MDSAA

Master Degree Program in

Data Science and Advanced Analytics

ACCURATE, TIMELY AND PORTABLE

Course-agnostic early prediction of student performance from LMS logs

Ricardo Miguel Costa Santos

Dissertation

presented as a partial requirement for obtaining the Master's Degree Program in Data Science and Advanced Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

2022

Title: Accurate, timely and portable

Subtitle: Course-agnostic early prediction of student performance from LMS logs

Ricardo Miguel Costa Santos

MDSAA

[This page should not be included in the digital version. Its purpose is only for the printed version

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

ACCURATE, TIMELY AND PORTABLE: COURSE-AGNOSTIC EARLY PREDICTION OF STUDENT PERFORMANCE FROM LMS LOGS

by

Ricardo Miguel Costa Santos

Dissertation report presented as a partial requirement for obtaining the Master’s degree in Advanced Analytics, with a Specialisation in Data Science / Business Analytics

Supervisor: Roberto Henriques, PhD

October 2022

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledged the Rules of Conduct and Code of Honor from the NOVA Information Management School.

Ricardo Santos

Lisbon, October 27th, 2022

DEDICATION

Gostaria de dedicar este documento a todos aqueles que, mesmo nem sempre concordando com as minhas escolhas, confiaram e apoiaram a decisão de me colocar numa segunda licenciatura para, posteriormente, perseguir o grau de Mestre:

Em primeiro lugar, à minha família, com especial destaque para os meus pais, irmãos e cunhadas,

Em segundo lugar, à Família Carreira, especialmente à D. Glória, ao Sr. António, à Filipa e ao Diogo por sempre me terem feito sentir como parte dos seus.

A todos os meus amigos que me aturaram a mim e aos meus devaneios nos bons e nos maus momentos.

Em último e mais importante lugar, à minha parceira e melhor amiga que tenho a sorte de ser também o amor da minha vida. Patrícia, nunca teria conseguido chegar aqui sem o teu apoio. Eu amo-te.

Para todos vós, perdoem-me os anos de atraso, mas aqui está.

Obrigado por tudo.

ACKNOWLEDGEMENTS

Não teria conseguido concluír este capítulo sem aqueles que, ao longo desta aventura no Mestrado em Data Science and Advanced Analytics, contribuíram de forma direta ou indireta para esta dissertação:

Ao Henrique, ao Gonçalo, à Raquel e à Andreia pelas suas contribuições durante os vários e muitas vezes exigentes trabalhos de grupo que tivemos ao longo deste percurso.

Ao Daniel, pelas horas em chamada no Teams durante a preparação para apresentações e entregas.

Ao Professor Roberto pela sua orientação, confiança, paciência e disponibilidade para prolongar reuniões muito para além daquilo que eram as suas possibilidades.

Ao Moe, ao Tiago e à Joana pelas dezenas de horas que dedicaram ao longo do ano às nossas reuniões para discussão das nossas respetivas dissertações. Estou certo que, sem o vosso contributo e sugestões, o meu trabalho seria certamente mais pobre e espero ansiosamente que, no final das nossas defesas, possamos todos ir celebrar o fim desta etapa.

Gostaria, ainda, de deixar um agradecimento especial àqueles que, não tendo dado um contributo visível para esta tese, tiveram uma indubitável influência na pessoa que sou hoje:

Ao Dr. Rui Carvalho e ao Ludgero Tavares, meus orientadores durante os meus tempos em Bioquímica, por todos os ensinamentos que me foram passaram aos longo dos anos em que interagimos.

À D. Aurora e ao Celestino da Pastelaria Flôr da Gala, que me deram o meu primeiro trabalho como empregado de balcão numa altura em que mais precisava e não tinha qualquer experiência profissional fora do contexto familiar.

Ao João Vaz e à Ecogestus que, em 2018, aos quais estou especialemnte grato por me terem dado a oportunidade de realizar os meus primeiros trabalhos qualificados e iniciado uma parceria que se tem prolongado até aos dias de hoje.

Aos amigos que deixei durante a minha passagem no programa de Trainees na Leroy Merlin com quem tanto aprendi: particularmente ao Tiago Barreiros, ao Gil Dias, Rui Toulson, Carlos Fernandes, André Fernandes e à Ana Amaral.

A todos vós, o meu muito e sincero obrigado.

ABSTRACT

Learning management systems are essential intermediaries between students and educational content in the digital era. Among other factors, the institutional adoption of such systems is meant to foster student engagement and lead to better educational outcomes in a scalable manner. However, a significant challenge facing educators and institutions is the timely identification of students who may require special attention and feedback. Early identification of students allows educators to provide necessary feedback and adopt suitable corrective measures. Therefore, a significant body of research has been dedicated to developing early warning systems with clickstream data. However, comprehensive studies that attempt prediction on multiple courses are few and far between. Moreover, most predictive models require sophisticated domain knowledge, data skills and computational power that may not be available in practice. In this work, we used an academic year's worth of data collected from all courses at a Portuguese information management school to perform two main experiments on two binary classification problems: the first being students at risk *vs* students not at risk and the second being high-performing students vs not high-performing students.

In the first experiment, we compared the performances obtained with traditional machine learning classifiers against majority class classifiers at multiple stages of course completion (more specifically, the 10%, 25%, 33%, 50% and 100% course completion thresholds). For both classification problems, performances on all metrics peaked when using all of the data collected throughout the course – 88.6% accuracy and 92.3% Area Under the Receiver Operating Characteristic (AUROC) using Random Forest (RF) for students at risk and 78.2% accuracy and 79.6% AUROC using ExtraTrees for high-performing students. Concerning early prediction, acceptable performances for classifying at-risk students are achieved as early as the 25% course duration threshold (72.8% AUROC using RF). Performances for high-performing students were generally lower, with AUROC at earlier stages peaking at the courses' midway point (64.4% AUROC using RF). Our second experiment deployed long-short term memory units (LSTM) trained with a time-dependent representation of a single feature (number of total clicks). While this approach achieved inferior performances, we argue that the more straightforward data pre-processing of this approach may represent a worthwhile tradeoff against relatively small losses in model performance, especially at earlier moments of prediction. We found the best tradeoff at 33% course duration – 64% AUROC against 74% AUROC using RF to predict at-risk students. To predict high-performing students, we found the best tradeoff to occur at 25% course duration (56% AUROC against 61% using RF).

Results obtained using a different set of logs validate the portability of our approach when it comes to static aggregate models. However, our deep learning approach did not generalize well on this data, which suggests that portability between courses using this approach may only be possible in specific instances.

KEYWORDS

Student performance; Learning Analytics; Machine Learning; Classification, Deep Learning

INDEX

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AdaBoost	Adaptive Boosting
ANN	Artificial Neural Network
AUROC	Area Under the Receiver Operating Characteristic
CART	Classification and Regression Tree
EDM	Educational Data Mining
EWS	Early Warning System
HEI	Higher Education Institutions
GBoost	Gradient Boosting
GDPR	General Data Protection Regulation
GPA	Grade Point Average
GRU	Gated Recurrent Units
KNN	K-Nearest Neighbours
LA	Learning Analytics
LAK	Learning Analytics and Knowledge Conference
LGBM	Light Gradient Boosting Machine
LMS	Learning Management Systems
LR	Logistic Regression
LSTM	Long Short-Term Memory Units
ML	Machine Learning
MLP	Multi-layer Perceptron
MOOC	Massive Open Online Course
MOODLE	Modular Object-Oriented Dynamic Learning Environment
NB	Naïve Bayes
Nova IMS	Nova Information Management School
OULAD	Open University Learning Analytics Dataset
PICO	Population/Intervention/Control/Outcome

PRISMA	Preferred Reporting Items for Systematic reviews and Meta-Analyses
RF	Random Forest
RFE	Recursive Feature Elimination
RFECV	Recursive Feature Elimination with Cross-Validation
RNN	Recurrent Neural Networks
ROC	Receiver Operating Characteristic
SMOTE	Synthetic Minority Oversampling Technique
SIS	Student Information System
SRL	Self-Regulated Learning
SVM	Support Vector Machines
URL	Uniform Resource Locator

1. INTRODUCTION

1.1. BIG DATA AND ANALYTICS IN THE CONTEXT OF HIGHER EDUCATION

In the present day, most of the data-intensive approaches to educational research are performed by either the Educational Data Mining (EDM) or the Learning Analytics (LA) research communities (Romero & Ventura, 2020). While formally distinct, both communities share similar goals (to solve problems using data to enhance educational practice), and each community's researchers use identical methods to reach said goals. Moreover, EDM and LA are relatively young communities that have experienced substantial growth over the past decade and are expected to continue to grow in the foreseeable future (Calvet Liñán & Juan Pérez, 2015; Romero & Ventura, 2020). Figure 1 presents the number of papers obtained in Google scholar when searching by EDM and LA.

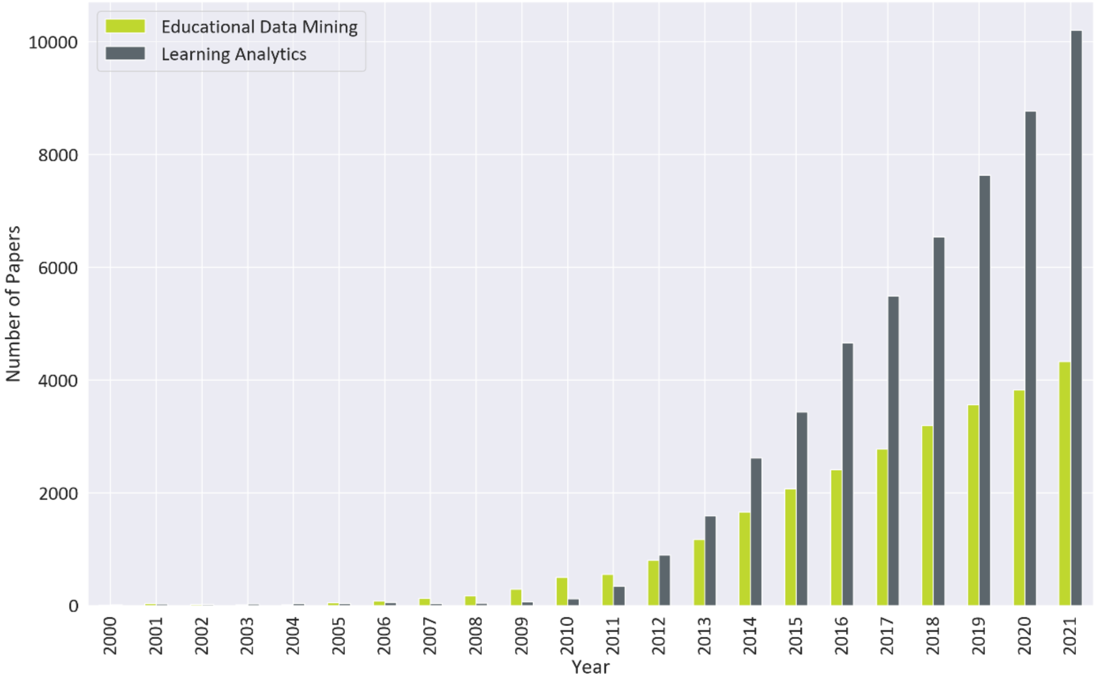


Figure 1.1 – Search results obtained for EDM and LA, by year, in Google Scholar[1]

Higher education institutions (HEI) have unprecedented means, in volume and variety, to digitally gather and store the behavioural footprint left by students as they interact with different university systems (Daniel, 2015). Relevant stakeholders (researchers, faculty members and administrators) recognise that this data has the potential to assist in the fulfilment of either institutional or educational goals (Jones et al., 2020; Romero & Ventura, 2020; Tsai et al., 2020). Thus, HEI data is a prime research subject in a broad spectrum of different educational problems and domains, with student performance being among the most researched (Aldowah et al., 2019; Khan & Ghosh, 2021).

1.2. STUDENT PERFORMANCE AND CLICKSTREAM DATA

To the best of our knowledge, student performance is used as an all-encompassing umbrella term that includes academic success and how academic success is measured (Baker & Yacef, 2009; Romero &

[1] For every year the between 2005 and 2021, the number returned search results for the term "Educational Data Mining" was collected. We repeated the same process for the term "Learning Analytics".

Ventura, 2010; Hellas et al., 2018). The importance of student performance intuitively stands out because metrics such as the Grade Point Average (GPA) are among the main observable proxies for overall academic and professional aptitude universities and private sector employers have while screening applications for scholarships, graduate programs, or employment opportunities (Grove et al., 2006; Imose & Barber, 2015). A student's GPA is computed as an average of the performances displayed across multiple curricular units (referred to as courses from this point onward). Educators use a plurality of internal assessments in each course to measure student performance and represent it as a grade (Shahiri et al., 2015). As most internal assessments of student performance occur at later stages of the course, the educator's ability to identify students who could benefit from timely feedback to adopt meaningful corrective measures is severely hindered (Chickering & Gamson, 1987).

Learning Management Systems (LMS) are feature-rich web applications where students and educators can, among other things, communicate remotely, share course materials or deliver assignments (Coates et al., 2005). Clickstream data are the timestamped records created whenever a student clicks on the LMS (Baker et al., 2020). From a constructivist perspective on learning, according to which learners are not mere recipients of concepts and ideas but active members who should interact with and extract knowledge from the available learning resources (Duffy & Cunningham, 1996), clickstream data, partial and noisy as it may be, encodes patterns of learning behaviour (Baker et al., 2020). Using different approaches, different authors have been able to, with moderate success, obtain early predictions of student performance using exclusively clickstream data (Chen & Cui, 2020; Conijn et al., 2017; Riestra-González et al., 2021).

1.3. OUTLINE

This work presents a basis for portable early warning systems (EWS) that identify at-risk and high-performing students using clickstream data from a Portuguese information management HEI. In more precise terms, the work investigates the following research questions:

1. Can features extracted from LMS data, on their own, predict student performance?
2. Is there a general set of rules/features that can inform academic performance across modalities and courses within NOVA IMS?
3. Can performance be inferred when, at most, 50% of the course is completed?
4. Do time-dependent representations of clickstream data yield comparable results in the early prediction of student outcomes?

To answer the research questions posited above, we use data from 3.2 million logs belonging to 1590 unique students attending 138 unique courses (9296 unique student-course pairings) to predict, at different stages of course completion (10%, 25%, 33%, 50% and 100%), each student’s exam performance. We use various supervised learning techniques to make predictions on a dataset built from the LMS logs. Moreover, we argue that, at least in some instances, adopting a single-feature temporal approach using long short-term memory units (LSTM) yields comparable results to those obtained using multiple features and traditional machine learning classifiers.

The remainder of this work is organised as follows: the next chapter reviews the relevant work that predicts student performance using LMS clickstream data. The third chapter presents our approach to addressing the research questions at hand. The fourth chapter assesses whether our results match the expected outcomes while discussing the work's most relevant implications. The fifth chapter summarises our main findings, and the sixth and final chapter outlines the work’s most significant limitations and future research paths.

2. LITERATURE REVIEW

When preparing for this chapter, we adapted the methods introduced by the Preferred Reporting Items for Systematic reviews and Meta-Analyses (PRISMA) checklist. PRISMA is a 27-item checklist that guides the elaboration of systematic literature reviews (Moher et al., 2009) that was initially designed to be used in healthcare but has seen adoption in other disciplines (Page, McKenzie, et al., 2021).

2.1. PRISMA GUIDELINES

As we do not consider this section to be an exhaustive literature review, our PRISMA adaptation focused primarily on defining objectives aligned with our research questions and developing an appropriate search strategy with clear and explicit criteria for eligibility.

2.1.1. Objectives

Our main objective is to identify and discuss the core approaches, techniques, and predictors used when predicting student performance using LMS data. We are especially interested in understanding:

1. What are the most common targets used in student performance prediction? Moreover, which predictors and techniques are the most successful in predicting student performance?
2. How is predictive power affected by portability (i.e. adoption of a general course-agnostic model) and timeliness of prediction (i.e. early prediction)?
3. How is LMS data organised for predictive purposes?

When appropriate, we identify the research gaps that ultimately led to the formulation of our research questions and highlight the unique contributions this work adds to the space.

2.1.2. Eligibility criteria

For this literature review, we looked at English-written research papers published until December 2021 in peer-reviewed academic journals and conference proceedings associated with data mining, machine learning and, more particularly, the EDM and the LA research communities. As a preliminary approach to identifying eligible research papers, we followed the guidance provided in PRISMA 2020 and adopted the Population/Intervention/Control/Outcome (PICO) structure (Page, Moher, et al., 2021). A brief description of the eligibility criteria for each PICO component is presented in table 2.1.

Table 2.1 – Eligibility criteria of research papers using the PICO framework

Component	Criteria
Population	Eligible surveys attempt to predict the performance of students attending higher education, either at the undergraduate or graduate level.
Intervention	At least partly, performance must be predicted at the course level using LMS log data. Therefore, surveys that do not use LMS or whose only focus is the prediction of a multi-course aggregate (e.g. GPA) were excluded.

Comparison	Surveys must use classification as a predictive method. In addition, there must be a baseline value (e.g. chance, results from the previous year, majority class or even results from other surveys) that allows comparison of results.
Outcome	The authors must have used one or more of the following metrics to assess model performance: Accuracy, Precision, Recall, F1-Score or area under the receiver operating characteristic (AUROC).

A preliminary selection of eligible research papers was conducted based on the title and number of citations. Afterwards, the inclusion or exclusion of each specific research paper was decided using a double-filter strategy similar to that used in López-Zambrano et al. (2021). Using this approach, research papers that did not meet the criteria for inclusion were either disregarded after reading the abstract (first filter) or after reading the full paper (second filter).

2.1.3. Information sources

Between September and December 2021, we extensively and iteratively searched for research papers in different international databases: Google Scholar, Scopus, Web of Science, ACM Digital Library and IEEE Xplore. In addition, we also looked at the conference proceedings of the International Conference on Educational Data Mining[2] and the International Learning Analytics and Knowledge Conference (LAK)[3].

2.1.4. Search strategy

Whenever a search was conducted, we resorted to using the following strings as search terms:

- "Higher education" AND ("Learning Analytics" OR "Educational Data Mining"),
- "Higher education" AND "Performance prediction" AND ("LMS" OR "MOODLE" OR "Clickstream"),
- "Student performance" AND ("Learning Analytics" OR "Educational Data Mining"),
- "Student performance" AND "Prediction" AND ("LMS" OR "MOODLE" OR "Clickstream"),
- "Academic performance" AND "Prediction" AND ("LMS" OR "MOODLE" OR "Clickstream")
- "Student performance" AND ("Portability" OR "Course-Agnostic")

Using this strategy, we found 103 potentially eligible research papers that advanced to the double-filter stage.

2.1.5. Selection process

A single reviewer was responsible for applying the double-filter strategy to all potentially eligible research papers and deciding on inclusion/exclusion according to the PICO criteria outlined in Table

[2] Available at https://educationaldatamining.org/conferences/ (last visited on the 15th of February 2022)

[3] The conference proceedings for LAK are published in the ACM Digital Library but can also be accessed via https://www.solaresearch.org/publications/conference-proceedings/ (last visited on the 15th of February 2022)

2.1. The entire selection process, from which 39 papers were selected as eligible for review, is summarised by the flowchart in Figure 2.1. A detailed literature review table of the selected papers can be found in Appendix A.

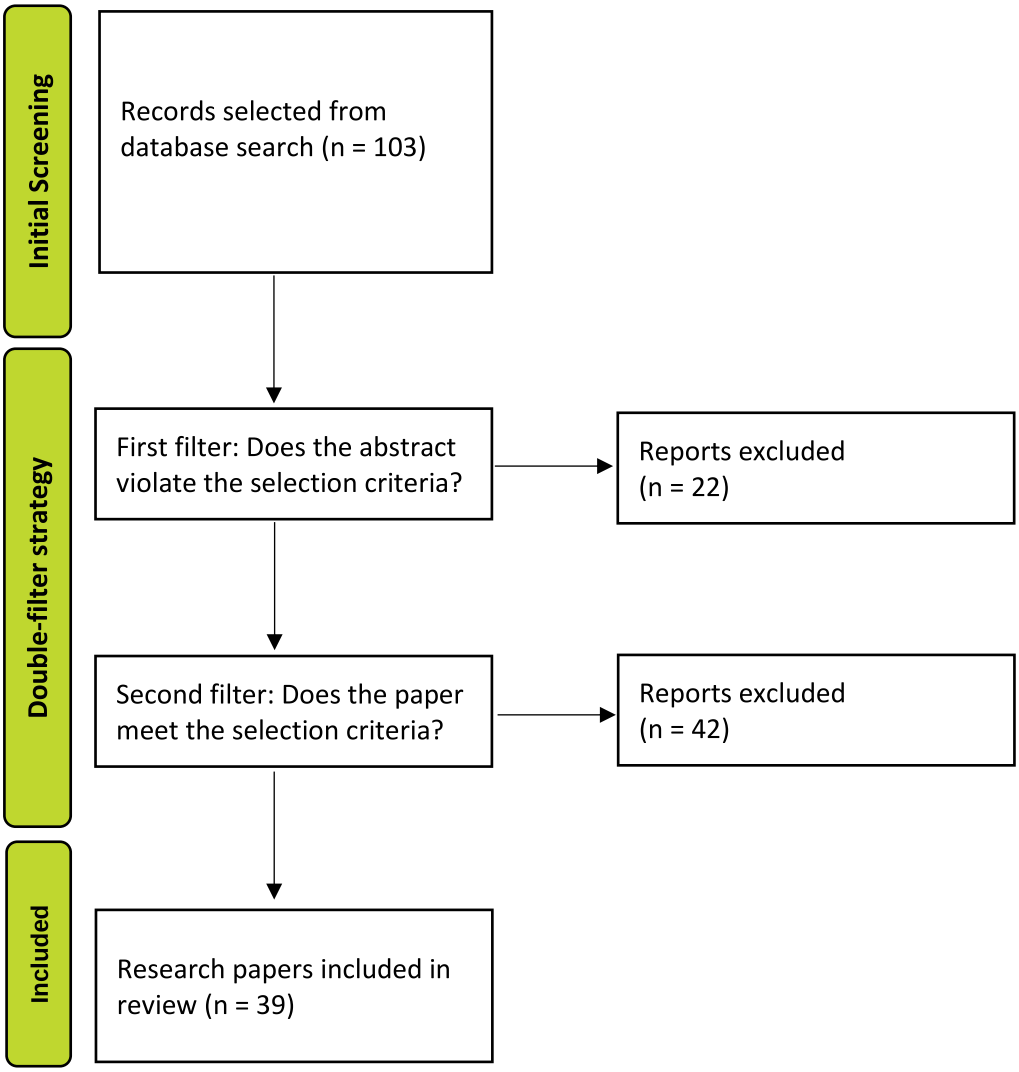

Initial Screening
Records selected from database search (n = 103)
Double-filter strategy
First filter: Does the abstract violate the selection criteria?
Reports excluded (n = 22)
Second filter: Does the paper meet the selection criteria?
Reports excluded (n = 42)
Included
Research papers included in review (n = 39)

Figure 2.1 – Summary of the selection process

2.2. Related work

The simultaneous use of data from different sources – documented in 16 of the 39 selected research papers – often results in increased predictive performance than LMS data alone. For example, models combining LMS features and student characteristics have been shown to outperform baseline models that solely use either student characteristics (Gašević et al., 2016; Sandoval et al., 2018) or LMS clickstreams (Adejo & Connolly, 2018; Waheed et al., 2020; R. Yu et al., 2020). In practice, as educators tend to have limited access to student data early on, their ability to leverage data to create effective early warning systems using data from multiple sources is not impossible (as shown by Kuzilek et al. (2015)) but is undoubtedly hindered. LMS clickstream is one of the primary data sources in student performance research. Often – as shown in 23 of the 39 research papers - LMS is the single data source used when making predictions. Following the PICO criteria in Table 2.1, the following sub-sections will cover the educational research landscape concerning LMS log data usage to predict student performance at the course level.

