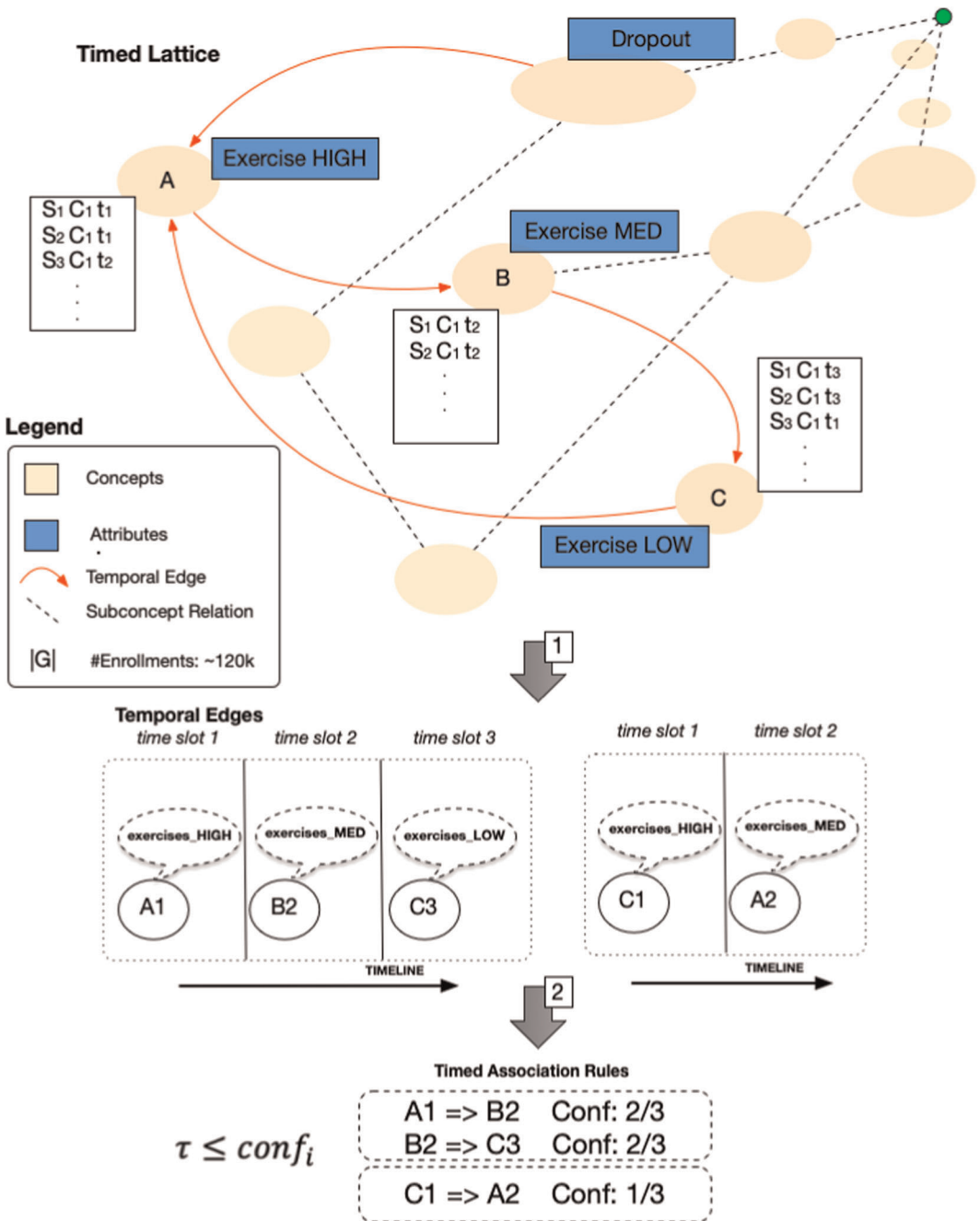


FIGURE 2 Algorithm execution example [Color figure can be viewed at wileyonlinelibrary.com]

The purpose of the KDD competition was to predict whether students would drop out in an instructor-paced course based on activities performed during the course itself. Students were labeled with “1” if a student has dropped out of the course and “0” otherwise. The label “1” was assigned if the student did not leave in 10 consecutive days a record of his/her activities during the course. There are 39 courses, and 120.542 enrollment IDs (student identifiers within a

course) spread among the courses. Among these students, only 20.70% did not dropout (label “0”), the 79.29% have abandoned the course (label “1”). The *KDD Cup 2015* data set contains four CSV files^{8,27}:

- The first one (`object.csv`) provides information about the courses and their content metadata.
- The second one (`enrollment_train.csv`) is a join table between enrollment data and course data.
- The third one (`log_train.csv`) contains data on students' activities.
- The last one (`true_train.csv`), for each pair of student-course, contains information about the dropout label (if a given student enrolled in a specific course dropped out such course or not).

