JOURNAL OF LEARNING ANALYTICS

SOLAR SOCIETY for LEARNING ANALYTICS RESEARCH

Volume 12(2), 259–278. https://doi.org/10.18608/jla.2025.8375

Interpretable Predictive Analytics for Online Learning: A Markov-Based Machine Learning Approach

Chaewon Lee[1], Lan Luo[2], Shelbi L. Kuhlmann[3], Robert D. Plumley[4], Abigail T. Panter[5], Matthew L. Bernacki[6], Jeffrey A. Greene[7] and Kathleen M. Gates[8]

Abstract

The increasing use of learning management systems (LMSs) generates vast amounts of clickstream data, opening new avenues for predicting learner performance. Traditionally, LMS predictive analytics have relied on either supervised machine learning or Markov models to classify learners based on predicted learning outcomes. Machine learning excels at pattern recognition but often overlooks temporal learning dynamics and obscures the reasoning behind predictions due to the black-box nature of many algorithms. Alternatively, Markov models provide an effective solution by capturing temporal learning dynamics for prediction, uncovering distinctive learning patterns between high and low performers. Despite these advantages, Markov model classification struggles with the heterogeneity of learning sequences, limiting its broad applicability. To address these limitations and bridge the gap between the two dominant approaches, we propose a hybrid framework: sequence-based Markov machine learning classification (seqMAC). Leveraging early-stage clickstream data, seqMAC provides an interpretable sequence classification method that captures critical behavioural transitions and identifies distinct learning patterns across performance groups. Tested on six LMS samples, seqMAC effectively identified at-risk students despite sequence heterogeneity, uncovering key predictive learning dynamics that differentiate performance groups. It also demonstrated promising generalizability, accurately identifying future at-risk students based on historical clickstream data.

Notes for Practice

- LMS predictive analytics, which use clickstream data to identify at-risk learners, have primarily relied on supervised machine learning or Markov models, both of which come with inherent limitations.
- We introduce seqMAC, a hybrid framework that combines the strengths of both approaches while addressing their limitations, effectively identifying at-risk students, capturing distinct learning patterns between performance groups, and enabling transition-based interpretations of key learning dynamics.
- The findings suggest seqMAC's potential to produce generalizable classifiers across academic terms, which, when applied over multiple academic terms, may better support at-risk learners by enabling accurate real-time alerts on behaviours needing improvement and guiding future LMS research on proactive interventions.

Keywords: Predictive learning analytics, learning management system, machine learning, Markov chain, explainable AI

Submitted: 11/02/2024 — **Accepted:** 01/05/2025 — **Published:** 30/08/2025

Corresponding author [1]Email: chaewon.lee@unc.edu Address: L. L. Thurstone Psychometric Laboratory, Department of Psychology and Neuroscience, The University of North Carolina at Chapel Hill, 235 E. Cameron Avenue, Chapel Hill, NC, 27599, USA. ORCID iD: https://orcid.org/0009-0008-0981-025X

[2]Email: lanl27@live.unc.edu Address: L. L. Thurstone Psychometric Laboratory, Department of Psychology and Neuroscience, The University of North Carolina at Chapel Hill, 235 E. Cameron Avenue, Chapel Hill, NC, 27599, USA. ORCID iD: https://orcid.org/0000-0003-0779-2808

[3]Email: s.kuhlmann@memphis.edu Address: Department of Psychology, Institute for Intelligent Systems, The University of Memphis, 400 Fogelman Dr., Memphis, Tennessee, 38152 USA. ORCID iD: https://orcid.org/0000-0002-6451-6765

[4]Email: rplumley@live.unc.edu Address: School of Education, The University of North Carolina at Chapel Hill, 100 E. Cameron Avenue, Chapel Hill, NC, 27514, USA. ORCID iD: https://orcid.org/0000-0001-8979-6276

[5]Email: panter@unc.edu Address L. L. Thurstone Psychometric Laboratory, Department of Psychology and Neuroscience, The University of North Carolina at Chapel Hill, 235 E. Cameron Avenue, Chapel Hill, NC, 27599, USA. ORCID iD: https://orcid.org/0000-0001-6914-8490

[6]Email: mlb@unc.edu Address: School of Education, The University of North Carolina at Chapel Hill, 100 E. Cameron Avenue, Chapel Hill, NC, 27514, USA. ORCID iD: https://orcid.org/0000-0003-1279-2829

[7]Email: jagreene@email.unc.edu Address: School of Education, The University of North Carolina at Chapel Hill, 100 E. Cameron Avenue, Chapel Hill, NC, 27514, USA. ORCID iD: https://orcid.org/0000-0003-4145-1847

[8]Email: kmgates@unc.edu Address: L. L. Thurstone Psychometric Laboratory, Department of Psychology and Neuroscience, The University of North Carolina at Chapel Hill, 235 E. Cameron Avenue, Chapel Hill, NC, 27599, USA. ORCID iD: https://orcid.org/0000-0002-1246-4529

1. Introduction

With digital technologies rapidly advancing, higher education institutions have increasingly adopted learning management systems (LMSs) such as Canvas and Sakai in recent years, enhancing learning, teaching, and learner–instructor interactions (Baker et al., 2020; Weaver et al., 2008). The growing influx of learners interacting with LMS generates ample clickstream data on learning behaviours, fuelling research on predicting learner outcomes and identifying those at-risk (Arizmendi et al., 2023; Bernacki, Chavez, & Uesbeck, 2020). Prior LMS prediction research mainly adopted classification approaches to categorize learners into low and high performers based on their predicted exam scores (e.g., Arizmendi et al., 2023; Bernacki, Chavez, & Uesbeck, 2020), course/degree completion (e.g., Alamri et al., 2019; J. Chen et al., 2024), or probabilistic evaluation of learning sequences (e.g., Fok et al., 2005; Gupta et al., 2022). In the existing body of work, supervised machine learning (ML) and Markov models have served as the primary research methodologies. ML has gained notable popularity due to its robust predictive power and efficiency in extracting meaningful patterns from data. However, most ML studies have relied on per-behaviour aggregated counts, overlooking the essential temporal dynamics that characterize LMS clickstreams. Although modern deep learning models like recurrent neural networks can handle sequential data, their black-box nature limits interpretability, making them less suitable for transparent learning analytics. Meanwhile, Markov models, including Markov chain models (MCMs) and hidden Markov models (HMMs), have proven effective in previous studies for analyzing LMS clickstream sequences in their original format to investigate learning dynamics (e.g., Fok et al., 2005; Geigle & Zhai, 2017; Gupta et al., 2022; Kokoç et al., 2021; Staufer et al., 2024; Sun et al., 2019; Tang et al., 2023; Tran & Hasegawa, 2022). In particular, Markov model classification has emerged as a prominent alternative to ML classification. Akin to ML models, it builds a predictive model from training data and evaluates its classification performance on test data. Additionally, it offers clear interpretations of distinct learning dynamics across performance groups, a unique advantage that ML approaches do not provide. However, its effectiveness is challenged by high inter-sequence heterogeneity, where learner behaviours vary significantly — not only in the order of behaviours but also in the presence or absence of certain behaviours. Specifically, "sparse" behaviours — those that occur infrequently and appear only in the test data — render Markov model classification ineffective, since these behaviours were not represented during training. These limitations call for an approach that retains the interpretability and sequence modelling capabilities of Markov models while overcoming their constraints, which directly motivated this study. At the same time, ML models, while not limited in the same way, often fail to capture temporal dynamics and lack transparency. This contrast underscores the need for a hybrid framework that integrates the strengths of both approaches while overcoming their limitations, prompting us to pose three critical research questions:

RQ1: Can learner performance be predicted using temporal dynamics in LMS clickstreams without being constrained by inter-sequence heterogeneity?

RQ2: Can key predictive learning dynamics be identified beyond static features like click event counts?

RQ3: What unique learning patterns distinguish different performance groups?

To address these questions, we propose a hybrid approach named sequence-based Markov machine learning classification (seqMAC). seqMAC is designed to synergize the predictive flexibility of ML with the interpretability and sequence modelling capabilities of Markov models. The novelty of seqMAC is twofold: 1) it uses sequence-wise MCMs to capture dynamic components from LMS clickstream sequences and leverage them for ML classification, and 2) it incorporates explainable AI techniques to identify critical behaviour transitions, shifting the focus of model interpretation from the static features to dynamic features. This study progresses as follows. Section 2 reviews related work based on supervised ML and Markov models and discusses their limitations. Section 3 introduces the seqMAC framework alongside the Markov model classification as a benchmark method. Section 4 describes the empirical LMS data and the model evaluation strategy. Section 5 presents results from 1) sample-level analysis and 2) generalizability analysis. In our sample-level analysis, we apply seqMAC separately to six LMS samples to build local classifiers and compare their prediction performance against the benchmark classifiers. We then use explainable AI to analyze key learning dynamics and conduct *t*-tests to examine distinctive learning patterns between low and high performers. In our generalizability analysis, we build a seqMAC global classifier from past student sequences and test its generalizability in predicting future student outcomes. Finally, Section 6 discusses the practical implications of seqMAC, along with methodological considerations for its use, limitations, and future research directions to further enhance its utility.

2. Related Work

2.1. Supervised Machine Learning Classification

ML classification has been at the forefront of LMS predictive analytics research. In this approach, ML algorithms build predictive models from training data to identify patterns within clickstream data and validate their performance against separate test data to predict learner performance. Over the years, ML-based LMS research has primarily relied on traditional algorithms like support vector machines and random forests as well as those with feature engineering capabilities like Elastic Net and

Lasso (e.g., Arizmendi et al., 2023; Bernacki, Chavez, & Uesbeck, 2020; Chui et al., 2020; Tamada et al., 2022; Van Goidsenhoven et al., 2020). Recently, deep learning algorithms have emerged as frontrunners, outperforming traditional algorithms (e.g., Al-Sulami et al., 2023; F. Chen & Cui, 2020; C.-A. Lee et al., 2021; Nanavaty & Khuteta, 2024; Tsiakmaki et al., 2020; Wen & Juan, 2023). However, most ML studies have rarely accounted for the dynamic nature of LMS sequences into predictive modelling. Instead, they have typically relied on per-behaviour aggregated counts collected at the individual level, overlooking the temporal dynamics embedded in these sequences. This limitation primarily stems from the constraints of many ML algorithms, which are ill-suited for handling sequences as input. Although some deep learning algorithms, such as recurrent neural networks, can process sequences, they often create black-box models, making it difficult for end-users to understand the internal reasoning behind the predictions (Rudin & Radin, 2019). This lack of transparency raises concerns about the reliability of the results since users cannot easily trace how specific features influence predictions (Ribeiro et al., 2016). To address this, explainable AI methods for sequence-based deep learning models (e.g., TimeSHAP; Bento et al., 2021) have been developed to interpret feature importance at specific time points. However, they focus on individual time points or intervals rather than evolving behavioural transitions, which are essential for interpreting sequential data in a wider scope.

2.2. Benchmark: Markov Model Classification

Methods that fall under the category of Markov models, such as MCMs and HMMs, have been utilized to analyze distinct learning dynamics between low and high performers and predict learner outcomes (e.g., Elmäng, 2020; Fok et al., 2005; Gupta et al., 2022; Witteveen & Attewell, 2017[1]). Markov models investigate sequences representing time-evolving random processes (i.e., Markov processes), where the current state depends solely on the immediately preceding state (Ibe, 2013; Rabiner & Juang, 1986). MCMs, the foundational form of Markov models, focus on analyzing "observed" events in LMS clickstreams, modelling learning events with the principle that a learner's behaviour at time t depends on their behaviour at time $t-1$ (Faisan et al., 2007; Heins & Stern, 2014). In contrast, HMMs posit that observed event sequences are governed by underlying "hidden" state sequences, where these hidden states presumably represent latent factors influencing observed events (Eddy, 2004). Markov model classification is effective for sequence analysis, providing advantages over ML classification by eliminating the need to convert LMS sequences to aggregated counts. It also allows the interpretation of distinctive learning patterns across performance groups. However, in practice, we encountered its limited applicability when individual LMS clickstream sequences were highly heterogeneous. This challenge directly prompted us to develop the seqMAC approach, with the Markov model classification method serving as the benchmark. In the upcoming subsection, we outline the operational aspects of this benchmark method and the specific scenario where its effectiveness is constrained.