2.2.1. Key variables when predicting performance

In this sub-section, we start by reviewing the different ways the literature defines course-level student performance to make predictions using classification models. Then, in the second part, we look into the predictive features used to predict performance independently of how performance is defined.

2.2.1.1. Targets

When it comes to the definition of targets for student performance at the course level, most of the reviewed works implement solutions that rely on one of the following key variables: **final mark** (used in 28 research papers), **exam mark** (used in 5) and **dropout** (also used in 4).

A student's final mark is a weighted average of the student's performance across all grading events undertaken in the course (e.g. quizzes, assignments or exams). To meet a course's minimum passing requirements, the student must have a final mark equal to or greater than a specific threshold. More than half of the research papers under review (20 in total) used that rationale to identify whether a student passed or failed the course (Baneres et al., 2019; Brooks et al., 2015; Buschetto Macarini et al., 2019; Calvo-Flores et al., 2006; Chui et al., 2020; Costa et al., 2017; Fahd et al., 2021; Gašević et al., 2016; Hasan et al., 2020; Helal et al., 2018; Kuzilek et al., 2015; López-Zambrano et al., 2020; Mahzoon et al., 2018; Romero, López, et al., 2013; Sandoval et al., 2018; Tsiakmaki et al., 2019, 2020; Waheed et al., 2020; Yu & Wu, 2021; Zacharis, 2018). A second group of researchers followed a similar, albeit more nuanced, rationale. Macfadyen & Dawson (2010), Saqr et al. (2017) and Zacharis (2015) opted not to stick to the rigid pass/fail paradigm and joined barely passing students with failing students into students at risk of failing. In a slightly different way, the binary decision boundaries drawn by Chen & Cui (2020), Huang et al. (2020) and R. Yu et al. (2020) discriminated between good and worse performance without much consideration for the final mark's impact on the student's success (e.g. the latter work only splits between having a final mark above or below the median grade). Moreover, the final group of approaches went beyond binary classification and into the realm of multi-class prediction. Chui et al. (2020) ran an experiment that discriminates students between *Passing*, *Barely Passing* and *Failing*. Aljohani et al. (2019) adapted the pass or fail approach by introducing the classes *Withdraw* and *Distinction*. These approaches were similar to the second experiment performed by Romero, Espejo et al. (2013) using discretised data, whose first experiment using continuous data treated each possible value of the final mark (integers from 0 to 10) as a discrete class. The use of the actual final mark as a target is more common in regression approaches (Gašević et al., 2016; Zacharis, 2015) which go beyond the scope of our work.

The grade obtained by the student in the course's final exam mark is also featured as a viable outcome variable that is converted into a classification target similarly to the final mark. The works under review using the exam mark either strictly follow the pass or fail dichotomy (Casey & Azcona, 2017; Fahd et al., 2021; Marras et al., 2021; Tomasevic et al., 2020) or make minor adaptations to where the decision boundary is drawn to also include bare passers as students at risk (Conijn et al., 2017). As the exam mark usually makes up for a significant proportion of the final mark, its use may seem unnecessary at first glance. However, as the exam grade is not directly computed from other graded records, its use allows, where applicable, intermediary grades to be used as performance predictors (Conijn et al., 2017).

The third featured outcome variable focuses on whether or not a student gives up on attending the course. Student attrition is a cause for concern for HEI (Adejo & Connolly, 2018), and the research efforts using this outcome were mainly focused on the development of tools to identify if a student will dropout or not (Adejo & Connolly, 2018; Tsiakmaki et al., 2019; Waheed et al., 2020; Whitehill et al., 2017; Xing & Du, 2019). In other works, dropout students were generally treated as students who failed, with only Aljohani et al. (2019) explicitly including *Withdraw* as a possible class in their multi-class implementation of the final mark into a pass or fail classifier.

The remaining research works assembled their classifiers using none of the previously mentioned outcome variables. The approach by Yu et al. (2019) discriminated between students who earned a certificate in a Massive Open Online Course (MOOC) and those who did not. A different approach was adopted by Yang et al. (2020), who first used the interaction patterns and homework scores to create 6 clusters of students and assigned a final mark to each cluster (from F to A) and then, in a second stage, used the resulting clusters as the prediction target. Riestra-González et al. (2021) also predicted targets obtained from an inferred final mark. In that case, the final mark was estimated from the available grades of assignments. Later, the inferred final mark was used to create three parallel classification problems: students at risk *vs* not at risk, pass *vs* fail and excellent students *vs* not excellent students.

2.2.1.2. Features

The literature showcases varied and distinct approaches concerning the LMS and the features extracted from them (Conijn et al., 2017). Feature engineering is regularly left to the researcher's discretion and is consequently limited by the researcher’s domain knowledge and available resources (Tomasevic et al., 2020; Tsiakmaki et al., 2019). Furthermore, different researchers provide different justifications for their LMS feature usage: some authors grounded their choice on features adopted by other researchers beforehand (Conijn et al., 2017)), and others opted to ground their choices solely on learning theory (e.g. the importance of forum usage by Romero, López, et al. (2013)), or self-regulated learning (SRL) theory by Gašević et al. (2016) and Saqr et al. (2017)). In other instances, the authors do not justify their choice of features (Calvo-Flores et al., 2006; Romero, Espejo, et al., 2013) or even go as far as not disclosing any specific features used for predictive modelling (Baneres et al., 2019; Chui et al., 2020). Moreover, there are some questions on whether a *de facto* general set of predictive features obtainable from LMS even exists, with the works by Gašević et al. (2016) and Conijn et al. (2017) using the small number of statistically significant features with consistent effects across multiple courses as an explicit argument against the use of general predictive models. These arguments against general models open the door to discussing the portability of the predictive models, which we will cover later in section 2.2.2.2.

Nevertheless, some discernible common trends have emerged from our reading of the published research works. A noteworthy standalone mention goes to grades obtained by the students in intermediary quizzes, assessments or assignments (which, going forward, we will refer to as partial grades). While this feature is not present in most works under review, it was found to be among the most influential when used (Conijn et al., 2017; Costa et al., 2017; Riestra-González et al., 2021).

An essential set of popular features is activity counts: the number of times a student performs a particular action. The use of features that count actions is overwhelmingly popular in the literature, as only 4 of the 39 papers under review do not explicitly refer to using this sort of feature in their predictive models (Baneres et al., 2019; Chui et al., 2020; Marras et al., 2021; Yang et al., 2020)[4]. Popular choices in the literature range from general features, such as the number of online sessions (used in 11 papers) or the total number of clicks (used in 8 papers), to other, more specific features, such as the number of clicks on course forums (used in 12 papers), the number of resources viewed (used in 13 papers) or even the number of assignments submitted (used in 9 papers). Unless under particular circumstances (e.g. evaluating the effects of forum interactions in performance as performed by Romero, López, et al. ((2013)), authors tend to extract features that reference different activities. For example, López-Zambrano et al. (2020) extracted activity counts from 50 log event types, while Aljohani et al. (2019) aggregated counts of log events into 20 different activities.

Time-related features are also relatively popular. Unlike activity counts, these features are generally computed from the timestamps to estimate the time a student spends engaging with the course contents (time on task). The most popular application we found in the literature was estimating the total time spent online (used in 11 papers). However, many authors have found other applications for this type of feature. Other use cases include, among others, finding how frequently a student accesses the LMS (Costa et al., 2017), the amount of time a student is not engaging with the system in a lab-session environment (Fahd et al., 2021) or how close to the deadline are students delivering their homework assignments (Yang et al., 2020). In total, time-related features were used in 20 research papers (Adejo & Connolly, 2018; Casey & Azcona, 2017; Chen & Cui, 2020; Conijn et al., 2017; Costa et al., 2017; Fahd et al., 2021; Hasan et al., 2020; Hu et al., 2014; Huang et al., 2020; Macfadyen & Dawson, 2010; Marras et al., 2021; Riestra-González et al., 2021[5]; Romero, Espejo, et al., 2013; Romero, López, et al., 2013; Saqr et al., 2017; Tomasevic et al., 2020; Whitehill et al., 2017; Yang et al., 2020; R. Yu et al., 2020; Zacharis, 2015).

The literature also has a place for other, more complex predictive features. For example, Marras et al. (2021) presented a new set of features based on *alignment*, *anticipation* and *strength* when using content, and Yu et al. (2019) created n-grams from the sequences of click types students made when watching pre-recorded video lectures. However, complex approaches also raise concerns over their practicality, which lead other authors to argue for approaches that are less reliant on intensive and expensive data pre-processing and feature engineering, with Chen & Cui (2020) obtaining promising results using clicks per day as the single feature.

2.2.2. Relevant distinctions in experimental design

Going past the differences in approach when it comes to the outcome variables and features used, the research works also differ in other aspects. In this sub-section, we start by reviewing how authors differ regarding when a prediction is made throughout the course duration. Then, we look at the portability (i.e. course-agnosticism) of the predictive models used in the literature. The sub-section ends with a review of researchers' possible data analysis strategies when manipulating clickstream data for predictive purposes.

[4] The work by Riestra-González et al. (2021) converts raw activity counts into relative variables.

[5] The work by Riestra-González et al. (2021) also converts time-related features into relative variables.

2.2.2.1. Moment of prediction

A significant body of work is dedicated to *post-hoc* predictions using data collected throughout the entire duration of the course - 15 of the 39 research papers under review exclusively made predictions (Adejo & Connolly, 2018; Calvo-Flores et al., 2006; Casey & Azcona, 2017; Chui et al., 2020; Gašević et al., 2016; Helal et al., 2018; Huang et al., 2020; López-Zambrano et al., 2020; Macfadyen & Dawson, 2010; Mahzoon et al., 2018; Romero, Espejo, et al., 2013; Tsiakmaki et al., 2020; Yang et al., 2020; Zacharis, 2015, 2018). Unfortunately, while this line of research has explored the potential of LMS features as effective predictors of student performance, it does not give educators the means to provide timely interventions to students in need.

The earlier the moment of prediction, the more time a student has to make adjustments. EWS aim to identify students of interest while there is still time to provide helpful feedback (Macfadyen & Dawson, 2010). In other words, early prediction requires the exclusive use of data generated up to the moment of prediction (Hu et al., 2014). It is possible to find research work predicting performance every week (Aljohani et al., 2019; Buschetto Macarini et al., 2019; Casey & Azcona, 2017; Conijn et al., 2017; Costa et al., 2017; Kuzilek et al., 2015; Marras et al., 2021; Whitehill et al., 2017; Xing & Du, 2019; C.-C. Yu & Wu, 2021; C.-H. Yu et al., 2019), after each assessment (Baneres et al., 2019; Kuzilek et al., 2015; Tomasevic et al., 2020) or at other stages of course completion (Brooks et al., 2015; Chen & Cui, 2020; Fahd et al., 2021; Hu et al., 2014; Riestra-González et al., 2021; Romero, Espejo, et al., 2013; Sandoval et al., 2018; Saqr et al., 2017; Tsiakmaki et al., 2020; Waheed et al., 2020; R. Yu et al., 2020). In most works, predictive performance tends to increase at later moments of prediction. However, reasonable tradeoffs between early prediction and predictive performance have been reached by the courses' halfway point (e.g. Riestra-González et al. (2021) reached AUROCs above 0.90 for all classification targets).

2.2.2.2. Portability

In the context of this work, we refer to portability as a model's ability to generalise to multiple courses. Works on performance prediction on a single course have been historically popular in the literature – as showcased in 17 papers (Adejo & Connolly, 2018; Buschetto Macarini et al., 2019; Calvo-Flores et al., 2006; Casey & Azcona, 2017; Chen & Cui, 2020; Fahd et al., 2021; Hu et al., 2014; Macfadyen & Dawson, 2010; Mahzoon et al., 2018; Marras et al., 2021; Romero, López, et al., 2013; Saqr et al., 2017; Xing & Du, 2019; Yang et al., 2020; C.-H. Yu et al., 2019; Zacharis, 2015, 2018). In the papers making predictions of multiple courses, some use the same model to make predictions on multiple courses and others use one predictive model for each course.

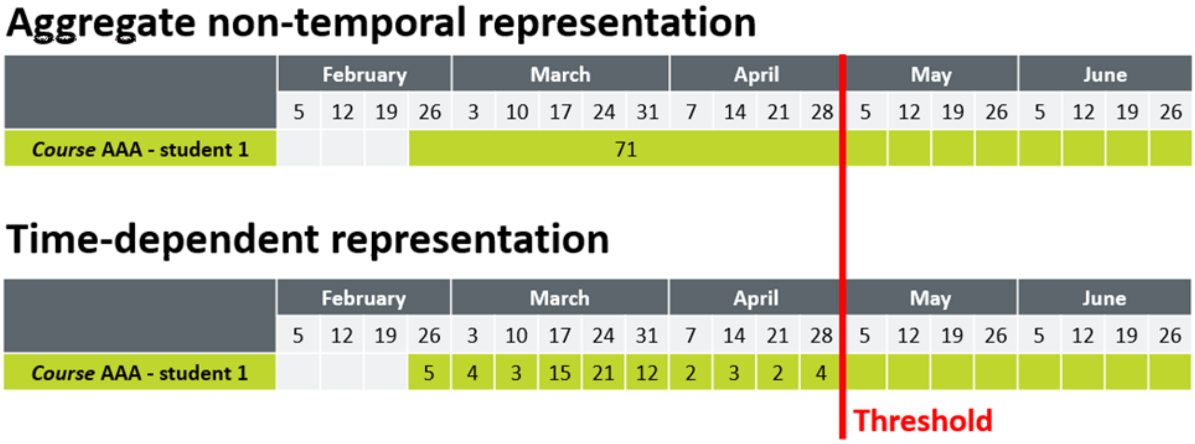
Four multi-course research works exclusively rely on course-specific approaches (Brooks et al., 2015; Chui et al., 2020; Huang et al., 2020; Tsiakmaki et al., 2019). In addition, a transfer learning approach by Tsiakmaki et al. (2020) found that the accuracy of a deep learning model trained on the course itself tended to be lower than that of models trained on data from another course.

Course-agnostic approaches assume that LMS interactions reflect patterns of behaviour transferrable from one course to another and that those patterns are predictive of performance. The development of general models is an active research topic that, over the years, has seen research works gradually improve in predictive performance: starting with an accuracy of 0.66 obtained by Romero, Espejo et al. (2013) on data from seven courses, Gašević et al. (2016) reaching 0.749 AUROC with a nine-course general Logistic Regression (LR)-based classifier model, Costa et al. (2017) managed F-scores over 0.80 at the courses' halfway point and 0.90 by the end of the courses using Support Vector Machines (SVM) on data from 2 courses. More recently, Aljohani et al. (2019) and Yu & Wu (2021) have achieved accuracies of 0.952 and 0.93 with deep learning models. However, there are also sceptical perspectives on the potential value of creating general models. For example, Gašević et al. (2016), closely followed by Conijn et al. (2017) and, to some extent, López-Zambrano et al. (2020) have all concluded that, for most general models, there is a significant dropoff in predictive performance in comparison to course specific models.

2.2.2.3. Strategies for data analysis

Due to their usefulness for this specific section, we adopted concepts used in a paper that did not meet our PICO criteria (Baker et al., 2020). The authors identify two primary data analysis strategies when working with clickstream data: the first being the use of *static aggregate representations* and the second being *time-dependent* or *sequence-dependent representations*. Figure 2.2 illustrates how, depending on the chosen data analysis strategy, the same feature of a hypothetical student is represented differently.

In static-aggregate representations, each student is treated as a flat multidimensional vector. While it loses the sequential aspects of the information, this representation is easier to work with in statistical analyses. Therefore, it is the most widely adopted, as demonstrated by the 34 research papers under review that adopted this format (Adejo & Connolly, 2018; Baneres et al., 2019; Buschetto Macarini et al., 2019; Calvo-Flores et al., 2006; Casey & Azcona, 2017; Chui et al., 2020; Conijn et al., 2017; Costa et al., 2017; Fahd et al., 2021; Gašević et al., 2016; Helal et al., 2018; Hu et al., 2014; Huang et al., 2020; Kuzilek et al., 2015; López-Zambrano et al., 2020; Macfadyen & Dawson, 2010; Marras et al., 2021; Riestra-González et al., 2021; Romero, Espejo, et al., 2013; Romero, López, et al., 2013; Sandoval et al., 2018; Saqr et al., 2017; Tomasevic et al., 2020; Tsiakmaki et al., 2019, 2020; Waheed et al., 2020; Whitehill et al., 2017; Xing & Du, 2019; Yang et al., 2020; C.-H. Yu et al., 2019; R. Yu et al., 2020; Zacharis, 2015, 2018).



Aggregate non-temporal representation

	February				March					April				May				June			
	5	12	19	26	3	10	17	24	31	7	14	21	28	5	12	19	26	5	12	19	26
Course AAA - student 1				71																	

Time-dependent representation

	February				March					April				May				June			
	5	12	19	26	3	10	17	24	31	7	14	21	28	5	12	19	26	5	12	19	26
Course AAA - student 1				5	4	3	15	21	12	2	3	2	4								

Figure 2.2 – Aggregate non-temporal representation and time-dependent representation of the same feature (e.g. number of clicks) using the same clickstream data

In time-dependent representations, each student is represented by a number of time series equal to the number of features. Researchers who adopt this format can access and extract latent information that would be lost otherwise. Historically, there is a comparatively low number of tools and techniques to work with when using this format. Moreover, retaining sequential patterns whilst creating representations amenable to working with the classifiers used on aggregate data is not trivial (Brooks et al., 2015; Mahzoon et al., 2018). In recent years, the rise in popularity of deep learning techniques like Recurrent Neural Networks (RNN) and LSTM has contributed to an increase in works in time-dependent representations (Aljohani et al., 2019; Chen & Cui, 2020; Yu & Wu, 2021).

2.2.3. Knowledge gaps

The use of LMS logs to predict student performance is extensively documented in the literature and has respectable results using multiple approaches. However, to the best of our knowledge, the work by Riestra-González et al. (2021) is the only institution-wide course-agnostic approach that exclusively uses LMS logs and partial grades in predictive modelling. Therefore, assessing whether similar results can be obtained using data from other institutions is relevant. Moreover, while the work showed promising results using a target computed from an inferred grade, it remains an open question whether using actual marks would work just as well.

Another open question is the ability of such course-agnostic models to work using approaches that are less reliant on extensive domain knowledge and pre-processing (i.e. what Sandoval et al. (2018) called low-cost variables). To that effect, we looked at the work by Chen & Cui (2020), who use a single feature in time-dependent representation to reach comparable performances to that of full-fledged multi-feature static aggregate representations on standard machine learning (ML) classifiers.

To that effect, our work introduces multiple novel contributions: first, it introduces a course-agnostic approach to early warning systems meant to predict exam performance in a sample of over 90 courses. Secondly, to our knowledge, our work has the largest sample of courses used to date when using deep learning to predict student performance. Finally, we introduce an early warning system that uses a time-dependent representation of a single feature in a course-agnostic setting.

3. METHODOLOGY

We used the data and methods described in this section to answer our research questions. Our manipulations and exploratory analyses were made using version 3.8.3 of the python programming language (McKinney, 2018). Moreover, most feature selection and traditional ML classifier models trained on static aggregate representations used the implementation available in the Scikit-learn (Pedregosa et al., 2011) package. All time-dependent deep learning implementations used PyTorch (Paszke et al., 2019). Our entire methodology can be found using the link https://github.com/RicardoSantos0/Msc_thesis.

3.1. DATA

Nova Information Management School (Nova IMS)[6] is an information management school that is part of Universidade Nova de Lisboa[7] that offers undergraduate, graduate and executive programs in Data Science, Information Management, and Information Systems and Technologies. In this work, we extracted the MOODLE logs for all courses taught at Nova IMS throughout the 2020/2021 academic year. In addition, we obtained the students' partial and final grades from the school's Student Information System (SIS). From the inner join of the information extracted from both systems, we obtained close to four million logs performed by 2140 unique students attending, in total, 222 courses. All student data were anonymized in compliance with the General Data Protection Regulation (GDPR). Moreover, the Nova IMS ethics committee reviewed and approved the study.