For the execution of the experiment, the data set has been split into a training set and a test set, according to the typical logic of 70–30 (70% for the training set and 30% for the test set). The training set is used to build the model (i.e., a set of rules representing behavioral patterns of students who made dropout). The test set, instead, is used to execute and evaluate such rules to predict the dropout behavior of students.

6.2 | Data preprocessing

The data set introduced in Section 6.1 must be preprocessed to obtain the *Formal Context* as described in Section 5. In Appendix A we summarize the features, to be included in the *Formal Context*, we selected from the relevant ones analysed in Section 5.1.

Figure 3 is a screenshot from Jupyter showing a fragment of the preprocessing results aimed to compute the selected features values. For instance, the field `avg_bt看_sessions` denotes the average time between the user's sessions (it measures the user's inactivity), while the field `problem` denotes the amount of the exercises solved by the user (it measures, in some sense, the autonomous study of the user, without considering the correctness of the solutions), for details see Appendix A. In particular, it is possible to see how the feature values change, for

enrollid	t	problem	video	access	wiki	discussion	navigate	page_close	avg_session	time_on_task_pc	avg_bt看_sessions	dropout
450	1	12	6	82	1	3	20	6	102.000000	0.666667	0.333333	1
450	2	0	0	5	0	0	2	2	0.500000	0.333333	0.500000	1
450	3	2	0	16	0	0	5	1	126.500000	0.333333	0.500000	1
450	4	0	0	0	0	0	0	0	0.000000	0.000000	6.000000	1
32216	1	4	2	54	2	2	20	3	226.800000	0.833333	0.166667	1
32216	2	0	0	4	0	0	3	2	9.000000	0.333333	0.500000	1
32216	3	0	0	0	0	0	0	0	0.000000	0.000000	6.000000	1
32216	4	0	0	0	0	0	0	0	119.000000	0.166667	0.833333	1
18728	1	10	10	33	0	2	6	27	559.000000	1.000000	0.000000	0
18728	2	14	17	47	1	2	10	38	915.333333	1.000000	0.000000	0
18728	3	0	7	32	0	0	0	24	1149.166667	1.000000	0.000000	0
18728	4	11	21	84	0	0	0	48	1002.666667	1.000000	0.000000	0

FIGURE 3 Preprocessed data set

each `enrollid` (note that the column represents the combination between a student and a course in which she is enrolled), along with the timeline (column t reports the number of the current time slot). Therefore, starting from the aforementioned information, the triples (s_{id}, c_{id}, t_i) , defined in the Section 5.2, can be constructed. Moreover, it is possible to observe changes in the behavior of a specific student by tracing the same `enrollid` value along consecutive t values. Indeed, in the Figure 3, are shown three students, two left the course (`dropout` = 1) and one passed it (`dropout` = 0). Differences between these students are underlined by feature values. For example, the dropout students have values near or equal to zero among all time slots. Instead, the no-dropout student starts the course with a certain level of engagement, keeping it constant overall time slots.

The next step is now to discretize the obtained feature values by, first, normalizing them in the range $[0, 1]$ and, then, positioning normalized values into one of the three plausible regions (L-LOW, M-MEDIUM, and H-HIGH). The number of intervals for scaling the features may be different, and we empirically choose 3 as it provided the best performance in terms of Precision-Recall measures for recognizing dropout. Once all values were discretized, a conceptual scaling by using the function \mathcal{I} (see Section 5.2) operation was applied to transform multivalued attributes in boolean attributes (those admitted by the FCA approach). Technically, such a process has been realized by using the *one-hot encoding* technique, where a categorical feature becomes an array whose size is the number of possible values for such features. The final result is a sparse matrix with zeros and ones, as shown in the Table 1.

