A Survey of Machine Learning Approaches for Student Dropout Prediction in Online Courses

BARDH PRENKAJ and PAOLA VELARDI, Sapienza University of Rome, Italy
GIOVANNI STILO, University of L'Aquila, Italy
DAMIANO DISTANTE and STEFANO FARALLI, University of Rome Unitelma Sapienza, Italy

The recent diffusion of online education (both MOOCs and e-courses) has led to an increased economic and scientific interest in e-learning environments. As widely documented, online students have a much higher chance of dropping out than those attending conventional classrooms. It is of paramount interest for institutions, students, and faculty members to find more efficient methodologies to mitigate withdrawals. Following the rise of attention on the Student Dropout Prediction (SDP) problem, the literature has witnessed a significant increase in contributions to this subject. In this survey, we present an in-depth analysis of the state-of-the-art literature in the field of SDP, under the central perspective, but not exclusive, of machine learning predictive algorithms. Our main contributions are the following: (i) we propose a comprehensive hierarchical classification of existing literature that follows the workflow of design choices in the SDP; (ii) to facilitate the comparative analysis, we introduce a formal notation to describe in a uniform way the alternative dropout models investigated by the researchers in the field; (iii) we analyse some other relevant aspects to which the literature has given less attention, such as evaluation metrics, gathered data, and privacy concerns; (iv) we pay specific attention to deep sequential machine learning methods—recently proposed by some contributors—which represent one of the most effective solutions in this area. Overall, our survey provides novice readers who address these topics with practical guidance on design choices, as well as directs researchers to the most promising approaches, highlighting current limitations and open challenges in the field.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Computing methodologies** → **Machine learning**; • **Applied computing** → **Distance learning**; **E-learning**; • **Information systems** → *Data mining*; *Information extraction*;

Additional Key Words and Phrases: Student dropout prediction, learning analytics, educational data mining

ACM Reference format:
Bardh Prenkaj, Paola Velardi, Giovanni Stilo, Damiano Distante, and Stefano Faralli. 2020. A Survey of Machine Learning Approaches for Student Dropout Prediction in Online Courses. *ACM Comput. Surv.* 53, 3, Article 57 (May 2020), 34 pages.
https://doi.org/10.1145/3388792

Authors' addresses: B. Prenkaj and P. Velardi, Sapienza University of Rome, Rome, Italy; emails: {prenkaj, velardi}@di.uniroma1.it; G. Stilo, University of L'Aquila, L'Aquila, Italy; email: giovanni.stilo@univaq.it; D. Distante and S. Faralli, University of Rome Unitelma Sapienza, Rome, Italy; emails: {damiano.distante, stefano.faralli}@unitelmasapienza.it.

0360-0300/2020/05-ART57 $15.00
https://doi.org/10.1145/3388792

1 INTRODUCTION

Student Dropout Prediction (SDP) is a research topic in the multidisciplinary field of Learning Analytics (LA) [40]. More precisely, it belongs to the area of Educational Data Mining (EDM) (see References [5, 27, 60] for an overview of this field). The specific objective of SDP is to analyse student dropout in distance learning environments by modelling student behaviour when interacting with e-learning platforms.

SDP should be treated with significant importance because of the vast amount of dropout rate from e-learning environments. Hence, it represents a new research line in online learning in general. Although online education systems started in mid-1990,[1] academia has posed little attention to the difficulties that "online students" experience during their studies in these institutions. The recent diffusion of online courses (especially Massive Open Online Courses—MOOCs), with their enormous number of enrolled students—out of which only a fraction completes their studies successfully—has led to increased attention to this problem. As a consequence, a growing number of online institutions have commenced considering the adoption of automated systems to predict their students' dropout decision. Automated strategies, in turn, have boosted the attention of researchers, particularly those in the machine learning area.

In this article, we present a comprehensive survey of recent advances in SDP, from the perspective of machine learning. In the rest of this section, we discuss the importance of SDP and clarify the differences between our survey and previous ones. Finally, we highlight the main contributions of our work.

1.1 Reasons for Caring about Student Dropout Prediction in Online Education

As reported in Reference [7], education—in particular online education—is one of the richest industries in the world.[2] However, despite the benefits of distance learning courses, institutions have a growing concern for low retention rates and, in general, low certification/graduation rates of these kinds of degrees. Students enrolled in online degree programs have a higher chance of dropping out than those attending a conventional classroom environment [12, 21, 23, 26, 35]. According to Reference [67], 40%–80% of online students drop out from online classes, bringing their retention rate to approximately 10%–20% lower than that of traditional universities [34]. Moreover, students can leave the course at any time without notice and further repercussions. Students can drop out at any time during the course. Therefore, it is of paramount interest for the institutions, students, and faculty members to find more efficient methodologies to mitigate the dropout phenomenon in e-learning environments [69]. From an institutional perspective, SDP strategies can lead to a substantial increase in retention and completion rates. Knowing which students are likely to abandon their studies helps distance learning institutions to develop intervention strategies to provide individually tailored support. Since dropouts cause significant economic waste, online universities have a clear interest in investing in this type of predictive action. Besides, from a prestige point-of-view, institutions who exhibit higher graduation rates—or higher retention rates—attract a higher number of students. It has been shown[3] that online institutions with the highest graduation rates also obtain a higher number of enrolled students with respect to (henceforth w.r.t.) the average enrollment number[4,5] in all tertiary education institutions in the US.

[1] https://www.uoc.edu/portal/en/index.html.

[2] https://www.forbes.com/pictures/feki45efjgh/no-12-other-schools-and-/.

[3] https://www.onlineschoolscenter.com/30-online-schools-highest-graduation-rates/.

[4] https://www.usnews.com/education/best-colleges/articles/2019-02-15/how-many-universities-are-in-the-us-and-why-that-number-is-changing.

[5] https://nces.ed.gov/fastfacts/display.asp?id=372.

Table 1. Qualitative Comparison with Other Surveys in the Literature, According to the Coverage of Relevant Dimensions Reflecting the Scope and Workflow of SDP Systems

		Field of study			Gathered Data			Student Modelling		Methods			Evaluation
		Education	Psychology	Computer Science	Polls	MOOCs	e-courses	Plain modelisation	Sequence labelling	Analytic examination	Classic learning	Deep learning	Metrics
Surveys	[47]	✓	✓	–	✓	–	–	–	–	–	–	–	–
	[77]	✓	–	–	✓	–	–	–	–	✓	–	–	–
	[73]	✓	–	–	✓	–	–	–	–	✓	–	–	–
	[58]	✓	✓	–	✓	–	–	–	–	–	–	–	–
	[49]	–	–	✓	–	–	–	–	–	–	✓	–	–
	[18]	–	–	✓	–	✓	–	–	–	–	✓	✓	–
	[28]	–	–	✓	–	✓	–	✓	✓	–	✓	✓	✓
	[68]	–	–	✓	–	–	–	–	–	–	✓	–	✓
	[this]	–	–	✓	–	✓	✓	✓	✓	✓	✓	✓	✓

In addition to its social and economic impact, SDP represents a new and interdisciplinary research problem involving both social and computer science. SDP has recently witnessed an increase in its contribution to the academia. The problem is particularly challenging, since student activities within an online learning platform are multiple and of a variegated nature. They are sequential when students engage in activities related to single courses. They are parallel, since students usually attend more than one course. They are interactive, because students benefit from participating in interactive activities through course forums and social networks. As a consequence, finding a model that suitably accounts for these multifaceted activities, as well as understanding their mutual influence in determining a student's performance, represents an open problem. Although the recent availability of high-performing deep machine learning methods offers the possibility of integrating sequential and parallel data, coping with the full complexity of student modelling in distance learning environments is still to be explored.

1.2 Differences between This Survey and Former Ones

SDP is still in its early stages, since significant attention towards online education has risen sharply in the past decade. Very few reviews provide a systematic summarisation of existing works and current progress within this particular domain.

One of the main differences between the present survey and previous ones is that we organise existing studies according to five dimensions, reflecting the scope and typical workflow of SDP: *Field of Study*, which affects the perspective and objectives of the study; type of *Gathered Data* used to analyse the problem; *Student Modelling* strategies employed to process the raw data; *Methods* to model and solve the SDP problem; and the adopted *Evaluation* measures. As argued in this section, we believe that these dimensions provide a better framework to categorise existing SDP studies. In the following, we describe each dimension in more detail and we show, as summarised in Table 1, that other surveys in the literature provide less systematic coverage of these aspects.

(1) **Field of Study**: It reflects the viewpoint from which research takes into consideration the SDP problem. In the literature, researchers tackle SDP according to either analytic or computational viewpoints. The former belongs to the fields of Psychology and Education, whereas the latter contributes to the domains of automated Data Analytics and Machine Learning. This aspect influences how research addresses the SDP problem. Surveys in Psychology and Education consider studies based on questionnaires and exploiting analytic methods to highlight the different dropout causes. Although in our survey we include some of the most cited analytical approaches [47, 58, 73, 77], we introduce the four dimensions to reflect the main steps of a computational approach to SDP.

(2) **Gathered Data**: Table 1 shows a sharp distinction among surveys in the fields of Psychology and Education and those in the field of Computer Science. The surveys in the former area [47, 58, 73, 77] approach the dropout phenomenon based on data collected from carefully designed polls and questionnaires, and use analytical methods to identify the main causes of dropout. The latter [18, 28, 48, 68] are concerned with studies where data are extracted primarily from e-learning platforms (MOOCs or e-courses). There is a substantial difference between SDP in a MOOC and that in online degrees. Two MOOCs do not interfere with each other and do not influence the dropout rates of one another. On the contrary, SDP becomes rather arduous for online degrees, because a course may have prerequisites taught in other courses. Additionally, SDP is harder in online degrees, because the decision to abandon can be influenced in a complex way by the student's experience in multiple parallel courses. Hence, the adopted predictive strategy has to cope with student dropout from the entire online degree rather than a specific course or module as in the MOOC scenario. Moreover, MOOCs are very short: their duration ranges from six to twelve weeks.[6] Also, they allow people to study a few hours a week and still be able to complete the course. For these reasons, the student *e-tivities* (defined as the actions performed in an e-learning platform) produce highly condensed time-series. Slight inactivity detected (i.e., gap in the time-series) for a certain student might be an indication of her propensity towards dropping out. Contrarily, online degree courses have a slower pace, implying sparser time-series for its enrolled students. Disengagement patterns of students might produce many false positives when resolving the SDP problem. Therefore, researchers must use novel and complex prediction strategies in online course scenarios. As highlighted in Table 1, most authors restrict their study to MOOCs. Our survey addresses online courses in general.

(3) **Student Modelling:** Works in the literature exploit student e-tivities to model the SDP problem as either a sequence labelling or a binary classification problem. In *sequence labelling*, the adopted model is built upon e-tivity time-series or evolving networks. This modelling scheme allows step-by-step monitoring of the student behaviour, henceforth contributing to the capability of the adopted model to identify real-time dropout cases. With *plain modelisation* instead, static data (e.g., student demographics and previous education information) are used to generate the input features for the chosen prediction strategy. This implies a flat approach during the prediction phase—i.e., to perform binary classification over time-independent information. These two alternative schemes considerably affect the capability to identify students prone to dropping out. However, available surveys do not consider this as a relevant dimension, except Reference [28], which considers the student modelling dimension in an alternative way. The authors organise the contributions in the literature based on the choice of the input features that, according

[6] http://desarrolloweb.dlsi.ua.es/moocs/structure-of-a-mooc.

to them, identify the building blocks of the predictive models. Therefore, they classify algorithms based on activities, demography, and discussion forums. The classification dimension that we propose is more general than providing a dense list of feature-based models. Nevertheless, we acknowledge Reference [28] to consider modelisation strategy as a relevant dimension in their survey.

(4) **Methods:** Research in the literature has used a plethora of methods to solve the SDP problem. In our survey, we group them into three classes.
   (a) *Analytic Examination* is based on "classical" statistical analysis of the data. The majority of these works perform correlation analysis to indicate whether some specific attribute is positively or negatively correlated with the dropout outcome. The surveys provided in References [73, 77] present detailed statistical studies to clarify the influence of several factors over the dropout decision in distance learning courses. However, these studies do not pertain to the computational field.
   (b) *Classic Machine Learning* encompasses off-the-shelf machine learning algorithms and their application to address the SDP problem.
   (c) *Deep Learning* includes models based on complex neural networks able to detect hidden patterns in the original data. Deep neural networks have shown to outperform classic prediction algorithms, especially when the quantity of data rises.

   Among the surveys in the same domain as ours, References [48, 68] give a general description of the prediction strategies used in the literature, but they narrow their focus to classic machine learning techniques (e.g., decision trees, naive Bayes, SVM, and so on). Differently, Dalipi et al. [18] enlist deep learning architectures together with a summary of open issues and challenges. With respect to Dalipi et al. [18], our survey of the adopted computational approaches goes well beyond a simple list, providing the relevant algorithmic details of the presented methodologies.

(5) **Evaluation:** To correctly evaluate the performances of a given model, several metrics are commonly used, such as accuracy, precision and recall, area under the ROC curve (AUC), and others. Since the data indicate a skewed dropout distribution—i.e., there are more dropouts than persisting students in online courses—proper evaluation metrics should be used to reflect the performances of the employed prediction strategy. Thus, a detailed examination of the critical issues of alternative metrics needs serious attention. Among all the analysed surveys, only References [18, 28] enlist the possible evaluation metrics used in the literature to assess the performances of the selected prediction strategy. Nevertheless, only Gardner and Brooks [28] delve, as in this survey, into discussing the utility of each metric.

With reference to Table 1, the main limitations of existing computer science surveys on SDP are the following:

(1) Many surveys do not consider diverse e-learning environments but limit themselves to MOOCs, which have different peculiarities compared to online degrees and can be effectively analysed with simpler models.
(2) The rapid progress in devising predictive algorithms (e.g., deep learning methods) able to cope with complex systems such as online learning environments makes previous surveys outdated.
(3) Existing surveys provide a summary of types of algorithms, datasets, and features used by simply enumerating these aspects without providing a critical analysis of limitations and open challenges in the field.