As we are considering courses that belong to different programs at both the undergraduate and graduate levels with different educators and grading schemas, we hypothesized that there is significant heterogeneity in MOODLE usage patterns between courses. Our hypothesis found support in Figure 3.1, which showcases the number of clicks made each week for each Nova IMS course: some courses have tens of thousands of clicks on MOODLE in a given week, while others do not go past 100 clicks. It should be noted that this heterogeneity occurred during an academic year where classes were held remotely due to Covid-19 protection measures and during which MOODLE was ubiquitously used across Nova IMS courses.

[6] https://www.novaims.unl.pt/en/

[7] https://www.unl.pt/en

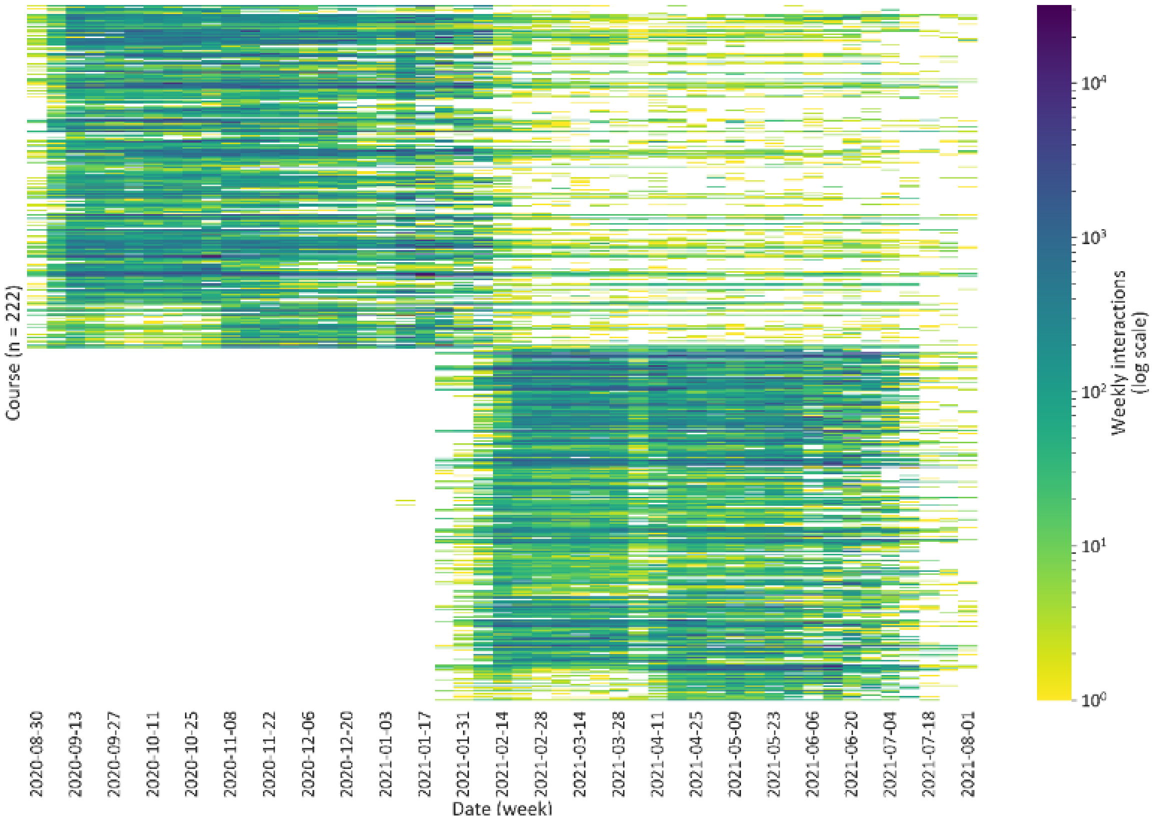


Figure 3.1 – Total number of LMS interactions by all students per week in each course

3.2. Data Pre-Processing

3.2.1. Setting target variables

We were provided with each student's exam mark and final mark. In the Portuguese system, grades vary between 0 and 20. To be approved, a student's final mark must be equal to or greater than 10. This work uses the exam mark as the outcome variable of interest. More specifically, we use the exam mark to create target variables for two different binary classification problems:

- **Students at risk vs students not at risk**: for this classification problem, all students who obtained an exam grade equal to 11 or less were labelled as students at risk (1), with the remaining student population being labelled as not at risk (0),
- **High-performing students vs Not high-performing students**: To compute this target, we calculate, for each course, the grade matching the 85th percentile in the course's distribution of grades as the decision threshold. Students whose exam marks surpassed the threshold were labelled as high-performing students (1), with the remaining students becoming not high-performing students (0).

3.2.2. Setting course duration thresholds

One of the most important factors to consider when creating an EWS is the setting at which points in time a prediction has to be made. An educator's ability to identify students of interest tends to be better when the prediction is made at later stages of the course. Conversely, it is in the interest of educators to identify students in need as early as possible. In this work, we set our early moments of prediction similarly to Riestra-González et al. (2021): with predictions made at 10%, 25%, 33% and 50% stages of course duration. In addition, we include a prediction after course completion. To calculate these duration thresholds, we used, for each course, the starting date and the regular season final exam date.

To ensure that predictions were made at the intended stages of course duration, we created a different set of logs for each course duration threshold – five in total. In each set of logs, we only kept the interactions made between the week before the start of the course and the corresponding threshold date. As we have pre-emptively created different sets of logs for each duration threshold, we note that all methods described from this point onward were performed five times in total – one for each course duration threshold.

3.2.3. Feature engineering – static aggregate representation

Our first step was to separate the logs by course. For each student in each course, we computed 31 predictive features from the logs, most previously used in works found in the literature. From the MOODLE logs, we computed 14 activity count features (total clicks – raw count and as a percentage, online sessions, forums clicks, forum posts, discussions viewed, course clicks, folder clicks, resources viewed, Uniform Resource Locators (URL) viewed, assessments started, assignments viewed and assignments submitted - raw count and as a percentage), 12 time-related features (total time online, largest period of inactivity and the relative starting time for each one of the first ten sessions) and five complex features (clicks/day, clicks/session, the average duration of each session and number of days with zero clicks – raw count and as a percentage).

As we have access to partial grades, we also computed the average of the student's partial grades and used it as a feature. The Nova IMS SIS stores each student's partial grades, but it neither stores what event the partial grade refers to nor when that event occurs. Therefore, computing this feature was trivial for the 100% course duration threshold but relatively challenging for shorter course completion thresholds. A preliminary naïve approach could assume a one-to-one relationship between submissions and partial grades (the first partial grade would refer to the first submission, the second partial grade to the second submission and the same would apply to the remaining partial grades). However, such an approach would not account for other relevant factors, such as not all partial grades resulting from individual quizzes or assignments. For example, we find it unlikely that all group members would have submitted the same group project separately. Ultimately, our solution associates partial grades with submissions but with some caveats. We start by assuming that in courses with multiple group projects, all submissions tend to be made by the same student and, for these students, that the ith submission corresponds to the event responsible for the ith partial grade. Thus, we chronologically ordered the submissions made by each student and assigned an ordinal number to each submission (where 1 represents that student's first submission to the course, 2 represents the student's second submission to the course and so on). Then, we considered that the delivery deadline for the ith grading event matches the median date of the students' ith submissions. Using this method, we assigned each partial grade to a date that was later used to decide whether a partial grade should be considered at each course duration threshold.

After computing all features, we had, for each duration threshold, a dataset with 11297 rows containing 1677 unique students attending, in total, 181 unique courses (each row representing a unique student-course pair). A complete list of the 32 features we computed and their description is presented in Appendix B’s Table B.1.

3.2.4. Course filtering

Using the dataset with the features extracted from the 100% duration logs, we filtered out all courses that met at least one of the following conditions: having less than 25 students; having the median student not make a click in at least 85% of the days of the course duration; having no at-risk or no-high performing students as defined in section 3.2.1. The resulting dataset had 9296 student-course pairings, representing 1590 unique students attending 138 courses.

For consistency purposes, we also filtered the remaining datasets (one for each course duration threshold) to ensure that all experiments' student population was identical. Table 3.1 shows how the population is distributed depending on the classification target.

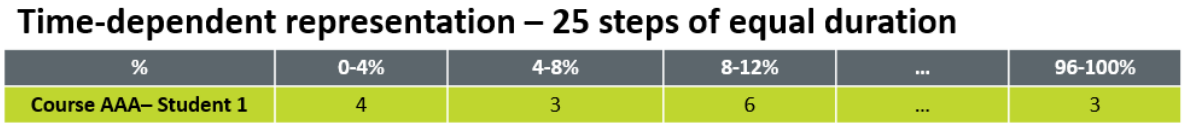
Table 3.1 – Class representation in each classification problem

Classification target	Yes	No
Students at risk	1872 (20%)	7424 (80%)
High-performing students	2574 (28%)	6722 (72%)

3.2.5. Feature engineering – time-dependent representation

While creating the datasets made of static aggregate representations, we parallelly created datasets featuring a time-dependent representation of the number of total clicks for each duration threshold (10%, 25%, 33%, 50% and 100%). Our goal was to make all sequences have the same number of steps, regardless of the student-course pair. However, our experimental setting included multiple courses whose duration could either be one trimester or one semester. Moreover, we also intended to create one dataset for each course duration threshold, increasing the problem's difficulty. Previous works using time-dependent representations considered a single course or a relatively small number of courses with similar durations. That simplicity allowed researchers to extract sequential data daily (Chen & Cui, 2020) or even weekly (Aljohani et al., 2019; Mahzoon et al., 2018; Yu & Wu, 2021). While intuitive, these approaches do not address our need to create sequences with the same number of steps from courses with different temporal lengths.

Our approach towards temporal representations of multiple courses involved treating time as a relative variable. For every course and duration threshold, we started from the logs generated in section 3.2. and computed the total time elapsed between the week before the start of the course and the course’s duration threshold date. Then, we divided that time into 25 splits, each representing 4% of the period under consideration and, in each split, counted the total clicks made by each student. A representative row of the datasets created using this approach is displayed in Figure 3.2.



Time-dependent representation – 25 steps of equal duration

%	0-4%	4-8%	8-12%	...	96-100%
Course AAA– Student 1	4	3	6	...	3

Figure 3.2 – Time-dependent representation of one feature (e.g. number of clicks) using our proposed split of the period in 25 steps of equal duration

3.3. Predictive models

In this section, we cover the predictive models used in this work. We start by covering the feature selection techniques and classification algorithms used for the datasets built on a static aggregate representation of clickstream data. Then, we introduce the structure of the deep learning model we trained using the time-dependent representation of the number of clicks. Finally, the section ends with a review of the metrics we chose to assess model performance.

3.3.1. Static aggregate representation

To answer our first, second and third research questions (and provide a baseline for comparison of our fourth research question), we used the methods described in this sub-section on the static datasets obtained from section 3.2.3. Each dataset, one per course duration threshold, contained the features presented in Appendix B's Table B.1.

3.3.1.1. Feature selection techniques

The use of feature selection techniques contributes to the creation of less computationally expensive models that tend to simultaneously have better predictive performances than models that apply no feature selection techniques (Chen & Cui, 2020; Hasan et al., 2020; Romero, Espejo, et al., 2013).

We used a multi-layered feature selection process with eight algorithms in this work. To not be excluded, a feature had to be selected by at least half of the algorithms. Seven of the algorithms used in this process used the respective Scikit-learn implementations (Pedregosa et al., 2011): Recursive Feature Elimination (Guyon et al., 2002) both in its simple form (RFE) and with cross-validation (RFECV); multiple forms of regression such as Ridge, Lasso, Logistic and ElasticNet; and Random Forest (RF), which is a tree-based ensemble method. The eighth algorithm we used was Light Gradient Boosting Machine (LGBM), a very efficient tree-based learning algorithm with no Scikit-learn implementation (Ke et al., 2017). A complete list of the feature selection algorithms and their respective hyper-parameters is available in Appendix C's Table C.1.

3.3.1.2. Classification algorithms

To predict student performance from the static aggregate data, we used the Scikit-learn implementations of ten traditional ML classification algorithms: K-Nearest Neighbors (KNN), LR, Naïve Bayes (NB), Classification and Regression Tree (CART), Multi-layer Perceptron (MLP) which is also commonly referred to as Artificial Neural Network (ANN), SVM, RF, ExtraTrees Classifier, Adaptive Boosting (AdaBoost) and Gradient Boosting (GBoost).

A complete list of the models and the hyper-parameters used is available in Appendix C's Table C.2. For CART, LR, MLP and SVM we adopted the hyper-parameters that Riestra-González et al. (2021)

determined to have the best results when discriminating between passing and failing students at the 50% course duration threshold.

3.3.2. Time-dependent representation

To address our fourth and final research question, we use a 25-step time-dependent representation of the number of clicks. We use LSTM networks to predict student performance from time-series LMS behaviour. Our implementation of LSTM uses the PyTorch package (Paszke et al., 2019).

3.3.2.1. Overview of LSTM networks

RNNs are a family of neural networks especially apt to handle sequential data. At the core, a RNN is deceptively simple: they take as input a vector *x* and output a different vector *h*. However, a crucial distinction of RNN from other neural networks is that for a given timestep *t*, the output (h_t) is not only influenced by the input of the current timestep (x_t) but also by all previous inputs as well. RRN have internal loops that allow information to flow from one timestep to the next (Rumelhart et al., 1986). As showcased in Figure 3.3, the RNN loop can also be understood as a sequence of multiple copies of the same cell that inherit information from the previous cells.

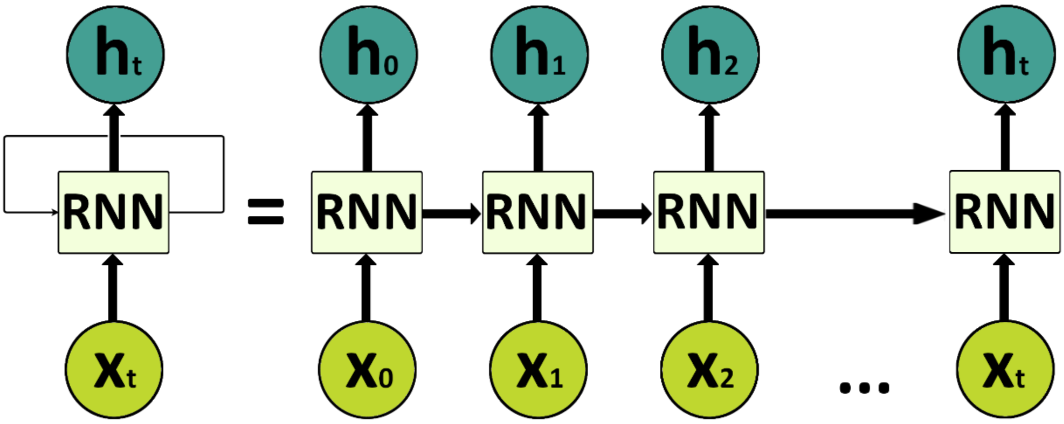


Figure 3.3 – Representation of a RNN

RNNs have historically exhibited good performances when working with small sequences. However, they are prone to suffer from either the exploding or vanishing gradient problems, which severely hinder the ability of RNNs to retain information across longer sequences (Pascanu et al., 2013). Hochreiter & Schmidhuber (1997) presented LSTM, an RNN-based architecture, to address the vanishing gradient problem. The advantage of LSTM is that they feature an internal cell state ***C,*** which can be thought of as the cell's memory, that flows between self-connected cells with minimal perturbations. At each timestep, the vanilla LSTM protected ***C*** via two multiplicative gates: an input gate (i_t) that softened the effect of irrelevant inputs and an output gate (o_t) that protected future units from inheriting irrelevant memories of the current cell. As a result, LSTM achieved state-of-the-art on previously unsolvable problems with O(n) complexity – the same as RNN.

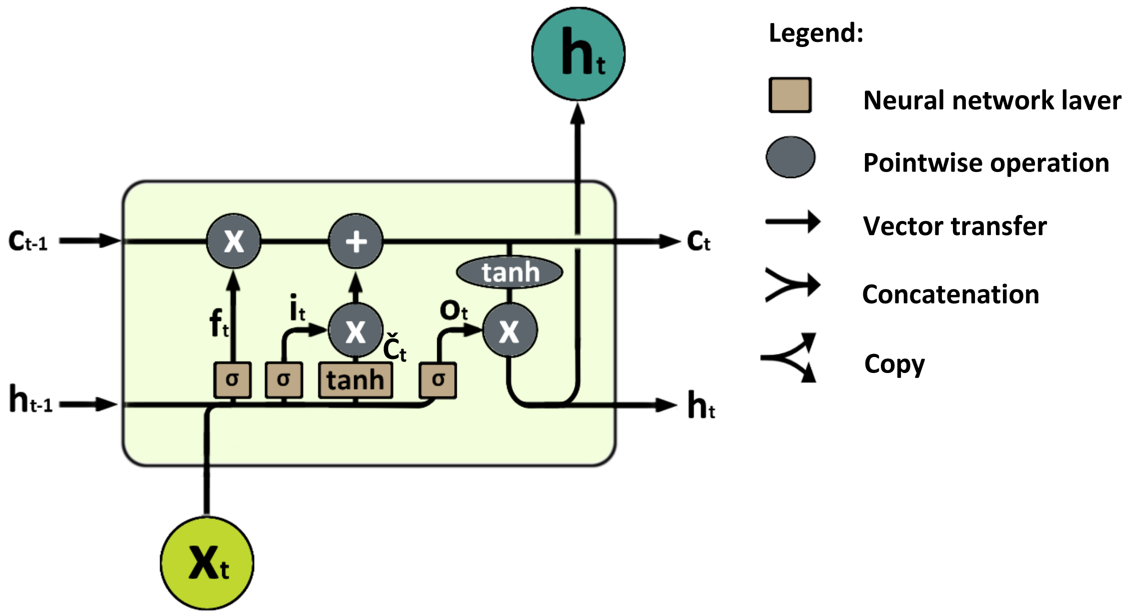


Figure 3.4 – LSTM architecture

The current form of LSTM, displayed in Figure 3.4, results from the contributions of multiple researchers. Its structure features the same input and output gates found in the vanilla LSTM with an added forget gate (f_t) introduced later by Gers et al. (2000). Let W_g and b_g represent the weights and biases of a specific gate. At a given timestep t, the current unit starts by receiving cell state $\boldsymbol{C_{t\text{-}1}}$ and hidden state $h_{t\text{-}1}$ from the preceding unit and input x_t and has to choose what information is to be kept and what information can be thrown away. The first step is to look at $\boldsymbol{C_{t\text{-}1}}$ and x_t and decide what information of the preceding cell states can be discarded, a decision made at the forget gate using equation (1).

$$f_t = \sigma(W_f \cdot [\,h_{t-1}, x_t] + b_f)$$

(1)

The second step is to decide what new information will update $\boldsymbol{C_t}$. That is decided by the input gate, described by equation (2). The results are combined with a vector of candidate values $\boldsymbol{\check{C}_t}$ described by equation (3). The new cell state $\boldsymbol{C_t}$ is determined by equation (4), combining the previous equations' results.

$$i_t = \sigma(W_i \cdot [\,h_{t-1}, x_t] + b_i)$$

(2)

$$\check{C}_t = \tanh(\,W_C \cdot [\,h_{t-1}, x_t] + b_C)$$

(3)

$$C_t = f_t C_{t-1} + i_t\,\check{C}_t)$$

(4)

The final step is deciding whether the current unit's cell state should be allowed to perturb future units. That is determined by the output gate, determined by equation (5). Finally, the new hidden state h_t results from the product between the output gate and the current cell state, as described by equation (6).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t \tanh(C_t) \tag{6}$$

3.3.2.2. Modelling student clicks with LSTM networks

In this work, we examine the potential of a single-feature LSTM network to make early predictions of student performance. To that end, we implement an architecture inspired by Chen & Cui (2020): the input nodes have a single LSTM layer on top that is followed by a 50% dropout layer and then a fully connected dense layer with a Softmax activation function that returns the probabilities for each class.

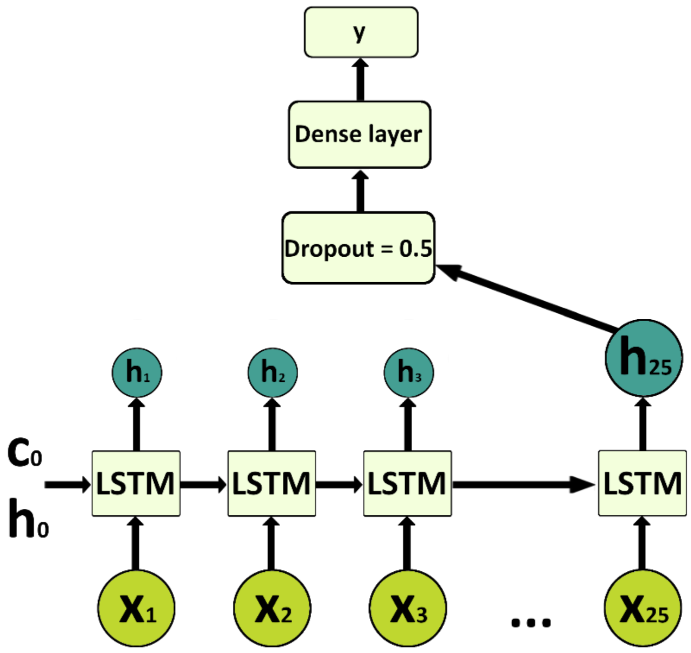


Figure 3.5 – 25-step implementation of LSTM to predict student performance from clickstream data

All of our deep learning implementations used an LSTM layer with 25 self-connected units to match the number of timesteps used in creating the time-dependent datasets documented in section 3.2.5. The weights for cell state C_0 and hidden state h_0 were initialized using Xavier (or Glorot) normal initialization (Glorot & Bengio, 2010), and only the final hidden state h_{25} was sent to the subsequent layers. The dense layer has 40 nodes which is the same size as the output of LSTM output. Training of the model was made in batches of 32 observations throughout 200 epochs. PyTorch's CrossEntropyLoss() was our loss function of choice and Adam was chosen as the optimizer (Kingma & Ba, 2017). A complete list with all hyper-parameters used for LSTM networks, most of them also used by (Chen & Cui, 2020), is presented in Appendix C's Table C.3.