Figure 4 shows a fragment of the preprocessed data set modelled as a *Formal Context*. In particular, in Figure 4 it is possible to note the temporal evolution of the behaviors of previous students in a given course. The first student (`enrollid` = 450), in the time slot $t = 1$, shows much time spent on tasks (`tot_pc_H`), but such amount decreases along the next consecutive time slots. Moreover, the average session (represented by attributes `avg_session_H`, `avg_session_M`, and `avg_session_L`) is stable around low values. The traced student made dropout for the considered course (`dropout` = 1). The second student (`enrollid` = 32216) seems to have similar behavior of the previous one, indeed, the student has a decrease in the time spent on task along the time slot and both have high values for time between sessions (`avg_bt看_session_H` = 1), in this case having high values means that the student does not connect often to the platform, indication of no-engagement, indeed the student has abandoned the course. The third (`enrollid` = 18728), instead, has connected to

<code>enrollid</code>	<code>t</code>	<code>dropout</code>	<code>avg_sess_L</code>	<code>avg_sess_M</code>	<code>avg_sess_H</code>	<code>tot_pc_L</code>	<code>tot_pc_M</code>	<code>tot_pc_H</code>	<code>avg_bt看_sess_L</code>	<code>avg_bt看_sess_M</code>	<code>avg_bt看_sess_H</code>
450	1	1	1	0	0	0	0	1	1	0	0
450	2	1	1	0	0	0	1	0	1	0	0
450	3	1	1	0	0	0	1	0	1	0	0
450	4	1	1	0	0	1	0	0	0	0	1
32216	1	1	1	0	0	0	0	1	1	0	0
32216	2	1	1	0	0	0	1	0	1	0	0
32216	3	1	1	0	0	1	0	0	0	0	1
32216	4	1	1	0	0	1	0	0	1	0	0
18728	1	0	0	1	0	0	0	1	1	0	0
18728	2	0	0	1	0	0	0	1	1	0	0
18728	3	0	0	0	1	0	0	1	1	0	0
18728	4	0	0	0	1	0	0	1	1	0	0

FIGURE 4 Formal context

the platform constantly keeping low the time of inactivity ($avg_btw_session_L = 1$) and the average time connected high ($avg_session_M = 1$ and $avg_session_H = 1$), at the end during course he kept highly constant the activity performed ($tot_pc_H = 1$).

6.3 | Experiment execution

The experiment has been conducted in two phases: (i) by implementing (in Python, NumPy, and Pandas) and executing (in Jupyter) the approach provided in Section 5; (ii) by exploiting the obtained *timed rules* to predict dropout behaviors for students and courses in the test set. The second phase is structured in different cycles (one for each considered rule confidence threshold). More in detail, for the first phase, the next step is to apply the proposed approach to extract the *time-based Behavior Rules*. For each generated rule, we calculated its confidence that measures, in some sense, the rule strength.

Taking a look at Figure 5, it is possible to see a fragment of the set of rules generated during the experiment. Such a figure reports the shape of the rule (antecedent and consequent) and the validity periods during the course. For instance, focusing on the rule with index 5, its