2.2.1. Benchmark Operations and Limitations

Markov model classification involves five steps:

1. Learners are categorized into two groups based on academic performance: high performers (group H) and low performers (group L).
2. Within each group, a training set (train^H & train^L) and a test set (test^H & test^L) are generated.
3. Group-specific Markov models (model^H & model^L), using either MCM or HMM, are built from their respective training sets.
4. The likelihood of each test sequence being associated with each group's Markov model is computed.
5. Each test sequence is classified into group H or L based on the higher likelihood.

The Benchmark method enables the interpretation of learning dynamics by comparing Markov model parameters between group H and L. It is noteworthy that Markov models were not originally designed for classification purposes but for in-depth exploration of sequences embodying Markov processes. In fact, Markov model classification may not always be applicable, especially when dealing with highly heterogeneous sequences. Here, "heterogeneous" denotes situations where certain behaviours are so "sparse" that they only appear in the "test" set. Essentially, the Markov classifier can predict the categories of test sequences only when the behaviours in the test set also exist in the training set, since their predictions rely on patterns identified during training. For instance, if the test sequences contain behaviours like "submitting homework (HW)," "taking quizzes (QZ)," and "attending discussions (DS)," but the training sequences only include "HW" and "QZ" and lack "DS," then "DS" is considered a sparse behaviour. Since the classifier has not encountered "DS" during training, it cannot predict test sequences including this behaviour. To address this limitation of the benchmark method, we aim to develop sequence classifiers that are consistently applicable and not constrained by the presence of sparse behaviours.

[1] Witteveen & Attewell (2017) was cited for implementing the benchmark method, despite using non-LMS digital transcript data.

3. Methods

In this section, we introduce seqMAC, a method for interpretable LMS predictive analytics that integrates the predictive power of supervised ML with the sequence analysis abilities of Markov models. seqMAC specifically leverages MCMs to extract dynamic components (i.e., transition probabilities) between observed behaviours from individual learning processes, which then serve as features for ML classification. Transition probabilities from a behaviour k to j are calculated by Equation 1.

$$T_{k \to j} = \frac{\text{number of transitions moving from a behavior } k \text{ to a behavior } j}{\text{number of transitions starting from a behavior } k} \quad (1)$$

LMS researchers have historically preferred HMMs over MCMs to emphasize latent states, such as learner mastery level, which may underpin observed learning behaviours in LMS clickstreams. In this hybridization, however, seqMAC incorporates MCMs instead of HMMs for two essential reasons. First, defining hidden states in educational research is challenging because multiple latent factors can influence learner behaviours, leading to ambiguous interpretations without a strong theoretical basis. Exploratory analysis may be required, but it may not always clarify latent states. Second, given the heterogeneous nature of individual learning processes, utilizing HMMs to derive "individual-level" transition probabilities risks measurement non-invariance. Measurement non-invariance denotes the situation when the same construct holds different meanings across individuals, leading to inconsistent or incomparable results (Putnick & Bornstein, 2016). Unlike MCMs, which calculate transition probabilities based on directly observed events, HMMs rely on transitions between inferred latent states, which may vary in meaning for different individuals. These discrepancies complicate comparing transition probabilities across learners and render them invalid as features for ML classification, undermining prediction reliability. In this regard, using HMM-derived transition probabilities as ML input requires strictly homogeneous learning processes across individuals, but this contradicts the recognized variability in individual learning processes (Caeiro-Rodríguez et al., 2005; Rizvi et al., 2019). Given these challenges, seqMAC uses MCMs to obtain transition probabilities between observed behaviours for ML classification.

3.1. seqMAC Framework

The objective of seqMAC is to leverage the inherent learning dynamics within LMS clickstreams to predict learner outcomes consistently and interpretably, addressing inter-sequence heterogeneity effectively. seqMAC operates through three distinct phases: 1) dynamic feature generation, 2) prediction, and 3) post-hoc interpretation. In this section, we detail each phase of seqMAC, including the methods and algorithms used.

3.1.1. Phase 1: Dynamic Feature Generation

As an initial step, seqMAC uses MCMs to extract dynamic features, represented as transition probabilities, from individual LMS clickstream sequences. Each sequence can have up to p behaviours, with p^2 representing the maximum number of possible transitions in a sample. seqMAC starts by constructing a null matrix of size $n \times p^2$, where n is the number of learners. Each learner's sequence is processed through an MCM to generate an individual transition probability matrix, with matrix dimensions varying depending on the number of behaviours each learner engages in. Transition probabilities are then mapped onto corresponding locations in the null matrix, with absent behavioural transitions assigned a probability of zero. Figure 1 illustrates the process of generating a dynamic feature matrix that captures the temporal learning dynamics of two learners in a unified representation, where A, B, and C represent unique learning behaviours. When the number of observed behaviours (p) is large, the dimensionality of the dynamic feature matrix, which scales with the number of possible transitions (p^2), can quickly outgrow the sample size. To mitigate this issue, we recommend applying data preprocessing if the number of possible transitions exceeds one-tenth of the sample size, following a general rule in ML (Banerjee et al., 2019). For this, we used logistic lasso regression to retain only key behaviours relevant to predicting student outcomes: reducing computational complexity and improving classifier performance by shrinking the coefficients of less significant predictors toward zero (Tibshirani, 1996). Using aggregated counts of each behaviour within the analyzed time window as predictors and binary final exam scores (see the following Section 3.1.2. for details) as the outcome, we identified distinct sets of key behaviours per sample from 42 total behaviours (see behaviours marked with ü in Table 1), suggesting inter-sample heterogeneity. The refined LMS sequences, containing only key behaviours for each sample, are then used to generate the dynamic feature matrix in Phase 1. Note that these refined sequences are also utilized for Markov benchmark modelling to allow a parallel comparative evaluation of the two approaches.

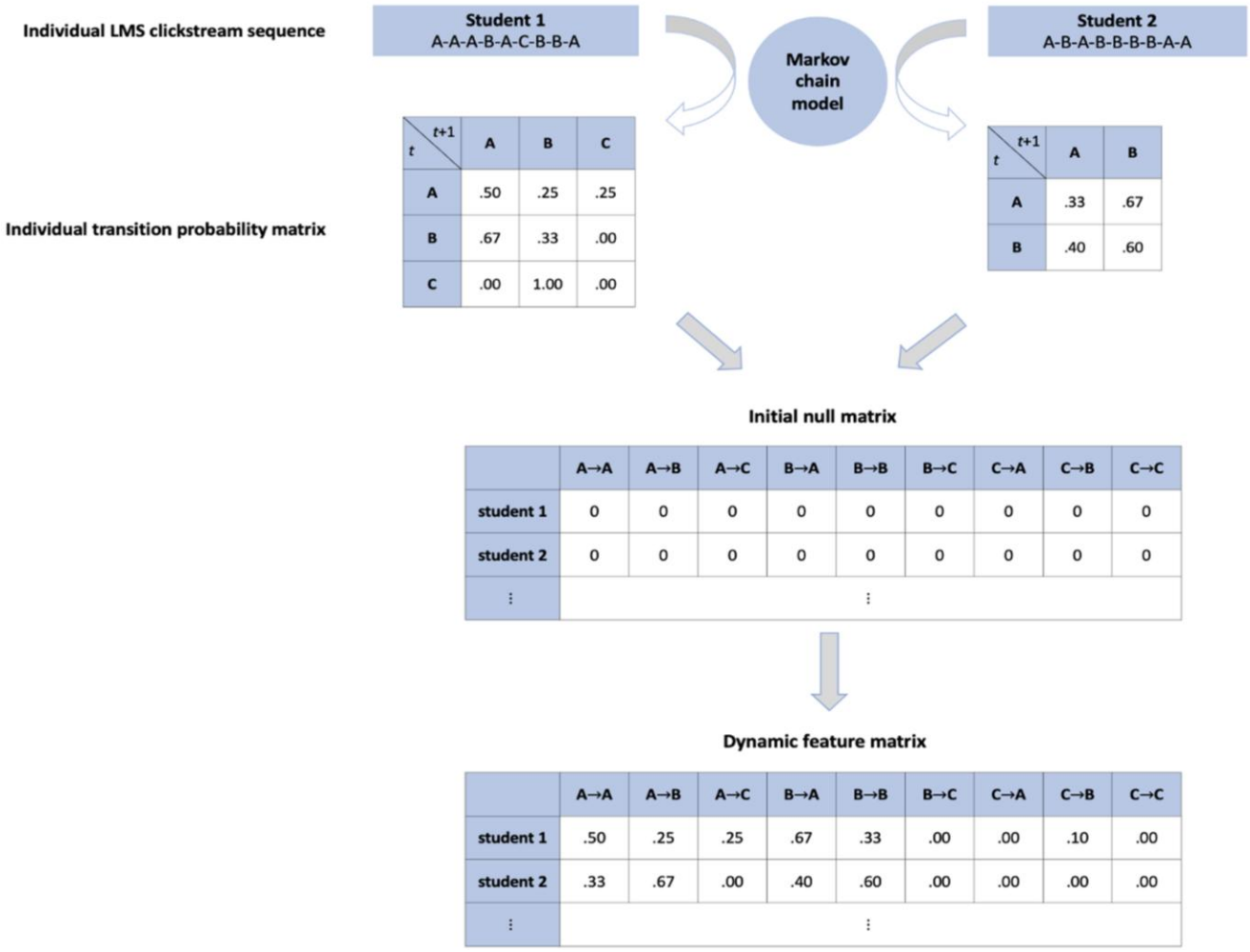


Figure 1. Illustration of dynamic feature matrix generation.

3.1.2. Phase 2: Prediction

In Phase 2, seqMAC leverages the individual dynamic features obtained from Phase 1 to perform predictive modelling with ML algorithms. Each feature is tagged with the student's performance category based on final exam scores (out of 100): group L (low performers; scores below 80) or group H (high performers; scores 80 and above). These individual dynamic features, along with the binary outcome, are then used as input for ML classification. Our rationale for selecting 80 as the performance cutoff stems from its authentic significance within the context of this study. In the setting where this research was conducted, 80 was the minimum required score for students to progress to the next course in their major; those who scored below this threshold had to retake the course. This threshold likely had a significant impact on student behaviours since they recognized the importance of achieving at least this score. To further assess seqMAC's general applicability, we tested alternative cutoffs at 60, 70, and 90, comparing its performance to Markov benchmark classifiers across these cutoff values. To determine the best ML algorithm through empirical testing, we applied both traditional ML and deep learning algorithms to explore a range of algorithms with different characteristics. For traditional ML, we applied tree algorithms (random forests [RF]; stochastic gradient boosting [SGB]) and support vector machines with linear (LSVM) and non-linear (RSVM) kernels. RF creates ensemble models by generating multiple decision trees on bootstrapped training data. It uses majority voting for classification and enhances tree diversity by randomly selecting feature subsets for each tree (Breiman, 2001). SGB sequentially fits weak classifiers, typically shallow decision trees. By subsampling without replacement in each iteration, it helps reduce overfitting and correlation among classifiers (Friedman, 2002). SVM seeks the optimal hyperplane that separates classes by maximizing the margin, which is the distance between the hyperplane and the nearest data points from each class, known as support vectors (James et al., 2013). LSVM is suited for linearly separable data, whereas RSVM maps non-linear data into higher dimensions, allowing it to form more complex boundaries. For deep learning, we used a deep feedforward neural network (DNN), which mimics human brain processes by using multiple hidden layers between input and output layers and learns complex patterns by optimizing weights and biases via forward and backward propagation (Hastie et al., 2017; C. Lee et al., 2025). The optimized hyperparameters for each algorithm are described in Supplementary Table 1.