1.3 Contributions of This Survey

The goal of this survey is to provide an in-depth and up-to-date study of the research in the area of SDP under the main but not unique perspective of machine learning.

As detailed in the previous section, our work is broader in scope and more systematic than other surveys available in the literature. In summary, the main contributions of this survey are the following:

(1) We propose a classification of the existing literature according to several dimensions that help the reader to easily classify alternative methodologies.
(2) We extend our analysis to both MOOC and e-courses, highlighting the additional complexity and challenges of SDP in the context of e-courses.
(3) We introduce a conceptual framework and a formal notation to organise and describe in a uniform way alternative strategies adopted by researchers in the field, thus facilitating a comparison among different models.
(4) To the best of our knowledge, we are the first to provide the reader with a summary of strengths and weaknesses of the employed methods, most promising research directions, and a guide to solving the SDP problem in practice, beginning from the choice of the raw data modelisation, prediction strategy, and evaluation metrics.
(5) Finally, we consider two additional, but quite relevant, issues such as the selection of available datasets and consideration of privacy concerns.

We organise the rest of this survey as follows: Section 2 briefly describes the relevant notation in the SDP domain and introduces a taxonomy of input modelisation and prediction strategies, which we use to organise existing literature. Section 3 gives details on how to model the student raw data into input features according to techniques exploited in the literature. Section 4 describes classic and deep learning strategies used for SDP. Section 5 provides a critical summary of the most widely used evaluation metrics to test the performances of proposed models and available datasets. Section 6 summarises open issues and promising directions in the research on SDP, and Section 7 presents our concluding remarks. Additionally, interested readers can refer to the online Supplementary Material for further details.

2 A FORMAL REPRESENTATION OF THE SDP PROBLEM

The literature addressing the SDP problem introduces several fundamental concepts that we formalise below for future research. Moreover, we present a conceptual taxonomy of adopted input modelling and prediction strategies.

2.1 Background Concepts and Definitions of SDP

As summarised in Section 1.2, researchers model students' e-tivities adopting either a recurrent (sequence labelling) or plain modelisation scheme. While a conventional classification problem assumes that the inputs are *independent and identically distributed (i.i.d.)*, the input features in a sequence labelling task are dependent. Without loss of generality, we here introduce time-aware definitions of the relevant entities and e-tivities within e-courses.

An online degree course generally contains one or more courses. We indicate with C the set of these courses.[7] A course $c \in C$ is composed of k_c sequential phases c_i $(1 \leq i \leq k_c)$ that begins at time $b(c_i)$ and finishes at time $f(c_i)$.

[7] The cardinality of the set of courses is equal to one when taking into consideration a single MOOC or e-course.

Table 2. Notation Used throughout the Survey with Its Corresponding Meaning

Relation	Notation	Meaning
Course	C	The set of all courses in an online degree
	c	A single course belonging to C
	c_i	The ith phase of course c ($1 \leq i \leq k$)
	$b(c_i)$	The time when phase c_i begins
	$f(c_i)$	The time when phase c_i finishes
Student	$\mathcal{S}$	The set of all enrolled students in an online degree
	$\mathcal{S}_c$	The set of students enrolled in course c
	s	A single student belonging to $\mathcal{S}_c$
E-tivity	$\mathcal{E}$	The set of all possible e-tivities
	$e_{c_i}^{t_j}$	A single e-tivity performed at time t_j in course phase c_i
	$w_s(t_b, t_f, c_i)$	The set of e-tivities that s performs in phase c_i in the time interval $[t_b, t_f]$
	$w_s(c_i)$	The set of e-tivities that s performs in phase c_i in the time interval $[b(c_i), f(c_i)]$
	$w_s(t_b, t_f, c)$	The set of e-tivities that s performs in course c in the time interval $[t_b, t_f]$

Upon enrollment, the students have to complete all the courses in an online degree to graduate.[8] Hence, $\mathcal{S}$ represents the set of all students in an online degree, whereas $\mathcal{S}_c \subseteq \mathcal{S}$ is the set[9] of those students that attend course c.

Interacting with the e-platform of a course c consists of performing some actions from the set of all permissible ones $\mathcal{E}$. Since e-tivities constitute the basis of SDP, we annotate the student $s \in \mathcal{S}_c$ as a temporal sequence of length m of its e-tivities throughout the i phases of course c: i.e., $s = \{e_{c_i}^{t_1}, \ldots, e_{c_i}^{t_m}\}$ where $e_{c_i}^{t_j}$ $1 \leq j \leq m$ is the e-tivity $e \in \mathcal{E}$ performed in the phase c_i at time t_j. Finally, for readability purposes, we propose an aggregation function of e-tivities, $w_s(t_b, t_f, c_i) = \bigcup_{t_b \leq t_j \leq t_f} e_{c_i}^{t_j}$, that a student s performs in phase c_i within a specific time interval $[t_b, t_f]$. According to this formulation, we can express all the e-tivities that s has performed in phase c_i as $w_s(c_i) = w_s(b(c_i), f(c_i), c_i)$. Furthermore, we extend the first formula to describe e-tivities of student s in the entire course: i.e., $w_s(t_b, t_f, c) = \bigcup_{c_i} w_s(t_b, t_f, c_i)$. This is useful when collecting the e-tivities that stretch to more than one course phase. Table 2 summarises the used notation and its corresponding meaning.

In the literature, SDP is a binary classification problem. In other words, zero indicates a persisting student and one represents a dropout student. We provide a formulation of the dropout problem both in a recurrent (sequential labelling) and plain (without temporal dependence) scenario. We discuss the benefits and possible drawbacks of each formulation.

(1) *Plain dropout formulation*: The student-platform interactions are independent in time, while in reality, student behaviour may change over time. Given the e-tivities $w_s(t_b, t_f, c)$ that $s \in \mathcal{S}_c$ performs, plain dropout determines whether s drops out or not regardless of how the e-tivities are sequenced in $[t_b, t_f]$.

[8] If we consider the scenarios of MOOCs and single online courses, then students have to finish a single course to obtain a completion certificate.

[9] $\mathcal{S}_c = \mathcal{S}$ in MOOCs and single e-courses.

(2) *Recurrent dropout formulation*: The recurrent dropout formulation consists of using information from previous course phases to decide a student's dropout status. When transiting from phase c_{i-1} to c_i, we share hidden information to efficiently monitor student behaviours in course c in real-time. Therefore, the label of s in phase c_i depends on the activities performed in the preceding phases c_{i-r}, $r \in \{1, \ldots, p \leq i-1\}$. The length p of the window used to consider previous phases determines how much of a student's past behaviour we want to consider. All this information passes as a hidden state to c_i.

The works in the literature adopt the previous two dropout formulations according to two different student modelisation strategies. Below, we provide their formal definitions.

Definition 2.1. Plain dropout: Given a time interval $[t_b, t_f]$, a student s is a dropout from course c if they do not survive until the end of the time span. In other words, s is a dropout if $\exists t_u \in [t_b, t_f]$ s.t. $w_s(t_u, t_f, c) = \emptyset$.

According to Definition 2.1, $w_s(t_b, t_f, c)$ returns a portion of the e-tivities of student s in course c. One can monitor the engagement of student s during the entire time span $[t_b, t_f]$. Nevertheless, this definition does not consider a phasal view of the course. Hence, the dropout condition does not depend on the information passed from phase c_{i-1} to c_i, as in the recurrent formulation. The dropout label is based on all the e-tivities $w_s(t_u, t_f, c)$ that s does after a certain point in time t_u.

Fei and Yeung [24] introduce three recurrent dropout definitions, which we now reformulate according to the notation previously introduced. While originally employed in MOOC contexts, we adapt them to other e-learning scenarios. Moreover, we generalise them by considering a phasic course unit instead of the weekly time-frame adopted in Reference [24]. Recall that k sequential phases $c_1, \ldots, c_k$ form a course c. The following definitions assume that the e-tivities of s are grouped according to $[b(c_i), f(c_i)]$ $\forall i \in [1, k]$ in course c.

Definition 2.2. Participation in the final course phase: A student s is a dropout if they do not persist until the last phase, c_k, of course c; otherwise, they are a persister. In other words, s is a dropout if $w_s(c_k) = \emptyset$.

Definition 2.3. Last phase of engagement: A student s is a dropout if they do not produce any e-tivities after the current phase c_i: i.e., s is a dropout if $w_s(c_i) \neq \emptyset \ \wedge \ \forall j \in [i+1, k] \ w_s(c_j) = \emptyset$. Notice that this definition is a generalisation of the previous one: i.e., we emulate Definition 2.2 by setting $i = k - 1$.

Definition 2.4. Participation in the next phase: A student s is a dropout if they do not have e-tivities in the next phase, c_{i+1}. Hence, s is a dropout if $w_s(c_{i+1}) = \emptyset \ \wedge \ i \neq k$.

Definitions 2.2 and 2.3 predict the final condition of the students. Contrarily, Definition 2.4 does not refer to the dropout status for the entire course c. Instead, it has a phasic view of the dropout status.[10] Thus, a student that does not finally dropout but has $w_s(c_{i+1}) = \emptyset$ is a dropout in phase c_i.

Table 3 summarises the advantages and disadvantages of the above definitions. Definition 2.2 disregards any fluctuations in student behaviour during the course phases. The works in the literature use it to design a model that predicts the survivability of students at the end of the course. Definition 2.3 requires only a partial view of the performed e-tivities to foretell the dropout label of student s. But, because it corresponds to the final status of dropout, it does not preemptively identify dropout cases. The last week of engagement might be too late to recover at-risk students. Finally, Definition 2.4 is suitable to perform real-time dropout identification and consistent

[10] For more clarification, check the example in Reference [24].

Table 3. Advantages and Disadvantages of the Three Dropout Definitions

Definition	Pro	Contra
Def. 2.2	- Simple student survivability model	- Final dropout status prediction - No preemptive identification
Def. 2.3	- Prediction using a partial view of student e-tivities	- Final dropout status prediction - No preemptive identification
Def. 2.4	- Ongoing dropout status prediction - Prediction using a partial view of student e-tivities - Real-time prediction - Temporarily inactive student recognition - Monitoring of student behaviour in each phase	- Complex prediction mechanism required

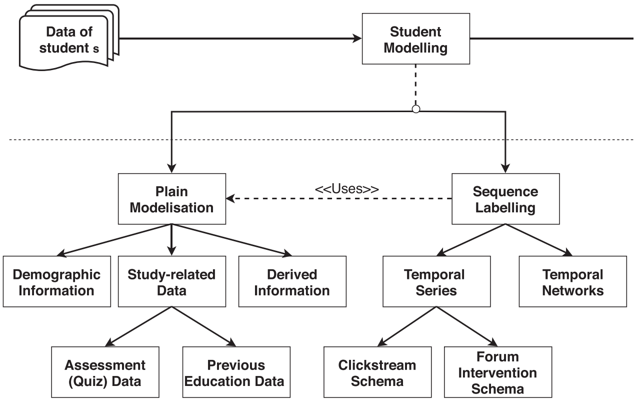

Fig. 1. A taxonomy of student modelling approaches.

monitoring of the behaviour of *s* at each phase. Nevertheless, its adoption requires implementing complex predictive strategies.

2.2 A Taxonomy of SDP Design Choices

When designing a student dropout predictor, we must select a suitable student model and a prediction strategy.

In Figures 1 and 2, we illustrate how these two phases of an SDP strategy have been used to classify the existing SDP literature. Note that the second picture is a continuation of the first one, but we split them for readability purposes. The upper parts of the figures illustrate the pipeline of designing a student dropout predictor. The lower parts are taxonomies of choices in each stage of the pipeline.

Specifically, Figure 1 shows *Plain Modelisation* and *Sequence Labelling*. The former mostly considers demographics, current study-related information, and their data derivations. The latter adopts two different structural approaches that, respectively, exploit temporal series and networks. Additionally, plain modelisation refers to time-invariant characteristics of the raw logs on student e-tivities; meanwhile, sequence labelling considers only time-variant features. Sequence labelling on temporal series is modelled in the form of clickstreams or forum interventions.

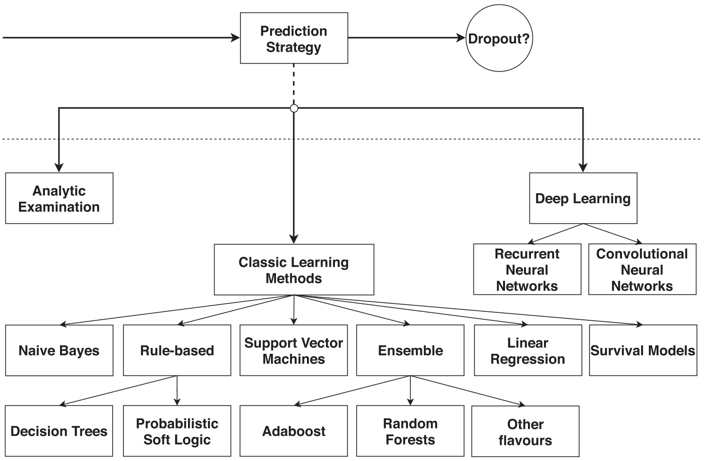

Fig. 2. (continues from Figure 1) A taxonomy of prediction strategy approaches.

Figure 2 illustrates the way we classify the prediction strategies. We have identified three different classes: *Analytic Examination*, *Classic Learning Methods*, and *Deep Learning*. The last two include off-the-shelf machine learning algorithms and complex deep learning strategies, respectively. The literature has over-explored classic machine learning, thus touching upon each class of algorithms depicted in the figure. Taking into consideration more recent developments of deep learning methods in SDP, current research focuses on adaptations of RNNs and CNNs. For completeness purposes, we include *Analytic Examination* that adopts statistical measures to determine the status of a student.