rules				
	antecedent	consequent	ts_start	ts_end
5	avg_bt看_sessions_low#avg_session_low#tot_pc_high	avg_bt看_sessions_low#avg_session_low#tot_pc_med	1	2
8	avg_bt看_sessions_low#avg_session_med#tot_pc_low	avg_bt看_sessions_low#avg_session_low#tot_pc_low	1	2
12	avg_bt看_sessions_low#avg_session_high#tot_pc_low	avg_bt看_sessions_low#avg_session_low#tot_pc_low	1	2
24	avg_bt看_sessions_low#avg_session_low#tot_pc_low	avg_bt看_sessions_low#avg_session_low#tot_pc_low	1	2
30	avg_bt看_sessions_low#avg_session_low#tot_pc_med	avg_bt看_sessions_low#avg_session_low#tot_pc_low	1	2
36	avg_bt看_sessions_high#avg_session_low#tot_pc_low	avg_bt看_sessions_high#avg_session_low#tot_pc_low	1	2
43	avg_bt看_sessions_low#avg_session_low#tot_pc_high	avg_bt看_sessions_low#avg_session_low#tot_pc_med	2	3
50	avg_bt看_sessions_high#avg_session_low#tot_pc_low	avg_bt看_sessions_high#avg_session_low#tot_pc_low	2	3
58	avg_bt看_sessions_low#avg_session_low#tot_pc_low	avg_bt看_sessions_low#avg_session_low#tot_pc_low	2	3
63	avg_bt看_sessions_low#avg_session_med#tot_pc_med	avg_bt看_sessions_low#avg_session_low#tot_pc_med	2	3
68	avg_bt看_sessions_low#avg_session_med#tot_pc_low	avg_bt看_sessions_low#avg_session_low#tot_pc_low	2	3
71	avg_bt看_sessions_low#avg_session_med#tot_pc_high	avg_bt看_sessions_low#avg_session_low#tot_pc_high	2	3
76	avg_bt看_sessions_low#avg_session_low#tot_pc_med	avg_bt看_sessions_low#avg_session_low#tot_pc_low	2	3
79	avg_bt看_sessions_low#avg_session_high#tot_pc_low	avg_bt看_sessions_low#avg_session_low#tot_pc_low	2	3
84	avg_bt看_sessions_low#avg_session_med#tot_pc_med	avg_bt看_sessions_low#avg_session_low#tot_pc_med	3	4
86	avg_bt看_sessions_low#avg_session_low#tot_pc_med	avg_bt看_sessions_low#avg_session_low#tot_pc_low	3	4
93	avg_bt看_sessions_high#avg_session_low#tot_pc_low	avg_bt看_sessions_low#avg_session_low#tot_pc_low	3	4
96	avg_bt看_sessions_high#avg_session_low#tot_pc_low	avg_bt看_sessions_high#avg_session_low#tot_pc_low	3	4
100	avg_bt看_sessions_low#avg_session_low#tot_pc_low	avg_bt看_sessions_low#avg_session_low#tot_pc_low	3	4
106	avg_bt看_sessions_low#avg_session_high#tot_pc_low	avg_bt看_sessions_low#avg_session_low#tot_pc_low	3	4
114	avg_bt看_sessions_low#avg_session_med#tot_pc_low	avg_bt看_sessions_low#avg_session_low#tot_pc_low	3	4
122	avg_bt看_sessions_low#avg_session_med#tot_pc_high	avg_bt看_sessions_low#avg_session_med#tot_pc_high	3	4

FIGURE 5 Time-based behavior rules

interpretation is that students behaving in the same way (low values for the feature `avg_btw_sessions`, low values for the feature `avg_session` and high values for the feature `tot_pc`) during the first time slot (value 1 for the attribute `ts_start`) and in the same way of the behavior expressed during the second time slot (value 2 for the attribute `ts_end`) are likely to make dropout before the end of the course.

The second phase of the experiment has been executed by applying the *Time-based Behavior Rules*, extracted from the training set, over the preprocessed log entries corresponding to the students in the test set. The trained model performance is obtained by measuring Precision and Recall of the truly raised dropout alert predicted by the system and false alarms (dropout yes or no) over the test set. The results are reported and discussed in the next section.

Now, let us briefly explain how the extracted rules are applied over new data. Looking at Figure 4, it is possible to match the behaviors of the students with the extracted rules shown in Figure 5, in particular, the object `enrollid = 32216` (i.e., a specific student in a given course) satisfies, in the early time slots, the first rule in the first row of the Figure 5. Hence, a dropout alert can be provided to the didactic staff. The same object also satisfies the rule at index 76 in the second and third time slots. This match leads to another alert for the didactic staff, and so forth.

In this case, during the evaluation, the prediction is correct because object `enrollid = 32216` is originally labeled with `dropout = 1`. Lastly, Figure 6 shows a graphical representation of the identified behavioral pattern extracted, in particular, this pattern concatenates the rules with index 5, 76, and 93.

6.4 | Results and discussion

To evaluate the performance of the trained model, multiple experiments have been carried out with different levels of rules' confidence. The rules set is built upon the training set that contains data of 66,906 enrollment IDs (unique identifier of student within a course) having abandoned the course. To validate the model, we used a test set composed of 28,675 enrollment IDs that dropped out courses and 7489 that did not do it. Table 2 shows the results of the test phase. In particular, *precision* and *recall* have been calculated to evaluate the classification task accomplished by the mined *timed rules* for each considered confidence level and for each time slot.