3.1.3. Phase 3: Post-hoc Interpretation

In Phase 3, seqMAC conducts post-hoc analysis to explore key learning transitions and distinct learning dynamics between low and high performers. Previous LMS predictive analytic work increasingly applies explainable AI, mainly focusing on interpreting key static features like aggregated behaviour counts, student demographics, and prior exam scores (e.g., Adnan et al., 2022; Afzaal et al., 2021; Jang et al., 2022). In contrast, seqMAC focuses on interpreting key dynamic features by analyzing behavioural transitions that occur in early learning processes using two model-agnostic explainable AI methods: 1) permutation feature importance (PFI) and 2) Shapley additive explainer (SHAP). PFI evaluates feature importance on classifier accuracy by shuffling each feature and observing changes in prediction error (Greenwell & Boehmke, 2020). A positive PFI score indicates that shuffling the feature worsened performance, meaning the feature is important. In contrast, a negative PFI score suggests the feature's removal could improve performance. SHAP calculates each feature's average contribution to predictions across all combinations, viewing predictions as cumulative effects of features (Lundberg et al., 2020; Shapley, 1953). To assess contributions across the student body rather than for a specific student, we computed global SHAP scores by averaging the absolute SHAP values from each student, capturing each feature's overall contribution to predicted outcomes (Greenwell & Boehmke, 2020). Additionally, we perform PFI and SHAP analyses on test data to evaluate feature importance during model validation. Finally, *t*-tests are performed on the group means of transition probabilities associated with key learning behaviours to identify distinct learning patterns between high and low performers. Figure 2 provides an overview of the seqMAC framework.

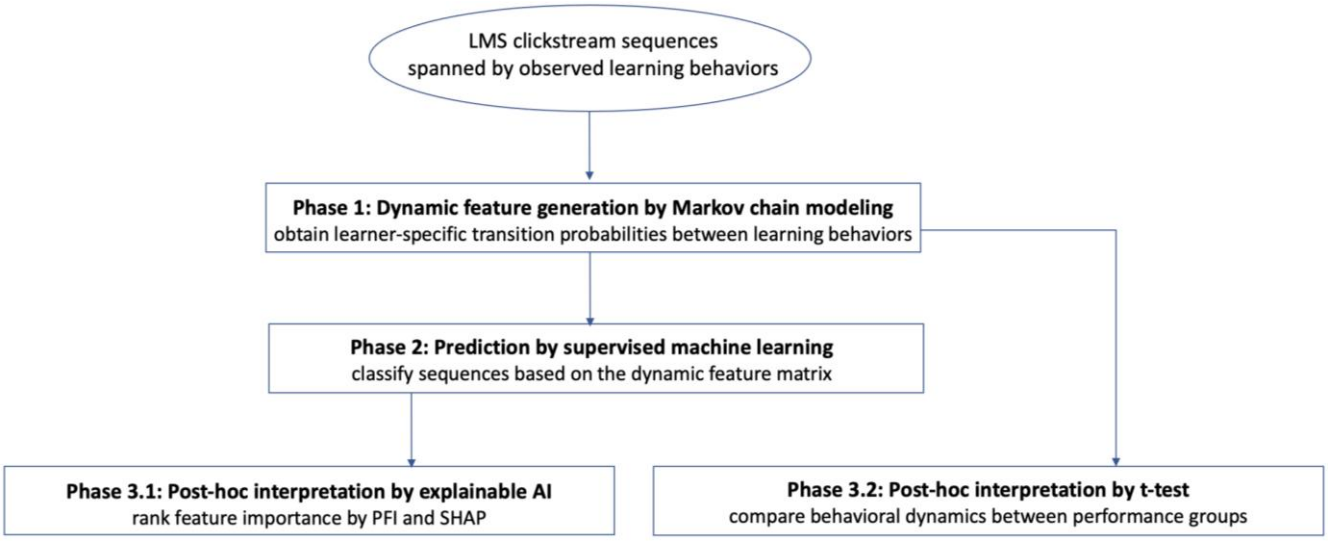


Figure 2. seqMAC framework.

4. Data and Evaluation Strategy

4.1. LMS Data[2]

LMS clickstreams are timestamped online learning click events automatically recorded as learners engage with pedagogical resources (Baker et al., 2020). These clickstreams naturally form sequences of unequal intervals, preserving temporal learning dynamics. In this study, clickstream data were collected from various LMS platforms, including Sakai, Pearson's Mastering Bio, and Piazza. The data were gathered from six sections of an undergraduate-level biology course across three terms: Spring 2019, Fall 2019, and Spring 2020 (two classes per term: 19SP-1, 19SP-2, 19FA-1, 19FA-2, 20SP-1, 20SP-2). Each term spanned 17 weeks (Spring 2019: Jan 9–May 7, Fall 2019: Aug 20–Dec 13, Spring 2020: Jan 8–May 5).

Figure 3 illustrates a subset of LMS clickstreams from three students analyzed in this study. Each row represents a click event. These timestamps span an entire semester and are organized into sequences after sorting and reorganizing the clickstreams into a wide format by student. Students were anonymized with opaque IDs. As the primary objective of predicting student outcomes is to proactively identify at-risk students, we trained classifiers on subsegments of sequences from the initial phase of learning, up to the completion of Lesson 3, roughly corresponding to the first four weeks of each term. Table 1 lists 35 key learning behaviours, such as syllabus review, homework submission, and quiz participation, selected from 42 total behaviours across six sections during data preprocessing. Seven behaviours, mainly related to practise and prior exams, were filtered out in this process as they tended to occur closer to exam periods, whereas our focus was on early-stage sequences.

[2] The R code and LMS dataset used in this study are available at https://osf.io/frwh9/

Figure 3. A sample of LMS clickstream data.

Table 1. Key LMS Learning Behaviours

Behaviour tag	Description	19SP-1	19SP-2	19FA-1	19FA-2	20SP-1	20SP-2
adlInfo.dwnld	Access course materials		✓		✓	✓	
adlRdgs.dwnld	Download reading materials	✓					
annc.read	Read course announcements				✓		
attend.SI.acCoach	1:1 academic coaching					✓	
attend.SI.bioCell	Biology review session					✓	
attend.SI.review	Review with TA/mentor		✓			✓	
attend.SI.tutoring	Biology peer tutoring				✓	✓	
attend.SI.wrCoach	1:1 writing coaching					✓	
caldr.read	View calendar events	✓					
caldr.revise	Edit calendar events		✓				
clsOln.dwnld	Download class outlines		✓		✓	✓	
forum.other	Other forum activities			✓			
forum.question	Pose forum question		✓				
forum.read	Read forum content					✓	
forum.response	Reply on forum			✓			
gradebook.read	View gradebook	✓		✓			✓
grq.dwnld	Download general reading questions					✓	
grq.newSubm	Start GRQ submission					✓	
grq.review	Review GRQ response	✓					
grq.reviseSubm	Edit GRQ draft	✓					
grq.submit	Submit GRQ				✓		
hw.hintlist	Use homework hints					✓	
hw.item	View homework item					✓	
hw.solution	Get homework solutions		✓		✓	✓	
hw.submCorr	Submit correct homework answer	✓	✓		✓	✓	✓
hw.submIncorr	Submit incorrect homework answer	✓	✓	✓	✓	✓	✓
lctSld.dwnld	Download lecture slides				✓	✓	
lsnPg.read	Load lesson page		✓				
prctExam.submit	Submit practice exam		✓				
prctExam.take	Start the practice exam					✓	
qz.submCorr	Submit correct quiz answer		✓				
qz.submIncorr	Submit incorrect quiz answer			✓	✓	✓	
selfRef.dwnld	Download self-reflection				✓		
stdyGd.dwnld	Download study guide				✓	✓	
syllabusPg.read	Access syllabus				✓	✓	

4.2. Heterogeneous Samples

In this study, we treated each section as an independent sample due to the observed heterogeneity across sections. Table 2 presents the descriptive statistics and demographic information for the LMS samples. First, the distribution of sequence lengths varied across sections (mean = 304–592 and SD = 46.3–92.8; see Supplementary Figure 1). Sections offered during the COVID-19 pandemic (20SP-1, 20SP-2) exhibited longer maximum sequence lengths than pre-pandemic sections, likely reflecting differences in learning styles due to different class formats (in-person vs. hybrid/remote). Second, lecture days varied by section, resulting in differences in lecture-day spacing and likely the order of learning behaviours. Third, there was a notable performance gap, with pre-pandemic sections having a higher proportion of students in group L compared to those during the pandemic. Figure 4 illustrates variations in behaviour transitions across samples, where each coloured square represents a transition from a behaviour in the row to a behaviour in the column. The colour gradation indicates sample-averaged transition probabilities. The 20SP-2 sample exhibited the fewest transitions, whereas 20SP-1 had the most. Differences in the distribution and intensity of coloured squares reflect variations in transition compositions and probabilities, collectively indicating high inter-sample heterogeneity.

Table 2. LMS Sample Descriptive Statistics and Demographics

		19SP-1	19SP-2	19FA-1	19FA-2	20SP-1	20SP-2
Descriptive Statistics							
Number of students		213	157	222	266	252	107
Instructor		A	B	B	C	A	B
Days of class		TT	MWF	MW	TT	TT	MW
Sequence length	min.	2	128	41	58	9	20
	max.	454	456	487	473	859	640
	mean	329	304	328	326	592	325
	standard deviation	61.0	47.6	46.3	49.1	92.8	72.4
Performance	group H	105	71	87	131	175	63
	group L	108	86	135	135	77	44
Demographic Information							
Female		0.734	0.625	0.680	0.714	0.729	0.610
First generation college student		0.192	0.270	0.247	0.229	0.207	0.229
Under-represented minorities in STEM		0.197	0.322	0.190	0.277	0.258	0.252
Ethnicity							
Caucasian		0.671	0.612	0.721	0.662	0.730	0.654
Asian		0.169	0.127	0.135	0.139	0.103	0.131
African American		0.099	0.185	0.095	0.154	0.131	0.112
Hispanic or Latinx		0.079	0.099	0.072	0.109	0.103	0.112
Non-Hispanic Latinx		0.052	0.051	0.050	0.053	0.064	0.047
Native American		0.019	0.038	0.041	0.015	0.028	0.028
Hawaiian or other Pacific Islanders		0.014	0.006	0.009	0.000	0.004	0.009
Not specified		0.024	0.038	0.041	0.023	0.028	0.019

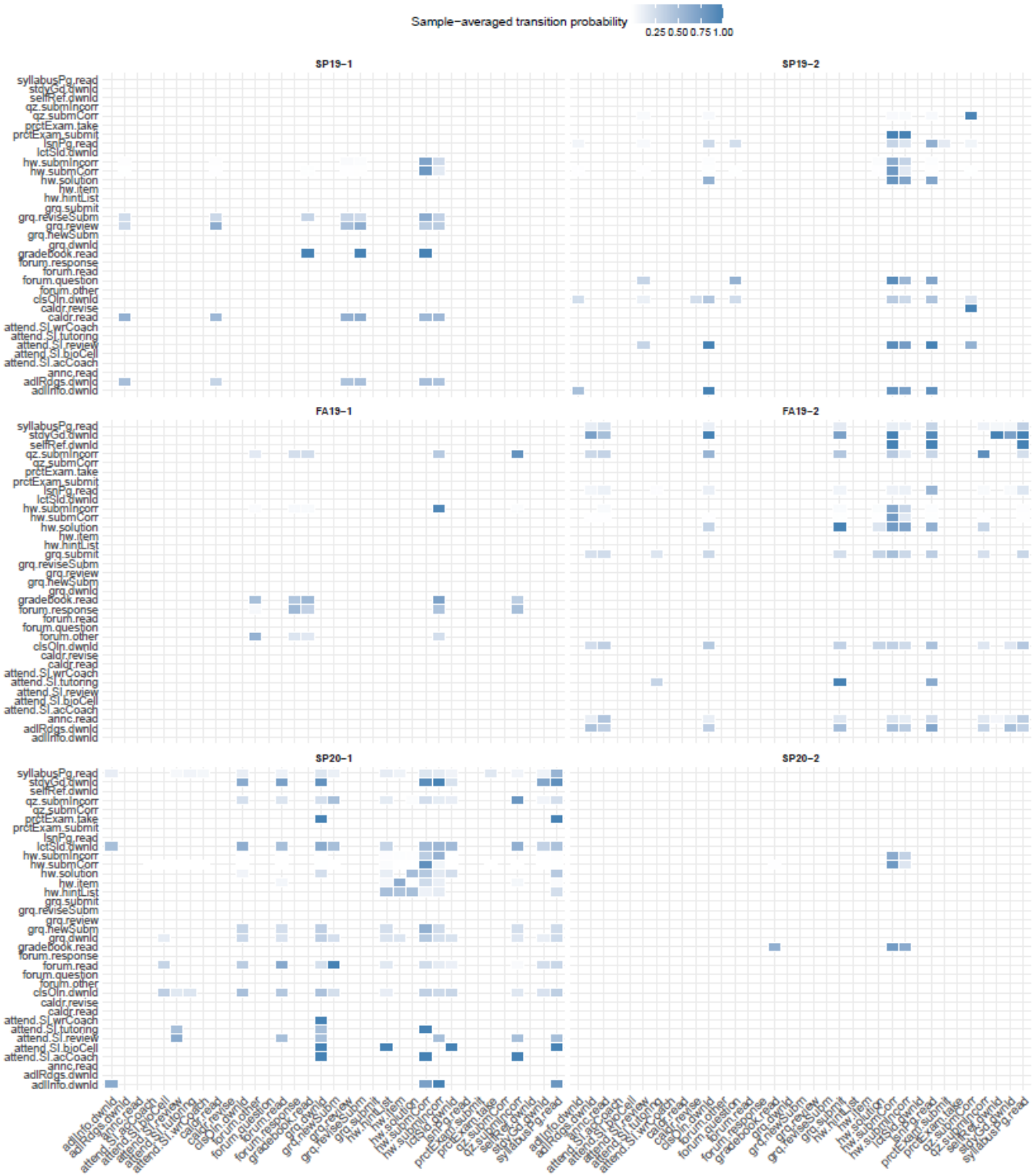


Figure 4. Inter-sample heterogeneity in behaviour transitions.