In Sections 3 and 4, we classify the SDP literature according to the taxonomy of Figures 1 and 2. We first explain the modelisation procedures formally, and then we enlist the literature papers. Because *Sequence Labelling* is more widely adopted, we study it more thoroughly. The same line of exposition applies to the prediction strategy.

3 STUDENT MODELLING

In this section, we survey student modelling features following the taxonomy introduced in Figure 1.[11] We begin by analyzing plain modelisation methodologies (Section 3.1) and, next, we consider the vaster class of sequence labelling methods in Sections 3.2 and 3.3.

3.1 Plain Modelisation

Plain modelisation exploits student demographic information and data related to the scores of tests/homework that students achieve in each course. Figure 3 shows a conceptual workflow for transforming raw data into flattened input features. The two vectors in the upper-left corner represent the demographic and "previous study" (e.g., pre-university data concerning previous education) information, respectively. The matrices depict the interactions of a student in phase $c_i \forall i \in [1, k]$. Data obtained from the e-tivity matrix concatenation of all phases are flattened into the vector f_c containing grades, exam failures, and various cumulative statistical features (steps 1.c and 2.a). Finally, all the vectors are merged (steps 3.a and 3.b) into the input feature vector used for the prediction strategy (step 4).

[11] For quicker reference, we also provide, in Section 2 of the Supplementary Material, a tabular version of the most popular student features considered in the literature, along with a list of papers adopting that particular feature.

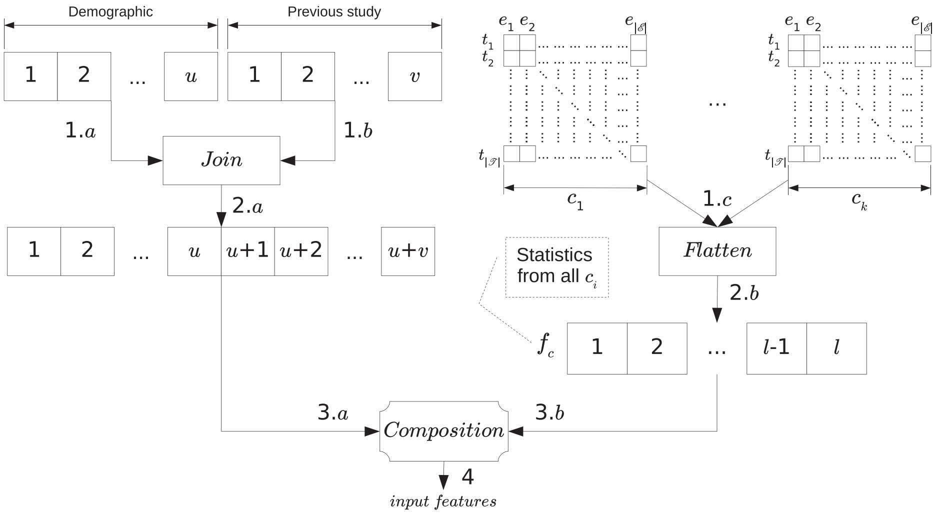

Fig. 3. Conceptual workflow illustrating the transformation of raw data in a plain modelisation format for student s. Time-dependent e-tivities related to each course phase c_i (the matrices in the upper left part of the figure) are flattened into a vector of study-related features f_c (i.e., statistics derived from all phases). These features are then composed with demographic information and other information related to previous studies (upper left part of the figure). The result is a flat feature vector.

SDP studies based on Plain Modelisation. The authors in References [42, 45, 75] use student demography and academic performance. Xenos et al. [75] also exploit student interviews and ad hoc questionnaires to identify other factors that lead to dropout. Kotsiantis et al. [44] extract only demographic data from the course of Informatics in the Hellenic Open University (HOU) to identify struggling students before the midterm of the academic year. The authors in Reference [43] use assessment scores to classify dropout HOU students. They mark a student as failing if they have not submitted one of the two required homework assignments. Lykourentzou et al. [53] consider time-varying and invariant attributes by examining demographic and previous study information. Nevertheless, they flatten the time-varying data in the raw logs to derive information such as grades, Grade Point Average (GPA), and the number of activities per course phase. Dekker et al. [20] look at pre-university data. Berens et al. [8] build a self-adjusting early detection system targeted for German universities. They employ personal, previous education, and current matriculation information to identify dropouts as soon as the first semester. Kovačić [46] performs an exhaustive study of the dropout phenomenon by exploring socio-demographic and study-environment variables. Similarly, Al-Radaideh et al. [1] use previous education information and some demographic data to shape their model. Jointly with the student responses to ad hoc questionnaires for several HarvardX courses, Robinson et al. [65] use general student demographic statistics.

Manhães et al. [54] propose WAVE, a monitoring architecture for student academic progress. Alongside other student-performance metrics, WAVE exploits exam information (e.g., GPA) to determine whether a student, attending the first semester, persists in the second one. Although Hu et al. [37] keep track of time-dependent information, they flatten the e-tivities into plain data for statistical analysis. Similarly, Gaudioso et al. [29] combine demographic data with information derived from the student interaction with the e-platform (e.g., total number of visits, number of sessions, percentage use of each of the learning resources). Wolff et al. [74] study the Virtual Learning Environment (VLE) clicks for each student enrolled in the Open University (OU). The authors enrich their dataset with tutor assessment grades, final exam marks, and demographic information.

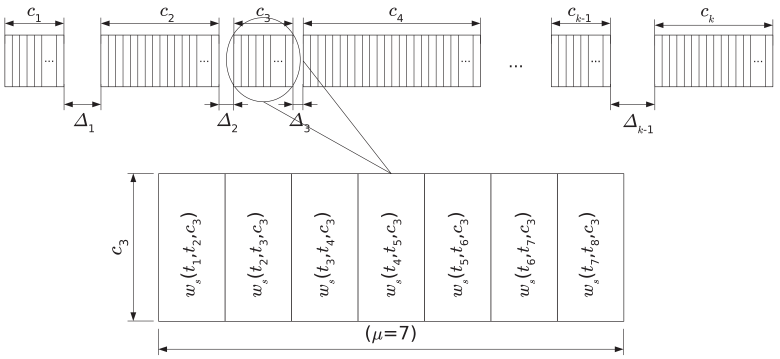

Fig. 4. Conceptual modelling of a sequential scheme of course c, with an example of consecutive and same-width time-slices in phase c_3. Note that periods of non-activity (the Δ_i) may have a different length.

Amnueypornsakul et al. [3] use quiz- and activity-related information. For each week, they transform the activity-related data of a student in a string consisting of e-tivities, identified with a letter. Gray and Perkins [31] model a student as a set of η Boolean observations indicating the attendance of a student at each point in time: i.e., $z = \{z_1, \ldots, z_\eta\}$. They utilise an ad hoc function that combines attendances and non-attendances of a student from the η observations to calculate engagement behaviour of a student: i.e., $BEM_\eta = \sum_{i=1}^{\eta}(-1)^{1-z_i}$. They choose the BEM scores for the first three weeks as well as demographic features to aid the information gain of their model. Finally, Feng et al. [25] extract student behaviour patterns. They engender statistically aggregated data from student clickstreams while simultaneously modelling user and course contextual information. They refer to demographic data for the student context; whereas, they represent the course context with its category (e.g., math, physics, and arts).

Finally, it is worth mentioning a number of studies that, although concerned with physical rather than online degrees, employ a modelisation strategy similar to those adopted for online courses/degrees [2, 16]. Ameri et al. [2] use demographics and pre-enrollment information. Additionally, they collect semester-wise attributes (e.g., GPA, credits passed/failed/dropped) from study-related data such as exams and other curricular activities. Chen et al. [16] exploit the same information as the previous work, but they use a lower quantity of features as input to their forecasting method (i.e., 13 [16] against 31 [2]).

3.2 Sequence Labelling with Temporal Series

Sequence labelling consists of shaping the raw data logs into a discrete temporal series that consists of the student e-tivities. A discrete-time temporal series [10] is a set of observations each one being recorded at a specific time t coming from a discrete set $\mathcal{T}$ of times. Essentially, we divide each phase c_i into μ consecutive temporal slices generally of the same duration where the end of the previous slice coincides with the beginning of the current one. Figure 4 illustrates the segmentation of the phases of course c in time-slices. Notice that there can be gaps between two consecutive phases c_i and c_{i+1}, here denoted as inter-phase delay Δ_i. Additionally, we portray the course phases with different duration to represent heterogeneity among them. In this example, we assume that phase c_3 has seven consecutive time-slices ($\mu = 7$).

Notice that the model presented generalises over what the current research studies do. In the literature, the courses have the same duration and no inter-phasic gaps (which are, instead, a useful indicator of future dropouts). Moreover, researchers adopt a time window of the same length of the duration of c_i. Referring to the notation of the previous example, t_b of the first slice and t_f of the last one correspond, respectively, to $b(c_i)$ and $f(c_i)$.

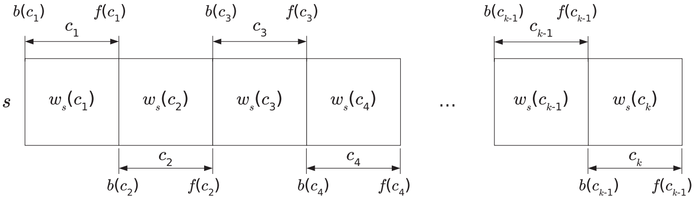

Fig. 5. A discrete-time temporal series where student s performs $|w_s(c_i)|$ e-tivities in phase c_i.

Figure 5 depicts the model. The squares represent the course phases, c_i. They contain the e-tivities of student s, $w_s(c_i)$. Notice that they can also include aggregation features calculated upon the e-tivities instead of just $w_s(c_i)$. Differently from the previous example, here, we exploit the whole engagement patterns for s in a phase to engender the dropout propensity of s.

We have identified two main approaches to temporal modelisation: i.e., *clickstream-based* and *forum intervention-based*. However, a minority of works use other approaches that derive from the previous ones.

(1) A clickstream-based schema relies on activities expressed in terms of clicking resources such as page-view and video-view logs. The click-data of a student s in phase c_i constitutes the collection of the same type clicks according to an aggregation function. For instance, the average number of lecture views might be one of the selected features for the predictive model. Notice that clickstreams do not include forum discussions such as commenting or starting threads nor assignment/project grades. Nevertheless, they incorporate forum click-events, such as liking or disliking forum comments and data related to assignment submissions.

(2) On the contrary, a forum intervention-based schema depends on information gathered from student discussions with their peers/tutors. Data used in this schema are composed of both structural (e.g., thread starters, and thread comments and replies) and temporal information. Generally, time series from the forum-derived data of s include statistics or NLP-derived metrics (e.g., total number of responses, text length/density) in each course phase.

SDP studies based on Clickstream-based schema. Kloft et al. [41] exploit page-view and video-view logs by performing a weekly aggregation (e.g., summing or averaging) of the identified features. Their strategy relies on extracting numerical features by capturing the user activity levels (e.g., number of requests) as well as some technical features coming from the used browser information. Ramesh et al. [64] use lecture views and quiz answers to model two forms of student engagement (i.e., active and passive) derived from user clicks. According to the authors, active engagement occurs when a student s attends lectures and submits quizzes/assignments. Meanwhile, passive engagement occurs when s scarcely attends lessons and does not submit the assigned homework. Hence, by using student commitment as a latent variable for their predictive model, the authors use student submissions to gauge the student survival rate. Similarly, Nagrecha et al. [59] use clickstream information from video viewing patterns. Each of the data characteristics included in their schema captures latent factors in the consumption of video contents.

Together with data derived from user clickstreams, the authors in References [33, 71] utilise homework submissions and grades. In detail, He et al. [33] compute statistics at the end of a particular phase (e.g., average attempts on each assignment done by week i) to train their weekly-based model. Taylor et al. [71] derive input statistics in the same way as presented in Reference [33], but they use the notions of *lag* and *lead* to train their model. Fei and Yeung [24] exploit lecture views, downloads, and quiz attempts. Li et al. [50] consider behavioural patterns such as accessing, viewing, and closing course resources, according to their time of occurrence in a certain course phase. Qiu et al. [61] include a list of statistical features such as the number of chapters a student browses and the total time they spend on watching videos. The authors in Reference [14] extract characteristics from weekly learning behaviour records.

Wang et al. [72] distinguish e-tivities coming from clickstream data according to click-source and click-type in a course phase. They transform these data into one-hot vectors and bit-wise sum the vectors belonging to the same time unit, feeding their result into the prediction model. Haiyang et al. [32] generate three time-series based on forum, video-lecture, and textual resource clicks. Afterwards, they sum up the numbers of clicks from each module on each day and align the total clicks of students. Ding et al. [22] use clickstream data corresponding to navigation and video lessons such as playing, pausing, and stopping.

Finally, Qiu et al. [62] model a temporal series as a one-dimensional grid data sampled at fixed time intervals. They transform the raw input according to a time window algorithm (DTTW), obtaining a time-behaviour dimensional matrix. The time dimension describes the time sequence relationship; whereas, the behaviour dimension describes the various behaviours in the corresponding period.

SDP studies based on Forum intervention-based schema. Models based on forum-derived features are suitable for distinguishing hugely engaging students from those that exhibit a less determined engagement with the course at hand. However, according to Reference [30], since only a tiny percentage of students engages in e-learning fora, the literature lacks studies that purely base their strategies on forum intervention. Because temporal series cannot model the structural patterns of forum discussions, research exploits forum-derived features to enrich clickstream-based models. Below, we report some studies based on clickstreams that use forum features to boost their performances.

Ramesh et al. [64] use linguistic characteristics (content sentiment) and structural typology to capture forum information. They include the subjectivity and polarity of each thread post to their final feature vector. Fei and Yeung [24] investigate forum views, thread initiations, posts, and comments. Finally, Qiu et al. [61] derive features from forum activities. They also consider the homophily correlation of a student expressed as the number of replies received from well-performing students. In this way, the students who received the replies are less prone to drop out.