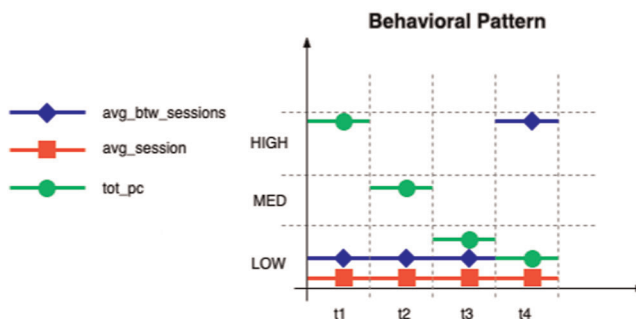


FIGURE 6 Graphical representation of student's behavior [Color figure can be viewed at wileyonlinelibrary.com]

In Table 2, the term *confidence* indicates the level of confidence of the considered rules, *rules* specifies the number of rules used for the classification (such rules have a confidence greater than or equal to the confidence level considered), and t_i indicates the cut of the test set data. For example, t_2 means that for the classification are used only data corresponding the first and the second time slot, t_3 means that the data used concern the first, second, and third time slot, and so on. Therefore, it is possible to notice that the precision and recall are high in the early time slots. The best results, in terms of precision, are obtained with the confidence level 0.4, where is possible to see that the precision increase with the number of the time slots considered, this means that at each step more information are used during the classification. Moreover, the weakest and least accurate rules are filtered out. On the flip side, the recall, indicating the percentage of true dropout enrollments (true-positive) among all the enrollments classified as dropout (both false-positive and true-positive), is the lowest. In detail, with a confidence level 0.4 it is caught the 65.47% of the overall enrollments associated to dropout in the original data set. But, despite this, with few data our model is able to have optimal results. Lastly, the overall approach has been also tested over the data set reporting only data related to students who completed the courses. The idea was to try to extract also behavior change patterns for such students. Honestly, the results were not positive like for the dropout-conducting patterns. Plausible explanations to this phenomenon are related to the available data and to the fact that the adopted features (to represent students' behavior) have

TABLE 2 Model evaluation

Confidence = 0.1, Rules = 63			
Measures	t_2	t_3	t_4
Precision	76.91%	76.96%	77.64%
Recall	95.69%	96.37%	96.41%
F-score	0.852	0.855	0.859
Confidence = 0.2, Rules = 48			
Measures	t_2	t_3	t_4
Precision	78.85%	79.26%	80.10%
Recall	89.82%	91.59%	91.81%
F-score	0.839	0.849	0.855
Confidence = 0.3, Rules = 35			
Measures	t_2	t_3	t_4
Precision	78.97%	81.69%	83.33%
Recall	69.07%	90.96%	89.54%
F-score	0.736	0.859	0.862
Confidence = 0.4, Rules = 22			
Measures	t_2	t_3	t_4
Precision	78.41%	81.84%	84.58%
Recall	53.65%	59.26%	65.47%
F-score	0.636	0.686	0.737

been selected to emphasize anomalous behaviors conducing to dropout because the main goal of the work was to support a DSS to early detect students at risk.

7 | APPLYING RULES TO IMPLEMENT A DSS

One of the main contributions of this study is the definition of DSS applied to educational platforms to improve their performance, effectiveness, and efficiency.²⁸ In the context of MOOCs, a DSS could help to increase the students' engagement, retain them, to avoid, preventing, or lessening dropout percentage. For the self-efficacy and the teacher's presence, a DSS can analyze the students' behaviors and notify the teacher, which are students at risk of dropout, but also explain such a result. Therefore, the teacher can provide timely and customized feedback to those students at risk. These alerts are useful to change and replan the course to simplify it and to encourage the students' self-efficacy and participation. The implementation of a DSS relies on the rules describing the temporal behavior of students trained through the workflow illustrated in Figure 7.

As shown in Figure 7A, the process input is the log file, including time-stamped students' entries, like clicks, watched videos, solved problems, and so on, gathered by the MOOC