4.3. Performance Evaluation Strategy

4.3.1. Benchmark Markov Classifiers

For benchmark evaluation, we followed the approach of Fok et al. (2005) and Gupta et al. (2022), using hold-out validation to construct benchmark Markov classifiers with HMMs. We used a 60% training and 40% test data split, using HMMs with three hidden states, which provided the highest accuracy when the benchmark method was applicable for classification.

4.3.2. seqMAC Classifiers

We evaluated seqMAC's performance through 1) sample-level analysis and 2) generalizability analysis. For the sample-level analysis, we applied seqMAC separately to six LMS samples, building local classifiers for each. Performance was assessed using 10-fold cross-validation, dividing each sample's dynamic feature matrix into 10 subsets, with one for validation and the rest for training, repeated 10 times. For the generalizability analysis, we constructed a global classifier from a combined historical sample and evaluated its predictive ability on two future samples using hold-out validation.

4.3.3. Model Evaluation Metrics

The chosen metrics for evaluating the performance of the benchmark and seqMAC classifiers align with our goal of proactively identifying at-risk students, who are treated as positive cases. We used sensitivity and F_β scores to assess how well positive cases were predicted, along with balanced accuracy as an overall summary metric. Sensitivity measures the ability to correctly identify positives among actual positives, indicating the extent of false negatives, where low performers are misclassified as not at-risk. However, sensitivity alone does not account for false positives — cases where high performers are misclassified as at-risk. To address this, we used the F_β score, which balances sensitivity and precision. Precision measures the proportion of correctly identified at-risk students among all predicted at-risk cases, indicating the rate of false positives. The β parameter allows adjustment of the relative importance of sensitivity versus precision. Since our primary goal was to evaluate the risk of false negatives — ensuring that at-risk students are not overlooked — we set β to 2, placing greater emphasis on sensitivity over precision. At the same time, balanced accuracy was used to ensure fair evaluation across class imbalances, preventing an overemphasis on positive predictions.

4.4. Software

The analysis was conducted in R 4.2.2 (R Core Team, 2022). Markov modelling was performed with the "RcppHMM" (Cardenas-Ovando et al., 2017) and "seqHMM" (Helske & Helske, 2019) packages. Supervised ML models (RF, SGB, and SVMs) were built using the "caret" package (Kuhn, 2008), while DNN models were built with the "keras" package (Kalinowski et al., 2024). For explainable AI, we utilized the "vip" (Greenwell & Boehmke, 2020) and the "fastshap" (Greenwell, 2024) packages. Logistic Lasso regression in data preprocessing was performed using "glmnet" (Friedman et al., 2010).

5. Results

5.1. Sample-Level Analysis

5.1.1. Prediction of Student Success

In this section, we evaluate the performance of seqMAC and the benchmark Markov classifiers (Figure 5). The benchmark method (shown in blue) performed effectively for only one sample (20SP-2), achieving 61% sensitivity, 63% F2 score, and 71% balanced accuracy. This success was likely due to the sample's low inter-sequence heterogeneity, as it contained only three key learning behaviours that were sufficiently distributed across both the training and the test data. This alignment satisfied the benchmark's requirement that all behaviours present in the test data must also appear in the training data. However, the benchmark method failed to generate predictions for the remaining five samples, where higher inter-sequence heterogeneity caused certain behaviours to appear only in the test data but not in the training data. Since these sparse behaviours were absent during training, the benchmark could not classify test sequences containing them. This indicates that the benchmark's functionality depends on low inter-sequence heterogeneity and struggles to handle substantial behavioural variation between training and test data.

In contrast, seqMAC consistently demonstrated classification ability in all samples, effectively managing inter-sequence heterogeneity. This success is attributed to seqMAC's representation of absent behaviours with zero transition probabilities, ensuring uniform feature vector lengths for all individuals. Figure 5 ranks seqMAC classifiers (shown in black) by balanced accuracy, with sample-representative classifiers highlighted in bold lollipop bars. Overall, tree-based algorithms outperformed DNN and SVM, with SGB performing best and LSVM worst. These results suggest SGB as the most suitable algorithm for predicting student outcomes in this biology class. The best performing classifier among the six was the SGB model for 20SP-2, which achieved 82% balanced accuracy, 80% F2 score, and 81% sensitivity, accurately identifying over 80% of low performers with high precision and effectively distinguishing between high and low performers. However, seqMAC's performance in 19FA-2 was moderate, with accuracy below 70%, possibly due to unrecognized behavioural differences not fully captured by the current algorithms. Exploring additional algorithms within seqMAC may help detect these missed patterns and improve performance. Additionally, higher within-sample variability, the presence of complex or noisier behaviour sequences, and underrepresentation of key transitions might have further limited seqMAC's ability to effectively capture essential patterns in this sample.

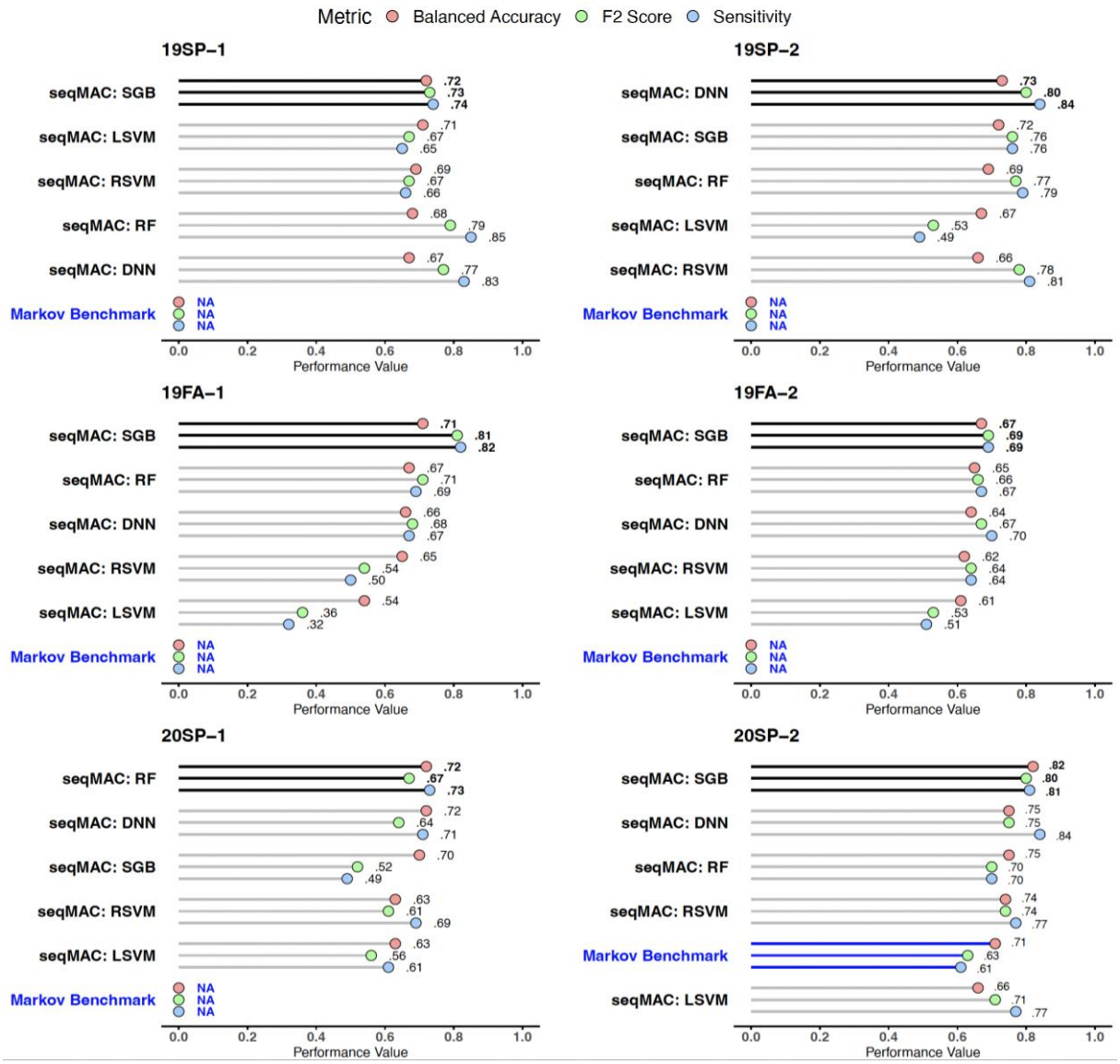


Figure 5. Comparison of seqMAC and benchmark classifier performance.

To further assess seqMAC's generalizability and its performance relative to the benchmark, we conducted additional analyses using alternative cutoff scores of 60, 70, and 90 (Supplementary Figure 2). seqMAC successfully generated predictions for all samples across these thresholds, whereas the benchmark method produced results for only one sample at a cutoff of 60, two at 70, and none at 90. Even when predictions were available, the benchmark consistently underperformed relative to seqMAC's top-performing models. Notably, at the 60 cutoff, seqMAC achieved even higher performance than at the primary threshold of 80, with balanced accuracy ranging from 77% to 91%. Although seqMAC demonstrated robust performance across all tested cutoff scores, we report results based on the 80 cutoff in the main text since it aligns with the actual passing criterion used in the course.

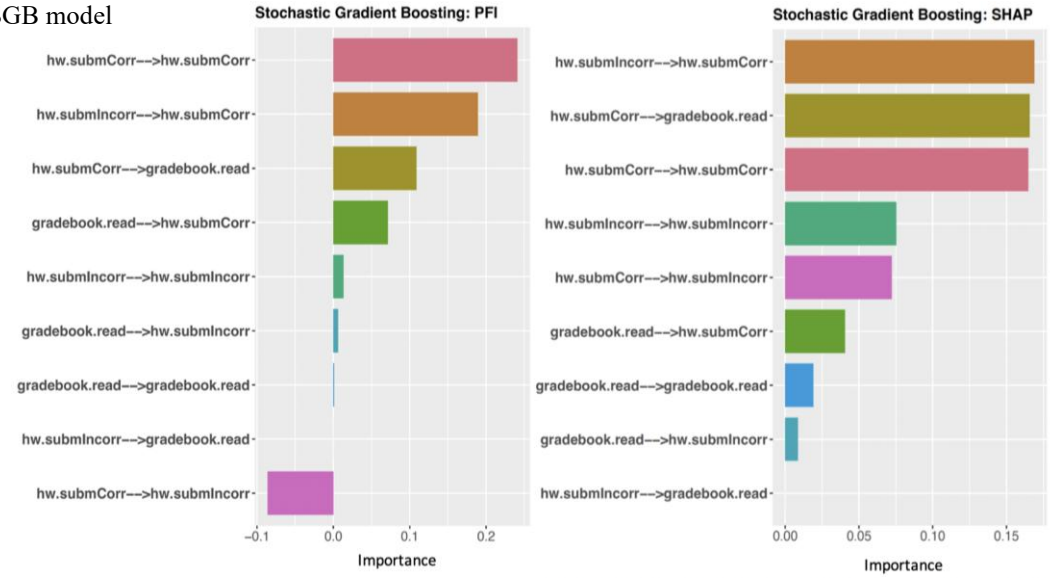
5.1.2. Influential Learning Dynamics

Explainable AI analyses provided distinct yet complementary insights, identifying features that enhance model performance through PFI and those that significantly contribute to predicted outcomes through SHAP. Figure 6 presents PFI and SHAP rankings from the top-performing seqMAC models for two samples, with bars of the same colour representing identical behavioural transitions. Notably, across most samples, four self- and cross-transitions related to correct ("hw.submCorr") and incorrect ("hw.submIncorr") homework submissions (hereinafter HW transitions) consistently ranked among the top 10 features by both methods (Supplementary Figure 3). This underscores their critical role in both classifier performance and outcome prediction. In the 20SP-2 SGB model (Figure 6A), both PFI and SHAP ranked "hw.submCorr → hw.submCorr" and "hw.submIncorr → hw.submCorr" among the top 3 features, indicating the strong impact of repeated correctness in homework submission and within-problem error correction on classification accuracy and predicted outcomes. Similarly, in the 19FA-2

SGB model, "hw.submIncorr → hw.submCorr" ranked highly by both methods. Beyond HW transitions, in the 20SP-2 SGB model, both PFI and SHAP consistently identified "hw.submCorr → gradebook.read" as another influential transition, suggesting that monitoring performance after completing an assignment significantly plays a key role. In the 19FA-2 SGB model, transitions between homework submissions and study preparation activities like class outline downloads ("ClsOln.dwnld") and lesson page read ("lsnPg.read") were commonly recognized as important by both PFI and SHAP.