3.3 Sequence Labelling with Temporal Networks

As stated in Reference [71], commenting and replying to threads may suggest a lower dropout susceptibility. To better model student engagement in fora, a few studies have exploited temporal networks. They capture both structural and temporal information of students that interact with their peers. There are three types of interactions in the forum:

- *Thread initialisation* characterises the action of creating a new argument. A thread θ can have $[0, n]$ comments. We use the notation of Reference [79] to denote the original poster of θ with $OP(\theta)$.
- *Comments* are posts directly correlated to the original thread message. A comment can have $[0, n]$ replies.
- *Reply messages* are responses to the comment messages. A reply message can have other nested replies.

We specialise the temporal-network definitions in References [6, 56] for the student interactions in fora to cope with the description of the methodologies that use this particular form of

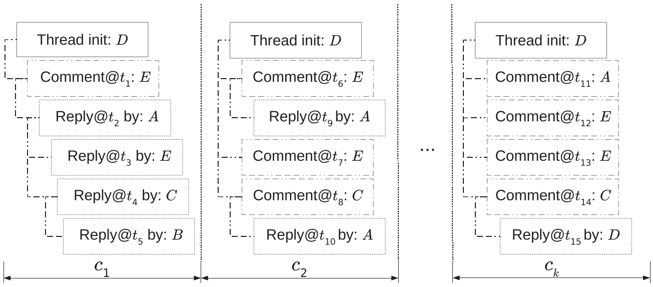

Fig. 6. Hierarchical view of the forum discussion in each course phase c_i. Here, we assume that only one thread θ has been opened and we trace the student interactions throughout the phases. The indentations denote the interactions between two students. In other words, the student in the innermost layer interacts with that in the layer immediately above it. We distinguish between the three interaction types of students. A whole-lined rectangle represents a thread starter, a dashed and dotted one a commenter, and a dashed one a replier.

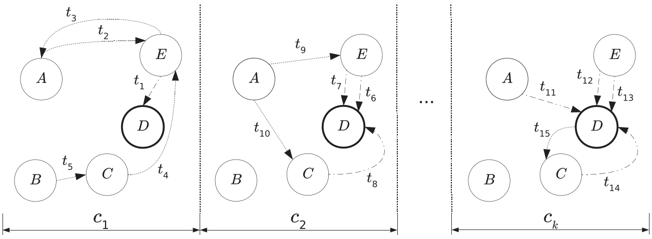

Fig. 7. The temporal multigraph G_{c_i} derived from the previous hierarchical representation of student interactions in each phase c_i. Notice that, in each c_i, we identify the thread starter with a bold circle. As before, we distinguish comment edges (dashed and dotted) from reply edges (dashed) for illustration purposes.

modelisation. We denote with $\Theta_c = \{\theta^1, \theta^2, .., \theta^x\}$ the set of all threads in course c. Recall that $\mathcal{S}_c$ is the set of students enrolled in course c. A forum-based social network for a thread $\theta^r \in \Theta_c$ during a time interval $[t_b, t_f]$ is a labelled multidigraph (i.e., directed graph with parallel edges [6]) $\mathcal{M}^{\theta^r}_{[t_b,t_f]} = (V, A, \ell_A)$ where $V \subseteq S_c$, A is the set of arcs (s', s'') derived from the following interactions:

- s' comments to thread θ^r that s'' has generated, or
- s' replies to a comment of s'' in thread θ^r ,

and $\ell_A : A \rightarrow \mathcal{T}$ is a function that maps the arcs (s', s'') to its relative interaction timestamp t. Clearly the social network interactions of phase c_i is $\mathcal{M}_{c_i} = \bigcup_{\theta^r \in \Theta_c} \mathcal{M}^{\theta^r}_{[b(c_i), f(c_i)]}$ and the interactions related to thread θ^r are captured by $\mathcal{M}^{\theta^r} = \bigcup_{1 \leq i \leq k} \mathcal{M}^{\theta^r}_{[b(c_i), f(c_i)]}$. For completeness purposes, we denote with $\mathcal{M}^{\theta^r}_{c_i} = \mathcal{M}^{\theta^r}_{[b(c_i), f(c_i)]}$.

Figures 6 and 7 illustrate an example of a temporal social network (labelled multidigraph) showing the timed interaction patterns between students. Suppose that the students initiate only a single forum thread in each phase c_i. Figure 6 depicts the hierarchical structural representation of the interaction of the students (A, B, C, D, E). Notice that, for each phase c_i, we depict only the threads opened in c_i. We do not want to trace the evolution of the thread influence in each phase, but rather check the engagement of students in time in different threads. To distinguish the three forum interaction types, we highlight them. We indicate the original poster of the thread with

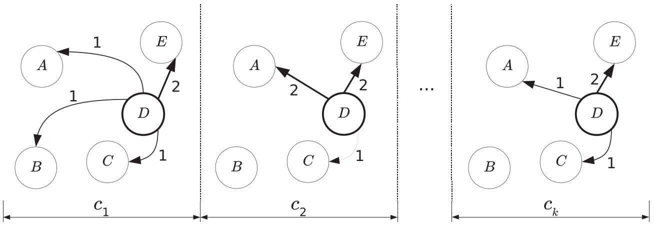

Fig. 8. Temporal network model highlighting the network structure in each course phase c_i according to Yang et al. [76]. Differently from our network formalisation in Figures 6 and 7, only thread starters (bold circled nodes) have outgoing arcs towards the other students. The edge labels indicate the number of times a thread starter has influenced another student into interacting in that particular thread in course phase c_i. Here, we do not distinguish between comment and reply edges. For visualisation purposes, the thickness of the direct edges depends on the number of interactions between its incident nodes.

a whole line, a comment with dashed and dotted lines, and a reply with a dashed line. Figure 7 depicts the hierarchical representation transformed into a labelled multidigraph. The labels in the edges depict the time of interaction between its endpoints. Notice that, by examining the phases $c_1, \ldots, c_k$, we can model the difference in message concentrations in the discussions. For instance, student B is inactive in phase c_2. The intuition here is that, if, e.g., B does not participate in the forum in the successive phases $c_3, \ldots, c_k$, then B is a potential dropout from the whole course. On the contrary, the other students maintain a linear pattern of interaction with one another; hence, they are likely to persevere.

In conclusion, temporal network models make it possible to exploit network-related metrics, such as betweenness and node centrality, to determine the departure decision for each student in each phase. It is furthermore possible to refine the criterion of considering an engaging student in the forum by discriminating her participation in the same thread in different time phases. Accordingly, a student that does not get involved in new thread discussions in the successive course phases can be considered a dropout.

SDP studies based on Temporal Networks. The models based on features derived exclusively from the forum attempt to solve the SDP problem by relying only on peer-to-peer interactions. Yang et al. [76] partition the enrolled students into cohorts according to their registration period. They reveal that students in later cohorts are more isolated, while those from earlier ones continue to interact. The authors model the relationships between thread starters and other students. According to them, the thread initiators have an outward connection to all students that participate in the discussion. Therefore, we need to transpose our network structure in each phase c_i. Furthermore, the authors do not model parallel edges to indicate multiple interactions between students. Instead, they label the edges with the number of interactions between their endpoints. Essentially, the network that the authors generate is equivalent to a transformation from $\mathcal{M}_{c_i}^{\theta^r}$ to a simple directed graph $G_{c_i}^{\theta^r} = (V', E)$ where $E = \{(s, s')|\exists s' \in V \wedge s = OP(\theta^r) \wedge s \neq s' \wedge deg_{out}^{\mathcal{M}}(s) \neq 0\}$ where $deg_{out}^{\mathcal{M}}(s)$ is the out-degree of s in $\mathcal{M}_{c_i}^{\theta^r}$ and the weighted function $\ell_E : E \rightarrow \mathbb{Z}^+$ depicts the out-degree, in the original multigraph, of the second element in $e \in E$.

Figure 8 illustrates the formalisation adopted in Reference [76], with reference to the hierarchical model introduced in Figure 7. The authors rely on social network analysis metrics, each providing different insights to the student engagement patterns.

Gitinabard et al. [30] consider two types of graph generation strategies. Based on Reference [11], the first type connects the students to all those that have previously participated in the discussion of the same thread. In this way, each user has an outgoing edge towards all the others except them-

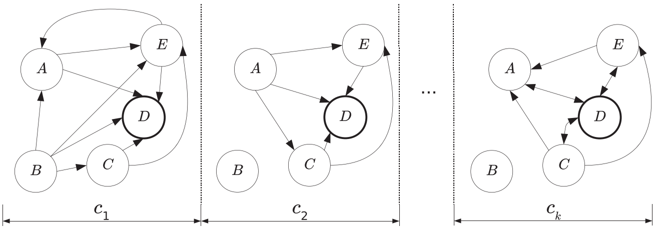

Fig. 9. Temporal graph in each phase c_i according to Brown et al. [11]. We connect each node with all the others that have previously participated in the discussion. For instance, consider phase c_2. According to the hierarchical interaction in Figure 6, first E comments to the initial thread message of D, thus G_{c_2} contains the edge (E, D). Then, E comments twice, but, since we do not model parallel edges, we do not G_{c_2} with arc (E, D) again. The comment of C to D induces the edge (C, D). Last, we model both replies of A to E and C, respectively, as two edges (A, C) and (A, E) the reply of C to the comment of D. The same reasoning stands behind the other phases. The representation of $OP(\theta)$ remains unaltered from the other figures.

selves. This method assumes that every participant in a thread has read all of the preceding posts first and is responding to all of them. The network that the authors generate is equivalent to a transformation from $\mathcal{M}_{c_i}^{\theta^r}$ to a simple directed graph $G_{c_i}^{\theta^r} = (V', E)$ where $E = \{(s, s') | \exists s, s' \in V \wedge s \neq s' \wedge min(\ell_A, s).t \geq min(\ell_A, s').t\}$ and $min(\ell_A, s) = \{t | \exists s' \in V \wedge (s, s') \in A \wedge t = \ell_A((s, s')) \wedge \forall s'' \in V \wedge (s, s'') \in A \rightarrow t \leq \ell_A((s, s''))\}$ returns the first timestamp of interaction of student s in thread θ^r. Figure 9 illustrates the interaction of users with all the previous participants in the thread discussion as described above. In this case, the authors do not model parallel edges nor do they label arcs according to a weight function. Nevertheless, one can extend this network structure by transforming it into a weighted graph.

The second strategy adopted in Gitinabard et al. [30] relies on thread-tracing networks. Based on the observation in Zhu et al. [79] that students use commenting and replies interchangeably, Gitinabard et al. [30] connect all the students in a thread to its original poster (initiator). Notice that the network built in this way has undirected edges, but for consistency purposes with $\mathcal{M}_{c_i}^{\theta^r}$, we employ bidirectional arcs. In fact, this network can be seen as a transformation from multidigraph $\mathcal{M}_{c_i}^{\theta^r}$ to a simple graph $G_{c_i}^{\theta_r} = (V', E)$ where $E = \{(s, s') | \exists s, s' \in V \wedge s \neq s' \wedge (s = OP(\theta^r) \vee s' = OP(\theta^r)) \wedge vis(s, s', \mathcal{M}) = 1\}$ and $vis(s, s', \mathcal{M})$ returns $(0, 1)$ according to a graph search (either breadth or depth search) from s to s' in $\mathcal{M}$. In other words, the transformed graph has bidirectional edges between $OP(\theta^r)$ and all the students that have participated in its discussion. According to this strategy, thread starters are at the centre of a star network. Although this approach traces a thread θ in each c_i to assess how the neighbourhood of $OP(\theta)$ changes, it can be applied as a commitment signal for the students therein. To show how the neighbourhood of the thread starter changes in time, we base the thread discussion on Figure 10. Additionally, Figure 11 depicts $\mathcal{G}_{c_i}^{\theta^r}$ based on the hierarchical interaction of Figure 10. Notice how the star network changes structure in each phase. This is useful to verify per-thread student interactions in different course phases. Finally, the authors in Reference [30] use similar metrics as Yang et al. [76] to distinguish persevering students from dropouts.

4 PREDICTION STRATEGY

In this section, we present a taxonomy of the prediction strategies to address/tackle the SDP problem. As previously seen in Figure 2, we organise the surveyed works according to the methods employed in identifying a dropout student. *Analytic Examination* (refer to Section 4.1) analyses the data and attempts to provide some insight based on pure statistical metrics. *Classic Learning*

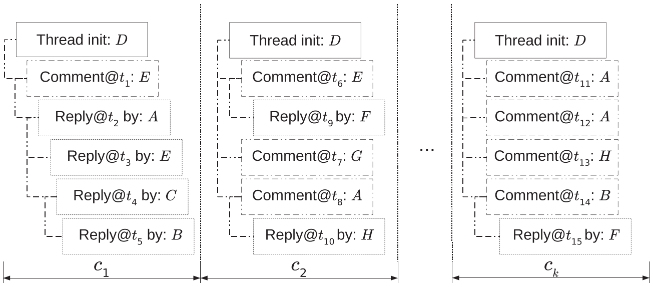

Fig. 10. Interaction of students in the same thread in different phases. The meaning of the rectangles is the same as in Figure 6. To show how the discussion of the same thread differs in the various phases, we assume that the number of students enrolled in c increases (from five to eight in the example).