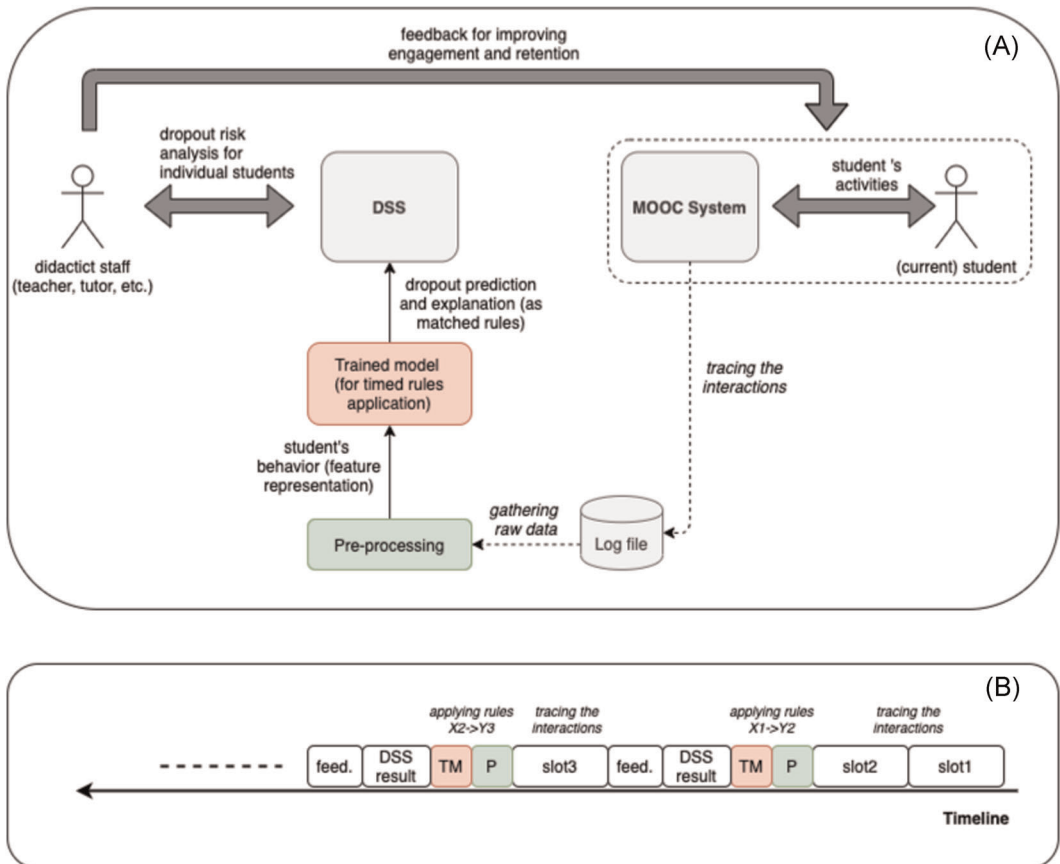


FIGURE 7 DSS architecture exploiting the trained model. (A) represents the general workflow of the proposed model, (B) details how the model works step-by-step along the timeline [Color figure can be viewed at wileyonlinelibrary.com]

platform. At the end of two consecutive time slots (Figure 7B), the log file is preprocessed and features for specific students calculated and discretized in the same way of the trained model. At this stage, such information can be compared against the trained model (the set of Time-based Behavior Rules obtained by applying the approach proposed in this study). The result is a set of students (whose features currently match at least one of the rules in the trained model) and their behaviors (summarized through features reported in Appendix A). Such students have been classified as being at risk of dropout. The DSS provides the didactic staff with such information allowing teachers, tutors, and so on, to deliver their feedback with the objective to increase the engagement level of the students at risk. The system continuously repeats these operations when collecting new students' data. Time is an important parameter because the same behavior can be risky in a given time slot and safe in a different time slot. This aspect is well supported by the proposed approach with *Time-based Behavior Rules* (see Section 5).

An example of this loop is shown in Figure 7B where students' data is elaborated in the second time slot for checking if rules, with the antecedent in time slot 1 and consequent in time slot 2, are satisfied. After the first two time slots, the above iteration is executed for each new time slot for rules with the antecedent in time slot 2 and consequent in time slot 3, and so forth.

8 | CONCLUSIONS AND FUTURE WORKS

This study proposes an approach to build an educational DSS to mitigate the dropout risk for MOOC students. The method is based on the extraction of behavioral patterns of the dropout students from historical data (log files of a MOOC platform). The novelties of the approach concern the modeling of these patterns through time-based rules. More in detail, such rules reflect the shift of the dropout students' behavior from a time slot to the next one. In this way, the recognition is context-dependent on the part of the course in which it manifested. The approach conceptualizes the students' behaviors by an aggregating group of indicators characterizing the students' trends. This aspect helps decision-makers to better reason about suitable actions to mitigate the risk. Experimentation reveals promising results.

In the future, the authors will investigate some extensions of the proposed work, for instance, using the subject matter knowledge to refine the timed rules and considering a richer set of features from which conceptualize and extract rules.

ACKNOWLEDGEMENTS

This study was partially supported by CORISA (Centro di Ricerca Sistemi ad Agenti), the authors were involved as consulting subcontractors for research project granted by the Italian Government (MISE—Ministero dello Sviluppo Economico) under the program PON I&C 2014-20 (HORIZON 2020—ID F/050371/02/X32) in the area of Educational Data Mining.