3.3.3. Model performance metrics

To evaluate the performance of our predictive models, we used four metrics. The mathematical definition of accuracy, precision and recall is shown in the following equations:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{7}$$

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

The fourth metric we used to assess model performance was AUROC, a popular model performance metric in education (as showcased by its usage in 14 of the 39 papers used in our literature review). In a binary classification problem, the receiver operating characteristic (ROC) curve plots the fraction of zeros misclassified as ones (false positive rate – defined in equation (10)) on the x-axis against the fraction of ones correctly assigned by the classifier (true positive rate or recall) on the y-axis. AUROC measures the classifier's ability to discriminate between classes correctly. AUROC varies between 0 and 1, with 0 meaning a classifier that got all predictions wrong, 0.5 meaning that the classifier cannot discriminate between classes (i.e. always predicts the same class) and 1 meaning that the classifier perfectly predicted all instances. In this work, we adopt the AUROC categorizations used by Gašević et al. (2016), where AUROC < 0.5, 0.5 ≤ AUROC < 0.7, 0.7 ≤ AUROC < 0.8, 0.8 ≤ AUROC < 0.9, and AUROC ≥ 0.9 respectively represent no discrimination, poor, acceptable, excellent, and outstanding discrimination.

Despite its popularity, the efficacy of the ROC curve and AUROC is not unanimous, as some authors in other fields have historically argued for its use (Provost et al., 1998) while others have argued against it (Drummond & Holte, 2004).

$$False\ positive\ rate = \frac{FP}{FP + TN} \tag{10}$$

3.4. Experimental design

Figure 3.6 outlines the general experimental design we adopted to answer our research questions using the data and algorithms introduced in previous sub-sections. In this work, we followed two distinct approaches: the first using a multi-feature static aggregate representation of the data extracted from Moodle logs and the second using a temporal representation of the total number of clicks. For each approach, we created a dataset for each moment of prediction (10%, 25%, 33%, 50% and 100% course duration). Then, we computed the classification targets: the first classification problem being students at risk vs students not at risk and the second being high-performing students vs students that are not high-performers and used them to train our predictive models. In total, each approach had ten different classification problems.

All experiments shared to following commonalities: all features were standardized with scikit-learn’s StandardScaler, training was performed with 30 repeats of stratified randomized 10-fold cross-validation (using RepeatedStratifiedKFold) and all model performances were compared against a baseline majority class classifier.

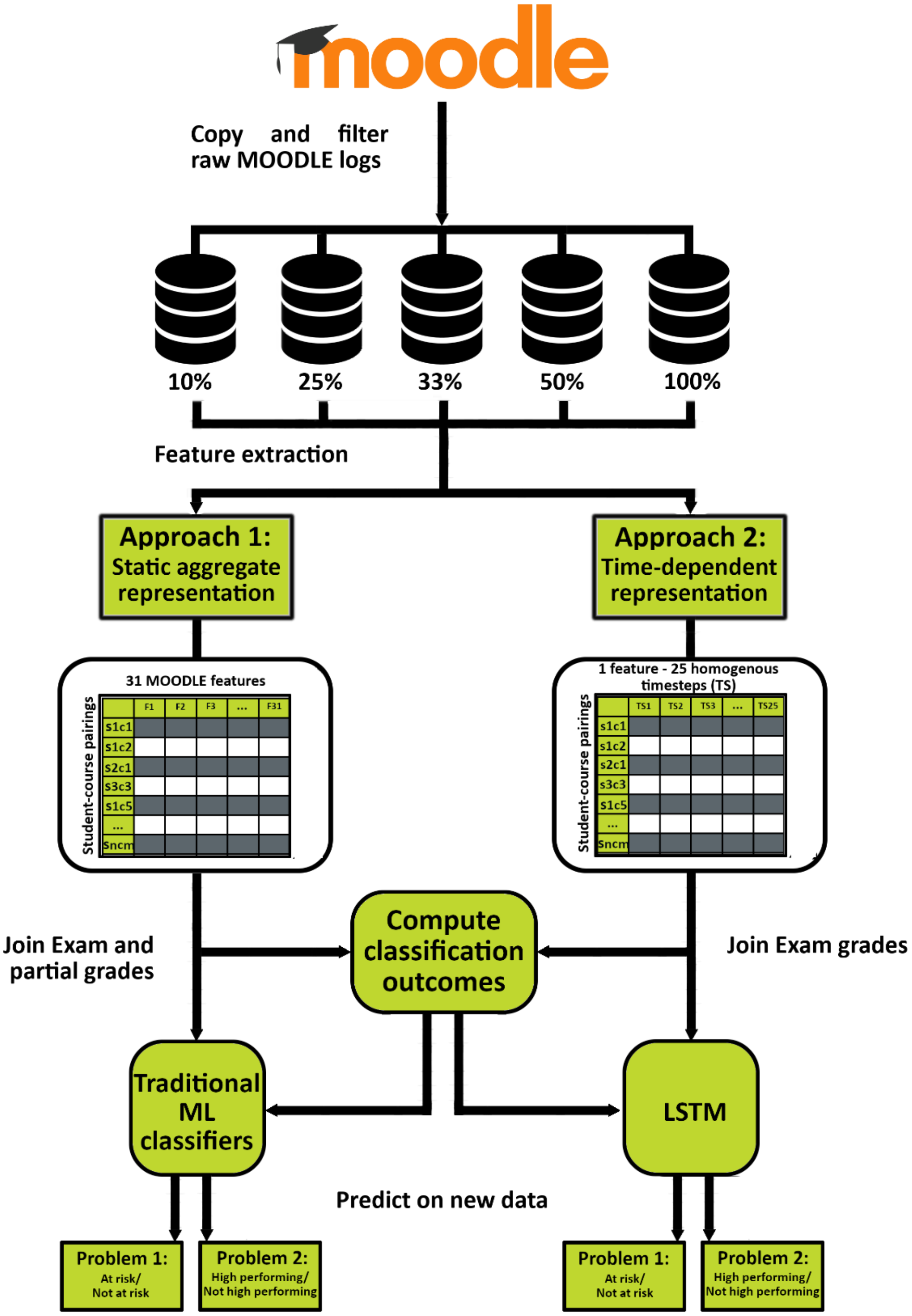

moodle
Copy and filter raw MOODLE logs
10%
25%
33%
50%
100%
Feature extraction
Approach 1: Static aggregate representation
Approach 2: Time-dependent representation
31 MOODLE features
Student-course pairings
F1 F2 F3 ... F31
s1c1 s1c2 s2c1 s3c3 s1c5 ... Sncm
1 feature - 25 homogenous timesteps (TS)
TS1 TS2 TS3 ... TS25
Join Exam and partial grades
Compute classification outcomes
Join Exam grades
Traditional ML classifiers
LSTM
Predict on new data
Problem 1: At risk/ Not at risk
Problem 2: High performing/ Not high performing
Problem 1: At risk/ Not at risk
Problem 2: High performing/ Not high performing

Figure 3.6 – Overview of the overall approach adopted throughout the work

3.4.1. Static aggregate representation

Figure 3.7 illustrates the four experiments made for each classification problem. The first experiment used all 32 features introduced in section 3.2.3. In the second experiment, partial grades were not considered a potential feature. The third and fourth experiments were repeats of the first and second experiments with a key distinction: we used Synthetic Minority Oversampling Technique (SMOTE) to handle class imbalances (Chawla et al., 2002). Standardization and feature selection were performed independently for each fold.

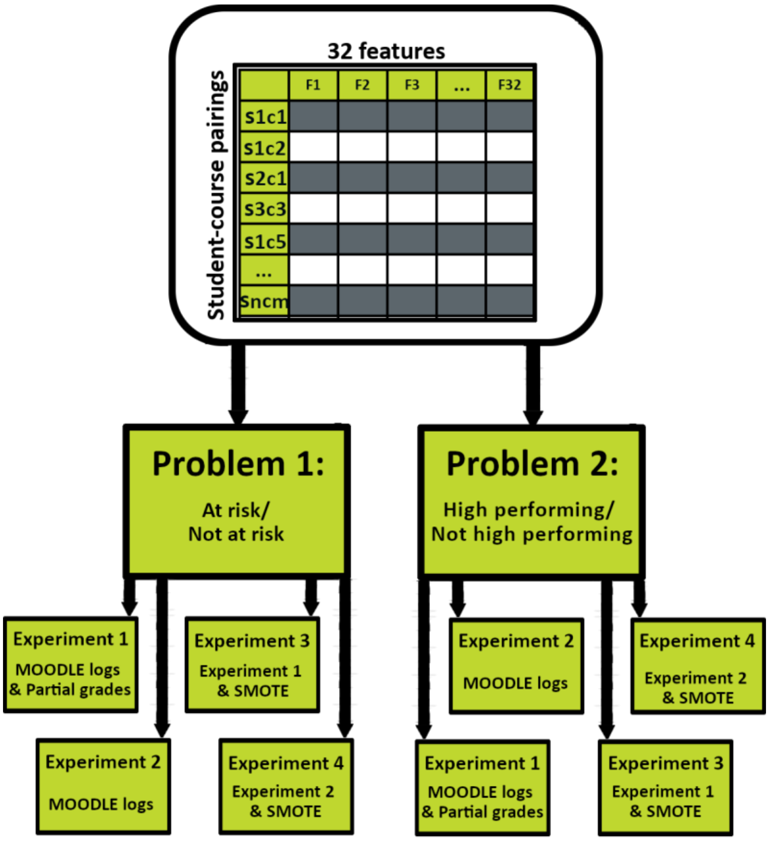


Figure 3.7 – Set of experiments performed for each moment of prediction using a static aggregate data representation

3.4.2. Time-dependent representation

Similarly to the static representation, different deep learning models were trained for each binary classification problem at each moment of prediction. Moreover, data standardization was performed independently for each fold.

As described in Figure 3.8, our modelling of time-dependent representations relied on two different experiments for each classification problem, with the first using the temporal representations of the number of clicks and the second using oversampled data.

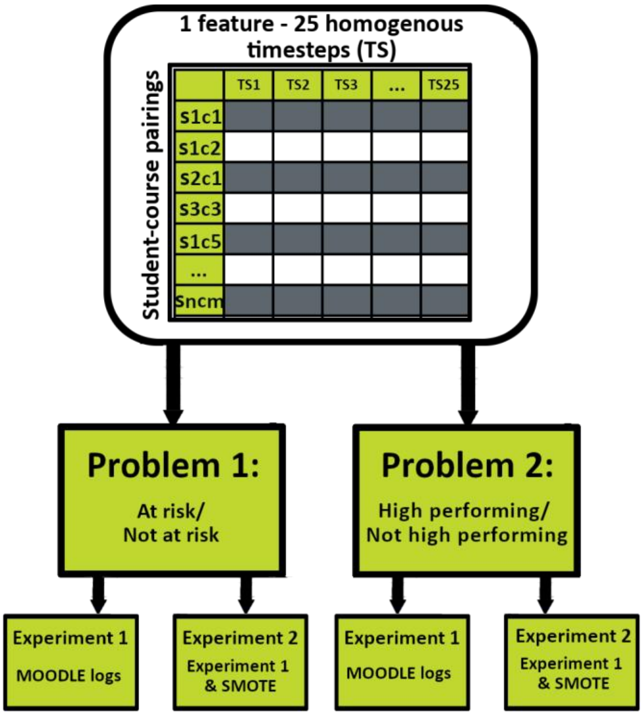


Figure 3.8 – Set of experiments performed for each moment of prediction using a time-dependent data representation

4. RESULTS AND DISCUSSION

In this section, we analyse the results obtained from the sets of experiments described in the previous sections and contextualize them to answer the research questions formulated in section 1.3. We created models aimed at addressing ten different classification problems. We ran 30 repeats of 10-fold stratified cross-validation on multiple ML classification algorithms for each classification problem. The average AUROC for all standard ML classifiers (i.e. those using static aggregate representations) are presented in Tables D.1 and D.2 of Appendix D.

4.1. LMS FEATURES AS PREDICTORS OF STUDENT PERFORMANCE

To address our first research question (*can features extracted from LMS data, on their own, predict student performance?*), we compared the performance of standard ML classifiers exclusively using LMS features (thus, not using partial grades) against a baseline majority class classifier. First, for each classification problem and moment of prediction, we selected the algorithm with the highest AUROC value (all AUROC values obtained for standard ML classifiers are presented in Tables D.1 and D.2 of Appendix D). Then, we used a paired-samples t-test to compare their accuracy and AUROC at the 0.05 and 0.01 significance levels.

4.1.1. Students at risk

As showcased in Table D.1, RF ubiquitously had the highest AUROC with respect to the prediction of students at risk. When assessing differences between our best classifier and the baseline, we note that statistically significant differences are noticeable starting from the early stages of course completion. Table 4.1 shows the accuracy and AUROC of RF at different moments of prediction. Differences in accuracy start being significant at the 25% course duration threshold (with accuracy = 0.803), which is consistent with the results obtained by Riestra-González et al. (2021), for whom the 25% course duration threshold marked the point in time where the dummy classifier started to be outperformed by the ML classifiers.

Table 4.1 – Model performance when predicting students at risk from LMS features using standard ML classifiers (averaged across 30 repeats of 10-fold cross-validation)

Performance metric (Baseline)	**10% duration (RF)**	**25% duration (RF)**	**33% duration (RF)**	**50% duration (RF)**	**100% duration (RF)**
Accuracy (0.799)	0.800	0.803**	0.808**	0.810**	0.817**
AUROC (0.5)	0.678**	0.713**	0.718**	0.730**	0.756**

Significance levels on the one-tailed paired-samples t-test against baseline classifier: * p < .05, ** p < .01

When considering AUROC, the differences between RF and the dummy classifier were significant from the 10% course duration threshold. However, these results should be interpreted cautiously because RF performance is compared against a classifier with no discriminative ability.

Overall, when looked at in isolation, the obtained recall scores tend to be low. For example, at 100% course duration, RF only detects 14.9% of students at risk. However, these tend to be compensated by relatively high precision scores (72.7% precision in the same example), which translates into acceptable AUROC (0.756 in the same example). Ultimately, the exclusive use of LMS features to predict students at risk yields statistically significant improvements in accuracy when compared to the baseline classifier. Moreover, our AUROC scores obtained at every moment hint at acceptable predictive capabilities of the models obtained exclusively from LMS data.

4.1.2. High-performing students

Similar to what we observed for the classifiers of at-risk students, RF had the highest average AUROC in all moments of prediction equal to or greater than the 25% course duration threshold. Moreover, it is a close second to LR at 10% course duration. Differences in accuracy, showcased in Table 4.2, start to be significant at the 25% course duration threshold, which is also consistent with the point in time at which the classifiers for excellent students started to outperform the baseline in work by Riestra-González et al. (2021).

Both recall and AUROC scores hint at LMS features having some predictive power to predict high-performing students, albeit comparatively lower than the displayed when predicting students at risk.

Table 4.2 – Model performance when predicting high-performing students from LMS features using standard ML classifiers (averaged across 30 repeats of 10-fold cross-validation)

Performance metric (Baseline)	**10% duration (LR)**	**25% duration (RF)**	**33% duration (RF)**	**50% duration (RF)**	**100% duration (RF)**
Accuracy (0.723)	0.723	0.725*	0.727**	0.729**	0.728**
AUROC (0.5)	0.593**	0.603**	0.606**	0.611**	0.634**

Significance levels on the one-tailed t-test against baseline classifier: * p < .05, ** p < .01

4.1.3. Introducing partial grades

While the results shown in the previous sub-sections indicate that LMS features have, on their own, low to acceptable degrees of discriminative power, the models we created did not consider all information we believe educators can reasonably access for an EWS. At any particular moment of prediction, educators not only have access to each student's clickstream logs but also know the partial grades obtained up to that point. Whenever used to predict student performance, partial grades have been documented among the most influential features in other works (Conijn et al., 2017; Costa et al., 2017; Riestra-González et al., 2021).

To assess whether the introduction of partial grades leads to better predictions, we compared the performances of the models that did not use partial grades against models that used partial grades. The results, showcased in Tables 4.3 for students at risk and Table 4.4 for high-performing students, confirm that the use of partial grades, in general, leads to improvements in the discriminative ability of the predictive models when compared against models that do not use partial grades.

Differences in discriminative performance between models using partial grades and models not using them are smaller at earlier stages but tend to increase at later moments of prediction. For example, when predicting students at risk at a 33% course duration threshold, the introduction of partial grades results in a 0.4 percentage point increase in accuracy (from 80.8% to 81.2%) and, when using all logs, the improvement in accuracy is 6.9 percentage points (from 81.7% to 88.6%). A similar trend is observable for classifiers of high-performing students. These results are consistent with observations from educational practice: grading events tend to be more common at later stages in the course duration. Therefore, the impact of partial grades as predictors of performance also tends to increase at later moments of prediction (Conijn et al., 2017).

Table 4.3 – Model performance when predicting students at risk from LMS features and partial grades using standard ML classifiers (averaged across 30 repeats of 10-fold cross-validation)

Performance metric	10% duration (RF)	25% duration (RF)	33% duration (RF)	50% duration (RF)	100% duration (RF)
Accuracy	0.802	0.804	0.812*	0.825**	0.886**
Precision	0.713	0.607	0.678	0.732**	0.817**
Recall	0.028*	0.086**	0.123**	0.205**	0.713**
AUROC	0.681	0.728**	0.749**	0.793**	0.923**

Significance levels on one-tailed t-test against model without partial grades: * p < .05, ** p < .01

Overall, features extracted from LMS clickstream exhibit the potential to help educators identify either students at risk or high-performing students. That potential can be enhanced by combining the LMS features with other data from other reasonably accessible sources of data, as is the case with the partial grades obtained throughout the course.

Table 4.4 – Model performance when predicting high-performing students from LMS features and partial grades using standard ML classifiers (averaged across 30 repeats of 10-fold cross-validation)

Performance metric	10% duration (LR)	25% duration (RF)	33% duration (GBoost)	50% duration (RF)	100% duration (GBoost)
Accuracy	0.723	0.724	0.731*	0.735**	0.782**
Precision	0.504**	0.562	0.572**	0.655**	0.668**
Recall	0.033	0.044**	0.110**	0.095**	0.423**
AUROC	0.598**	0.611**	0.624**	0.644**	0.782**

Significance levels on one-tailed t-test against model without partial grades: * p < .05, ** p < .01

4.2. Predictors of course performance at Nova IMS

Our second research question (*is there a general set of rules/features that can inform academic performance across modalities and courses within NOVA IMS?*) focuses on identifying whether there is a consistent set of features that are found to be important predictors across classification targets and moments of prediction. To that effect, we analysed the outputs of our feature selection protocol for each different classifier. Table 4.5 lists the 14 features selected for all course-agnostic classification problems. These findings support the existence of a common set of features that can be used to predict performance in a course-agnostic context.

Table 4.5 – List of features selected that were selected in all classification problems

Selected Features
Online sessions
Total clicks
Clicks (%)
Clicks/day
Clicks/session
Resources viewed
URL views
Course clicks
Days with 0 clicks
Days with 0 clicks (%)
Total time online (min)
Average duration of sessions (min)
Largest period of inactivity (h)
1st session (%)

Table 4.6 lists the remaining features that were selected for different classification problems. While not ubiquitous, four features (folder clicks, forum clicks, the start of 2nd session and the start of 3rd session) were selected in at least eight classification problems. A common trend between these features and Table 4.5 is that they represent general interactions that we expected to be measurable for most students across most courses. Moreover, these are among the most widely adopted features in the literature (Table B.1 in Appendix B). Another relevant feature is the average of the partial grades, although it only becomes ubiquitously important from the 33% course duration threshold onward, which is consistent with the findings presented in section 4.1.3.

We found the heterogeneity observed in the set of features that disclose the moment when each of the first ten logins to be particularly interesting. First, knowledge of when the first login occurred was considered important for all targets and moments of prediction. Then, the second and third logins were always relevant to identify students at risk but not as much when identifying high-performing students. Surprisingly, knowing when the remaining logins are is comparatively less important, especially in identifying high-performing students. We expected that more frequent and evenly spread login patterns would be relevant predictors of better students (Conijn et al., 2017; Riestra-González et al., 2021). At the end of the spectrum, we have forum posts as the least selected feature, closely followed by assignment submissions, discussions viewed and assessments started. These features have been proven helpful in other works (Macfadyen & Dawson, 2010; Sandoval et al., 2018), but their effectiveness has been observed in course-specific approaches and not in course-agnostic contexts (Conijn et al., 2017; Gašević et al., 2016).

Table 4.6 – Remaining features selected for each classification problem

	10% duration	25% duration	33% duration	50% duration	100% duration
Students at risk/ Students not at risk	Folder clicks Forum clicks 2nd session (%) 3rd session (%) Assignment views 9th session (%) 4th session (%) 5th session (%) 6th session (%) 7th session (%)	Folder clicks Forum clicks 2nd session (%) 3rd session (%) Partial grades Assignment views 9th session (%) 4th session (%) 5th session (%) 6th session (%) 10th session (%) 8th session (%) Assessments started	Folder clicks Forum clicks 2nd session (%) 3rd session (%) Partial grades Assignment views 4th session (%) 5th session (%) 6th session (%) 7th session (%) 10th session (%) 8th session (%)	Folder clicks Forum clicks 2nd session (%) 3rd session (%) Partial grades Assignment views 9th session (%) 4th session (%) 7th session (%) 10th session (%) Discussions viewed	2nd session (%) 3rd session (%) Partial grades Submissions (%)
High-performing students/ Not high-performing students	Folder clicks 2nd session (%) 3rd session (%) 9th session (%) 4th session (%) 5th session (%) 6th session (%)	Folder clicks Forum clicks 2nd session (%) 3rd session (%) Assignment views 9th session (%) 4th session (%) 5th session (%) 6th session (%) 7th session (%) 10th session (%)	Forum clicks Partial grades 9th session (%) 7th session (%)	Folder clicks Forum clicks 2nd session (%) 3rd session (%) Partial grades Assignment views 9th session (%) 6th session (%) 7th session (%) 10th session (%) Assignments submitted	Folder clicks Forum clicks Partial grades Assignment views 5th session (%) 10th session (%) 8th session (%)

4.3. Early prediction of student performance

In previous sections, we established that LMS features could assist educators in identifying either at-risk or high-performing students. Moreover, using partial grades significantly improved the discriminative ability of our models, especially at later moments of prediction (Tables 4.3 and 4.4). For example, using all logs and partial grades, our best classifier for students at risk achieved an accuracy > 0.85, a recall > 0.70, a precision > 0.8 and an AUROC > 0.90. When predicting at the 50% course threshold, performances on most metrics were still respectable, even if comparatively lower. In broad strokes, an affirmative answer to our third research question (*Can performance be inferred when, at most, 50% of the course is completed?*) seems trivial. However, at that same moment of prediction, our best classifiers could only correctly identify 20% of students at risk and 10% of high-performing students at the courses' midway point, raising questions about our models' usefulness as EWS.