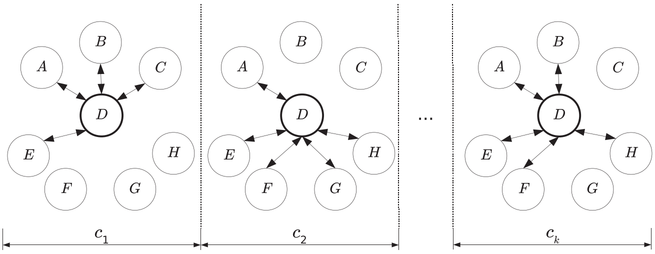

Fig. 11. An example of time-evolving social network for the same thread θ in all the course phases. The temporal graphs in each phase c_i are based on the structure proposed in Reference [79]. Note that the nodes connected to the $OP(\theta)$—bold border node—form a strongly connected component. The cardinality of the strongly connected component with its centre in $OP(\theta^r)$ indicates the popularity of θ^r. We maintain the bidirectional edges for consistency with our modelisation of $\mathcal{M}_{c_i}^{\theta^r}$.

Methods (refer to Section 4.2) exploit a variety of typical machine learning algorithms. *Deep Learning* (refer to Section 4.3) handles more complex strategies based on recurrent and convolutional neural networks. Because of the high performances of deep learning strategies when the data quantity increases,[12] we concentrate on discussing their peculiarities. Readers that need more background on Machine Learning methods adopted in the SDP domain may refer to the Supplementary Material, Section 1.

4.1 Analytic Examination

Studies in this research area have adopted a purely statistical approach to predict the dropout state of a student. Generally, these studies collect data from various sources and then perform a correlation analysis between the extracted feature and the dropout label. Besides correlation studies, they delve into providing insight on the statistical distributions of selected characteristics: e.g., distribution of dropout students according to their cohort or their assignment submissions.

Notice that, because the analytic examination is descriptive and not predictive of future dropouts, its intended use is a better interpretation of the data to devise appropriate intervention strategies. Furthermore, being time-insensitive methods, they do not guarantee stable findings, since patterns of dropout might change in time.

[12] In SDP, we log every student interaction with the course platform. Therefore, we produce an excessive amount of information.

Studies based on Analytic Examination. Xenos et al. [75] concentrate their analysis on a four-year bachelor's degree in Informatics at the Hellenic Open University (HOU). The authors collect CS-related information to identify high-risk students. The data include communication means, computer usage, and previous computer science training. Moreover, they gather student answers via questionnaires and telephone interviews about the perceived reasons of dropout. In this way, they analyse the dropout causes to implement intervention strategies to cope with them. Because HOU permits its students to repeat a failed course module, the authors cope with four categories of students, among whom only two are considered dropouts. Based on these two dropout categories, they perform correlation studies concerning the student profiles generated from the data collected. They give statistically significant information about the correlation of assignment submission and previous education with the dropout decision. They stretch their study further by researching the reasons behind the dropout decision of each student. They interview dropouts and categorise the dropout reasons according to several aspects that include professional and family issues. They conclude that dropout students wrongly estimate the burden of studying while working. Additionally, a minority of them feel that their tutor did not assist them in understanding the course material and completing their assignments.

More recently, Wolff et al. [74] investigated the General Unary Hypotheses Automaton (GUHA), which is a method of automatic generation of hypotheses based on empirical data. It generates hypotheses from the data based on the initial parameters:

- *Confidence*, which delineates the probability that a generated hypothesis correctly classifies the labels;
- *Support*, which is the minimum percentage of the rules to fit the generated rule;
- *Maximum number of antecedents*, which corresponds to the number of literals occurring in the left part of the implication.

The authors show that failing to submit the fourth assessment leads to complete abandonment of the course. However, the results are accurate when applied across different presentations of the same module. In other words, the varying nature of module design means that the rules of GUHA are unlikely to hold in an inter-module scenario.

4.2 Classic Machine Learning Prediction Strategies

The majority of the surveyed works address the SDP problem by relying on off-the-shelf Machine Learning (ML) algorithms. The literature extensively uses classic machine learning algorithms/models, because a plethora of programmatic frameworks supports them. Except for the works relying on *survival analysis* and *probabilistic soft logic*, all the other studies adopt a plain modelisation schema to structure the raw data. Therefore, the articles surveyed hereafter differ on the choice of the algorithmic family (macro-category) and on the features taken into consideration with appropriate pre-processing techniques.

This section follows the same format of Section 3. In particular, for each type of model in Figure 2 under the *Classic Learning Methods* category, we provide its formal description and subsequently enlist the literature studies that exploit it. Although research engages in confronting several prediction models, we report only the strategy (and combinations) of the best performing one. Notice that we group methods based on Decision Trees and Probabilistic Soft Logic under the *Rule-based* class. All the models above identify rules that help to classify new instances. Instead, we situate methods based on Random Forests and AdaBoost under the *Ensemble* category.

Works based on Linear Regression. Gitinabard et al. [30] use decision trees to select features corresponding to maximum information gain. Subsequently, they use a logistic regression model to distinguish between dropouts and persisters. He et al. [33] propose two methods based on linear regression: sequentially and simultaneously smoothed linear regression, respectively, LR-SEQ and LR-SIM. Both of the proposed strategies use transfer learning approaches to use the previous week's knowledge to help learn smoothed probabilities for the current week. The authors split courses into weekly phasic views. LR-SIM is an improvement upon LR-SEQ because, in the latter, early inaccurate predictions cannot benefit from the knowledge learned in subsequent weeks, hence sabotaging later models. Therefore, LR-SIM learns models for all weeks simultaneously. Robinson et al. [65] use a lasso-regularised variant of the logistic regression, which operates as a penalty for complexity whose size the authors determine empirically. Taylor et al. [71] use logistic regression to predict dropout. They apply two temporal concepts to structure their prediction strategy: i.e., *lead* and *lag*. Lead represents how many weeks in advance to predict dropout; whereas, lag represents how many weeks of historical data the classifier has available to make the prediction. For instance, when considering a lead of five and a lag of three, the first three weeks of data are used to predict the upcoming five weeks. Therefore, the data corresponding to the first three weeks of a course become the training set and the dropout value for the eighth week becomes the label to predict.

Works based on SVM. Following an initial principal component analysis that indicates that the dropouts can be better separated, Kloft et al. [41] trained a weekly SVM. They state that "history" features coming from previous weeks are useful until a particular point in time. Amnueypornsakul et al. [3] train an SVM classifier based on the RBF kernel for each of their two models (i.e., specific and general case models). The authors distinguish between dropout and inactive student cases. Dropouts compose the set of students that do not interact at all. Inactive students still attend the course at hand, but their interaction sequence is coarse-grained.

Works based on Naive Bayes. Gaudioso et al. [29] employed several rule-based decision tree algorithms (JRip, J48, and PART), but finally they conclude that the most accurate model is the Naive Bayes classifier. Similarly, the best performing and statistically most significant classification model in Reference [44] is the Naive Bayes classifier. This classifier works best in all the dataset variants the authors have considered (i.e., demographics data only and demographics together with the first homework results). Kotsiantis et al. [45] and Manhães et al. [54] choose Naive Bayes classifier not only because of its superior performances but also because of its short computational time. Last, Li et al. [50] exploit *n* Naive Bayes classifier to train the components of a multi-view Semi-Supervised Learning (SSL) architecture [13]. The authors follow two steps in multi-training. First, they train the components on a subset of the dataset. Afterwards, they move the unlabelled examples, which have high confidence upon prediction from the *n* classifiers, into the labelled set of instances. This two-phase process continues until there are no more unlabelled data.

Works based on Decision Trees. Al-Radaideh et al. [1] test ID3 and C4.5. The former performs better when executing a hold-one-out experiment, whereas the latter outperforms the other in a 10-fold cross-validation method. Kovačić [46] exploits two different decision tree classifiers, Chi-square Automatic Interaction Detection (CHAID) [38] and CART [9], which has a slightly better performance. Because Dekker et al. [20] want to distinguish between misclassification types, they boost accuracy with cost-sensitive learning. They use a cost matrix as input to the new meta-classifier (i.e., CART algorithm) whose goal is to increase the weight of false negatives over false positives. Kostopoulos et al. [42] adopt an SSL approach over the C4.5 algorithm [63]. The authors use the Tri-Training [78] strategy to produce the best performing variant for their dataset. Nagrecha et al. [59] confront interpretable methods such as decision trees against non-interpretable strategies (e.g., random forests and gradient boosted trees). The model the authors choose has the best performance and the greatest interpretability. Therefore, decision trees emerge as the model satisfying both criteria.

Works based on Probabilistic Soft Logic. Ramesh et al. [64] use the modelisation schema described above to construct relevant PSL rules using logical connectives. The produced model associates these rules with student survival. The authors build two PSL models to resolve the SDP problem. The first—denoted as *direct*—infers the student survival solely from observable features. The second—denoted as *latent*—infers the survival as a hidden variable. The rules produced vary from one model to the other. In more detail, the direct model exploits one or more observable behaviour features to predict student survival. Hence, observable features directly imply survival. Contrarily, the latent model includes hidden variables based on patterns of student engagement. Because these variables cannot be directly measured, the authors treat student engagement as a latent variable by associating observable variables (features) to a certain form of engagement. The authors demonstrate that the latent model outperforms the direct one.

Works based on Neural Networks. Chen et al. [14] base their work on a novel combination of decision trees with Extreme Learning Machine (ELM). They transform results obtained from the decision tree employed into a neural network. They perform feature selection using decision trees based on maximal information gain and weight them based on their impact on the leaf nodes. Because the root node has the best classification ability being connected to all nodes, it has the most significant impact. They call this procedure the *enhancement layer*. The subsequent layer—*mapping layer*—transforms the decision tree into an ELM, which is a single hidden layer neural network. The input neurons are the root and internal nodes of the tree. The hidden neurons correspond to the leaf nodes. They connect an input neuron to the hidden neurons if it has an impact on its corresponding leaf in the tree. Therefore, they link the input neuron analogous to the root node to each hidden neuron. Meanwhile, other input neurons might be connected only to some hidden neurons. Hence, the authors set to zero the weights of a connectionless neuron pair.

Ensembles: Works based on Random Forests. Gray and Perkins [31] use random forests, which relies on the degree program as the discriminatory cohort to determine the dropout cases. Haiyang et al. [32] exploit Time-Series Forests (TSF) [36]. A TSF employs a combination of entropy gain and distance measure to evaluate the different splits. Besides capturing important temporal characteristics, it reveals which course module (or period) affects the most the learning progress of a certain student s.

Ensembles: Works based on AdaBoost. Berens et al. [8] combine the prediction powers of linear regression, neural networks, and random forests to distinguish at-risk students and persisters. Berens et al. [8] use a simple multivariate model for the linear regression $y_{s,j} = \beta_0 + \beta_1 \vec{d_s} + \beta_2 z_{s,j} + \epsilon_{s,j}$ where s and j denote the students and the semester, respectively, and $\vec{d_s}$ is the vector composing demographic data. In this equation, $z_{s,j}$ denotes the time-varying performance data of student s in semester j. The neural network component is composed of three layers (including the input) with a final sigmoid output layer. The input layer has 31 neurons. The first hidden fully connected layer has 16, whereas the second fully connected has 8. The hidden layer weights are randomly initialised in the range $[-1, +1]$. The activation of each layer is the sigmoid function. The last component is a random forest, because decision trees tend to overfit data being a non-parametric ML technique. Hu et al. [37] use two variants of boosting with decision trees. In other words, the authors rely on the CART and C4.5 algorithms to build the components of the boosting algorithm. The authors decide to use the CART algorithm as the base component of the boosting schema, because, according to the experimentation phase, it has a lower false-positive rate.

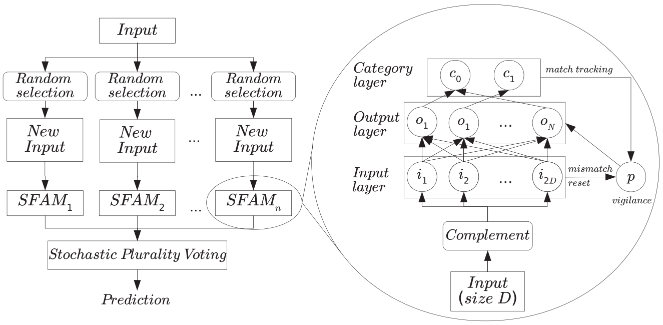

Fig. 12. The PESFAM architecture composed of *n* components.

Works based on other ensemble methods. Kotsiantis et al. [43] use an ensemble of WINNOW [51], 1-Nearest Neighbour, and Naive Bayes. WINNOW is a linear online algorithm similar to a perceptron. The only difference stands on the weight updating rule. In WINNOW, the weights of relevant features increase exponentially; meanwhile, those of irrelevant features decrease exponentially. However, 1-Nearest Neighbour assigns an instance to the class of its closest neighbour in the feature space.

Lykourentzou et al. [53] use a neural network, SVM, and a probabilistic ensemble simplified fuzzy ARTMAP (PESFAM) [52] as the components of the ensemble model. Having covered enough content about the first two strategies, we provide the reader with the information on how a PESFAM works. PESFAM combines simplified fuzzy ARTMAP (SFAM) modules whose merging policy is the probabilistic plurality voting strategy. Inspired by the PESFAM illustration in Reference [53], we delineate in Figure 12 the same architecture composed of *n* SFAM components in the student dropout scenario. The left part of the figure depicts the PESFAM ensemble. As with other ensembles, the dataset instances are randomly selected with replacement to generate a new dataset representing the *New input* rectangles. They represent different “views” of the original input dataset. Each component then produces prediction labels for each input instance. A majority voting strategy stabilises the final labels for the instances. The right part of the figure represents the detailed view of an SFAM module of the ensemble. An SFAM is a three-layered neural network: input, output, category. There is a connection between the input nodes and those in the output layer. Every output layer node points to a single category node, which depicts one of the two classes (dropout or not). For further details about the training and testing phases of an SFAM network, we refer the reader to References [39, 53].

Works based on Survival Models. The algorithmic choice of Gitinabard et al. [30] is a logistic regression, however, they perform a survival analysis, finding that both behavioural and social features have a high hazard ratio. Survival analysis is known to provide less biased estimates than least-squares linear regression. Hence, Yang et al. [76] conducted this analysis in a stage-based way to determine the factors with the most substantial effect on student dropout. They exploit the hazard rate to describe the impact of the identified factors on attrition. Both References [2] and [16] use Cox proportional hazard regression [17] as their survival model to predict dropout labels in their degrees.