Meanwhile, discrepancies between PFI and SHAP also emerged for certain features. Here we describe a few notable examples. In the 20SP-2 SGB model (Figure 6A), SHAP assigned "hw.submCorr → hw.submIncorr" a relatively high rank (5th), indicating its moderate contribution to predicted outcomes, whereas PFI assigned it a negative value, suggesting, rather, it deteriorated model performance. In the 19FA-2 SGB model (Figure 6B), such discrepancies were even more pronounced, as seen in the wider spread of differing coloured bars and their varying ranks, indicating greater divergence between features identified as important by PFI versus SHAP. PFI ranked "clsOln.dwnld → clsOln.dwnld" and "lsnPg.read → clsOln.dwnld" as the most influential transitions, suggesting that batch downloading class outlines and accessing the lesson-related task page before downloading the class outline effectively improves classifier performance. The former likely reflects effective organizational and planning strategies, whereas the latter represents SRL tied to task definition and preparation. However, SHAP primarily ranked HW transitions as the top features and highlights others that PFI did not, such as "lsnPg.read → lsnPg.read," meaning repetitive access to the task page, and "clsOln.dwnld → lsnPg.read," the reverse transition of the most important PFI-ranked feature. These discrepancies highlight the complementary nature of PFI and SHAP, showing that different feature sets influence classifier performance and predicted outcomes.

A) 20SP-2 SGB model



B) 19FA-2 SGB model

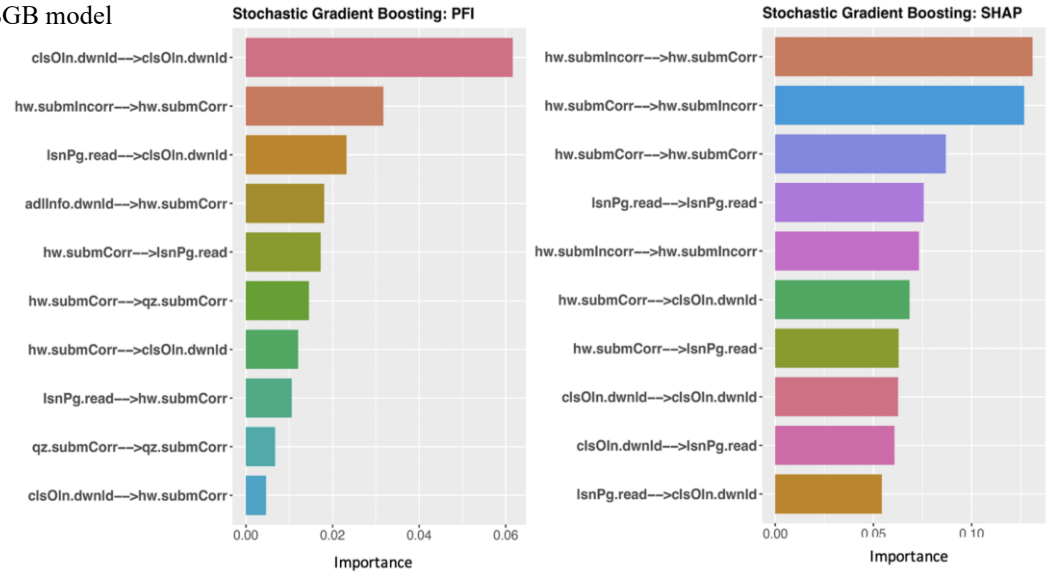


Figure 6. Feature importance ranking by PFI and SHAP.

5.1.3. Distinct Learning Patterns Between Low and High Performers

We used *t*-tests to compare the mean transition probabilities between groups H and L, examining significant differences in learning patterns. As an example, we present results on nine transition probabilities related to key learning behaviours in the 20SP-2 sample ("gradebook.read," "hw.submCorr," "hw.submIncorr"). To address multiple comparisons, a Bonferroni-corrected significance level ($\alpha = .05/9 \approx .0056$) was applied. Figure 7 visualizes the network of relative behaviour patterns between group H and L. Circles represent key behaviours. Solid lines indicate significant transitions; dashed lines indicate non-significant ones. Red arrows denote higher transition probabilities for group H, blue for group L. No path from "incorrect homework submission" to "gradebook reading" is shown due to zero variance. Notably, all HW transition probabilities exhibited statistically significant group differences. Group L showed significantly lower transitions to correct homework submissions from both incorrect (prob = 0.599) and correct (prob = 0.742) submissions compared to group H (prob = 0.695 and prob = 0.834). Conversely, Group L showed significantly higher transitions to incorrect homework submission from both incorrect (prob = 0.377) and correct (prob = 0.229) submissions compared to Group H (prob = 0.305 and prob = 0.163). This suggests that low-performing students had greater inertia toward incorrect homework submissions and weaker inertia toward correct homework submissions, whereas high-performing students exhibited the reverse trends. These patterns remained consistent across all other samples (Supplementary Table 2).

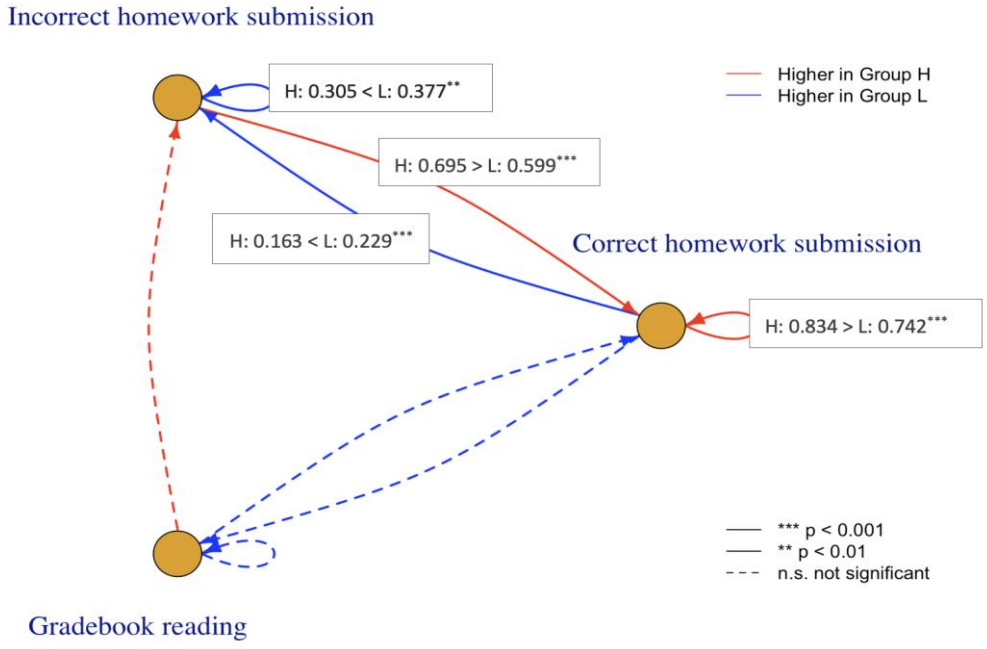


Figure 7. Contrasting learning patterns of high vs. low performers.

5.2. Generalizability Analysis

Our study concluded with a generalizability analysis of the seqMAC global classifier, trained on past cohort data to predict future student outcomes. In our sample-level analyses, we found SGB classifiers generally outperform others, with HW transitions as key features. Leveraging these two insights, we developed the global seqMAC classifier, using SGB as the model-building algorithm and HW transition probabilities as universal features. Specifically, SGB was trained on four HW transition probabilities from three previous 2019 samples (19SP-1, 19SP-2, 19FA-2), excluding 19FA-1 due to its lack of "hw.submCorr" in the key learning behaviours. SGB's performance was then independently validated on two future samples from spring 2020 (20SP-1, 20SP-2). The global seqMAC classifier demonstrated 84% sensitivity, 68% F2 score, and 62% balanced accuracy for 20SP-1, and 88% sensitivity, 77% F2 score, and 66% balanced accuracy for 20SP-2. These results indicate that the global classifier effectively identified 84–88% of low-performing students in future samples. The gap between sensitivity and balanced accuracy suggests the global classifier is more effective at identifying low-performing students than high-performing ones. While this marks an initial step toward a universally applicable classifier across academic terms, seqMAC showed promise in building generalizable classifiers to predict future learner outcomes by leveraging historical data and insights gained from individual sample analyses. Figure 8 summarizes seqMAC's performance, presenting the top local classifiers for each sample as well as the global classifier.

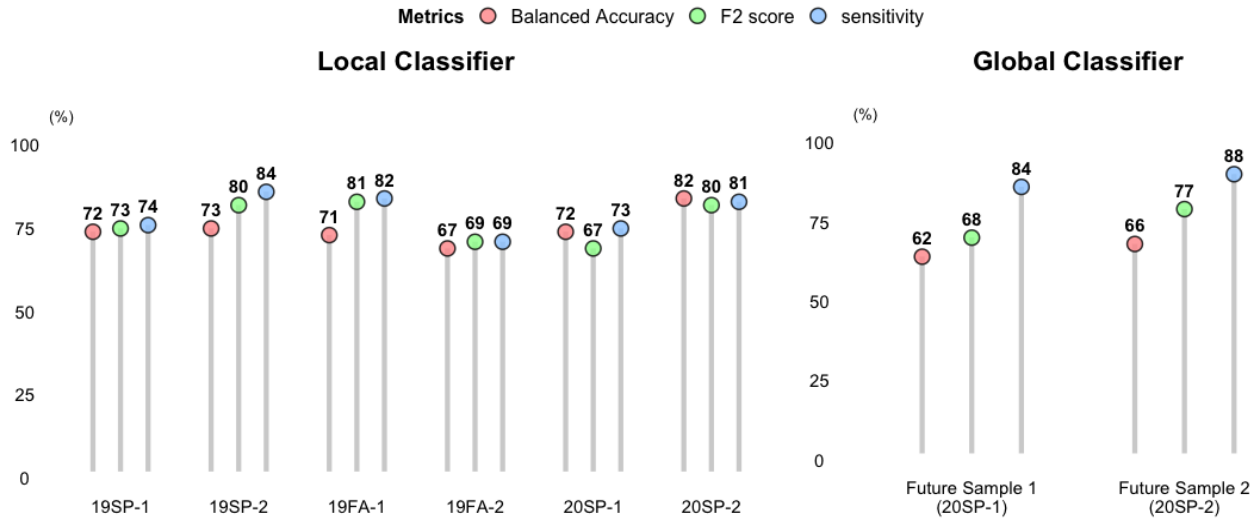


Figure 8. seqMAC summary performance.