The use of SMOTE turned into an elegant solution with immediate improvements in our models' ability to identify students of interest. Tables 4.7 and 4.8 present the performance metrics obtained for each best early classifier using SMOTE and compare them with the best performing models presented in Tables 4.3 and 4.4, respectively. In general, the models we trained using oversampled data had worse accuracy and AUROC than those trained without oversampled data, an observation consistent with other works (Riestra-González et al., 2021; Romero, Espejo, et al., 2013). However, they also had much higher recall values. A decent compromise using SMOTE is achieved at the 25% course duration threshold where AUROC remains similar, but our classifiers can identify 50% of the students of interest (for both at-risk and high-performing students).

The models we trained using oversampled data have shown greater potential in identifying students of interest at earlier stages (as indicated by the recall values). However, this increased potential is also tied to a higher proneness to predict false positives (and consequently lower accuracy and precision). On the contrary, models trained without using SMOTE are less capable of detecting students of interest but much more precise.

Table 4.7 – Model performance when predicting students at risk from LMS features and partial grades using standard ML classifiers and SMOTE (averaged across 30 repeats of 10-fold cross-validation)

Performance metric	10% duration (RF)	25% duration (RF)	33% duration (RF)	50% duration (RF)
Accuracy	0.733**	0.732**	0.736**	0.764**
Precision	0.354**	0.376**	0.387**	0.436**
Recall	0.396**	0.500**	0.527**	0.577**
AUROC	0.674**	0.725	0.741**	0.788**

Significance levels on two-tailed t-test against the best model with partial grades: * p < .05, ** p < .01

When comparing against the general LA research landscape, the results obtained by our traditional ML classifiers are on par with the ones obtained by other authors. For example, at the 50% duration threshold, our best classifiers for students at risk (RF with 0.788 AUROC on the version trained on oversampled data and 0.793 on the model trained without it) achieved AUROC scores above the general models developed by Gašević et al. (2016) and Helal et al. (2018) – respectively 0.749 and 0.700 by the end of the course. Moreover, other works trained on data from a single course are outperformed by our predictions at the 50% duration threshold, such as the 0.69 AUROC obtained by Saqr et al. (2017) at the course's midway point or all classifiers used by Chen & Cui (2020).

Table 4.8 – Model performance when predicting high-performing students from LMS features and partial grades using standard ML classifiers and SMOTE (averaged across 30 repeats of 10-fold cross-validation)

Performance metric	10% duration (LR)	25% duration (ExtraTrees)	33% duration (ExtraTrees)	50% duration (ExtraTrees)
Accuracy	0.592**	0.581**	0.599**	0.611**

Precision	0.341**	0.342**	0.353**	0.365**
Recall	0.508**	0.552**	0.537**	0.548**
AUROC	0.596	0.608	0.624**	0.632**

Significance levels on two-tailed t-test against the best model with partial grades: * p < .05, ** p < .01

The results obtained using either approach (training without oversampling or with oversampling) support our proposition that student performance can be inferred using only data collected up to the moment of prediction. Ultimately, deciding which approach is better suited for a given situation depends on the educators' goals for implementing the EWS and the cost associated with wrong predictions. We argue that models trained with SMOTE are better suited for identifying students at risk to provide timely feedback. An EWS should be able to detect as many students of interest as possible, as the cost of false negatives is not receiving feedback and taking corrective action. Moreover, a small number of false positives is almost innocuous for students incorrectly receiving the intervention. Conversely, if the goal is to provide more challenging content to high-performing students, we argue that the cost of a false positive is much greater than the potential costs associated with false negatives. Therefore, a more conservative and precise approach would be better suited in these cases, which, in the case of our models, is the use of models trained without SMOTE.

4.4. Early prediction using LSTM networks

Converting LMS logs to a format amenable to standard ML classifiers is not trivial. Effective pre-processing and manipulation of data require domain knowledge, data skills and computational resources that may not be accessible to educators. In this section, we analyze the potential of a single-feature temporal representation of the number of clicks as a predictor of student performance. We rely on the datasets created in section 3.2.5 and use LSTM networks to identify students at risk or high-performing students. Figure 4.1 compares, for each moment of early prediction, the model performances obtained from the single feature LSTM network against those obtained by each of the best traditional ML classifiers presented in Tables 4.3 through 4.8. A complete display of the comparison between models, including differences in the 100% duration threshold, is presented in Tables E.1 and E.2 found in Appendix E.

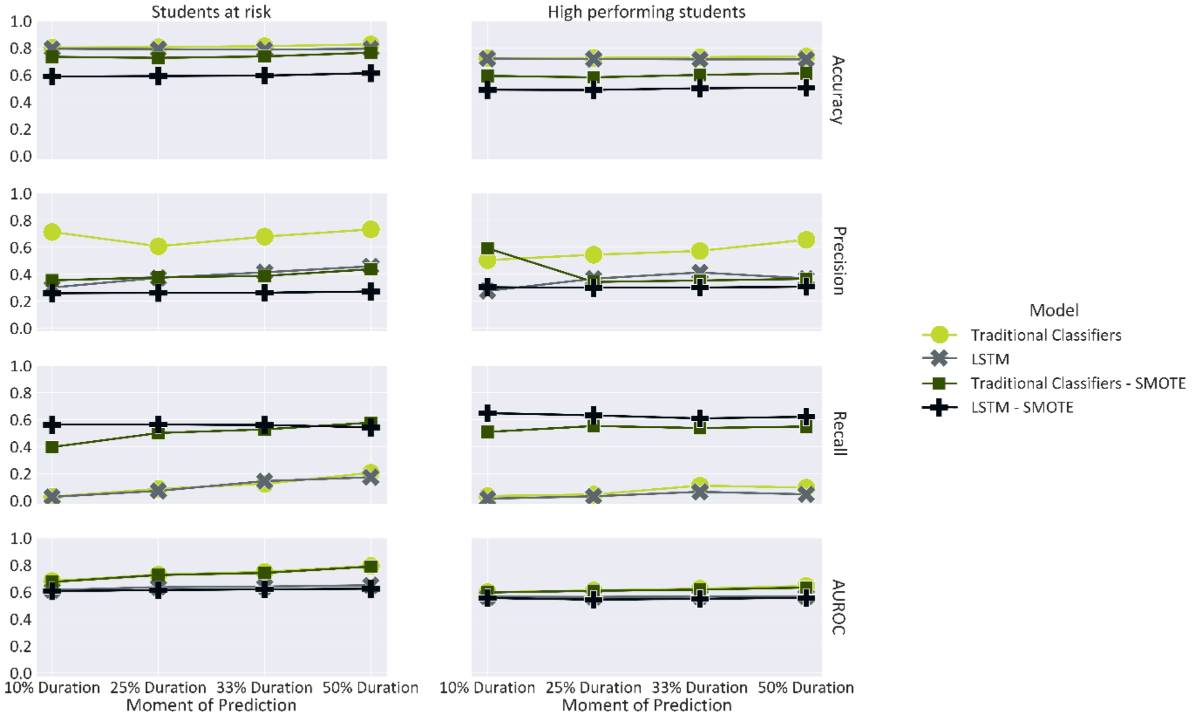


Figure 4.1 – Comparison between model performances on early prediction between the best traditional ML classifiers and the single-feature LSTM

Among all combinations of targets and moments of early prediction, AUROC values vary between 0.54 (high-performing students at the 25% duration threshold using oversampled data) and 0.65 (students at risk at the 50% duration threshold without oversampling), which places our LSTM models as having poor discriminative power (Gašević et al., 2016). With minimal optimisation, our LSTM networks obtained results comparable to the ones published by Yu & Wu (2021) (who reported 67% accuracy by the courses' midway point) and Chen & Cui (2020) (who reported AUROC values varying between 0.596 and 0.682 on their education course test data). Nevertheless, our results are below the validation performances (all AUROC values above 0.7) reported by Chen & Cui (2020) and the 90% accuracy achieved by Aljohani et al. (2019) by the courses' 25% duration threshold (the 10th week on courses with 38 weeks).

One possible justification for the reported differences in performance is the amount of data used. Our approach relies on a powerful deep learning algorithm (LSTM) using a single, easily extractable feature. Educators without the resources or expertise can avoid the need to perform sophisticated pre-processing or feature engineering. The works by Aljohani et al. (2019) and Yu & Wu (2021) rely on similar algorithms while using additional features. Therefore, even if, at some stages, we reach comparable performances to those obtained by the latter, our models are expected to underperform more sophisticated deep learning models.

Other possible justifications involve differences in experimental design, particularly sample size and the number of nodes. The work by Chen & Cui (2020) uses a single course and 41 students per fold, whereas each of our folds considers close to 1000 students and multiple courses. Even if we set aside the self-evident differences in sample size, our observations are still compatible with previous claims of course-agnostic models performing worse than models trained on a single course (Conijn et al., 2017; Gašević et al., 2016; López-Zambrano et al., 2020). For the number of nodes, the works found in the literature use weekly (Aljohani et al., 2019; Mahzoon et al., 2018; Yu & Wu, 2021) or daily time intervals (Chen & Cui, 2020). In our work, we use a fixed window of 25 timesteps, each node representing 4% of the time passed from the start of the course to the moment of prediction. Treating time as a relative variable allows the creation of course-agnostic models that capture sequential dependencies from multiple courses with different hypothetical lengths. However, the differences in course length are not highlighted using this approach. For example, 4% of a trimestral course is, in absolute terms, close to half of the time passed in 4% of a semestral course. When counting the number of clicks made in a timestep, we treat periods of different lengths as similar. Thus, the ability of LSTM to capture patterns detectable when working with absolute time intervals may be hindered.

When analyzing the results obtained from our LSTM in the context of Nova IMS data, we observed trends similar to those previously detected for traditional classifiers. Accuracy, recall and AUROC values tend to increase over time. Moreover, models trained with oversampling had lower accuracy and precision than the ones trained without using SMOTE whilst having a significantly higher recall and close to the same AUROC. While our LSTM models were generally on par with popular traditional ML classifiers such as SVM, NB and or CART, they consistently underperformed their corresponding best traditional classifier on most metrics. However, while statistically significant, the differences are relatively small, especially with respect to accuracy and recall. When it comes to accuracy, the most prominent differences found between models trained without SMOTE were observed at the 50% course duration threshold: 3.2 percentage points for students at risk (from 82.5% for RF to 79.3% for LSTM) and 2 percentage points for high performing students (73.5% for LR to 71.5% for LSTM). In models trained using SMOTE, the differences in accuracy increased substantially (to close to 20 percentage points) for both targets at all moments of prediction. The most considerable difference in recall between the best traditional classifier and the LSTM model when classifying students at risk was 3 percentage points, also observed at the 50% duration threshold (20.5% for RF to 17.2% for LSTM). When using models trained using SMOTE, LSTM models have higher recall than the traditional ML counterpart.

In objective terms, our analysis of model performances alone advises against adopting the single feature LSTM instead of RF, Gboost or ExtraTrees for most instances. Arguably, using LSTM at 10%, 25% or the 33% course duration threshold can be justified by simpler pre-processing and feature extraction with minimal performance losses compared to the corresponding best classifier. For example, when comparing the classifiers for students at risk at the 25% duration threshold, the LSTM and RF were separated by 1 percentage point for accuracy values (respectively, 79% against 80%) and recall values (0.071 against 0.086).

4.5. A Follow-up Study for Generalizability

In order to verify the potential for generalizability of our EWS, we performed a follow-up analysis on a different set of data. Our goal with this section was to validate the promising results obtained with our static aggregate representations and to obtain additional information concerning our single feature deep learning approach.

4.5.1. Data description

For this section, we used the Oviedo University data presented by Riestra-González et al. (2021). The raw data included over 47 million log entries belonging to 29602 unique students and was collected throughout the 2014/2015 academic year. The main reason behind the choice of this data was the scale of the number of students and courses it contained. Moreover, as the data includes all courses taught at the University of Oviedo, there is a large variability in click patterns and course durations.

We started by downloading the MySQL dump the authors made available[8] and replicated the initial pre-processing and filtering steps. As neither course duration nor the outcome variable are provided in the raw logs, we estimated both by adopting the same methodology as the authors of the paper. Then, to create new static aggregate datasets for each target and moments of early prediction[9], we replicated the feature extraction and filtering criteria we adopted on the Nova IMS data in sections 3.2.3 and 3.2.4. Likewise, for the temporal representations, we also followed the methods described in section 3.2.5. At the end of the process, we had eight static aggregate datasets (one for each target at the 10%, the 25%, the 33% and the 50% course duration thresholds) and eight time-dependent datasets. Each dataset had 13857 rows, with each row representing a unique student-course pair. Table 4.9 summarizes the class imbalances for each classification problem. For the remainder of the work, we strictly followed the analytical procedures described for the Nova IMS data.

Table 4.9 – Class representation in each classification problem

Classification target	Yes	No
Students at risk	5280 (38%)	8577 (62%)
High-performing students	2751 (20%)	11106 (80%)

4.5.2. Results and discussion

Figure 4.2 compares the model performances obtained for each model on the Oviedo University datasets. The results validate our approach's potential to identify both students at risk and high-performing students. Across all metrics, the model performances obtained by our best static aggregate models tend to be better at later moments of prediction, with an excellent discriminative ability (AUROC between 0.8 and 0.9) being achieved from the 25% course duration threshold onward when identifying students at risk and from the 33% course duration threshold for high-performing students. Furthermore, as in our previous results obtained with Nova IMS data, the models trained with oversampled data have lower precision and higher recall than the non-oversampled models, ultimately resulting in similar AUROC. A complete display of the comparison between models is presented in Tables E.3 and E.4 found in Appendix E.

[8] Available at https://github.com/moisesriestra/moodle-early-performance-prediction (last visited on the 23rd of April 2022)

[9] As the outcome variable used by Riestra-González et al. (2021) is estimated using a regression from assignment grades, we did not consider the 100% duration threshold for this data.

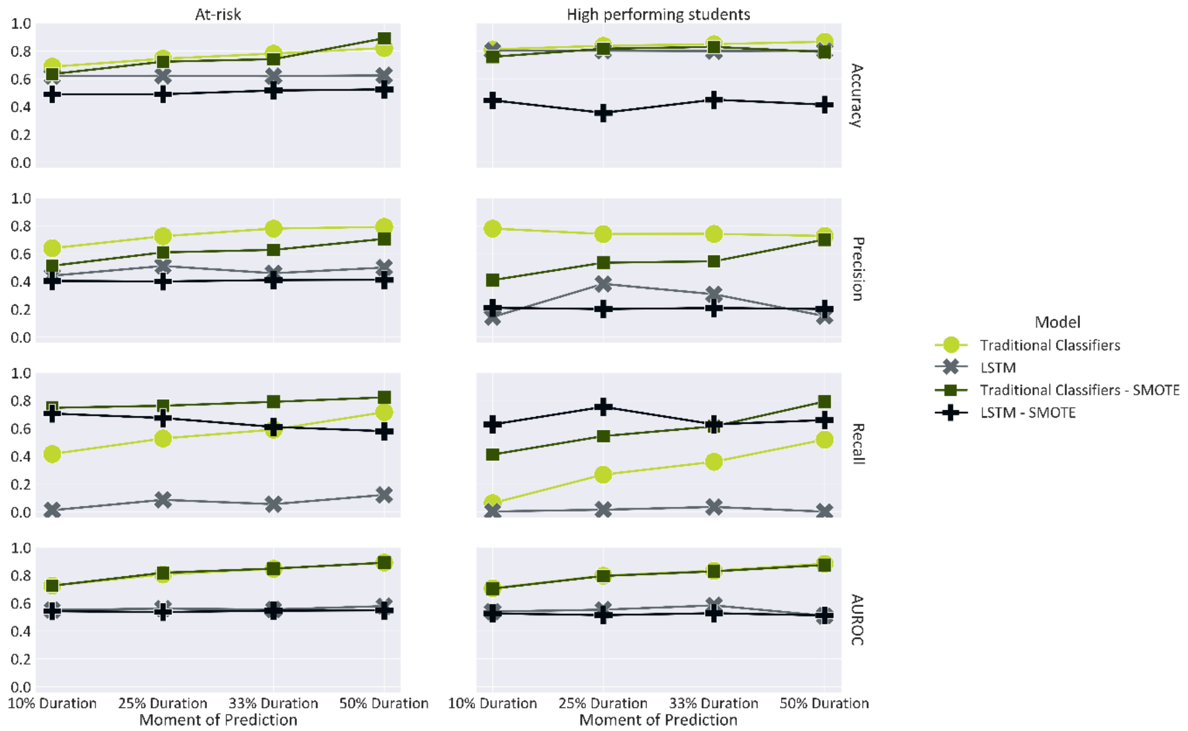


Figure 4.2 – Comparison between model performances on early prediction between the best traditional ML classifiers and the single-feature LSTM on a new dataset

Whilst static aggregate model performances with the Oviedo University datasets were comparatively higher than those obtained with Nova IMS data, it had worse performances when using the 25-timestep temporal representations of the number of clicks. The low discriminative ability of our models trained using this approach is highlighted by AUROC scores below 0.60 for all targets and moments of prediction. For these datasets, we consider that there is no point where a reasonable tradeoff between loss in performance and easier data pre-processing can be achieved.

Overall, the results we obtained from the Oviedo University data match those we obtained from the Nova IMS data as the models created using static aggregate data representation achieved significantly better performances than the single-feature temporal representations.

5. CONCLUSIONS

In the post-Covid era, LMS and other virtual learning environments play a vital role in delivering educational content to students, making them invaluable tools for educators and HEI. A major challenge facing educators is the timely identification of students at risk of failing. Early identification allows educators to provide feedback and develop corrective measures to prevent student failure (Hu et al., 2014; Macfadyen & Dawson, 2010). A similar rationale for EWS may be applied to the early identification of students who excel and whose development would benefit from more challenging materials (Riestra-González et al., 2021).

In this work, we used LMS log data collected throughout the 2020/2021 academic year at a portuguese information management school and created different predictive models to identify at-risk or high-performing students across multiple courses and at multiple stages of course completion. First, for each course, we calculated the moment in time that would represent the 10%, 25%, 33%, 50% and 100% course duration threshold and created a different set of logs for each moment of prediction. Then, we created two types of dataset: the first being datasets containing an aggregate non-temporal representation of the log data and the second being datasets present using a time-dependent representation of the number of clicks. In total, we created 20 datasets: two forms of data representation for two different classification problems, each with five moments of prediction.

Predictive models trained with the datasets constructed using an aggregate non-temporal logic achieved excellent to outstanding discriminative abilities when using all data: 0.923 AUROC when predicting students at risk using RF and 0.796 when predicting high-performing students with GBoost. In addition, our models had respectable performances at earlier moments of prediction, especially in the students at risk classification problem: from 0.728 AUROC when making predictions at 25% of course duration up to 0.793 AUROC at 50% of course duration. However, when predicting high-performing students, model performances were comparatively lower across the board, with the best AUROC (0.644) being attained using RF at the 50% course duration threshold.

To demonstrate that the information encoded into a temporal representation of data can yield comparable without requiring such extensive domain knowledge or data skills, we trained LSTM networks with the 25-timestep temporal representation of a single feature (the number of clicks). Unlike previous studies (Chen & Cui, 2020), our implementation of LSTM networks using sequences of the number of clicks did not outperform other traditional ML classifiers at any moment of prediction. However, the results from Nova IMS students suggest that there is potential for the use of these single feature datasets, especially at the earliest stages. For both the classification of students at risk and the classification of high-performing students, the results obtained at the 10%, 25% and 33% course duration with LSTM point towards much simpler data pre-processing at the expense of relatively small losses in performance (below 5 percentage points in accuracy and recall and below 10 percentage points in AUROC). In particular, we find that the best moment of prediction using LSTM to predict students at risk is after one third of the course has been completed - 0.64 AUROC against the 0.74 AUROC obtained with RF. Furthermore, for high-performing students, that tradeoff occurs after one fourth of the course is completed - 0.563 AUROC against 0.611 AUROC obtained with RF.

Finally, we replicated our experiments on another set of logs obtained from Oviedo University. Our results validated our previous observations with respect to the performance of the traditional ML classifiers trained with data using a static aggregate representation. For both students at risk and high-performing students, excellent discriminative performances are achieved at early moments of prediction: the classification of students at risk achieves AUROC greater than 0.80 from the 25% course threshold onward, and the classification of high-performing students does the same starting from the 33% course threshold. Peak performances are achieved at the courses' midway point, where RF achieves 0.89 AUROC when distinguishing between students at risk and students not at risk. Likewise, at that moment of prediction, GBoost achieved 0.88 AUROC when identifying high-performing students. Unfortunately, the promising signs displayed by the temporal representation of the number of clicks observed with Nova IMS data did not generalize to this dataset, as LSTM performances are lower than 0.6 AUROC across the board.

Ultimately, the findings presented herein provide insights into using traditional ML classifiers and deep learning into early predictions of student performance within a course-agnostic context. Our results with traditional ML classifiers on different datasets support the view that student activity on LMS is predictive of student performance across different courses and educational contexts. Furthermore, our deep learning results indicate that portability may be possible in specific contexts. All models and reusable Python source code are freely available at the following link https://github.com/RicardoSantos0/Msc_thesis.