4.3 Deep Learning Prediction Strategies

With the advent of deep learning, the dropout research area has benefited with multiple proposals of sophisticated strategies to solve the SDP problem. We can adopt deep learning approaches because of the size of available datasets, and because some algorithms have been shown to perform particularly well on sequential information. Models proposed in the literature exploit the raw logs that the interaction of students with the e-platform generates. Moreover, deep strategies are convenient, because, unlike classic learning techniques, they avoid most manual feature engineering problems, thus facilitating the task of inexperts in the e-learning domain.

As shown in Figure 2 and, specifically, in sub-classes of *Deep Learning*, we have identified two families of deep learning approaches used in the literature: i.e., recurrent and convolutional neural networks. SDP studies have not adopted deep strategies until recently. So, the works we enlist in the following sections are far less numerous than those belonging to the classic learning methods discussed above. To analyse these works, we follow the same structure of the previous section.

Works based on RNNs. Although the majority of the surveyed works base their prediction models on LSTM cells, Wang et al. [72] exploit the RNN as the last predictive layer of their overall architecture. The choice of an RNN is because, in their model, past e-tivities affect current student e-tivities. E-tivities closer in time have a more dominant influence than those that are distant. The authors exploit this characteristic by using an RNN among different time slices. The activation function for the hidden state is the ReLU function except the last one, which is a sigmoid function.

Works based on LSTMs. The authors in Reference [24] propose two different Input-Output Hidden Markov Models (IOHMMs) for their approach. Finally, they demonstrate that a variant of an RNN with 20 LSTM cells is the best performing model. Ding et al. [22] use a combination of LSTMs and Auto-Encoders (AEs) to learn compact and effective representations from raw data. They choose an unsupervised approach to extract underlying representations of the input sequences they use. Then, an LSTM intertwined with AEs combines the sequence-to-sequence learning framework [70]. As an AE requires, the authors use two LSTMs: an encoder and a decoder LSTM. The input sequence $(x_1, x_2, \ldots, x_{i-1})$ is fed into the encoder that learns a hidden representation z forwarded to the decoder. The decoder LSTM reads the original sequence—with the exception of x_1—in reverse order and it outputs $(\tilde{x}_{i-1}, \tilde{x}_{i-2}, \ldots, \tilde{x}_1)$ as a reconstruction of the original input. The authors try to predict student performance with only behavioural patterns from the first i course phases. For this, the predictive power of the initial model is not optimal. Instead, they exploit transfer learning from finished courses to cope with ongoing courses and real-time dropout identification. In this way, they can use the features after course phase c_i to the decoder part. The prediction happens at the encoder level only using the first i course phases. The authors, hence, add a predicting decoder parallel to the reconstruction decoder. The newly added component to the overall architecture aims to predict the sequences of features in the later course phases $(x_i, \ldots, x_k)$. Finally, the authors use the learnt representations as new feature inputs to a fully connected one hidden layer neural network to distinguish between dropout students and persisting students.

Works based on CNNs. The core of the CFIN system presented in Reference [25] relies on a one-dimensional CNN, while the prediction is handled by XGBoost [15]. The authors incorporate context information, including user and course data, into a single prediction schema. Thus, because contextual data correlates with learning activities, the authors rely on CNN to learn a context-aware representation for each activity feature—a process called context-smoothing. Three crucial steps compose CFIN: context-smoothing, attention mechanism, and prediction component. Feature augmentation, embedding, and fusion form the first step. These procedures come before the convolution part, which learns a context-aware representation for each selected e-tivity feature. The authors exploit an attention mechanism[13] to model attention-based interactions for e-tivity features using contextual information. Then, a dense neural network receives the produced

[13] We invite the reader to check the work of Bahdanau et al. [4] for more details on the attentive networks in the context of neural machine translation.

Table 4. Model Selection Matrix Based on the Student Modelling and Prediction Strategy Taxonomies in Figures 1 and 2

				Student Modelling						
				Plain Modelisation				Sequence Labelling		
				Demography	Study-related		Derived	Temporal Series		Temporal Networks
					Assessments	Previous education		Clickstream	Forum	
Prediction Strategy	Analytic Examination			[75]	[74, 75]		[74]			
	Classic Learning Methods	Naive Bayes		[29, 44, 45]	[45]		[29, 54]	[50]		
		Linear Regression					[65]	[33, 71]		[30]
		Rule-based	DTrees	[1, 42, 46]	[42]	[1, 20]	[46]	[59]		
			PSL					[64]	[64]	
		Survival Analysis								[76]
		SVM			[3]		[3]	[41]		
		Neural Networks						[14]		
		Ensemble	Adaboost	[8]	[8]	[8]	[37]			
			Random Forests				[31]	[32]		
			Other	[53]	[43, 53]	[53]	[53]			
	Deep Learning	RNN	Plain RNN					[72]		
			LSTM					[22, 24]	[24]	
		CNN					[25]	[62, 72]		

weighted-sum vector from the attention mechanism, and it predicts the outcome label. Qiu et al. [62] propose DP-CNN to tackle the SDP problem. The convolution stage follows an end-to-end approach. It is composed of several convolutional layers and a last fully connected layer with a sigmoid function. Precisely, DP-CNN consists of a four-layer network with two convolutional layers and two fully connected ones. To preserve all the feature information, the authors do not use pooling layers. Because the input that the authors consider is two-dimensional, the convolution kernel is also two-dimensional. The kernel activation function is a ReLU. The third layer of the architecture also uses a ReLU activation function, whereas the last one—the predictive layer—relies on a sigmoid function.

As mentioned above, Wang et al. [72] utilise a simple RNN as the last layer of their prediction strategy, ConRec. The authors use a CNN to extract features for the e-tivity matrices in each time slice. Therefore, the first four layers of the architecture are convolution and pooling layers. The fifth layer is a fully connected neural network that fuses features that the convolutional and pooling layers extract. This layer generates one vector for each time slice. The RNN takes these vectors and engages in determining the instances' dropout class labels.

4.4 Impact of Student Modelling on Prediction Strategy

In practice, the choice of student modelling and prediction strategies make an impact on each other. In this section, we provide a summary in Table 4 to suggest papers to consult for further information. Each cell of the table points to the works that correspond to the different hierarchical sub-branches of the pipeline in Figures 1 and 2. By consulting the architectures presented in these papers, the reader can pinpoint the desired model and apply it to their project.

The table complies with the observations that we made in the previous sections. The majority of works in the literature tackle the SDP problem relying on plain modelisation for the raw data.

Referring to the plain modelisation procedure (modelled in Figure 3), notice that the most used prediction strategies are Naive Bayes and Decision Trees. Both these off-the-shelf classifiers are easy to use. In other words, there is a plethora of available libraries that have classic machine learning classifiers already implemented. In addition to its simplicity, a decision tree classifier is interpretable. Hence, the rules engendered are usable for designing intervention schemata for at-risk students. Note that poor interpretability is a recognised problem of deep methods that has recently attracted the attention of researchers in machine learning. However, the SDP literature has not yet considered this issue.

Notice also that in the sequence labelling part of the table the academia concentrates mainly on clickstream information. Research has paid less attention to forum-derived information and the modelisation via temporal networks, because only a minor percentage of students intervene in forum discussions [66]. Thence, the few papers [24, 64] that exploit forum intervention data incorporate clickstream information. The other works based on temporal networks [30, 76] are among the few to obtain competitive results and performances w.r.t. the articles approaching SDP from a pure clickstream point-of-view. Last, among deep learning strategies, some works have utilised RNNs and CNNs to distinguish between dropouts and persisters. Only one paper uses a combination of RNNs and CNNs to tackle the SDP problem [72]. Notice that published works using deep methods are far less than those adopting classic learning ones. Table 4 demonstrates that the upper quadrant is much more populated than the lower one. Therefore, researchers might profit from the scarcity of deep learning methods in SDP to develop novel techniques in this research area.

5 EVALUATION MEASURES AND DATASETS

We use evaluation measures for a predictive model to assess how well the model's predictions reflect the ground truth of a particular dataset, and to estimate performances on future unseen cases. In general, several well-known evaluation metrics are used in statistics, machine learning, and deep learning. However, in the SDP scenario, given the unbalanced nature[14] of the data, only a fraction of these methods can be used. In this section, we discuss the advantages and drawbacks of the measures used in the SDP literature.

Research considers SDP as a binary classification problem. Note that even a probabilistic output must be eventually converted into a binary decision, i.e., whether or not a student is at the risk of dropout. We have identified several measurements that the surveyed studies exploit based on classification problems.[15] Hereafter, we report a description of these metrics, and we summarise their usage in Table 5:

(1) *Accuracy.* Pioneer works [42–46] relying on simple off-the-shelf machine learning algorithms employ only accuracy as a means to assess their models' performances. Accuracy solely can be misleading. In SDP, considering the class imbalance problem, a model can predict the value of the majority class for all predictions and achieve a high classification accuracy. Nevertheless, this model is not useful in the problem domain. Therefore, it is mandatory to apply additional measurements to evaluate the classifier properly. Other papers [2, 3, 8, 20, 32, 37, 41, 46, 53, 54] also use different metrics to obtain a more realistic view of the model's performances.

(2) *Confusion Matrix and F-measure.* To tackle the inefficiency of accuracy to capture good classifiers, the studies in literature rely on rates calculated from the confusion matrix [3, 8, 20, 31, 37, 50, 53, 54, 62, 72, 74] (e.g., Precision, Recall, and their derivatives such as the

[14] The number of persisters and dropouts is not the same. Some courses might have more completers than dropouts. Others might have the opposite scenario.

[15] Some works also exploit regression-related metrics, adopting them into a classification scenario.

Table 5. The List of Evaluation Measurements That the Surveyed Articles Use

	Measurements	Works
Rates	False Negative	[37, 54]
	False Positive	[31, 37, 54]
	True Negative	[8, 54]
	True Positive (Recall)	[3, 8, 20, 31, 50, 53, 54, 62, 72, 74]
	Precision	[3, 8, 20, 31, 50, 53, 62, 72, 74]
	Accuracy	[1–3, 8, 20, 32, 37, 41–46, 53, 54]
F_β	$\beta = 1$	[2, 3, 14, 16, 25, 30, 31, 50, 62, 72, 74]
	$\beta = 1.5$	[20]
	AUCROC	[2, 14, 16, 24, 25, 30, 31, 59, 62, 64, 65, 71, 72]
	AUCPR	[16, 64]
	Cohen's Kappa	[3]
	MSE	[22]
	MAE	[2]
	OPER/UPER	[2]

F-scores). Notice that the majority of the studies that rely on precision and rate-based metrics exploit the F-score measurement. However, the F-score does not provide any intuitive explanation of the performances. However, a large number of samples—as is the case of SDP—does not affect precision. Precision measures the number of true positives out of the samples predicted as positives (i.e., true positives and false positives together), and it does not depend on the number of samples. A mixture of recall, precision, and F-score gives an intuition of the overall performances of a classifier. Because the general formula of the F-score is $F_\beta = (1 + \beta^2) \times \frac{P \times R}{\beta^2 \times P + R}$ where P and R denote, respectively, the precision and recall, one can exploit several choices of β to measure the relationship between P and R. The most common choice of β is 1.

(3) *AUCROC and AUCPR.* Recall and false positive rate (FPR) constitute the building blocks of the ROC curve. They measure the ability to separate dropouts from persisting students.[16] The ROC is a probability curve, and AUCROC tells how much a model is capable of distinguishing between classes. The higher the AUCROC [2, 14, 16, 24, 25, 30, 31, 59, 62, 64, 65, 71, 72], the better a model is at predicting dropouts as 1s and persisters as 0s. Therefore, the works that rely on AUCROC combine the true positive and false positive rates to assess the model's predicting power. However, AUCPR, used by References [16, 64], is a more suitable measure, since it is more resistant to the class imbalance problem w.r.t. AUCROC [19]. Differently from the ROC curve, the Precision-Recall (PR) curve does not account for true negatives, because they are not components of either the recall or the precision formula.[17]

(4) *Cohen's Kappa.* Only a minority of research exploits the Cohen's Kappa score [3]. This measure manages both multi-class and imbalanced class problems. Cohen's Kappa represents the inter-rater reliability score (taking into account also an agreement by chance) of two models (classifiers). Therefore, researchers could exploit this metric to verify the alliance between their classifier and that corresponding to the ground truth. In this way, one

[16] https://towardsdatascience.com/what-metrics-should-we-use-on-imbalanced-data-set-precision-recall-roc-e2e79252aeba.

[17] http://www.chioka.in/differences-between-roc-auc-and-pr-auc/.

determines how close are the predictions of the selected model with the original labels. Although −1 indicates no agreement and +1 perfect agreement, the reader can consult Reference [55] when choosing an acceptable range of the kappa score.

(5) *Mean Squared Error (MSE).* It measures the average of the squares of the errors represented as the difference between the predicted values and the ground truth: i.e., MSE = $\frac{1}{n}\sum_{i=1}^{n}(\tilde{y}_i - y_i)^2$ where $\tilde{y}_i$ represents the ith predicted value and y_i is the ith true label. In this context, it determines how well the model fits the data. A perfect MSE score is zero, but, since algorithms contain some randomness, MSE scores are strictly positive. We consider as optimal those models whose MSE is close to zero. Ding et al. [22] employs this metric for their performances.

(6) *Mean Absolute Error (MAE).* It is the average of absolute errors: i.e., MAE = $\frac{1}{n}\sum_{i=1}^{n}|\tilde{y}_i - y_i|$ where $\tilde{y}_i$ and y_i have the same meaning as those mentioned in MSE. It treats both underestimating and overestimating of the true value in the same manner. As with MSE, MAE is also a negatively oriented metric, meaning that lower scores are better. Ameri et al. [2] use MAE.