ENDNOTES

*<https://www.biendata.xyz/competition/kddcup2015/>.

†<https://numpy.org/>.

‡<https://pandas.pydata.org/>.

§<https://next.xuetangx.com/>.

ORCID

Blundo Carlo  <http://orcid.org/0000-0002-8225-3401>
 Fenza Giuseppe  <http://orcid.org/0000-0002-4736-0113>
 Fuccio Graziano  <http://orcid.org/0000-0001-9213-5284>
 Loia Vincenzo  <http://orcid.org/0000-0003-4807-8942>
 Orciuoli Francesco  <http://orcid.org/0000-0001-6899-4396>

REFERENCES

1. Liyanagunawardena TR, Adams AA, Williams SA. MOOCs: a systematic study of the published literature 2008–2012. *Int Rev Res Open Distributed Learn*. 2013;14(3):202-227.
2. Jung Y, Lee J. Learning engagement and persistence in massive open online courses (MOOCs). *Comput Educ*. 2018;122:9-22. <https://doi.org/10.1016/j.compedu.2018.02.013>
3. Yu H, Miao C, Leung C, White TJ. Towards AI-powered personalization in MOOC learning. *NPI Sci Learn*. 2017;2(1):1-5.
4. Gütl C, Rizzardini RH, Chang V, Morales M. Attrition in MOOC: lessons learned from drop-out students. In: Uden L, Sinclair J, Tao YH, Liberona D, eds. *Learning Technology for Education in Cloud. MOOC and Big Data*. Cham: Springer International Publishing; 2014: 37-48.
5. Kloft M, Stiehler F, Zheng Z, Pinkwart N. Predicting MOOC dropout over weeks using machine learning methods. In: *Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs*; 2014:60-65. <https://doi.org/10.3115/v1/W14-4111>
6. Boyer S, Gelman BU, Schreck B, Veeramachaneni K. Data science foundry for MOOCs. In: *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. Paris, France; 2015:1-10. <https://doi.org/10.1109/DSAA.2015.7344825>
7. Knowledge Discovery & Data Mining. MOOC dataset from KDD Cup 2015; 2015. <http://data-mining.philippe-fournier-viger.com/the-kddcup-2015-dataset-download-link/>
8. Ardchir S, Talhaoui MA, Jihal H, Azzouazi M. Predicting MOOC dropout based on learner's activity features. *Int J Eng Technol*. 2018;7:124-126.
9. Pigeau A, Aubert O, Prié Y. Success prediction in MOOCs: a case study. In: *Paper presented at the International Conference on Educational Data Mining (EDM)*, July 2-5, 2019. Montreal, Canada; 2019.
10. Qiu L, Liu Y, Liu Y. An integrated framework with feature selection for dropout prediction in massive open online courses. *IEEE Access*. 2018;6:71474-71484. <https://doi.org/10.1109/ACCESS.2018.2881275>
11. Li Y, Fu C, Zhang Y. When and who at risk? Call back at these critical points. *EDM*. 2017.
12. Feng W, Tang J, Liu TX. Understanding dropouts in MOOCs. *Proc AAAI Conf Artif Intell*. 2019;33(1): 517-524. <https://doi.org/10.1609/aaai.v33i01.3301517>
13. Chen J, Feng J, Sun X, Wu N, Yang Z, Chen S. MOOC dropout prediction using a hybrid algorithm based on decision tree and extreme learning. *Machine, Mathematical Problems in Engineering*; 2019. <https://doi.org/10.1155/2019/8404653>
14. Peng D, Aggarwal G. Modeling MOOC dropouts. *Entropy*. 2015; 10(114): 1-5.
15. Luo Y, Zhou G, Li J, Xiao X. Study on MOOC scoring algorithm based on Chinese University MOOC learning behavior data. *Heliyon*. 2018;4(11):e00960. <https://doi.org/10.1016/j.heliyon.2018.e00960>
16. Ganter B, Wille R. *Formal Concept Analysis: Mathematical Foundations*. 1st ed. Secaucus, NJ: Springer-Verlag New York, Inc; 1997.
17. De Maio C, Fenza G, Loia V, Senatore S. Hierarchical web resources retrieval by exploiting fuzzy formal concept analysis. *Inf Process Manage*. 2012;48(3):399-418. <https://doi.org/10.1016/j.ipm.2011.04.003>
18. Elzinga P, Wolff KE, Poelmans J. Analyzing chat conversations of pedophiles with temporal relational semantic systems. In: *2012 European Intelligence and Security Informatics Conference (EISIC)*. IEEE; 2012: 242-249. <https://doi.org/10.1109/EISIC.2012.12>
19. Wolff KE. Temporal concept analysis explained by examples. In *CDUD'11-Concept Discovery in Unstructured Data*; 2011:104.