6. LIMITATIONS AND RECOMMENDATIONS FOR FUTURE WORKS

Our work has some relevant limitations we intend to address in future works. A first concern is the data's ability to generalize, as this survey's primary set of logs concerns a single academic year in an information management school. In addition, due to Covid-19 security measures, all courses worked similarly to online courses, which made 2020/2021 a particularly unique academic year. LMS are expected to be the primary interface between students and educational content in a completely online context. However, that may not necessarily be the case in a typical year with an open campus and face-to-face classes. While our analysis of the data collected throughout an entire university in a different pre-Covid academic year shows promising results concerning our static aggregate models' ability to generalize, further validation on Nova IMS data across different academic years would be a welcome addition.

A second relevant limitation of our survey concerns our data preprocessing and feature extraction strategies. As extensively discussed in previous sections, these tasks are not trivial and require significant domain knowledge and data skills. In this work, our primary focus was extracting features that enjoy widespread adoption in the literature. However, promising experimental features such as those introduced by Marras et al. (2021) were not considered, even though their introduction could potentially lead to more accurate predictive models. A similar limitation concerns our choice of 25 timesteps for the temporal representation of the number of clicks, an approach that had promising results with the Nova IMS data but did not perform as well with the Oviedo University data. Multiple possible alternatives can be considered for future works: from developing more elaborate filtering strategies that only maintain courses with relatively similar course durations to having a different number of timesteps be considered for different moments of predictions.

There is also room for improvement in our deep learning approach. More particularly, the number of clicks is not the only feature that exhibits predictive power using a time-dependent representation and future works should consider the addition of other, simple to obtain, sequences of activity counts (Aljohani et al., 2019; Yu & Wu, 2021). Furthermore, predictive performance can likely be enhanced with more elaborate deep learning algorithms such as bi-directional LSTM (Graves & Schmidhuber, 2005).

Our final highlight-worthy limitation concerns model optimization. The experiments performed in this work were, for the most part, exploratory. While we adopted the hyper-parameters used in other works (Chen & Cui, 2020; Riestra-González et al., 2021), minimal effort was placed into additional fine-tuning of hyper-parameters. It is not self-evident that the best hyper-parameters in other works are the most adequate for our data and future experiments should place a more explicit emphasis on model optimization and fine-tuning.

7. BIBLIOGRAPHY

Adejo, O. W., & Connolly, T. (2018). Predicting student academic performance using multi-model heterogeneous ensemble approach. *Journal of Applied Research in Higher Education*, *10*(1), 61–75. https://doi.org/10.1108/JARHE-09-2017-0113

Aldowah, H., Al-Samarraie, H., & Fauzy, W. M. (2019). Educational data mining and learning analytics for 21st century higher education: A review and synthesis. *Telematics and Informatics*, *37*, 13–49. https://doi.org/10.1016/j.tele.2019.01.007

Aljohani, N. R., Fayoumi, A., & Hassan, S.-U. (2019). Predicting At-Risk Students Using Clickstream Data in the Virtual Learning Environment. *Sustainability*, *11*(24), 7238. https://doi.org/10.3390/su11247238

Baker, R., Xu, D., Park, J., Yu, R., Li, Q., Cung, B., Fischer, C., Rodriguez, F., Warschauer, M., & Smyth, P. (2020). The benefits and caveats of using clickstream data to understand student self-regulatory behaviors: Opening the black box of learning processes. *International Journal of Educational Technology in Higher Education*, *17*(1), 13. https://doi.org/10.1186/s41239-020-00187-1

Baker, R., & Yacef, K. (2009). *The State of Educational Data Mining in 2009: A Review and Future Visions*. https://doi.org/10.5281/ZENODO.3554657

Baneres, D., Rodriguez, M. E., & Serra, M. (2019). An Early Feedback Prediction System for Learners At-Risk Within a First-Year Higher Education Course. *IEEE Transactions on Learning Technologies*, *12*(2), 249–263. https://doi.org/10.1109/TLT.2019.2912167

Brooks, C., Thompson, C., & Teasley, S. (2015). A time series interaction analysis method for building predictive models of learners using log data. *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, 126–135. https://doi.org/10.1145/2723576.2723581

Buschetto Macarini, L. A., Cechinel, C., Batista Machado, M. F., Faria Culmant Ramos, V., & Munoz, R. (2019). Predicting Students Success in Blended Learning—Evaluating Different Interactions Inside Learning Management Systems. *Applied Sciences*, *9*(24), 5523. https://doi.org/10.3390/app9245523

Calvet Liñán, L., & Juan Pérez, Á. A. (2015). Educational Data Mining and Learning Analytics: Differences, similarities, and time evolution. *RUSC. Universities and Knowledge Society Journal*, *12*(3), 98. https://doi.org/10.7238/rusc.v12i3.2515

Calvo-Flores, M. D., Galindo, E. G., Jiménez, M. C. P., & Pérez, O. (2006). Predicting students' marks from Moodle logs using neural network models. *Current Developments in Technology-Assisted Education*, *1*(2), 586–590.

Casey, K., & Azcona, D. (2017). Utilizing student activity patterns to predict performance. *International Journal of Educational Technology in Higher Education*, *14*(1), 4. https://doi.org/10.1186/s41239-017-0044-3

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953

Chen, F., & Cui, Y. (2020). Utilizing Student Time Series Behaviour in Learning Management Systems for Early Prediction of Course Performance. *Journal of Learning Analytics*, *7*(2), 1–17. https://doi.org/10.18608/jla.2020.72.1

Chickering, A. W., & Gamson, Z. F. (1987). Seven principles for good practice in undergraduate education. *AAHE Bulletin*, *39*(7), 3–7.

Chui, K. T., Fung, D. C. L., Lytras, M. D., & Lam, T. M. (2020). Predicting at-risk university students in a virtual learning environment via a machine learning algorithm. *Computers in Human Behavior*, *107*, 105584. https://doi.org/10.1016/j.chb.2018.06.032

Coates, H., James, R., & Baldwin, G. (2005). A critical examination of the effects of learning management systems on university teaching and learning. *Tertiary Education and Management*, *11*, 19–36.

Conijn, R., Snijders, C., Kleingeld, A., & Matzat, U. (2017). Predicting Student Performance from LMS Data: A Comparison of 17 Blended Courses Using Moodle LMS. *IEEE Transactions on Learning Technologies*, *10*(1), 17–29. https://doi.org/10.1109/TLT.2016.2616312

Costa, E. B., Fonseca, B., Santana, M. A., de Araújo, F. F., & Rego, J. (2017). Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses. *Computers in Human Behavior*, *73*, 247–256. https://doi.org/10.1016/j.chb.2017.01.047

Daniel, B. (2015). Big Data and analytics in higher education: Opportunities and challenges: The Value of Big Data in Higher Education. *British Journal of Educational Technology*, *46*(5), 904–920. https://doi.org/10.1111/bjet.12230

Drummond, C., & Holte, R. C. (2004). What ROC Curves Can't Do (and Cost Curves Can). *ROCAI*, 19–26.

Fahd, K., Miah, S. J., & Ahmed, K. (2021). Predicting student performance in a blended learning environment using learning management system interaction data. *Applied Computing and Informatics*, *ahead-of-print*(ahead-of-print). https://doi.org/10.1108/ACI-06-2021-0150

Gašević, D., Dawson, S., Rogers, T., & Gasevic, D. (2016). Learning analytics should not promote one size fits all: The effects of instructional conditions in predicting academic success. *The Internet and Higher Education*, *28*, 68–84. https://doi.org/10.1016/j.iheduc.2015.10.002

Gers, F. A., Schmidhuber, J. A., & Cummins, F. A. (2000). Learning to Forget: Continual Prediction with LSTM. *Neural Computation*, *12*(10), 2451–2471. https://doi.org/10.1162/089976600300015015

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Aistats*, *9*, 249–256.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, *18*(5–6), 602–610. https://doi.org/10.1016/j.neunet.2005.06.042

Grove, W. A., Wasserman, T., & Grodner, A. (2006). Choosing a Proxy for Academic Aptitude. *The Journal of Economic Education*, *37*(2), 131–147. https://doi.org/10.3200/JECE.37.2.131-147

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning*, *46*(1–3), 389–422.

Hasan, R., Palaniappan, S., Mahmood, S., Abbas, A., Sarker, K. U., & Sattar, M. U. (2020). Predicting Student Performance in Higher Educational Institutions Using Video Learning Analytics and Data Mining Techniques. *Applied Sciences*, *10*(11), 3894. https://doi.org/10.3390/app10113894

Helal, S., Li, J., Liu, L., Ebrahimie, E., Dawson, S., Murray, D. J., & Long, Q. (2018). Predicting academic performance by considering student heterogeneity. *Knowledge-Based Systems*, *161*, 134–146. https://doi.org/10.1016/j.knosys.2018.07.042

Hellas, A., Ihantola, P., Petersen, A., Ajanovski, V. V., Gutica, M., Hynninen, T., Knutas, A., Leinonen, J., Messom, C., & Liao, S. N. (2018). Predicting academic performance: A systematic literature review. *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 175–199. https://doi.org/10.1145/3293881.3295783

Hu, Y.-H., Lo, C.-L., & Shih, S.-P. (2014). Developing early warning systems to predict students' online learning performance. *Computers in Human Behavior*, *36*, 469–478. https://doi.org/10.1016/j.chb.2014.04.002

Huang, A. Y. Q., Lu, O. H. T., Huang, J. C. H., Yin, C. J., & Yang, S. J. H. (2020). Predicting students' academic performance by using educational big data and learning analytics: Evaluation of classification methods and learning logs. *Interactive Learning Environments*, *28*(2), 206–230. https://doi.org/10.1080/10494820.2019.1636086

Imose, R., & Barber, L. K. (2015). Using undergraduate grade point average as a selection tool: A synthesis of the literature. *The Psychologist-Manager Journal*, *18*(1), 1–11. https://doi.org/10.1037/mgr0000025

Jones, K. M. L., Asher, A., Goben, A., Perry, M. R., Salo, D., Briney, K. A., & Robertshaw, M. B. (2020). "We're being tracked at all times": Student perspectives of their privacy in relation to learning analytics in higher education. *Journal of the Association for Information Science and Technology*, *71*(9), 1044–1059. https://doi.org/10.1002/asi.24358

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances in Neural Information Processing Systems*, *30*, 3149–3157.

Khan, A., & Ghosh, S. K. (2021). Student performance analysis and prediction in classroom learning: A review of educational data mining studies. *Education and Information Technologies*, *26*(1), 205–240. https://doi.org/10.1007/s10639-020-10230-3

Kingma, D. P., & Ba, J. (2017). *Adam: A Method for Stochastic Optimization* (arXiv:1412.6980). arXiv. http://arxiv.org/abs/1412.6980

Kuzilek, J., Hlosta, M., Herrmannova, D., Zdrahal, Z., Vaclavek, J., & Wolff, A. (2015). OU Analyse: Analysing at-risk students at The Open University. *Learning Analytics Review*, *LAK15*(1), 1–16.

López-Zambrano, J., Lara, J. A., & Romero, C. (2020). Towards Portability of Models for Predicting Students' Final Performance in University Courses Starting from Moodle Logs. *Applied Sciences*, *10*(1), 354. https://doi.org/10.3390/app10010354

López-Zambrano, J., Lara Torralbo, J. A., & Romero, C. (2021). Early Prediction of Student Learning Performance Through Data Mining: A Systematic Review. *Psicothema*, *33.3*, 456–465. https://doi.org/10.7334/psicothema2021.62

Macfadyen, L. P., & Dawson, S. (2010). Mining LMS data to develop an "early warning system" for educators: A proof of concept. *Computers & Education*, *54*(2), 588–599. https://doi.org/10.1016/j.compedu.2009.09.008

Mahzoon, M. J., Maher, M. L., Eltayeby, O., Dou, W., & Grace, K. (2018). A Sequence Data Model for Analyzing Temporal Patterns of Student Data. *Journal of Learning Analytics*, *5*(1). https://doi.org/10.18608/jla.2018.51.5

Marras, M., Vignoud, J. T. T., & Käser, T. (2021). Can Feature Predictive Power Generalize? Benchmarking Early Predictors of Student Success across Flipped and Online Courses. *14th International Conference on Educational Data Mining*, 11.

McKinney, W. (2018). *Python for data analysis: Data wrangling with pandas, NumPy, and IPython* (Second edition). O'Reilly Media, Inc.

Moher, D., Liberati, A., Tetzlaff, J., Altman, D. G., & and the PRISMA Group. (2009). Preferred Reporting Items for Systematic Reviews and Meta-Analyses: The PRISMA Statement. *Physical Therapy*, *89*(9), 873–880. https://doi.org/10.1093/ptj/89.9.873

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., … Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ*, n71. https://doi.org/10.1136/bmj.n71

Page, M. J., Moher, D., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., … McKenzie, J. E. (2021). PRISMA 2020 explanation and elaboration: Updated guidance and exemplars for reporting systematic reviews. *BMJ*, n160. https://doi.org/10.1136/bmj.n160

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training Recurrent Neural Networks. *ICML'13: Proceedings of the 30th International Conference on International Conference on Machine Learning*, *28*, 1310–1318.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *33rd Conference on Neural Information Processing Systems*, 12.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., & Cournapeau, D. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Provost, F., Fawcett, T., & Kohavi, R. (1998). The Case Against Accuracy Estimation for Comparing Induction Algorithms. *Proceedings of the Fifteenth International Conference on Machine Learning*, 9.

Riestra-González, M., Paule-Ruíz, M. del P., & Ortin, F. (2021). Massive LMS log data analysis for the early prediction of course-agnostic student performance. *Computers & Education*, *163*, 104108. https://doi.org/10.1016/j.compedu.2020.104108

Romero, C., Espejo, P. G., Zafra, A., Romero, J. R., & Ventura, S. (2013). Web usage mining for predicting final marks of students that use Moodle courses. *Computer Applications in Engineering Education*, *21*(1), 135–146. https://doi.org/10.1002/cae.20456

Romero, C., López, M.-I., Luna, J.-M., & Ventura, S. (2013). Predicting students' final performance from participation in on-line discussion forums. *Computers & Education*, *68*, 458–472. https://doi.org/10.1016/j.compedu.2013.06.009

Romero, C., & Ventura, S. (2010). Educational Data Mining: A Review of the State of the Art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *40*(6), 601–618. https://doi.org/10.1109/TSMCC.2010.2053532

Romero, C., & Ventura, S. (2020). Educational data mining and learning analytics: An updated survey. *WIREs Data Mining and Knowledge Discovery*, *10*(3). https://doi.org/10.1002/widm.1355

Rumelhart, D. E., Hintont, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533–536.

Sandoval, A., Gonzalez, C., Alarcon, R., Pichara, K., & Montenegro, M. (2018). Centralized student performance prediction in large courses based on low-cost variables in an institutional context. *The Internet and Higher Education*, *37*, 76–89. https://doi.org/10.1016/j.iheduc.2018.02.002

Saqr, M., Fors, U., & Tedre, M. (2017). How learning analytics can early predict under-achieving students in a blended medical education course. *Medical Teacher*, *39*(7), 757–767. https://doi.org/10.1080/0142159X.2017.1309376

Tomasevic, N., Gvozdenovic, N., & Vranes, S. (2020). An overview and comparison of supervised data mining techniques for student exam performance prediction. *Computers & Education*, *143*, 103676. https://doi.org/10.1016/j.compedu.2019.103676

Tsai, Y.-S., Rates, D., Moreno-Marcos, P. M., Muñoz-Merino, P. J., Jivet, I., Scheffel, M., Drachsler, H., Delgado Kloos, C., & Gašević, D. (2020). Learning analytics in European higher education—Trends and barriers. *Computers & Education*, *155*, 103933. https://doi.org/10.1016/j.compedu.2020.103933

Tsiakmaki, M., Kostopoulos, G., Kotsiantis, S., & Ragos, O. (2019). Implementing AutoML in Educational Data Mining for Prediction Tasks. *Applied Sciences*, *10*(1), 90. https://doi.org/10.3390/app10010090

Tsiakmaki, M., Kostopoulos, G., Kotsiantis, S., & Ragos, O. (2020). Transfer Learning from Deep Neural Networks for Predicting Student Performance. *Applied Sciences*, *10*(6), 2145. https://doi.org/10.3390/app10062145

Waheed, H., Hassan, S.-U., Aljohani, N. R., Hardman, J., Alelyani, S., & Nawaz, R. (2020). Predicting academic performance of students from VLE big data using deep learning models. *Computers in Human Behavior*, *104*, 106189. https://doi.org/10.1016/j.chb.2019.106189

Whitehill, J., Mohan, K., Seaton, D., Rosen, Y., & Tingley, D. (2017). Delving Deeper into MOOC Student Dropout Prediction. *ArXiv:1702.06404 [Cs]*. http://arxiv.org/abs/1702.06404

Xing, W., & Du, D. (2019). Dropout Prediction in MOOCs: Using Deep Learning for Personalized Intervention. *Journal of Educational Computing Research*, *57*(3), 547–570. https://doi.org/10.1177/0735633118757015

Yang, Y., Hooshyar, D., Pedaste, M., Wang, M., Huang, Y.-M., & Lim, H. (2020). Predicting course achievement of university students based on their procrastination behaviour on Moodle. *Soft Computing*, *24*(24), 18777–18793. https://doi.org/10.1007/s00500-020-05110-4

Yu, C.-C., & Wu, Y. (Leon). (2021). Early Warning System for Online STEM Learning—A Slimmer Approach Using Recurrent Neural Networks. *Sustainability*, *13*(22), 12461. https://doi.org/10.3390/su132212461

Yu, C.-H., Wu, J., & Liu, A.-C. (2019). Predicting Learning Outcomes with MOOC Clickstreams. *Education Sciences*, *9*(2), 104. https://doi.org/10.3390/educsci9020104

Yu, R., Li, Q., & Fischer, C. (2020). Towards Accurate and Fair Prediction of College Success: Evaluating Different Sources of Student Data. *International Educational Data Mining Society*, 10.

Zacharis, N. Z. (2015). A multivariate approach to predicting student outcomes in web-enabled blended learning courses. *The Internet and Higher Education*, *27*, 44–53. https://doi.org/10.1016/j.iheduc.2015.05.002

Zacharis, N. Z. (2018). Classification and Regression Trees (CART) for Predictive Modeling in Blended Learning. *International Journal of Intelligent Systems and Applications*, *10*(3), 1–9. https://doi.org/10.5815/ijisa.2018.03.01

APPENDIX A. LITERATURE REVIEW TABLE

In total, 39 research papers met the selection criteria laid out in section 2.1. This table summarises the main attributes of the research work used throughout our literature review. All of them share the following commonalities:

- All works use features extracted from LMS clickstream to predict student performance (independently of how performance is defined),
- Predict performance on courses intended for higher education students (the course itself may be face-to-face, MOOC or blended),
- All works use at least one of the following model performance metrics: Accuracy, Precision, Recall, F-score or AUROC.

It should be noted that, while some of these references go beyond the scope of our initial criteria (that may range from, e.g. adopting other model performance metrics to making a full-fledged analysis of student groups via clustering), our summarisation efforts were directed towards presenting the elements that fall under the main scope of this work. The consequence of this choice is that, for any given reference, our summary will not cover the entire body of work published in each specific paper.