(7) *Underestimated/Overestimated Prediction Error Rate (UPER/OPER).* To mitigate the underestimating and overestimating effect that MAE lacks on the predicted values, Ameri et al. [2] introduce UPER and OPER. The authors define UPER as the fraction of the underestimated prediction output over the entire prediction error. In other words, $UPER = \frac{\sum_{i=1}^{n} I(\tilde{y}_i < y_i)}{\sum_{i=1}^{n} I(\tilde{y}_i < y_i) + \sum_{i=1}^{n} I(\tilde{y}_i > y_i)}$ where $I(\tilde{y}_i < y_i) = 1$ if $\tilde{y}_i < y_i$ and $I(\tilde{y}_i > y_i) = 1$ if $\tilde{y}_i > y_i$. The indicator functions $I(\tilde{y}_i < y_i)$ and $I(\tilde{y}_i > y_i)$ are equal to zero if their conditions are not satisfied. Therefore, $\sum_{i=1}^{n} I(\tilde{y}_i < y_i)$ is the number of underestimated instances and $\sum_{i=1}^{n} I(\tilde{y}_i > y_i)$ is the number of overestimated instances. OPER is the opposite of UPER: i.e., $OPER = 1 - UPER$. The authors exploit these two metrics to estimate the semester in which a student drops out. Nevertheless, researchers might consider UPER and OPER if they treat SDP as a regression problem. i.e., the label of each student now represents the probability that they drop out.

We note that a suitable comparison of the performances of different methods summarised in this survey is not hindered by the diversity of the adopted evaluation measures, but rather by the scarce tendency to consider common benchmarks, with very few exceptions. To help researchers compare their predicted results with available baselines, we summarize in Table 6 the datasets that have been adopted in the surveyed works.

The KDD Cup competition[18] is one of the few common benchmarks adopted in the literature. In this competition, systems have been compared on well-known online courses provided by famous MOOCs such as Coursera and *edX*. The goal was to predict the probability that a student would drop out of a course in 10 days. The competition used AUCROC to evaluate the submissions. Chen et al. [14], Feng et al. [25], Li et al. [50], Qiu et al. [62], and Wang et al. [72] exploit the mentioned benchmark dataset. As reported in Table 5, we can compare References [14, 25, 72], as they use AUCROC for the evaluation. Instead, we can also compare References [25, 50, 62, 72], because they all rely on the F-1 score. From the first triple of papers, Feng et al. [25] is the winner with an AUCROC of 0.9093. Whereas, the most performing strategy from the second quadruple is Li et al. [50], with an F-1 score of 0.9241. However, we note that predicting a student dropout with a 10-day anticipation might not allow in practice the adoption of any retention strategy by the institution.

For completeness, we mention two studies concerned with traditional degrees that used data on freshmen enrolled at Wayne State University [2] and George Mason University [16]. The former

[18] https://biendata.com/competition/kddcup2015/.

Table 6. The Datasets Used in the Literature

Type	Resource	Study	Available link
MOOC	HarvardX	[65]	None
	edX	[22]	https://www.edx.org/course/introduction-to-java-programming-part-1-2 https://www.edx.org/course/introduction-to-java-programming-part-2-2
		[24]	https://www.edx.org/course/introduction-to-java-programming-part-1-2 https://www.edx.org/course/introduction-to-java-programming-part-2-2
		[30]	https://www.edx.org/course/big-data-and-education
		[59]	https://www.edx.org/course/i-heart-stats-learning-love-statistics-notredamex-soc120x
	MITx	[71]	https://www.edx.org/xseries/mitx-circuits-and-electronics
	Coursera	[24]	https://www.coursera.org/learn/gastronomy
		[30]	None
		[64]	https://www.coursera.org/learn/surviving-disruptive-technologies https://www.coursera.org/learn/genes
		[76]	None
	KDDCup15	[14, 25, 50, 62, 72]	http://data-mining.philippe-fournier-viger.com/the-kddcup-2015-dataset-download-link/
	EMNLP 2014	[3, 41]	None
e-course	CS in HOU	[42–45, 75]	https://www.eap.gr/en/courses/216-computer-science/course-structure/1556-pli10-introduction-to-informatics
	Programming I in YU	[1]	None
	Information Systems in OP	[46]	None
	Information Literacy & Information Ethics	[37]	None
	Computer Networks & Communications NTUA	[53]	None
	Web Design NTUA	[53]	None
Online degree	Electrical ENG in TU/e	[20]	None
	ENG in UFRJ	[54]	None
	State University in NRW	[8]	None
	PUAS in NRW	[8]	None
	UK Bangor University	[31]	None
	Open University	[32, 74]	https://analyse.kmi.open.ac.uk/open_dataset

does not distinguish between students of different degrees and uses all the enrolled freshmen for training. Instead, Chen et al. [16] study the dropout phenomenon by discriminating between freshmen registered in nine major STEM degrees (e.g., mathematics, physics, chemistry).

Besides KDDCup15 and the Open University, all the other datasets are not publicly available and require that researchers contact data providers to access them. These studies often devise prediction strategies tailored for their own institutions. Thus, they do not release sensitive data about their enrolled students. We note that the publication of sensitive data is to be strictly compliant with privacy regulations; for this reason, in Section 3 of Supplementary Material, we analyse critical aspects related to privacy infringement mitigation.

6 OPEN CHALLENGES

The student dropout research area presents several challenges not addressed yet. We dedicate this section to summarise open issues and promising research directions that emerged during the analysis of the state-of-the-art literature in the field.

(1) *Finding standard benchmarks and evaluation strategies.* Due to the absence of standard benchmarks, we have experienced some difficulty in comparing different approaches, even if they adopt similar modelling strategies. Surveyed studies tend to compare themselves with simple baselines rather than state-of-the-art solutions, making a direct comparison between strategies an arduous task. Even when they use data coming from the same provider (e.g., Coursera) they adopt different evaluation metrics and baselines. To help benchmark future solutions, in Section 5, we provided (Table 6) an index of available datasets. Researchers can consult it when evaluating and comparing their performances with those proposed in the literature. Clearly, the field would also greatly benefit from the systematic organisation of public challenges such as the previously mentioned KDDCup15.

(2) *Deep sequential methods should be better explored.* Institutions are interested in identifying dropout students as soon as possible. Therefore, techniques that do not exploit the sequential nature of students' behaviour are less suited, since they are unable to cope with real-time requirements. As discussed in Section 2.1, information from previous course phases is relevant to decide whether a particular student is likely to drop out in the future. Deep learning methods are better suited for sequence labelling, thanks to their built-in hidden state modelling. For example, a CNN combines both structural and temporal features into a convoluted hidden representation of the student behaviour; whereas, an RNN generates a hidden vector that requires the calculation of its current state at each time step. A further argument in favour of deep methods is that they perform better when large datasets are available, as is often the case in educational data mining.

    To date, deep learning in SDP is still not sufficiently explored. Among the issues hindering the adoption of deep methods, one of the main problems is poor interpretability (see Reference [57] for a thorough survey). Interpretability is critical in the e-learning field, because the problem is not so much predicting, but preventing, dropouts. In absence of an explanation of the student features and learning "motifs" that supported a dropout prediction, effective prevention strategies cannot be prescribed. Interpretive deep methods[19] are an emerging research topic that may open the possibility of more extensive applications in SDP and is therefore a suggested field of study. In general, we believe that deep learning may provide an ample opportunity of contributing to the academia with innovative ideas.

(3) *More sophisticated models are needed to cope with e-degrees.* As more and more universities offer online degree courses, it is also necessary to address the more complex case of e-degrees. As emphasised in the previous sections,[20] the literature does not focus on online degrees, rather, it copes with the fast-paced and short-term MOOCs. Modelling e-degrees is particularly challenging, since the decision to abandon studies for a student may arise from complex interactions between the results of sequential and parallel activities. For this reason, modelling e-degrees is not a simple extension of MOOCs. For example, the authors in References [1, 42, 44–46] exploit simple time-invariant features to train their models, achieving mediocre accuracy scores. Because their dataset contains descriptive static features, Hu et al. [37] reach optimal performances in terms of accuracy. Kotsiantis et al. [43] try to adopt a three-layered neural network with time-unvarying features, failing at surpassing the baseline's accuracy. Last, Lykourentzou et al. [53] have a boost in accuracy when including more course phases in the input features. However, because we aim at performing real-time predictions, their plain modelisation strategy is not suitable

[19] Among which *attentive* mechanisms are gaining popularity, such as Feng et al. [25], among others.
[20] The reader can notice this phenomenon in the last two quadrants of Table 6.

for proactive student monitoring. To foster better modelling of e-degrees, in this survey, we introduced a general notation that may help scholars to understand the implications of extending their methods to the case of students attending multiple parallel courses.

(4) *Models should consider the duration of e-tivities.* Last, temporal lags on completing e-tivities—such as assignment submissions—are essential for detecting dropouts. The surveyed works do not consider this facet of SDP, which requires modelling intervals without e-tivities and the amount of time that a student engages in the same type of action. Note that this issue is particularly crucial for corporate universities, for which the problem, rather than the risk of dropout, is an excessive completion time for professional refresher courses. Future developments should consider deep sequential methods that focus not only on the dropout phenomenon but also on the time that students take to complete course stages.[21] As an example, researchers might use a modelisation like the one presented in Figure 4, which considers inter-stage gaps that can be tailored ad hoc for each student.

7 CONCLUSIONS

Nowadays, education—and even more, online education—is one of the top industries in the world. E-learning systems are increasingly gaining popularity, because of several benefits of learning anywhere, anyplace, and anytime. Not only public and private universities, but also many corporations have adopted e-learning systems for employee training and learning, to create a collaborative learning environment. However, despite the benefits of distance learning courses, educational institutions have a growing concern for low retention rates and, in general, low certification/graduation rates of these kinds of degrees. Students enrolled in online degree programs have a higher chance of dropping out than those attending a conventional classroom environment. For corporate universities, the main issue is the lack of a motivation to learn for employees and the inability to decipher one's preferred learning style. It may lead to poor results in terms of efficacy and alignment of employee needs with strategic organisational goals. Therefore, it is of paramount importance for both public and private institutions to increase the efficiency and efficacy of learning outcomes, both for social and economic reasons.

Early prediction of students at the risk of dropout is one of the challenging tasks faced by researchers in the learning analytics field. The purpose of this survey was to present an in-depth analysis of student dropout prediction (SDP) in e-learning environments, under the central perspective, but not exclusive, of Machine Learning. We organised existing literature according to a hierarchical classification that follows the workflow of design choices in SDP. Furthermore, we introduced a formal notation to uniformly describe the alternative dropout models adopted by researchers in the field, and to model both MOOCs and the more complex case of e-degrees. Besides synthesising the most promising predictive strategies presented in the literature, we analysed several additional, and yet crucial, issues such as the evaluation strategy, the identification of data sets available for comparative studies, and the consideration of privacy issues.

Overall, our survey provides novice readers who address the SDP problem with practical guidance on possible design choices and researchers with an overview of open challenges and most promising research directions on the topic.

REFERENCES

[1] Qasem A. Al-Radaideh, Emad M. Al-Shawakfa, and Mustafa I. Al-Najjar. 2006. Mining student data using decision trees. In *Proceedings of the International Arab Conference on Information Technology (ACIT'2006)*. 1–5.

[21] In MOOCs, students complete a module when they submit and successfully pass the assignment.

[2] Sattar Ameri, Mahtab J. Fard, Ratna B. Chinnam, and Chandan K. Reddy. 2016. Survival analysis based framework for early prediction of student dropouts. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 903–912.

[3] Bussaba Amnueypornsakul, Suma Bhat, and Phakpoom Chinprutthiwong. 2014. Predicting attrition along the way: The UIUC model. In *Proceedings of the EMNLP Workshop on Analysis of Large Scale Social Interaction in MOOCs*. Association for Computational Linguistics, 55–59.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[5] Behdad Bakhshinategh, Osmar R. Zaiane, Samira ElAtia, and Donald Ipperciel. 2018. Educational data mining applications and tasks: A survey of the last 10 years. *Educ. Inf. Technol.* 23, 1 (2018), 537–553.

[6] V. K. Balakrishnan. 1997. *Schaum's Outline of Graph Theory: Including Hundreds of Solved Problems*. McGraw Hill Professional, New York, NY.

[7] Papia Bawa. 2016. Retention in online courses: Exploring issues and solutions—A literature review. *Sage Open* 6, 1 (2016), 2158244015621777.

[8] Johannes Berens, Kerstin Schneider, Simon Görtz, Simon Oster, and Julian Burghoff. 2019. Early detection of students at risk - Predicting student dropouts using administrative student data from German Universities and machine learning methods. *Journal of Educational Data Mining* 11, 3 (2019), 1–41. http://doi.org/10.5281/zenodo.3594771

[9] Leo Breiman. 2017. *Classification and Regression Trees*. Routledge, Abingdon, Oxfordshire, UK.

[10] Peter J. Brockwell, Richard A. Davis, and Matthew V. Calder. 2002. *Introduction to Time Series and Forecasting*. Vol. 2. Springer, New York, NY.

[11] Rebecca Brown, Collin Lynch, Yuan Wang, Michael Eagle, Jennifer Albert, Tiffany Barnes, Ryan Shaun Baker, Yoav Bergner, and Danielle S. McNamara. 2015. Communities of performance & communities of preference. In *CEUR Workshop Proceedings*, Vol. 1446. CEUR-WS.

[12] Vicki Carter. 1996. Do media influence learning? Revisiting the debate in the context of distance education. *Open Learn. J. Open, Dist. e-Learn.* 11, 1 (1996), 31–40.

[13] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. 2010. *Semi-Supervised Learning*. The MIT Press, Cambridge, MA.