6. Discussion

6.1. Practical Implications and Considerations

6.1.1. Advancing Current LMS Predictive Analytics

With online learning now firmly established as essential (Greenhow et al., 2022), there is a pressing demand for appropriate analytic tools to handle LMS clickstream data (Bernacki, 2018). Specifically, proactively identifying at-risk learners stands as a critical aspect of LMS predictive tools (Macfadyen & Dawson, 2010; Riestra-González et al., 2021). To address this, we proposed sequence-based Markov machine learning classification (seqMAC), a hybrid framework designed to predict end-of-course performance from early LMS clickstreams. By integrating Markov chain models (MCMs) with supervised machine learning (ML), seqMAC leverages the strengths of both approaches while overcoming their limitations. seqMAC was specifically developed to address a major limitation of the Markov benchmark method, which becomes non-functional when faced with high inter-sequence heterogeneity, especially when the test data includes behaviours unseen in the training data (i.e., sparse behaviours). Our analysis demonstrated that seqMAC outperformed the benchmark approach in applicability, successfully generating predictions for all samples, whereas the benchmark failed in 5 out of 6 samples. In the one sample where both methods were functional, the best-performing seqMAC model outperformed the benchmark across all evaluated metrics (Figure 5). Notably, seqMAC maintained robust performance across various cutoff scores (Supplementary Figure 2). seqMAC consistently generated predictions across all samples, whereas the benchmark failed to produce predictions for most or all samples. When both methods were functional, the top seqMAC models outperformed the benchmark. This adaptability of seqMAC is critical given the diverse nature of individual learning processes (Caeiro-Rodríguez et al., 2005; Rizvi et al., 2019). In addition to its adaptability, seqMAC extends the analytical capabilities of count-based ML approaches by moving beyond their limited temporal focus, which has been prevalent in LMS predictive analytics (e.g., Arizmendi et al., 2023; Bernacki, Chavez, & Uesbeck, 2020; F. Chen & Cui, 2020; Tsiakmaki et al., 2020). In the first phase, seqMAC generates a dynamic feature matrix in cross-sectional form, preserving the temporal dynamics of LMS sequences through transition probabilities between learning behaviours. This design allows the wide range of available ML algorithms — typically suited for cross-sectional data — to remain applicable, while still capturing evolving learning dynamics for predictive modelling by ML.

Another significant implication of seqMAC is its ability to enhance model interpretability. The demand of interpretable models is growing in LMS predictive analytics, where trust in results is essential for informed interventions (Kay et al., 2023; Larrabee Sønderlund et al., 2019). Although explainable AI is increasingly used in this field, it has primarily focused on static features, such as learner demographics and click event counts, which have been the primary input for count-based ML approaches. Deep learning models for sequential data, such as long short-term memory networks and transformers, offer strong predictive capabilities but often lack interpretability. When paired with explainable AI techniques like TimeSHAP (Bento et al., 2021), they enable timestamp-based interpretations, measuring feature importance at specific time points or intervals. However, they do not effectively capture broader behavioural transitions in a human-interpretable way, limiting their ability to provide insights into how learners progress through sequences of actions over time. In contrast, seqMAC offers transition-based interpretations by tracking behavioural shifts over time with an emphasis on key learning transitions. This capability uniquely positions seqMAC to bridge the gap between traditional ML approaches, which rely on static aggregated features,

and deep learning approaches, which provide only timepoint- or interval-based interpretations. Since seqMAC is designed for both sequence classification and transition-based interpretation, neither traditional ML nor deep learning approaches serve as meaningful baselines. Given that Markov models naturally provide transition-based interpretation by comparing classified group learning dynamics — similar to how seqMAC performs post-hoc group comparisons — the Markov classification method serves as the most appropriate baseline. Although higher-order or multi-layered variations of Markov models could serve as alternative baselines, their fundamental approach to sequence classification remains the same as the models used here, and a discussion of these methodological variations falls outside the scope of this paper.

Beyond model interpretability, explainable AI analysis with seqMAC may also provide insights into sample characteristics. Our study revealed varied predictive features across samples, indicating inter-sample heterogeneity. Methodologically, this variation likely originated from preprocessing, where logistic lasso regression was applied before seqMAC modelling to address the high dimensionality of the dynamic feature matrix. This process identified distinct sets of key behaviours per sample, leading to variability in predictive features. However, what fundamentally caused this heterogeneity remains an open question. Validation, a key process in learning analytics (Winne, 2020), requires multiple data sources like multimodal learning analytics, to support inferences. Interpretations should extend beyond a single source, considering how learning context, such as participants and external conditions, affect engagement. Conducting such an investigation, however, falls beyond the scope of the current study. With LMS clickstreams as our sole source, we cautiously infer potential sources of heterogeneous predictive features from instructional design intentions, though these insights remain speculative and should be seen as an example of what explainable AI can reveal about samples within the seqMAC framework. Our preliminary examination suggests that the predictive strength varies across samples depending on transition category. Assessment-related transitions consistently showed strong predictive validity (e.g., all instructors, particularly Instructor B), whereas SRL-related transitions were predictive only in certain cases (e.g., particularly Instructors A and C). These differences may reflect variations in how instructors scaffolded SRL and adapted to instructional contexts, such as the shift to emergency online learning due to COVID-19 or differing class schedules. Researchers interpreting AI findings, beyond the goals of demonstrating this new technique, should incorporate observational methods and direct interviews with educators and students to triangulate well-supported justifications.

Post-hoc findings from seqMAC carry practical implications for educational interventions. Leveraging its interpretability, seqMAC can reveal critical transitions in learning processes that are universally important within a course. Targeting these transitions may help researchers identify at-risk learners early and design effective interventions that reinforce those essential shifts, ultimately improving learning outcomes. For example, seqMAC identified that at-risk students tended to exhibit fewer transitions from incorrect to correct homework submissions. If a student's transition probability falls below a reference point, such as the historical average for high-performing students, seqMAC would flag this as a critical issue, enabling proactive interventions. Researchers have tested various intervention strategies to support struggling STEM learners (e.g., Bernacki, Chavez, & Uesbeck, 2020; Theobald, 2021; Zepeda et al., 2015), with many focused on enhancing SRL skills (Greene et al., 2024). These skills may help students recognize low homework accuracy (Janssen & Lazonder, 2024) and apply better strategies to improve it (McDaniel & Einstein, 2020). Interventions providing more direct support for domain- or course-specific learning (e.g., Aleknavičiūtė et al., 2023) would also be possible.

To date, only a handful of these interventions have been deployed using predictive models (e.g., Bernacki, Vosicka, & Utz, 2020; Cogliano et al., 2022). The accuracy of these predictive models is paramount in predict-and-intervene initiatives because misclassifications, such as misclassifying an at-risk learner as not at-risk or incorrectly identifying a learner as at-risk when they are not, can lead to compounded problems for learners. For instance, misclassified learners may miss critical opportunities for improvement due to the absence of necessary interventions or be needlessly distracted from their focal learning tasks by unnecessary interventions. Further, when algorithms are predictive but not interpretable, they offer limited guidance on how to support learners effectively. Therefore, the current study focuses on methodological innovations to improve the versatility of predictive models without being constrained by heterogeneous LMS clickstreams while enhancing prediction accuracy and interpretability, before turning attention to intervention design and deployment.

6.1.2. Methodological Considerations

In applying seqMAC, researchers may encounter four key methodological considerations: 1) analysis window, 2) algorithm selection, 3) performance cutoff, and 4) model evaluation metrics. First, for the analysis window, it is recommended to choose one that captures at least one complete learning cycle. The ideal endpoint will depend on factors like course type (e.g., college or online learning) and duration (e.g., regular semester or summer session). For instance, in a typical academic setting, a complete learning cycle could be from the start of a lesson to the completion of a major assessment like assignments or quizzes, where learners demonstrate understanding of the material covered during that period. It would also be helpful to consider when to identify underperformers — whether before midterms or finals. Our analysis window covered about the first four weeks, up to Lesson 3, with the ultimate goal of offering early interventions before midterms in future applications.

Second, for algorithm selection, it can be beneficial to test a diverse set of algorithms with different characteristics (e.g., linear/non-linear models, tree-based models/neural networks) to evaluate performance under various data conditions. Researchers can then focus on the algorithms that consistently perform well within their specific research context. Third, for the performance cutoff, selecting the most appropriate point for dividing learners into high and low performers is crucial, as it directly impacts the early identification of at-risk learners. Setting the cutoff too high may flag more students than necessary, whereas setting it too low could delay interventions for those who need help. The cutoff may vary depending on the course-specific grading system and learning objectives. In our empirical study, where the course had an absolute passing grade of 80, we used this as the authentic performance cutoff, classifying students scoring below this threshold as low performers. In contrast, courses with more lenient grading might use a lower cutoff, such as 60. For courses without a fixed passing grade, where grades are based on relative performance among learners, researchers might experiment with different cutoff points in pilot studies. Using an adaptive approach, they could adjust and update the cutoff based on early results, ongoing performance data, and instructor feedback, refining it for accurate assessments and timely interventions.

Lastly, for model evaluation metrics, although high sensitivity is important to avoid missing at-risk learners, we also recommend prioritizing summary metrics, such as balanced accuracy and area under the curve (AUC), to ensure that the model remains unbiased and generalizable across different academic terms. Focusing solely on sensitivity can result in over-alerting high-performing learners, diverting their attention from areas that genuinely need improvement and causing unnecessary interventions that waste time and limited resources. Balanced accuracy is particularly useful in cases where there is an uneven class distribution. AUC could be another useful summary metric, though it was not used in this study due to the nature of the Markov benchmark method, which classifies individuals based on comparisons at the individual level — assessing their likelihoods of belonging to the high- or low-performing Markov model — rather than using a universal threshold for all individuals. For these reasons, we prioritized balanced accuracy when selecting the top-performing model for each sample.

6.2. Limitations and Future Research Directions

Notwithstanding the benefits, the current seqMAC framework has limitations. First, we did not apply deep learning algorithms like long short-term memory networks or transformers, which are typically used for sequential data to preserve temporal structures and dependencies. Instead, we chose deep feedforward neural networks, better suited for the cross-sectional feature matrix generated in Phase 1 of seqMAC. seqMAC abstracts temporal learning dynamics as transition probabilities for each learner, with each row in the matrix serving as features for ML modelling in Phase 2. Given this cross-sectional format, LSTMs and transformers — designed for time-ordered sequences — were less suitable and not directly comparable to the algorithms chosen for this study. Second, seqMAC relies on the memoryless Markov property of MCMs to capture behavioural transitions between adjacent observed events. To incorporate longer memory into modelling, higher-order models could be considered (Raftery, 1985), but determining the appropriate order and interpreting the results may become increasingly complex. Third, seqMAC focuses exclusively on observed events, thereby precluding any inference from hidden states. We willingly accepted this trade-off in order to avoid potential measurement non-invariance in transition probabilities associated with hidden states and the resulting peril of invalid predictions. Fourth, seqMAC does not currently address outliers and class imbalances. Some LMS samples used in this study contained outliers, such as extremely long or short sequences, which might have compromised prediction accuracy for those samples. Exploring and incorporating outlier detection and resampling techniques tailored to sequential data (e.g., Aggarwal, 2017; Gan et al., 2023; Helske & Helske, 2019) may help mitigate these issues.

A major future step for enhancing seqMAC is developing a thresholding system that leverages group-averaged transition probabilities to predict learner outcomes. As discussed in Section 6.1.1, historically averaged transition probabilities for key behaviours among high-performing learners could serve as an initial guide for detecting when a learner's behaviour deviates from the expected norm. More robust methods like Youden's index and the diagnostic odds ratio (Hajian-Tilaki, 2018) could refine this thresholding system by balancing sensitivity (correctly identifying those at-risk) and specificity (correctly identifying those not at-risk). With more data collected, thresholds could be adaptively updated to reflect evolving patterns, improving accuracy and reliability while keeping the system flexible to changing learning behaviours. Ideally, the adaptive thresholding system could support a two-stage real-time forecasting approach: first categorizing learners with the seqMAC global classifier, followed by affirming or adjusting classifications using the system to improve prediction accuracy.