Table A.1 – Literature review table

Reference	Population	Data sources	Target variable	Moment of prediction	Data representation	Best performance
Calvo-Flores et al. (2006)	1 course 240 students	LMS	**Final mark:** Pass/Fail	End of course	Static aggregate representation	**Accuracy:** 0.802 (ANN)
Macfadyen & Dawson (2010)	1 course 118 students	LMS	**Final mark:** At risk: <60% Not at risk: >60%	End of course	Static aggregate representation	**Accuracy:** 0.737 (LR) **Precision:** 0.703 (LR) **Recall:** 0.809 (LR)
Romero, Espejo, et al. (2013)	7 courses 438 students	LMS	**Final mark:** 0-10	End of course	Static aggregate representation	**Geometric mean of Accuracy:** 0.660 (NNEP)
Romero, López, et al. (2013)	1 course 114 students	LMS Message scores Survey	**Final mark:** Pass/Fail	Middle of course End of course	Static aggregate representations	**Middle of course** **Accuracy:** 0.824 (SMO/NB) **F-score:** 0.821 (SMO)

						End of course **Accuracy:** 0.903 (BayesNet/NB) **F-score:** 0.895 (BayesNet/NB)
Hu et al. (2014)	1 course 300 students	LMS	**Final mark:** Pass/Fail	4 weeks 8 weeks 13 weeks (end)	Static aggregate representations	**4 weeks** **Accuracy:** 0.972 (AdaBoost+CART/ AdaBoost+J48) **8 weeks** **Accuracy:** 0.978 (AdaBoost+CART)
Brooks et al. (2015)	4 courses 350k students	LMS	**Final Mark:** Pass/Fail	Throughout time End of course	Time-dependent representation	**End of course** **Accuracy:** >0.93 all courses (J48)
Kuzilek et al. (2015)	2 courses Unspecified number of students	LMS Student characteristics	**Final mark:** Pass/Fail	Results show prediction after each assessment The system is presented as able to predict every week	Static aggregate representation	**After second assessment** **Precision:** 0.885 **Recall:** 0.493 **F-score:** 0.574 (Average of 4 classifiers) **After fourth assessment** **Precision:** 0. 934 **Recall:** 0.250 **F-score:** 0.387 (Average of 4 classifiers)
Zacharis (2015)	1 course 134 students	LMS	**Final mark:** At-risk: <5.5/10 Not at risk: >5.5/10	End of course	Static aggregate representation	**Accuracy:** 0.813 (LR)

Gašević et al. (2016)	9 courses 4134 students	LMS Student characteristics	**Final mark:** Pass/Fail	End of course	Static aggregate representation	**General model** **AUROC:** 0.749 (LR) **Worst single course model** **AUROC:** 0.765 (LR) **Best single course model** **Accuracy:** 0.915 (LR)
Casey & Azcona (2017)	1 course 111 students	LMS	**Exam mark:** Pass/Fail	Every week End of course	Static aggregate representation	**At 10th week** **AUROC:** 0.80(CART) **End of course (16th week)** **AUROC:** 0.85 (CART)
Conijn et al. (2017)	17 courses 4989 students	LMS	**Exam mark:** Pass: >5.5/10 Fail: <5.5/10	Every week End of course	Static aggregate representation	**At 3rd week** **Accuracy:** 0.67 (LR) **End of course (10th week)** **Accuracy:** 0.69 (LR)
Costa et al. (2017)	2 courses 262 online & 141 campus students	LMS Student characteristics	**Final mark:** Pass/Fail	Every week End of course	Static aggregate representation	**At 3rd week** **F-score:** 0.83 (SVM) **End of course (5th week)** **F-score:** 0.92 (SVM)
Saqr et al. (2017)	1 course 133 students	LMS	**Final mark:** At-risk: <65% Not at-risk: >65%	Middle of course End of course	Static aggregate representation	**Middle of course** **AUROC:** 0.69 (LR) **End of course** **AUROC:** 0.90 (LR)

Whitehill et al. (2017)	40 courses 530k students	LMS Student characteristics	Dropout/ No dropout	Every week End of course	Static aggregate representation	**At 4th week** **AUROC:** 0.87 (LR - Trained on proxy labels) **End of course (8th week)** **AUROC:** 0.91 (LR - Trained on same course)
Adejo & Connolly (2018)	1 course 141 students	LMS Student characteristics Survey	Dropout/ No Dropout	End of course	Static aggregate representation	**Percentage of Accurate Predictions (PAP):** 0.83 (SVM – trained on survey data) **Precision:** 0.796 (Ensemble SVM+CART+ANN) **Recall:** 0.780 (Ensemble SVM+CART+ANN) **F-score:** 0.777 (Ensemble DVM+CART+ANN)
Helal et al. (2018)	Unspecified number of courses 4010 students	LMS Student characteristics	**Final mark:** Pass/Fail	End of course	Static aggregate representation	**Precision:** 0.68 (NB) **Recall:** 0.39 (NB – trained on LMS only) **F-score**: 0.48 (NB) **AUROC:** 0.70 (J48)
Mahzoon et al. (2018)	1 course 91 students	LMS Student characteristics Sentiment analysis	**Final mark:** Pass/Fail	End of course	Time-dependent representation	**Static baseline model** **Accuracy:** 0.849 (SVM) **Temporal model** **Accuracy:** 0.956 (Progressive classification – trained on LMS)
Sandoval et al. (2018)	Unspecified number of courses	Academic history	**Final mark:** Pass/Fail	Middle of course	Static aggregate representation	**Middle of course** **Average accuracy:** 0.844 (RF)

	21314 students	LMS Student characteristics		End of course		**PAP:** 0.578 (RF) **AF-Score:** 0.961 (RF) **RF-Score:** 0.514 (RF) **End of course** **Average accuracy:** 0.845 (RF) **PAP:** 0.583 (RF) **AF-Score:** 0.960 (RF) **RF-Score:** 0.525 (RF)
Zacharis (2018)	1 course 352 students	LMS	**Final mark:** Pass/Fail	End of course	Static aggregate representation	**Accuracy:** 0.991 (CART)
Aljohani et al. (2019)	7 courses 32593 students	LMS	**Final mark:** Withdraw/ Fail/Pass/Distinction	Multiple thresholds of course completion End of course	Time-dependent representation	**At 5th week** **Accuracy:** 0.802 (LSTM) **At 10th week** **Accuracy:** 0.900 (LSTM) **End of course (38th week)** **Accuracy:** 0.952 (LSTM)
Baneres et al. (2019)	608 courses 316k students	Data Mart (similarities with LMS) Grades of assessments	**Final mark:** Pass/Fail	Every assessment up to 90% course duration	Static aggregate representation	**Middle of course** **Accuracy:** 0.896 (SVM) **Recall:** 0.793 (NB) **90% completion threshold** **Accuracy:** 0.924 (SVM) **Recall:** 0.793 (NB)

Buschetto Macarini et al. (2019)	1 course 89 students	LMS	**Final mark:** Pass/Fail	Every week up to 50% course duration	Static aggregate representation	**Average across all weeks** **AUROC:** 0.920 (RF- DB5) **AUROC:** 0.961 (RF- DB5 using SMOTE)
Tsiakmaki et al. (2019)	3 courses 591 students	LMS	Dropout/ No dropout **Final mark:** Pass/Fail	Every month End of course	Static aggregate representation	**Physical chemistry course** **Dropout** **3rd month** **AUROC:** 0.863 (AutoWeka-LMT) **End of course (6th month)** **AUROC:** 0.896 (AutoWeka-LMT) **Pass/Fail** **3rd month** **Accuracy:** 0.812 (AutoWeka-LMT) **End of course (6th month)** **AUROC:** 0.816 (AutoWeka-PART)
Xing & Du (2019)	1 course 3617 students	LMS	Dropout/ No dropout	Every week up to the week before the final assignment	Static aggregate representation	**At 4th week** **Accuracy:** 0.966 (ANN) **AUROC:** 0.960 (ANN) **A week before the end of the course (7th week)**

						Accuracy: 0.974 (ANN) **AUROC:** 0.984 (ANN)
C.-H. Yu et al. (2019)	1 course 590 students	LMS	Earn certificate/ Not earn certificate	Every week End of course	Static aggregate representation	The authors did not publish weekly results **End of course** **Accuracy:** 0.955 (ANN)
Chen & Cui (2020)	1 course 668 students	LMS	**Final mark:** Good: B- or more Poor: C+ or less	Multiple thresholds of course completion	Time-dependent representation	**At 28th day** **AUROC:** 0.713 (LSTM) **At 42nd day** **AUROC:** 0.734 (LSTM) **At 56th day** **AUROC:** 0.752 (LSTM) **End of course (70th day)** **AUROC:** 0.738 (LSTM)
Chui et al. (2020)	7 courses 32593 students	LMS	**Final mark:** Pass/Fail **Final mark:** Fail/ Marginal Pass/Pass	End of course	Static aggregate representation	**Pass/Fail** **Accuracy:** [0.922, 0.938] (RTV-SVM) **Fail/Marginal Pass/Pass** **Accuracy:** [0.913, 0.935] (RTV-SVM)
Hasan et al. (2020)	2 courses 772 students	Degree history LSM	**Final mark:** Pass/Fail	End of course	Static aggregate representation	**Accuracy:** 0.883 (RF – equal width transformation and Information Gain selection) **AUROC:** 0.933 (RF – equal width transformation and

						Information Gain Ratio selection)
Huang et al. (2020)	7 courses Unspecified number of students	LMS Ebook reading behaviours	**Final mark:** **U1:** High class: >60 Low class: <60 **U2 and U3:** High class: >70 Low class: <70	End of course	Static aggregate representation	**Results for U1:** **Accuracy:** 0.88 (ANN) **Precision:** 0.90 (Gaussian NB) **Recall:** 0.88 (ANN) **F-Score:** 0.87 (LR; NB) **AUROC:** 0.86 (NB)
López-Zambrano et al. (2020)	24 courses 3235 students	LMS	**Final mark:** Pass: >5/10 Fail: <5/10	End of course	Static aggregate representation	**Experiment 1** **Best group: Computer** **AUROC:** 0.896 (J48) **Worst group: Engineering** **AUROC:** 0.576 (J48) **Experiment 2** **Best group: Low MOODLE use** **AUROC:** 0.758 (J48) **Worst group: High MOODLE use** **AUROC:** 0.576 (J48)
Tomasevic et al. (2020)	2 courses 3166 students	Degree history LMS Student characteristics	**Exam mark:** Pass/Fail	After each assessment End of course	Static aggregate representation	**At 3rd assessment** **F-score:** 0.86 (ANN – train on all sources) **End of course**

						F-score: 0.97 (ANN – train on assessments and activity logs)
Tsiakmaki et al. (2020)	5 courses 866 students	LMS Student characteristics	**Final mark:** Pass: >5/10 Fail: <5/10	End of course	Static aggregate representation	**Average accuracy:** 0.861 (ANN – epoch 100)
Waheed et al. (2020)	7 courses 32593 students	LMS Student characteristics	**Final mark**: Pass/Fail Withdraw/Pass Distinction/Fail Distinction/Pass	Every quarter End of course	Static aggregate representation	**Pass/Fail** **At 2nd quarter** **Accuracy:** 0.816 (ANN) **End of course** **Accuracy:** 0.845 (ANN) **Withdraw/Pass** **At 2nd quarter** **Accuracy:** 0.860 (ANN) **End of course** **Accuracy:** 0.845 (ANN) **Distinction/Fail** **At 2nd quarter** **Accuracy:** 0.816 (ANN) **End of course** **Accuracy:** 0.864 (ANN) **Distinction/Pass** **At 2nd quarter** **Accuracy:** 0.805 (ANN) **End of course** **Accuracy:** 0.805 (ANN)

Yang et al. (2020)	1 course 242 students	LMS	**Inferred grade:** A/B/C/D/E/F	End of course	Static aggregate representation	**Accuracy:** 0.846 (L-SVM – trained on categorical features 5 folds) **Precision:** 0.870 (L-SVM – trained on categorical features 10 folds) **F-score:** 0.857 (L-SVM – trained on continuous features 15 folds)
R. Yu et al. (2020)	10 courses 2090 students	LMS Student characteristics Survey	**Final mark:** Above median/ Below median	5 weeks (unspecified course duration)	Static aggregate representation	**Accuracy:** 0.675 (SVM – trained on all data sources)
Fahd et al. (2021)	1 course 122 students	LMS	**Exam mark:** Pass/Fail	After 6 lab sessions	Static aggregate representation	**Accuracy:** 0.857 (RF) **Precision:** 0.857 (RF) **Recall:** 0.857 (RF) **F-score:** 0.843 (RF)
Marras et al. (2021)	1 course 214 students	LMS	**Exam mark:** Pass: >4/6 Fail: <4/6 Above course average/ Below course average	Every week	Static aggregate representation	**Average Balanced Accuracy:** 0.64 (RF) **Average Pass Recall:** 0.78 (RF) **Average Fail Recall:** 0.43 (RF) **Average AUROC:** 0.43 (RF)
Riestra-González et al. (2021)	699 courses 15944 students	LMS	**Inferred grade:** At-risk: <2.5/10 Not at-risk: >2.5/10 Pass: <5/10	Multiple thresholds of course completion	Static aggregate representation	Showing results at 50% course duration **At-risk** **Accuracy:** 0.902 (MLP) **F-score:** 0.938 (MLP)

			Fail: >5/10 Excellent: >8.5/10 Not excellent: <8.5/10			**AUROC:** 0.958 (MLP) **Pass/Fail** **Accuracy:** 0.872 (MLP) **F-score:** 0.894 (MLP) **AUROC:** 0.947 (MLP) **Excellent** **Accuracy:** 0.901 (MLP) **F-score:** 0.942 (DT) **AUROC:** 0.935 (RF)
Yu & Wu (2021)	3 courses 234 students	LMS	**Final mark:** Pass/Fail	Every week	Time-dependent representation	**At 9th week** **Accuracy:** 0.67(RNN) **Precision:** 0.46(LSTM) **Recall:** 0.67 (RNN) **F-score:** 0.53(RNN) **End of course (18th week)** **Accuracy:** 0.93 (RNN; CNN) **Precision:** 0.87 (RNN; GRU) **Recall:** 0.95 (LSTM) **F-score:** 0.87 (RNN)

APPENDIX B. FEATURES USED

This section presents an overview of the aggregate features extracted from the Nova IMS MOODLE logs. Each feature name is accompanied by a short description of the feature and a list of research works where that specific feature, or similar, was used previously.

Table B.1 – Aggregate predictive features

Features Extracted (unit)	Description	Used in

Total clicks (n)	Number of clicks made in the course	Buschetto Macarini et al. (2019);
		Chen & Cui (2020);
		Conijn et al. (2017);
		Saqr et al. (2017);
		Tsiakmaki et al. (2019);
		Whitehill et al. (2017);
		R. Yu et al. (2020);
		Zacharis (2015)

Clicks (% of course total)	Number of clicks made in the course, relative to total clicks all students made in the course	Riestra-González et al. (2021)

Online sessions (n)	Number of online sessions	Calvo-Flores et al. (2006);
		Casey & Azcona (2017);
		Chen & Cui (2020);
		Conijn et al. (2017);
		Gašević et al. (2016);
		Hu et al. (2014);
		Macfadyen & Dawson (2010);
		Saqr et al. (2017);
		Tomasevic et al. (2020);
		C.-H. Yu et al. (2019);
		Zacharis (2015)

Clicks/session (n)	Total clicks / Online sessions	Adapted from Buschetto Macarini et al. (2019)
Clicks/day (n)	Total clicks/ number of days	Adapted from Buschetto Macarini et al. (2019)

Forum clicks (n)	Number of clicks on the course forum	Adejo & Connolly (2018);
		Aljohani et al. (2019);
		Brooks et al. (2015);
		Chen & Cui (2020);
		Costa et al. (2017);
		Gašević et al. (2016);
		Helal et al. (2018);
		López-Zambrano et al. (2020);
		Saqr et al. (2017);

		Tomasevic et al. (2020);
		Tsiakmaki et al. (2020);
		Whitehill et al. (2017);
		Xing & Du (2019)

Discussions viewed (n)	Number of discussions and course forum posts viewed	Conijn et al. (2017);
		Macfadyen & Dawson (2010);
		Romero, Espejo, et al. (2013);
		Sandoval et al. (2018);
		Whitehill et al. (2017)

Forum posts (n)	Number of posts and replies in discussions and course forum	Conijn et al. (2017);
		Helal et al. (2018);
		Hu et al. (2014);
		Huang et al. (2020);
		Macfadyen & Dawson (2010);
		Romero, Espejo, et al. (2013);
		Romero, López, et al. (2013);
		Sandoval et al. (2018);
		Saqr et al. (2017);
		Whitehill et al. (2017);
		Yu & Wu (2021);
		Zacharis (2015)

Folder clicks (n)	Number of clicks on folders	López-Zambrano et al. (2020); Tsiakmaki et al. (2020)

Resources viewed (n)	Number of course educational resources viewed	Adejo & Connolly, (2018);
		Aljohani et al. (2019);
		Calvo-Flores et al. (2006);
		Conijn et al. (2017);
		Gašević et al., (2016);
		Hu et al. (2014);
		López-Zambrano et al. (2020);
		Sandoval et al. (2018);
		Saqr et al. (2017);
		Tsiakmaki et al. (2019),
		Tsiakmaki et al. (2020);
		Zacharis (2015);
		Zacharis (2018)

URLs viewed (n)	Number of clicks on external links	Aljohani et al. (2019);
		Conijn et al. (2017);
		Macfadyen & Dawson (2010);
		Sandoval et al. (2018);
		Zacharis (2015)

Course clicks (n)	Number of clicks on course pages	Aljohani et al. (2019);
		Conijn et al. (2017);
		Helal et al. (2018);

		López-Zambrano et al. (2020);
		Saqr et al. (2017);
		Tsiakmaki et al. (2019)

Assessments started (n)	Number of assessments and quiz attempts started on MOODLE	Adejo & Connolly (2018);
		Brooks et al. (2015);
		Conijn et al. (2017);
		Helal et al. (2018);
		Macfadyen & Dawson (2010);
		Sandoval et al. (2018);
		Saqr et al. (2017);
		C.-H. Yu et al. (2019);
		Zacharis (2015);
		Zacharis (2018)

Assignments viewed (n)	Number of assignment page views	Conijn et al., (2017);
		Gašević et al. (2016);
		López-Zambrano et al. (2020);
		Macfadyen & Dawson (2010);
		Mahzoon et al. (2018);
		Riestra-González et al. (2021);
		Sandoval et al. (2018);
		Tsiakmaki et al. (2020);
		Xing & Du (2019)

Assignments submitted (n)	Number of assignments submitted (either via direct or Turnitin submission)	Conijn et al., (2017);
		Gašević et al. (2016);
		López-Zambrano et al. (2020);
		Macfadyen & Dawson (2010);
		Mahzoon et al. (2018);
		Riestra-González et al. (2021);
		Sandoval et al. (2018);
		Tsiakmaki et al. (2020);
		Xing & Du (2019)

Submissions (% of course total in period)	Number of submissions made in the course, relative to total submissions all students made in the course	Riestra-González et al. (2021)

Total time online (min)	Sum of the duration of all online sessions undertaken by the student	Adejo & Connolly (2018);
		Casey & Azcona (2017);
		Chen & Cui (2020);
		Conijn et al. (2017);
		Hu et al. (2014);
		Macfadyen & Dawson (2010);
		Saqr et al. (2017);
		Tomasevic et al., (2020);
		R. Yu et al. (2020);
		Zacharis, (2015)

Average duration of online sessions (min)	Total time online / Online sessions	Chen & Cui (2020);
		Conijn et al. (2017);
		Fahd et al. (2021);
		Hu et al. (2014)

Largest period of inactivity (h)	Largest temporal interval between consecutive online sessions	Conijn et al. (2017)
Days with 0 clicks (n)	Difference between the total number of days in the period and the number of days with at least one click (as used by Xing & Du, (2019)	-
Days with 0 clicks (% of period)	**Days with 0 clicks** in percentage form	-
Start time of nth session 1st Session: Time of first login as % of course duration … 10th Session: Time of tenth login as % of course duration	These are ten features: one for each of the first ten logins made by the student. It is calculated by the stage of course completion the login was made: The variable takes negative values if the login is made before the course start, 0% at the start of the course date and 100% at the end of course date.	Adapted from Riestra-González et al. (2021)

Average of partial grades (n)	Average of the partial glades obtained by the student	Baneres et al. (2019);
		Conijn et al. (2017);
		Costa et al. (2017);
		Riestra-González et al. (2021);
		Tomasevic et al. (2020)

APPENDIX C. HYPER-PARAMETERS

In this appendix, we disclose the hyper-parameters used for the feature selection and classification algorithms used throughout this work. The default value was used whenever a hyper-parameter is not mentioned. Table C.1 displays the hyper-parameters of the feature selection algorithms, Table C.2 displays the hyper-parameters used to train the traditional ML classifiers and Table C.3 displays the hyperparameters used for training the LSTM models.

Table C.1 – Hyper-parameters used in feature selection algorithms

Feature Selection Algorithm	Hyper-parameters

RFE	estimator = DecisionTreeClassifier()
	step = 2

RFECV	estimator = DecisionTreeClassifier()
	step = 1
	cv = 5

LR	penalty = l2
	max_features = 32

RF	n_estimators = 100
	max_features = 32

LGBM	n_estimators=500
	learning_rate=0.05
	num_leaves=32
	colsample_bytree=0.2
	reg_alpha=3
	reg_lambda=1
	min_split_gain=0.01
	min_child_weight=40

ElasticNet	cv = 5
	random_state = 123

Lasso Regression	random_state = 123
Ridge Regression	random_state = 123

Table C.2 – Hyper-parameters used on standard ML classifiers

ML Classifier	Hyper-parameters

KNN	n_neighbors=10
	weights='distance'

LR	tol=1e-05
	solver='liblinear'
	penalty='l1'
	max_iter =200

NB	-

MLP	alpha=0.01
	hidden_layer_sizes = (20,20)
	activation = 'relu'
	solver = 'adam'
	learning_rate = 'adaptive'
	verbose = 0
	learning_rate_init = 0.02

CART	criterion='gini'
	max_depth=10
	class_weight = 'balanced'

SVM	tol = 0.01
	probability = True
	gamma='scale'
	kernel='rbf'
	C = 1

RF	max_depth = 10
	random_state = 15
	n_estimators=500
	min_samples_leaf = 3

ExtraTrees	n_estimators=175
	criterion='entropy'
	max_depth = 10
	min_samples_split= 50

AdaBoost	n_estimators = 95
	learning_rate = 0.8
	random_state = 15

GBoost	n_estimators=175
	learning_rate=0.1
	random_state=15

Table C.3 – LSTM hyper-parameters

Hyper-parameter	Value
Number of epochs	200
Hidden state size	40
LSTM layers	1
Loss function	CrossEntropyLoss
Batch size	32
Dense layer activation function	LogSoftmax
h_0 initialization	Xavier normal
c_0 initialization	Xavier normal
Optimizer	Adam
Initial learning rate	0.01
ReduceLROnPlateau factor	0.1
ReduceLROnPlateau patience	10 epochs
ReduceLROnPlateau cooldown	20 epochs

APPENDIX D. MODEL PERFORMANCE: TRADITIONAL ML CLASSIFIERS

We publish all AUROC model performances obtained on all traditional ML classifiers in this appendix. In total, we created models to solve ten different classification problems: one per target for each moment of prediction. Moreover, we performed four different experiments for each classification problem:

Experiment 1: predictions are made using LMS features and partial grades.

Experiment 2: predictions are made using LMS features.

Experiment 3: predictions are made using LMS features and partial grades. The minority class is oversampled using SMOTE.

Experiment 4: predictions are made using LMS features. The minority class is oversampled using SMOTE.

Table D.1 displays the results for predicting the students at risk classification problem. Table D.2 does the same for the prediction of high-performing students.