20. Wolff KE. Temporal concept analysis. In: *ICCS-2001 International Workshop on Concept Lattices-Based Theory, Methods and Tools for Knowledge Discovery in Databases*. Palo Alto (CA): Stanford University; 2001: 91-107.
21. Neouchi R, Tawfik AY, Frost RA. Towards a temporal extension of formal concept analysis. In: *Advances in Artificial Intelligence*. Springer; 2001:335-344. https://doi.org/10.1007/3-540-45153-6_33
22. Wolff KE. Towards a conceptual system theory. In: *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics, SCI*; 2000:124-132.
23. De Maio C, Fenza G, Loia V, Parente M. Time aware knowledge extraction for microblog summarization on twitter. *Inform Fusion*. 2016;28:60-74.
24. Wang DX, Hu XG, Liu XP, Wang H. Association rules mining on concept lattice using domain knowledge. In: *2005 International Conference on Machine Learning and Cybernetics*, Vol. 4. Guangzhou, China: IEEE; 2005:2151-2154. <https://doi.org/10.1109/ICMLC.2005.1527301>
25. Moreno-Marcos PM, Muñoz-Merino PJ, Maldonado-Mahauad J, Pérez-Sanagustín M, Alario-Hoyos C, Kloos CD. Temporal analysis for dropout prediction using self-regulated learning strategies in self-paced MOOCs. *Comput Educ*. 2020;145:103728. <https://doi.org/10.1016/j.compedu.2019.103728>
26. Hershkovitz A, Nachmias R. Developing a log-based motivation measuring tool. In: *EDM*; 2008:226-233.
27. Zhang T, Yuan B. Visualizing MOOC user behaviors: a case study on xuetangx. In: Yin H, Gao Y, Li B, eds. *Intelligent Data Engineering and Automated Learning—IDEAL 2016*. Cham: Springer International Publishing; 2016:89-98.
28. Telem M. DSS in educational organizations. *Comput Educ*. 1990;14(1):61-69. [https://doi.org/10.1016/0360-1315\(90\)90023-Z](https://doi.org/10.1016/0360-1315(90)90023-Z)
29. Hong B, Wei Z, Yang Y. Discovering learning behavior patterns to predict dropout in MOOC. In: *2017 12th International Conference on Computer Science and Education (ICCSE)*; 2017:700-704. <https://doi.org/10.1109/ICCSE.2017.8085583>

How to cite this article: Blundo C, Fenza G, Fuccio G, Loia V, Orciuoli F. A time-driven FCA-based approach for identifying students' dropout in MOOCs. *Int J Intell Syst*. 2021;1-23. <https://doi.org/10.1002/int.22414>

APPENDIX A

Feature name	Comments	Source	Selected
Active days	Days in which the student is connected to course	5,13	Yes
Time spent on course's resources	Time spent on course's materials, as problems, videos, and so on	6	No
Time on tasks of course (Tot_pc)	Time spent in front of PC making course's tasks	26	Yes
Average pace of activity	A measure of the effective work done during the course	26	Yes
Student's sessions	Number of time the student logged in	5	No
Average of session duration (avg_session)	A measure of working time	26	Yes
Average time of inactivity (avg_bt看_session)	The opposite of average session duration	26	Yes
Pace of word marking	Word marking speed	26	No
Game activities percentage	Related to performance-goal orientation	26	No
Exam/problem activities percentage	Tendency to make problems or to take exams	26	Yes
Rate of user dropout	Dropped-out courses on total chosen courses	29	No
Accesses	Total accesses to the course objects except videos and problems	5,10,13	Yes
Problems performed	Total number of problems completed	6,10,13	Yes
Video	Total number of video watched	5,10,13	Yes
Forum posts	Total number of posts in forum threads	10,13	Yes
Course info	Number of navigation to course info	5,10,13	Yes
Page closure	Total times of course's page closure	10,13	Yes
Navigation	Navigation through the course's contents	10,13	Yes
Courses selected by user	Number of courses attended	29	No
Correct submitted problems	A count of only correct problems	6	No
Analysis on video played by student	Analysis on events made during video (stop/pause/rewind)	5	No