Declaration of Conflicting Interest

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding Acknowledgments

This research was supported by the U.S. National Science Foundation Award DUE-1821594. The opinions, findings, and conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Adnan, M., Uddin, M. I., Khan, E., Alharithi, F. S., Amin, S., & Alzahrani, A. A. (2022). Earliest possible global and local interpretation of students' performance in virtual learning environment by leveraging explainable AI. *IEEE Access*, *10*, 129843–129864. https://doi.org/10.1109/ACCESS.2022.3227072

Afzaal, M., Nouri, J., Zia, A., Papapetrou, P., Fors, U., Wu, Y., Li, X., & Weegar, R. (2021). Explainable AI for data-driven feedback and intelligent action recommendations to support students self-regulation. *Frontiers in Artificial Intelligence*, *4*, Article 723447. https://doi.org/10.3389/frai.2021.723447

Aggarwal, C. C. (2017). *Outlier analysis* (2nd ed.). Springer Cham. https://doi.org/10.1007/978-3-319-47578-3_10

Alamri, A., Alshehri, M., Cristea, A., Pereira, F. D., Oliveira, E., Shi, L., & Stewart, C. (2019). Predicting MOOCs dropout using only two easily obtainable features from the first week's activities. In A. Coy, Y. Hayashi, & M. Chang (Eds.), *Intelligent tutoring systems: 15th international conference, ITS 2019, Kingston, Jamaica, June 3–7, 2019, proceedings* (pp. 163–173). Springer Cham. https://doi.org/10.1007/978-3-030-22244-4_20

Aleknavičiūtė, V., Lehtinen, E., & Södervik, I. (2023). Thirty years of conceptual change research in biology: A review and meta-analysis of intervention studies. *Educational Research Review*, *41*, Article 100556. https://doi.org/10.1016/j.edurev.2023.100556

Al-Sulami, A., Al-Masre, M., & Al-Malki, N. (2023). Deep learning to predict at-risk students' achievement in a preparatory-year English courses. In *2023 1st international conference on advanced innovations in smart cities (ICAISC)* (pp. 1–6). IEEE. https://doi.org/10.1109/ICAISC56366.2023.10085097

Arizmendi, C. J., Bernacki, M. L., Raković, M., Plumley, R. D., Urban, C. J., Panter, A. T., Greene, J. A., & Gates, K. M. (2023). Predicting student outcomes using digital logs of learning behaviors: Review, current standards, and suggestions for future work. *Behavior Research Methods*, *55*(6), 3026–3054. https://doi.org/10.3758/s13428-022-01939-9

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), Article 13. https://doi.org/10.1186/s41239-020-00187-1

Banerjee, I., de Sisternes, L., Hallak, J., Leng, T., Osborne, A., Durbin, M., & Rubin, D. (2019). *A deep-learning approach for prognosis of age-related macular degeneration disease using SD-OCT imaging biomarkers*. arXiv. https://doi.org/10.48550/arXiv.1902.10700

Bento, J., Saleiro, P., Cruz, A. F., Figueiredo, M. A. T., & Bizarro, P. (2021). TimeSHAP: Explaining recurrent models through sequence perturbations. In F. Zhu, B. C. Ooi, C. Miao, H. Wang, I. Skrypnyk, W. Hsu, & S. Chawla (Eds.), *KDD '21: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (pp. 2565–2573). ACM Press. https://doi.org/10.1145/3447548.3467166

Bernacki, M. L. (2018). Examining the cyclical, loosely sequenced, and contingent features of self-regulated learning: Trace data and their analysis. In D. H. Schunk & J. A. Greene (Eds.), *Handbook of self-regulated learning and performance* (2nd ed., pp. 370–387). Routledge.

Bernacki, M. L., Chavez, M. M., & Uesbeck, P. M. (2020). Predicting achievement and providing support before STEM majors begin to fail. *Computers & Education*, *158*, Article 103999. https://doi.org/10.1016/j.compedu.2020.103999

Bernacki, M. L., Vosicka, L., & Utz, J. C. (2020). Can a brief, digital skill training intervention help undergraduates "learn to learn" and improve their STEM achievement? *Journal of Educational Psychology*, *112*(4), 765–781. https://doi.org/10.1037/edu0000405

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32. https://doi.org/10.1023/A:1010933404324

Caeiro-Rodríguez, M., Anido-Rifón, L., & Llamas-Nistal, M. (2005). Improving the modelling of heterogeneous learning activities. In *Proceedings of the eighth IFIP world conference on computers in education.* https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=d58281c31156cd266c12cee3f5129783730b6044

Cardenas-Ovando, R. A., Noguez, J., & Rangel-Escareno, C. (2017). *RcppHMM: Rcpp hidden Markov model* (R version 1.2.2) [Computer software]. R Foundation for Statistical Computing. https://cran.r-project.org/web/packages/RcppHMM/

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

Chen, J., Fang, B., Zhang, H., & Xue, X. (2024). A systematic review for MOOC dropout prediction from the perspective of machine learning. *Interactive Learning Environments*, *32*(5), 1642–1655. https://doi.org/10.1080/10494820.2022.2124425

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*, Article 105584. https://doi.org/10.1016/j.chb.2018.06.032

Cogliano, M., Bernacki, M. L., Hilpert, J. C., & Strong, C. L. (2022). A self-regulated learning analytics prediction-and-intervention design: Detecting and supporting struggling biology students. *Journal of Educational Psychology 114*(8), 1801–1816. https://doi.org/10.1037/edu0000745

Eddy, S. R. (2004). What is a hidden Markov model? *Nature Biotechnology*, *22*, 1315–1316. https://doi.org/10.1038/nbt1004-1315

Elmäng, N. (2020). Sequence classification on gamified behavior data from a learning management system: Predicting student outcome using neural networks and Markov chain [Master's Thesis, University of Skövde]. DiVA Portal. https://urn.kb.se/resolve?urn=urn:nbn:se:his:diva-18654

Faisan, S., Thoraval, L., Armspach, J.-P., & Heitz, F. (2007). Hidden Markov multiple event sequence models: A paradigm for the spatio-temporal analysis of fMRI data. *Medical Image Analysis*, *11*(1), 1–20. https://doi.org/10.1016/j.media.2006.09.003

Fok, A. W. P., Wong, H. S., & Chen, Y. S. (2005). Hidden Markov model based characterization of content access patterns in an e-learning environment. In *2005 IEEE international conference on multimedia and expo* (pp. 201–204). IEEE. https://doi.org/10.1109/ICME.2005.1521395

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, *38*(4), 367–378. https://doi.org/10.1016/S0167-9473(01)00065-2

Friedman, J. H., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, *33*(1), 1–22. https://doi.org/10.18637/jss.v033.i01

Gan, W., Chen, L., Wan, S., Chen, J., & Chen, C.-M. (2023). Anomaly rule detection in sequence data. *IEEE Transactions on Knowledge and Data Engineering, 35*(12), 12095–12108. https://doi.org/10.1109/TKDE.2021.3139086

Geigle, C., & Zhai, C. (2017). Modeling student behavior with two-layer hidden Markov models. *Journal of Educational Data Mining, 9*(1), 1–24. https://doi.org/10.5281/zenodo.3554623

Greene, J. A., Bernacki, M. L., & Hadwin, A. F. (2024). Self-regulation. In P. A. Schutz & K. R. Muis (Eds.), *Handbook of educational psychology* (4th ed., pp. 314–334). Routledge. https://doi.org/10.4324/9780429433726-17

Greenhow, C., Graham, C. R., & Koehler, M. J. (2022). Foundations of online learning: Challenges and opportunities. *Educational Psychologist*, *57*(3), 131–147. https://doi.org/10.1080/00461520.2022.2090364

Greenwell, B. M. (2024). *Fastshap: Fast approximate Shapley values* (R package version 0.0.7) [Computer software]. R Foundation for Statistical Computing. https://CRAN.R-project.org/package=fastshap

Greenwell, B. M., & Boehmke, B. C. (2020). Variable Importance Plots: An introduction to the vip package. *The R Journal, 12*(1), 343–366. https://journal.r-project.org/archive/2020/RJ-2020-013/RJ-2020-013.pdf

Gupta, A., Garg, D., & Kumar, P. (2022). Mining sequential learning trajectories with hidden Markov models for early prediction of at-risk students in e-learning environments. *IEEE Transactions on Learning Technologies*, *15*(6), 783–797. https://doi.org/10.1109/TLT.2022.3197486

Hajian-Tilaki, K. (2018). The choice of methods in determining the optimal cut-off value for quantitative diagnostic test evaluation. *Statistical Methods in Medical Research, 27*(8), 2374–2383. https://doi.org/10.1177/0962280216680383

Hastie, T., Tibshirani, R., & Friedman, J. (2017). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer.

Heins, K., & Stern, H. (2014). A statistical model for event sequence data. *Proceedings of Machine Learning Research, 33,* 338–346. https://proceedings.mlr.press/v33/heins14.html

Helske, S., & Helske, J. (2019). Mixture hidden Markov models for sequence data: The seqHMM package in R. *Journal of Statistical Software, 88*(3), 1–32. https://doi.org/10.18637/jss.v088.i03

Ibe, O. C. (2013). *Markov processes for stochastic modeling* (2nd ed.). Elsevier.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning: With applications in R*. Springer.

Jang, Y., Choi, S., Jung, H., & Kim, H. (2022). Practical early prediction of students' performance using machine learning and eXplainable AI. *Education and Information Technologies*, *27*(9), 12855–12889. https://doi.org/10.1007/s10639-022-11120-6

Janssen, N., & Lazonder, A. W. (2024). Meta-analysis of interventions for monitoring accuracy in problem solving. *Educational Psychology Review*, *36*(3), Article 96. https://doi.org/10.1007/s10648-024-09936-4

Kalinowski, T., Falbel, D., Allaire, J. J., Chollet, F., RStudio, Google, Tang, Y., Van Der Bijl, W., Studer, M., & Keydana, S. (2024). *R interface to "Keras"* (Version 2.15.0) [Computer software]. R Foundation for Statistical Computing. https://cran.r-project.org/web/packages/keras/keras.pdf

Kay, J., Kummerfeld, B., Conati, C., Porayska-Pomsta, K., & Holstein, K. (2023). Scrutable AIED. In B. du Boulay, A. Mitrovic, & K. Yacef (Eds.), *Handbook of artificial intelligence in education* (pp. 101–126). Edward Elgar.

Kokoç, M., Akçapınar, G., & Hasnine, M. N. (2021). Unfolding students' online assignment submission behavioral patterns using temporal learning analytics. *Educational Technology & Society*, *24*(1), 223–235. https://www.jstor.org/stable/26977869

Kuhn, M. (2008). Building predictive models in R using the caret package. *Journal of Statistical Software*, *28*(5), 1–26. https://doi.org/10.18637/jss.v028.i05

Larrabee Sønderlund, A., Hughes, E., & Smith, J. (2019). The efficacy of learning analytics interventions in higher education: A systematic review. *British Journal of Educational Technology*, *50*(5), 2594–2618. https://doi.org/10.1111/bjet.12720

Lee, C., Gates, K. M., Chun, J., Al Kontar, R., Kamali, M., McInnis, M. G., & Deldin, P. (2025). Suicide risk estimation in bipolar disorder using N200 and P300 event-related potentials and machine learning: A pilot study. *Journal of Affective Disorders Reports*, *20*, Article 100875. https://doi.org/10.1016/j.jadr.2025.100875

Lee, C.-A., Tzeng, J.-W., Huang, N.-F., & Su, Y.-S. (2021). Prediction of student performance in massive open online courses using deep learning system based on learning behaviors. *Educational Technology & Society*, *24*(3), 130–146. https://www.jstor.org/stable/27032861

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, *2*, 56–67. https://doi.org/10.1038/s42256-019-0138-9

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

McDaniel, M. A., & Einstein, G. O. (2020). Training learning strategies to promote self-regulation and transfer: The knowledge, belief, commitment, and planning framework. *Perspectives on Psychological Science*, *15*(6), 1363–1381. https://doi.org/10.1177/1745691620920723

Nanavaty, S., & Khuteta, A. (2024). A deep learning dive into online learning: Predicting student success with interaction-based neural networks. *International Journal of Intelligent Systems and Applications in Engineering*, *12*(1), 102–107. https://www.ijisae.org/index.php/IJISAE/article/view/3769

Putnick, D. L., & Bornstein, M. H. (2016). Measurement invariance conventions and reporting: The state of the art and future directions for psychological research. *Developmental Review*, *41*, 71–90. https://doi.org/10.1016/j.dr.2016.06.004

Rabiner, L., & Juang, B. (1986). An introduction to hidden Markov models. *IEEE ASSP Magazine*, *3*(1), 4–16. https://doi.org/10.1109/MASSP.1986.1165342