Table D.1 – AUROC measures for predictions of students at risk (traditional ML classifiers)

Moment of prediction	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Algorithm
10%	0.638 +/-0.029	0.635 +/-0.029	0.626 +/-0.03	0.625 +/-0.03	KNN
	0.664 +/-0.023	0.658 +/-0.023	0.662 +/-0.023	0.656 +/-0.023	LR
	0.629 +/-0.024	0.617 +/-0.024	0.627 +/-0.024	0.616 +/-0.024	NB
	0.647 +/-0.023	0.651 +/-0.024	0.639 +/-0.025	0.640 +/-0.025	MLP
	0.622 +/-0.024	0.613 +/-0.026	0.539 +/-0.022	0.611 +/-0.024	CART
	0.619 +/-0.024	0.62 +/-0.024	0.612 +/-0.025	**0.670 +/-0.023**	SVM
	0.681 +/-0.023	**0.678 +/-0.023**	**0.674 +/-0.024**	0.668 +/-0.023	RF
	0.66 +/-0.023	0.653 +/-0.024	0.653 +/-0.023	0.647 +/-0.023	AdaBoost
	0.677 +/-0.024	0.668 +/-0.024	0.671 +/-0.024	0.666 +/-0.024	GBoost
	0.668 +/-0.023	0.663 +/-0.024	0.664 +/-0.023	0.656 +/-0.025	ExtraTrees
25%	0.692 +/-0.026	0.678 +/-0.024	0.679 +/-0.025	0.663 +/-0.024	KNN
	0.684 +/-0.023	0.682 +/-0.021	0.682 +/-0.023	0.679 +/-0.021	LR
	0.636 +/-0.025	0.628 +/-0.023	0.633 +/-0.025	0.626 +/-0.023	NB
	0.692 +/-0.026	0.676 +/-0.023	0.677 +/-0.027	0.664 +/-0.024	MLP
	0.655 +/-0.029	0.633 +/-0.023	0.647 +/-0.026	0.638 +/-0.025	CART
	0.678 +/-0.028	0.659 +/-0.023	0.712 +/-0.026	0.697 +/-0.021	SVM
	0.728 +/-0.024	**0.713 +/-0.022**	**0.725 +/-0.023**	**0.711 +/-0.021**	RF
	0.705 +/-0.025	0.691 +/-0.023	0.691 +/-0.025	0.679 +/-0.022	AdaBoost
	0.721 +/-0.025	0.709 +/-0.022	0.714 +/-0.025	0.704 +/-0.022	GBoost
	0.707 +/-0.025	0.695 +/-0.023	0.707 +/-0.026	0.694 +/-0.023	ExtraTrees

33%	0.700 +/-0.025	0.679 +/-0.022	0.686 +/-0.026	0.666 +/-0.023	KNN
	0.694 +/-0.023	0.691+/-0.022	0.693 +/-0.023	0.689 +/-0.022	LR
	0.651 +/-0.025	0.640 +/-0.023	0.651 +/-0.025	0.637 +/-0.023	NB
	0.715 +/-0.027	0.678 +/-0.023	0.698 +/-0.027	0.668 +/-0.023	MLP
	0.674 +/-0.027	0.637 +/-0.026	0.676 +/-0.027	0.642 +/-0.024	CART
	0.708 +/-0.027	0.670 +/-0.024	0.724 +/-0.025	0.700 +/-0.021	SVM
	0.749 +/-0.025	**0.718 +/-0.022**	**0.741 +/-0.025**	**0.714 +/-0.022**	RF
	0.727 +/-0.025	0.694 +/-0.021	0.711 +/-0.026	0.681 +/-0.024	AdaBoost
	0.743 +/-0.025	0.712 +/-0.022	0.734 +/-0.025	0.707 +/-0.022	GBoost
	0.728 +/-0.026	0.699 +/-0.023	0.724 +/-0.025	0.696 +/-0.023	ExtraTrees
50%	0.732 +/-0.024	0.690 +/-0.025	0.723 +/-0.023	0.679 +/-0.023	KNN
	0.693 +/-0.023	0.686 +/-0.022	0.693 +/-0.023	0.685 +/-0.022	LR
	0.659 +/-0.025	0.645 +/-0.024	0.657 +/-0.025	0.642 +/-0.024	NB
	0.764 +/-0.024	0.695 +/-0.023	0.748 +/-0.024	0.682 +/-0.023	MLP
	0.700 +/-0.026	0.646 +/-0.026	0.720 +/-0.022	0.651 +/-0.026	CART
	0.755 +/-0.023	0.696 +/-0.025	0.761 +/-0.022	0.715 +/-0.022	SVM
	0.793 +/-0.020	**0.730 +/-0.022**	**0.788 +/-0.020**	**0.724 +/-0.022**	RF
	0.768 +/-0.021	0.708 +/-0.023	0.761 +/-0.022	0.686 +/-0.023	AdaBoost
	0.789 +/-0.021	0.726 +/-0.022	0.785 +/-0.021	0.721 +/-0.022	GBoost
	0.767 +/-0.022	0.704 +/-0.023	0.766 +/-0.022	0.706 +/-0.023	ExtraTrees
100%	0.845 +/-0.017	0.724 +/-0.024	0.839 +/-0.018	0.714 +/-0.024	KNN
	0.781 +/-0.018	0.708 +/-0.023	0.783 +/-0.018	0.707 +/-0.023	LR
	0.719 +/-0.025	0.678 +/-0.023	0.716+/-0.025	0.677 +/-0.024	NB
	0.906 +/-0.015	0.720 +/-0.025	0.898+/-0.014	0.712 +/-0.024	MLP
	0.834 +/-0.024	0.676 +/-0.026	0.843+/-0.022	0.678 +/-0.025	CART
	0.888 +/-0.014	0.699 +/-0.025	0.894+/-0.012	0.727+/-0.023	SVM
	0.923 +/-0.011	**0.756 +/-0.023**	0.921+/-0.011	**0.752+/-0.023**	RF
	0.907 +/-0.013	0.720 +/-0.024	0.906+/-0.012	0.704+/-0.024	AdaBoost
	0.923 +/-0.011	0.751 +/-0.023	**0.922+/-0.011**	0.742+/-0.024	GBoost
	0.904 +/-0.013	0.723 +/-0.023	0.897+/-0.014	0.724+/-0.023	ExtraTrees

Table D.2 – AUROC measures for predictions of high-performing students (traditional ML classifiers)

Moment of prediction	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Algorithm
10%	0.550 +/-0.025	0.551 +/-0.023	0.544 +/-0.025	0.547 +/-0.022	KNN
	0.598 +/-0.02	**0.593 +/-0.023**	**0.596 +/-0.02**	**0.590 +/-0.023**	LR
	0.572+/-0.023	0.574 +/-0.022	0.574 +/-0.022	0.574 +/-0.022	NB
	0.557+/-0.024	0.559 +/-0.025	0.548 +/-0.023	0.550 +/-0.024	MLP
	0.539 +/-0.022	0.539 +/-0.025	0.544 +/-0.023	0.545 +/-0.024	CART
	0.56 +/-0.025	0.558 +/-0.021	0.539 +/-0.025	0.579 +/-0.021	SVM
	0.593 +/-0.023	0.591 +/-0.023	0.585 +/-0.023	0.586 +/-0.022	RF
	0.580 +/-0.021	0.578 +/-0.022	0.580 +/-0.021	0.580 +/-0.022	AdaBoost
	0.577+/-0.021	0.580 +/-0.022	0.567 +/-0.02	0.568 +/-0.022	GBoost
	0.592+/-0.023	0.592 +/-0.022	0.589+/-0.022	**0.590 +/-0.023**	ExtraTrees
25%	0.569 +/-0.023	0.564 +/-0.022	0.560 +/-0.022	0.557 +/-0.024	KNN
	0.609 +/-0.022	0.602 +/-0.021	0.607 +/-0.022	**0.601 +/-0.022**	LR
	0.583 +/-0.023	0.579 +/-0.022	0.584 +/-0.023	0.581 +/-0.022	NB
	0.566 +/-0.023	0.566 +/-0.023	0.561 +/-0.023	0.559 +/-0.023	MLP
	0.540 +/-0.025	0.539 +/-0.024	0.556 +/-0.021	0.550 +/-0.024	CART
	0.573 +/-0.023	0.573 +/-0.021	0.590 +/-0.022	0.589 +/-0.022	SVM
	0.611 +/-0.023	**0.603 +/-0.021**	0.603 +/-0.022	0.598 +/-0.022	RF
	0.599 +/-0.023	0.591 +/-0.022	0.595 +/-0.023	0.591 +/-0.023	AdaBoost
	0.601 +/-0.023	0.587 +/-0.021	0.598 +/-0.022	0.593 +/-0.022	GBoost
	0.611 +/-0.022	0.603 +/-0.022	**0.608 +/-0.022**	0.600 +/-0.022	ExtraTrees
33%	0.584 +/-0.022	0.569 +/-0.024	0.574 +/-0.021	0.560 +/-0.023	KNN
	0.609 +/-0.022	0.603 +/-0.022	0.607 +/-0.022	0.602 +/-0.022	LR
	0.586 +/-0.023	0.580 +/-0.023	0.588 +/-0.023	0.582 +/-0.022	NB
	0.575 +/-0.024	0.570 +/-0.024	0.568 +/-0.023	0.564 +/-0.024	MLP
	0.562 +/-0.026	0.548 +/-0.022	0.556 +/-0.024	0.552 +/-0.022	CART
	0.585 +/-0.022	0.570 +/-0.024	0.600 +/-0.021	0.590 +/-0.021	SVM
	0.619 +/-0.022	**0.606 +/-0.022**	0.613 +/-0.022	0.600 +/-0.021	RF
	0.607 +/-0.022	0.593 +/-0.021	0.602 +/-0.021	0.590 +/-0.022	AdaBoost
	0.624 +/-0.021	0.597 +/-0.02	0.611 +/-0.02	**0.601 +/-0.023**	GBoost
	0.619 +/-0.022	0.603 +/-0.023	**0.617+/-0.022**	0.600 +/-0.022	ExtraTrees
50%	0.598 +/-0.022	0.586 +/-0.021	0.590 +/-0.022	0.582 +/-0.022	KNN
	0.615 +/-0.021	0.608 +/-0.021	0.613 +/-0.021	0.607 +/-0.021	LR
	0.593 +/-0.022	0.593 +/-0.022	0.591 +/-0.023	0.593 +/-0.021	NB
	0.594 +/-0.024	0.581 +/-0.023	0.588 +/-0.022	0.576 +/-0.025	MLP
	0.581 +/-0.024	0.551 +/-0.024	0.567 +/-0.024	0.554 +/-0.024	CART

	0.601 +/-0.023	0.582 +/-0.024	0.614 +/-0.022	0.603 +/-0.022	SVM
	0.644 +/-0.021	**0.611 +/-0.021**	0.631 +/-0.021	0.607 +/-0.021	RF
	0.631 +/-0.021	0.599 +/-0.022	0.622 +/-0.021	0.599 +/-0.021	AdaBoost
	0.641 +/-0.022	0.603 +/-0.022	0.630 +/-0.021	0.600 +/-0.021	GBoost
	0.635 +/-0.022	**0.611+/-0.021**	**0.632 +/-0.022**	**0.610 +/-0.022**	ExtraTrees
100%	0.636 +/-0.022	0.596 +/-0.022	0.623 +/-0.022	0.587 +/-0.021	KNN
	0.636 +/-0.022	0.617 +/-0.022	0.634 +/-0.022	0.616 +/-0.022	LR
	0.609 +/-0.021	0.599 +/-0.021	0.608 +/-0.021	0.600 +/-0.022	NB
	0.700 +/-0.025	0.591 +/-0.023	0.673 +/-0.023	0.584 +/-0.022	MLP
	0.704 +/-0.024	0.563 +/-0.023	0.708 +/-0.022	0.569 +/-0.025	CART
	0.669 +/-0.02	0.591 +/-0.023	0.690 +/-0.019	0.618 +/-0.021	SVM
	0.782 +/-0.018	**0.634 +/-0.021**	0.756 +/-0.019	0.621 +/-0.021	RF
	0.782 +/-0.017	0.609 +/-0.020	0.755 +/-0.018	0.607 +/-0.021	AdaBoost
	0.796 +/-0.017	0.622+/-0.020	**0.781 +/-0.017**	0.616 +/-0.020	GBoost
	0.725 +/-0.02	0.626+/-0.021	0.720 +/-0.021	**0.626 +/-0.021**	ExtraTrees

APPENDIX E. COMPARISON BETWEEN TRADITIONAL ML CLASSIFIERS AND SINGLE FEATURE LSTM

In this appendix, we showcase the model performances on all metrics (Accuracy, Precision, Recall and AUROC) for our LSTM models at every moment of prediction. Moreover, we also include the performances for each of the corresponding best traditional classifiers. Rows in bold highlight rows where LSTM is a comparable alternative to the best standard ML classifier. Tables E.1 and E.2 show, respectively, the results obtained using Nova IMS data for students at risk and high-performing students. Tables E.3 and E.4 do the same for the second dataset used in Riestra-González et al. (2021)

Table E.1 – Model performances for students at risk using Nova IMS data (LSTM and best traditional ML classifiers)

Moment	Model	Accuracy	Precision	Recall	AUROC
10%	**RF**	0.802 +/-0.023	0.713 +/-0.208	0.028 +/-0.013	0.681 +/-0.023
	RF – SMOTE	0.733 +/-0.024	0.354 +/-0.029	0.396 +/-0.041	0.674 +/-0.024
	LSTM	**0.793 +/-0.006**	**0.301 +/-0.185**	**0.027 +/-0.020**	**0.614 +/-0.024**
	LSTM - SMOTE	0.651 +/-0.028	0.259 +/-0.017	0.562 +/-0.050	0.606 +/-0.024
25%	**RF**	0.804 +/-0.024	0.607 +/-0.120	0.086 +/-0.028	0.728 +/-0.024
	RF – SMOTE	0.725 +/-0.023	0.376 +/-0.027	0.500 +/-0.043	0.725 +/-0.023
	LSTM	**0.790 +/-0.008**	**0.372 +/-0.129**	**0.071 +/-0.040**	**0.636 +/-0.025**
	LSTM - SMOTE	0.590 +/-0.054	0.261 +/-0.016	0.563 +/-0.053	0.614 +/-0.024
33%	**RF**	0.812 +/-0.025	0.678 +/-0.086	0.123 +/-0.027	0.749 +/-0.025
	RF – SMOTE	0.736+/-0.025	0.387 +/-0.029	0.527 +/-0.050	0.741 +/-0.022
	LSTM	**0.786 +/-0.011**	**0.414 +/-0.090**	**0.143 +/-0.043**	**0.640 +/-0.024**
	LSTM - SMOTE	0.594 +/-0.027	0.262 +/-0.017	0.559 +/-0.052	0.619 +/-0.025
50%	**RF**	0.825 +/-0.020	0.723 +/-0.023	0.205 +/-0.035	0.793 +/-0.020
	RF – SMOTE	0.764 +/-0.020	0.436 +/-0.028	0.577 +/-0.040	0.788 +/-0.020
	LSTM	0.793 +/-0.010	0.458 +/-0.102	0.172 +/-0.055	0.624 +/-0.027
	LSTM - SMOTE	0.612 +/-0.037	0.271 +/-0.021	0.542 +/-0.063	0.682 +/-0.023
100%	**RF**	0.886 +/-0.011	0.817 +/-0.035	0.563 +/-0.044	0.923 +/-0.011
	GBoost – SMOTE	0.849 +/-0.011	0.588 +/-0.024	0.844 +/-0.032	0.921 +/-0.011
	LSTM	0.792 +/-0.011	0.468+/-0.069	0.210 +/-0.040	0.680 +/-0.025
	LSTM - SMOTE	0.609 +/-0.040	0.275+/-0.021	0.576 +/-0.073	0.638 +/-0.028

Table E.2 – Model performances for high-performing using Nova IMS data (LSTM and best traditional ML classifiers)

Moment	Model	Accuracy	Precision	Recall	AUROC
10%	**LR**	0.723 +/-0.020	0.504 +/-.0137	0.033 +/-0.012	0.598 +/-0.020
	LR – SMOTE	0.592 +/-0.020	0.592 +/-0.034	0.508 +/-0.034	0.596 +/-0.020
	LSTM	**0.720 +/-0.005**	**0.275 +/-0.217**	**0.013 +/-0.014**	**0.563 +/-0.021**
	LSTM - SMOTE	0.489 +/-0.051	0.303 +/-0.015	0.647 +/-0.101	0.555 +/-0.023
25%	**RF**	0.725 +/-0.023	0.544 +/-0.117	0.044 +/-0.016	0.611 +/-0.023
	ExtraTrees – SMOTE	0.581 +/-0.022	0.342 +/-0.019	0.552 +/-0.035	0.608 +/-0.022
	LSTM	**0.717 +/-0.007**	**0.366 +/-0.161**	**0.031 +/-0.022**	**0.563 +/-0.023**
	LSTM - SMOTE	**0.487 +/-0.054**	**0.299 +/-0.014**	**0.631 +/-0.082**	**0.543 +/-0.023**
33%	**GBoost**	0.731 +/-0.021	0.572 +/-0.076	0.110 +/-0.021	0.624 +/-0.021
	ExtraTrees – SMOTE	0.599 +/-0.022	0.353 +/-0.020	0.537 +/-0.034	0.617 +/-0.022
	LSTM	0.715 +/-0.009	0.413 +/-0.103	0.064 +/-0.028	0.565 +/-0.021
	LSTM - SMOTE	0.500 +/-0.036	0.301 +/-0.016	0.608 +/-0.070	0.549 +/-0.026
50%	**RF**	0.735 +/-0.021	0.655 +/-0.021	0.095 +/-0.019	0.644 +/-0.021
	ExtraTrees – SMOTE	0.611 +/-0.022	0.365 +/-0.020	0.548 +/-0.037	0.632 +/-0.022
	LSTM	0.715+/-0.008	0.368 +/-0.141	0.046 +/-0.029	0.567 +/-0.022
	LSTM - SMOTE	0.505 +/-0.025	0.306 +/-0.013	0.621 +/-0.047	0.557 +/-0.021
100%	**GBoost**	0.782 +/-0.017	0.668 +/-0.034	0.423 +/-0.029	0.796 +/-0.017
	ExtraTrees – SMOTE	0.763 +/-0.017	0.575 +/-0.029	0.560 +/-0.034	0.781 +/-0.017
	LSTM	0.708 +/-0.012	0.401 +/-0.102	0.117 +/-0.049	0.589 +/-0.022
	LSTM - SMOTE	0.473 +/-0.050	0.304 +/-0.017	0.694 +/-0.082	0.560 +/-0.026

Table E.3 – Model performances for students at risk using the Riestra-González et al. (2021) data (LSTM and best traditional ML classifiers)

Moment	Model	Accuracy	Precision	Recall	AUROC
10%	**MLP**	0.686 +/-0.018	0.638 +/-0.036	0.416+/-0.051	0.726 +/-0.016
	RF – SMOTE	0.633 +/-0.015	0.512 +/-0.013	0.747 +/-0.023	0.725 +/-0.015
	LSTM	0.618 +/-0.003	0.441 +/-0.252	0.012 +/-0.021	0.552 +/-0.017
	LSTM - SMOTE	0.489 +/-0.029	0.403 +/-0.012	0.706 +/-0.075	0.542 +/-0.018
25%	**GBoost**	0.743 +/-0.013	0.724 +/-0.023	0.527 +/-0.026	0.806 +/-0.013
	RF – SMOTE	0.722 +/-0.013	0.608 +/-0.018	0.762 +/-0.026	0.818 +/-0.013
	LSTM	0.621 +/-0.008	0.511 +/-0.107	0.087 +/-0.031	0.561 +/-0.019
	LSTM - SMOTE	0.489 +/-0.023	0.399 +/-0.011	0.674 +/-0.072	0.536 +/-0.019
33%	**RF**	0.780 +/-0.011	0.779 +/-0.020	0.591 +/-0.024	0.847 +/-0.011
	RF – SMOTE	0.740+/-0.011	0.626 +/-0.015	0.790 +/-0.021	0.848 +/-0.011
	LSTM	0.618 +/-0.007	0.458 +/-0.151	0.056 +/-0.042	0.555 +/-0.019
	LSTM - SMOTE	0.515 +/-0.025	0.409 +/-0.013	0.610 +/-0.083	0.546 +/-0.018
50%	**RF**	0.820 +/-0.010	0.791 +/-0.018	0.716 +/-0.023	0.891 +/-0.010
	RF – SMOTE	0.890 +/-0.020	0.705 +/-0.017	0.822 +/-0.018	0.890 +/-0.009
	LSTM	0.624 +/-0.009	0.499 +/-0.144	0.123 +/-0.070	0.579 +/-0.025
	LSTM - SMOTE	0.523 +/-0.027	0.411 +/-0.017	0.579 +/-0.090	0.548 +/-0.023

Table E.4 – Model performances for high-performing students using the Riestra-González et al. (2021) data (LSTM and best traditional ML classifiers)

Moment	Model	Accuracy	Precision	Recall	AUROC
10%	**RF**	0.810 +/-0.018	0.779 +/-0.100	0.063 +/-0.019	0.707 +/-0.018
	RF – SMOTE	0.757 +/-0.015	0.409 +/-0.055	0.412 +/-0.067	0.704 +/-0.018
	LSTM	0.801 +/-0.002	0.145 +/-0.268	0.003 +/-0.006	0.539 +/-0.020
	LSTM - SMOTE	0.445 +/-0.148	0.211 +/-0.018	0.628 +/-0.216	0.526 +/-0.023
25%	**RF**	0.836 +/-0.017	0.740 +/-0.046	0.268 +/-0.026	0.797 +/-0.017
	RF – SMOTE	0.815 +/-0.017	0.534 +/-0.029	0.544 +/-0.036	0.795 +/-0.017
	LSTM	0.801 +/-0.003	0.382 +/-0.244	0.016 +/-0.017	0.553 +/-0.022
	LSTM - SMOTE	0.355 +/-0.079	0.202 +/-0.009	0.754 +/-0.113	0.514 +/-0.027
33%	**RF**	0.848 +/-0.016	0.741 +/-0.040	0.360 +/-0.030	0.832 +/-0.016
	RF – SMOTE	0.828 +/-0.016	0.544 +/-0.027	0.614 +/-0.034	0.828 +/-0.016
	LSTM	0.798 +/-0.005	0.308 +/-0.203	0.036 +/-0.038	0.584 +/-0.027
	LSTM - SMOTE	0.449 +/-0.092	0.209 +/-0.015	0.628 +/-0.125	0.528 +/-0.031
50%	**Gboost**	0.865 +/-0.012	0.726 +/-0.032	0.519 +/-0.034	0.883 +/-0.012
	GBoost – SMOTE	0.791 +/-0.011	0.700 +/-0.016	0.792 +/-0.020	0.874 +/-0.011
	LSTM	0.801 +/-0.001	0.151 +/-0.281	0.002 +/-0.004	0.509 +/-0.024
	LSTM - SMOTE	0.518 +/-0.130	0.224 +/-0.023	0.545 +/-0.180	0.543 +/-0.028