[14] Jing Chen, Jun Feng, Xia Sun, Nannan Wu, Zhengzheng Yang, and Sushing Chen. 2019. MOOC dropout prediction using a hybrid algorithm based on decision tree and extreme learning machine. *Math. Probl. Eng.* 2019 (2019). https://doi.org/10.1155/2019/8404653

[15] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 785–794.

[16] Yujing Chen, Aditya Johri, and Huzefa Rangwala. 2018. Running out of STEM: A comparative study across STEM majors of college students at-risk of dropping out early. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*. ACM, 270–279.

[17] David R. Cox. 1972. Regression models and life-tables. *J. Roy. Statist. Soc. Series B (Methodol.)* 34, 2 (1972), 187–202.

[18] Fisnik Dalipi, Ali Shariq Imran, and Zenun Kastrati. 2018. MOOC dropout prediction using machine learning techniques: Review and research challenges. In *Proceedings of the IEEE Global Engineering Education Conference (EDUCON'18)*. IEEE, 1007–1014.

[19] Jesse Davis and Mark Goadrich. 2006. The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd International Conference on Machine Learning*. ACM, New York, NY, 233–240.

[20] Gerben W. Dekker, Mykola Pechenizkiy, and Jan M. Vleeshouwer. 2009. Predicting students drop out: A case study. In *Proceedings of the International Conference on Educational Data Mining (EDM'09)*.

[21] David P. Diaz. 2000. *Comparison of Student Characteristics, and Evaluation of Student Success, in an Online Health Education Course*. Ph.D. Dissertation. Nova Southeastern University.

[22] Mucong Ding, Kai Yang, Dit-Yan Yeung, and Ting-Chuen Pong. 2018. Effective feature learning with unsupervised learning for improving the predictive models in massive open online courses. arXiv:1812.05044.

[23] William Doherty. 2006. An analysis of multiple factors affecting retention in web-based community college courses. *Internet High. Educ.* 9, 4 (2006), 245–255.

[24] Mi Fei and Dit-Yan Yeung. 2015. Temporal models for predicting student dropout in massive open online courses. In *Proceedings of the IEEE International Conference on Data Mining Workshop (ICDMW'15)*. IEEE, 256–263.

[25] Wenzheng Feng, Jie Tang, and Tracy Xiao Liu. 2019. Understanding dropouts in MOOCs. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'19)*.

[26] Karen Frankola. 2001. Why online learners drop out. *Workf. Costa Mesa* 80, 10 (2001), 52–61. Retrieved from http://www.workforce.com/feature/00/07/29.

[27] S. Hari Ganesh and A. Joy Christy. 2015. Applications of educational data mining: A survey. In *Proceedings of the International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS'15)*. IEEE, 1–6.

[28] Josh Gardner and Christopher Brooks. 2018. Student success prediction in MOOCs. *User Model. User-Adapt. Interact.* 28, 2 (2018), 127–203.

[29] Elena Gaudioso, Miguel Montero, and Felix Hernandez-Del-Olmo. 2012. Supporting teachers in adaptive educational systems through predictive models: A proof of concept. *Exp. Syst. Applic.* 39, 1 (2012), 621–625.

[30] Niki Gitinabard, Farzaneh Khoshnevisan, Collin F. Lynch, and Elle Yuan Wang. 2018. Your actions or your associates? Predicting certification and dropout in MOOCs with behavioral and social features. arXiv:1809.00052.

[31] Cameron C. Gray and Dave Perkins. 2019. Utilizing early engagement and machine learning to predict student outcomes. *Comput. Educ.* 131 (2019), 22–32.

[32] Liu Haiyang, Zhihai Wang, Phillip Benachour, and Philip Tubman. 2018. A time series classification method for behaviour-based dropout prediction. In *Proceedings of the IEEE 18th International Conference on Advanced Learning Technologies (ICALT'18).* IEEE, 191–195.

[33] Jiazhen He, James Bailey, Benjamin I. P. Rubinstein, and Rui Zhang. 2015. Identifying at-risk students in massive open online courses. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence.*

[34] Michael Herbert. 2006. Staying the course: A study in online student satisfaction and retention. *Online J. Dist. Learn. Admin.* 9, 4 (2006), 300–317.

[35] Erin Heyman. 2010. Overcoming student retention issues in higher education online programs. *Online J. Dist. Learn. Admin.* 13, 4 (2010).

[36] Deng Houtao, Runger C. George, Tuv Eugene, and Martyanov Vladimir. 2013. A time series forest for classification and feature extraction. *Inf. Sci.* 239 (2013), 142–153.

[37] Ya-Han Hu, Chia-Lun Lo, and Sheng-Pao Shih. 2014. Developing early warning systems to predict students' online learning performance. *Comput. Human Behav.* 36 (2014), 469–478.

[38] Gordon V. Kass. 1980. An exploratory technique for investigating large quantities of categorical data. *J. Roy. Statist. Soc.: Series C (Appl. Statist.)* 29, 2 (1980), 119–127.

[39] Tom Kasuba. 1993. Simplified fuzzy ARTMAP. *AI Expert* 8, 11 (1993).

[40] Usha Keshavamurthy and H. S. Guruprasad. 2014. Learning analytics: A survey. *Int. J. Comput. Trends Technol.* 18, 6 (2014).

[41] Marius Kloft, Felix Stiehler, Zhilin Zheng, and Niels Pinkwart. 2014. Predicting MOOC dropout over weeks using machine learning methods. In *Proceedings of the EMNLP Workshop on Analysis of Large Scale Social Interaction in MOOCs.* 60–65.

[42] Georgios Kostopoulos, Sotiris Kotsiantis, and Panagiotis Pintelas. 2015. Estimating student dropout in distance higher education using semi-supervised techniques. In *Proceedings of the 19th Panhellenic Conference on Informatics.* ACM, ACM, New York, NY, 38–43.

[43] Sotiris Kotsiantis, Kiriakos Patriarcheas, and Michalis Xenos. 2010. A combinational incremental ensemble of classifiers as a technique for predicting students' performance in distance education. *Knowl.-based Syst.* 23, 6 (2010), 529–535.

[44] Sotiris Kotsiantis, Christos Pierrakeas, and Panagiotis Pintelas. 2003. Preventing student dropout in distance learning using machine learning techniques. In *Proceedings of the International Conference on Knowledge-based and Intelligent Information and Engineering Systems.* Springer, New York, NY, 267–274.

[45] Sotiris Kotsiantis, Christos Pierrakeas, Ioannis Zaharakis, and Panagiotis Pintelas. 2003. *Efficiency of Machine Learning Techniques in Predicting Students Performance in Distance Learning Systems.* University of Patras Press, 297–306.

[46] Zlatko J. Kovačić. 2010. Early prediction of student success: Mining student enrollment data. In *Proceedings of the Informing Science & IT Education Conference.* Citeseer.

[47] George D. Kuh. 2009. The national survey of student engagement: Conceptual and empirical foundations. *New Direct. Inst. Res.* 2009 (12 2009), 5–20. DOI: https://doi.org/10.1002/ir.283

[48] Anupama S. Kumar and M. N. Vijayalakshmi. 2012. Mining of student academic evaluation records in higher education. In *Proceedings of the International Conference on Recent Advances in Computing and Software Systems.* IEEE, 67–70.

[49] Mukesh Kumar, A. J. Singh, and Disha Handa. 2017. Literature survey on educational dropout prediction. *Int. J. Educ. Manag. Eng.* 7, 2 (2017), 8.

[50] Wentao Li, Min Gao, Hua Li, Qingyu Xiong, Junhao Wen, and Zhongfu Wu. 2016. Dropout prediction in MOOCs using behavior features and multi-view semi-supervised learning. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'16).* IEEE, 3130–3137.

[51] Nick Littlestone and Manfred K. Warmuth. 1994. The weighted majority algorithm. *Inf. Comput.* 108, 2 (1994), 212–261.

[52] Chu Kiong Loo and M. V. C. Rao. 2005. Accurate and reliable diagnosis and classification using probabilistic ensemble simplified fuzzy ARTMAP. *IEEE Trans. Knowl. Data Eng.* 17, 11 (2005), 1589–1593.

[53] Ioanna Lykourentzou, Ioannis Giannoukos, Vassilis Nikolopoulos, George Mpardis, and Vassili Loumos. 2009. Dropout prediction in e-learning courses through the combination of machine learning techniques. *Comput. Educ.* 53, 3 (2009), 950–965.

[54] Laci Mary Barbosa Manhães, Sérgio Manuel Serra da Cruz, and Geraldo Zimbrão. 2014. WAVE: An architecture for predicting dropout in undergraduate courses using EDM. In *Proceedings of the 29th ACM Symposium on Applied Computing*. ACM, New York, NY, 243–247.

[55] Mary McHugh. 2012. Interrater reliability: The kappa statistic. *Biochem. Med.: Časopis Hrvatskoga društva medicinskih biokemičara / HDMB* 22 (10 2012), 276–282. DOI: https://doi.org/10.11613/BM.2012.031

[56] Othon Michail. 2016. An introduction to temporal graphs: An algorithmic perspective. *Internet Math.* 12, 4 (2016), 239–280.

[57] Christoph Molnar. 2018. Interpretable machine learning. Retrieved from https://christophm.github.io/interpretable-ml-book.

[58] Michael Morgan, Matthew Butler, Neena Thota, and Jane Sinclair. 2018. How CS academics view student engagement. In *Proceedings of the 23rd ACM Conference on Innovation and Technology in Computer Science Education*. ACM, New York, NY, 284–289.

[59] Saurabh Nagrecha, John Z. Dillon, and Nitesh V. Chawla. 2017. MOOC dropout prediction: Lessons learned from making pipelines interpretable. In *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 351–359.

[60] Alejandro Peña-Ayala. 2014. Educational data mining: A survey and a data mining-based analysis of recent works. *Exp. Syst. Applic.* 41, 4 (2014), 1432–1462.

[61] Jiezhong Qiu, Jie Tang, Tracy Xiao Liu, Jie Gong, Chenhui Zhang, Qian Zhang, and Yufei Xue. 2016. Modeling and predicting learning behavior in MOOCs. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*. ACM, New York, NY, 93–102.

[62] Lin Qiu, Yanshen Liu, Quan Hu, and Yi Liu. 2019. Student dropout prediction in massive open online courses by convolutional neural networks. *Soft Comput.* 23 (2019), 10287–10301. https://doi.org/10.1007/s00500-018-3581-3

[63] Ross J. Quinlan. 2014. *C4.5: Programs for Machine Learning*. Elsevier, New York, NY.

[64] Arti Ramesh, Dan Goldwasser, Bert Huang, Hal Daume III, and Lise Getoor. 2014. Learning latent engagement patterns of students in online courses. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*.

[65] Carly Robinson, Michael Yeomans, Justin Reich, Chris Hulleman, and Hunter Gehlbach. 2016. Forecasting student achievement in MOOCs with natural language processing. In *Proceedings of the 6th International Conference on Learning Analytics & Knowledge*. ACM, New York, NY, 383–387.

[66] Carolyn Rose and George Siemens. 2014. Shared task on prediction of dropout over time in massively open online courses. In *Proceedings of the EMNLP Workshop on Analysis of Large Scale Social Interaction in MOOCs*. 39–41.

[67] Belinda G. Smith. 2010. *E-learning Technologies: A Comparative Study of Adult Learners Enrolled on Blended and Online Campuses Engaging in a Virtual Classroom*. Ph.D. Dissertation. Capella University.

[68] Dagim Solomon. 2018. Predicting performance and potential difficulties of university students using classification: Survey paper. *Int. J. Pure Appl. Math.* 118, 18 (2018), 2703–2707.

[69] Denise E. Stanford-Bowers. 2008. Persistence in online classes: A study of perceptions among community college stakeholders. *J. Online Learn. Teach.* 4, 1 (2008), 37–50.

[70] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems*. 3104–3112.

[71] Colin Taylor, Kalyan Veeramachaneni, and Una-May O'Reilly. 2014. Likely to stop? Predicting stopout in massive open online courses. arXiv:1408.3382.

[72] Wei Wang, Han Yu, and Chuyan Miao. 2017. Deep model for dropout prediction in MOOCs. In *Proceedings of the 2nd International Conference on Crowd Science and Engineering*. ACM, New York, NY, 26–32.

[73] Pedro A. Willging and Scott D. Johnson. 2009. Factors that influence students' decision to dropout of online courses. *J. Asynch. Learn. Netw.* 13, 3 (2009), 115–127.

[74] Annika Wolff, Zdenek Zdrahal, Andriy Nikolov, and Michal Pantucek. 2013. Improving retention: Predicting at-risk students by analysing clicking behaviour in a virtual learning environment. In *Proceedings of the 3rd International Conference on Learning Analytics and Knowledge*. ACM, New York, NY, 145–149.

[75] Michalis Xenos, Christos Pierrakeas, and Panagiotis Pintelas. 2002. A survey on student dropout rates and dropout causes concerning the students in the course of informatics of the Hellenic Open University. *Comput. Educ.* 39, 4 (2002), 361–377.

[76] Diyi Yang, Tanmay Sinha, David Adamson, and Carolyn Penstein Rosé. 2013. Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. In *Proceedings of the NIPS Data-driven Education Workshop*, Vol. 11. Curran Associates, Inc., 14.

[77] Eran Yukseltur and Fethi Ahmet Inan. 2006. Examining the factors affecting student dropout in an online learning environment. *Turk. Online J. Dist. Educ.* 7, 3 (2006), 76–88.

[78] Zhi-Hua Zhou and Ming Li. 2005. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Trans. Knowl. Data Eng.* 17, 11 (2005), 1529–1541.

[79] Mengxiao Zhu, Yoav Bergner, Yan Zhan, Ryan Baker, Yuan Wang, and Luc Paquette. 2016. Longitudinal engagement, performance, and social connectivity: A MOOC case study using exponential random graph models. In *Proceedings of the 6th International Conference on Learning Analytics & Knowledge.* ACM, New York, NY, 223–230.

Received December 2019; revised March 2020; accepted March 2020