Raftery, A. E. (1985). A model for high-order Markov chains. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *47*(3), 528–539. https://doi.org/10.1111/j.2517-6161.1985.tb01383.x

R Core Team. (2022). *R: A language and environment for statistical computing* (R version 4.2.2) [Computer software]. R Foundation for Statistical Computing. https://www.R-project.org/

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In B. Krishnapuram, M. Shah, A. Smola, C. Aggarwal, D. Shen, & R. Rastogi (Eds.), *KDD '16: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135–1144). https://doi.org/10.1145/2939672.2939778

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

Rizvi, S., Rienties, B., & Khoja, S. A. (2019). The role of demographics in online learning; A decision tree based approach. *Computers & Education*, *137*, 32–47. https://doi.org/10.1016/j.compedu.2019.04.001

Rudin, C., & Radin, J. (2019). Why are we using black box models in AI when we don't need to? A lesson from an explainable AI competition. *Harvard Data Science Review*, *1*(2). https://doi.org/10.1162/99608f92.5a8a3a3d

Shapley, L. S. (1953). A value for n-person games. In H. W. Kuhn & A. W. Tucker (Eds.), *Contributions to the theory of games* (Vol. II, pp. 307–318). Princeton University Press. https://doi.org/10.1515/9781400881970-018

Staufer, S., Bugert, F., Hauser, F., Grabinger, L., Ezer, T., Nadimpalli, V. K., Bittner, D., Röhrl, S., & Mottok, J. (2024). Tyche algorithm: Markov models for generating learning paths in learning management systems. In L. Gómez Chova, C. González Martínez, & J. Lees (Eds.), *INTED2024 proceedings: 18th international technology, education and development conference* (pp. 4195–4205). IATED Academy. https://doi.org/10.21125/inted.2024.1080

Sun, D., Cheng, G., Xu, P., Zheng, Q., & Chen, L. (2019). Using HMM to compare interaction activity patterns of student groups with different achievements in MPOCs. *Interactive Learning Environments*, *27*(5-6), 766–781. https://doi.org/10.1080/10494820.2019.1610780

Tamada, M. M., Giusti, R., & de Magalhães Netto, J. F. (2022). Predicting students at risk of dropout in technical course using LMS logs. *Electronics*, *11*(3), Article 468. https://doi.org/10.3390/electronics11030468

Tang, Y., Li, Z., Wang, G., & Hu, X. (2023). Modeling learning behaviors and predicting performance in an intelligent tutoring system: A two-layer hidden Markov modeling approach. *Interactive Learning Environments*, *31*(9), 5495–5507. https://doi.org/10.1080/10494820.2021.2010100

Theobald, M. (2021). Self-regulated learning training programs enhance university students' academic performance, self-regulated learning strategies, and motivation: A meta-analysis. *Contemporary Educational Psychology*, *66*, Article 101976. https://doi.org/10.1016/j.cedpsych.2021.101976

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Tran, T. M., & Hasegawa, S. (2022). Using Markov chain on online learning history data to develop learner model for measuring strength of learning habits. In D. G. Sampson, D., Ifenthaler, & P. Isaías (Eds.), *Proceedings of the international conference on cognition and exploratory learning in the digital age.* International Association for Development of the Information Society. https://eric.ed.gov/?id=ED626882

Tsiakmaki, M., Kostopoulos, G., Kotsiantis, S., & Ragos, O. (2020). Transfer learning from deep neural networks for predicting student performance. *Applied Sciences*, *10*(6), Article 2145. https://doi.org/10.3390/app10062145

Van Goidsenhoven, S., Bogdanova, D., Deeva, G., Broucke, S. v., De Weerdt, J., & Snoeck, M. (2020). Predicting student success in a blended learning environment. In C. Rensing, H. Drachsler, V. Kovanović, N. Pinkwart, M. Scheffel, & K. Verbert (Eds.), *LAK '20: Proceedings of the tenth international conference on learning analytics & knowledge* (pp. 17–25). ACM Press. https://doi.org/10.1145/3375462.3375494

Weaver, D., Spratt, C., & Nair, C. S. (2008). Academic and student use of a learning management system: Implications for quality. *Australasian Journal of Educational Technology*, *24*(1). https://doi.org/10.14742/ajet.1228

Wen, X., & Juan, H. (2023). Early prediction of students' performance using a deep neural network based on online learning activity sequence. *Applied Sciences*, *13*(15), Article 8933. https://doi.org/10.3390/app13158933

Winne, P. H. (2020). Construct and consequential validity for learning analytics based on trace data. *Computers in Human Behavior*, *112*, Article 106457. https://doi.org/10.1016/j.chb.2020.106457

Witteveen, D., & Attewell, P. (2017). The college completion puzzle: A hidden Markov model approach. *Research in Higher Education*, *58*(4), 449–467. https://doi.org/10.1007/s11162-016-9430-2

Zepeda, C. D., Richey, J. E., Ronevich, P., & Nokes-Malach, T. J. (2015). Direct instruction of metacognition benefits adolescent science learning, transfer, and motivation: An in vivo study. *Journal of Educational Psychology*, *107*(4), 954–970. https://doi.org/10.1037/edu0000022

Supplementary Table 1. Optimized hyperparameters of ML algorithms adopted in the seqMAC framework

Algorithms	Parameters	Grid	19SP-1	19SP-2	19FA-1	19FA-2	20SP-1	20SP-2
RF	mtry	seq(1,sqrt(number of features)+2, 1)	4	11	4	12	20	5
SGB	n.trees	seq(10,100,10)	50	90	30	100	50	90
	interaction.depth	seq(5,30,5)	15	5	10	15	10	10
	shrinkage	seq(0.1,0.3,len = 20)	.237	.268	.237	.247	.132	.195
	n.minobsinnode	seq(1, 11, 2)	5	1	7	1	9	1
DNN	size	c(16, 32, 64)	32	32	16	32	16	32
	lambda	c(0.1,1)	0.1	0.1	1	0.1	0.1	0.1
	batch_size	c(128, 256)	256	256	128	128	256	256
	decay	c(0.1, 0.5)	0.5	0.5	0.5	0.5	0.5	0.1
	learning rate	0.1						
	activation	ReLu						
	epoch	30						
	optimizer	Adam						
LSVM	cost	c(.001, .01, .05, .1, .5, 5,10, 50,100,1000)	50	5	.01	.001	.1	50
RSVM	cost	c(10^-4,10^-3, 10^-2, 10^-1, 1, 5, 10, 10^2, 10^3, 10^4)	100	5	1000	10	1	1000
	sigma	c(.001, .005, .01, .05, .1,.5, .9)	.01	.5	.9	.05	.01	.05

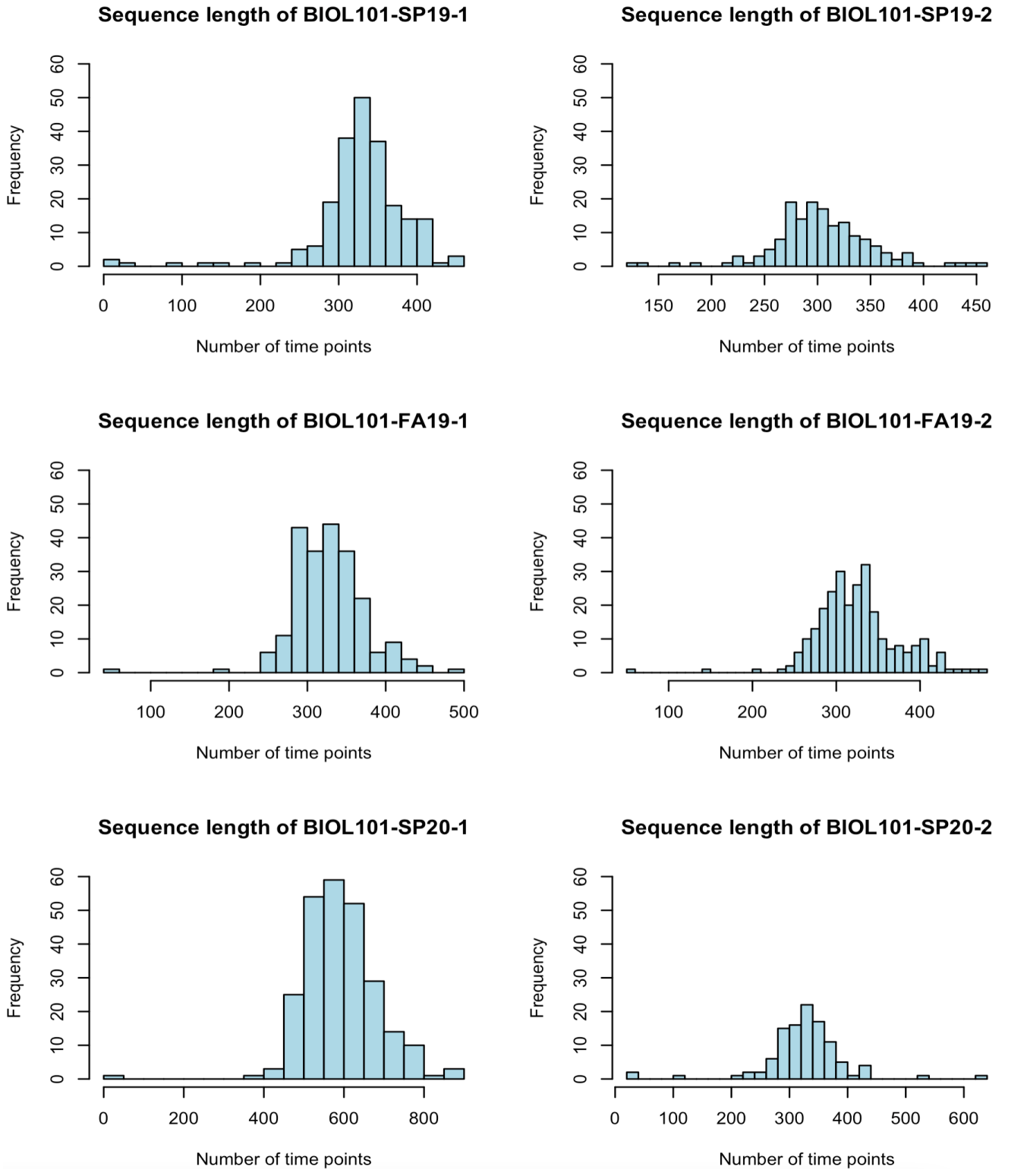
Note. During hyperparameter optimization for SGB and DNN, the large number of hyperparameters and their extensive ranges presented challenges such as prolonged training times and the complexity of identifying optimal combinations. To efficiently explore the search space, we initially employed a random search strategy to identify promising regions. Once these regions were identified, we conducted a grid search to precisely tune the hyperparameter values. Specifically, regarding the DNN learning rate, we observed that excessively low values significantly extended training time. Consequently, we began with a learning rate of 0.001 and progressively increased it to 0.01 and 0.1. Our experiments demonstrated that a learning rate of 0.1 generally outperformed the other tested values, leading us to adopt 0.1 while optimizing the remaining hyperparameters. For the DNN activation function and optimizer, we chose ReLU and Adam, as they are the default options in Keras and are widely recognized for their ability to enhance model performance and training efficiency. The number of epochs in DNN modeling was set to 30 to ensure sufficient training while minimizing the risk of overfitting, as we did not observe significant performance improvements beyond this point. Additionally, we set conservative values for the hidden layer sizes to prevent overfitting, as our samples had fewer observations than typically required for deep learning. This was to avoid overly complex models given the smaller sample sizes.

Supplementary Table 2. Comparison of behavioral dynamics between groups for homework submission

Behavioral transition	sample	signif.	*p*-values	mean prob. group L	mean prob. group H	group mean comparison
hw.submCorr→ hw.submCorr	19SP-1	****	4E-06	.800	.843	L < H
	19SP-2	****	1E-06	.778	.837	L < H
	19FA-1	-	-	-	-	-
	19FA-2	***	.0004	.770	.812	L < H
	20SP-1	*	.0171	.856	.888	L < H
	20SP-2	***	.0007	.742	.834	L < H
hw.submCorr→ hw.submIncorr	19SP-1	****	2E-06	.183	.138	L > H
	19SP-2	****	1E-06	.202	.144	L > H
	19FA-1	-	-	-	-	-
	19FA-2	****	2E-06	.201	.153	L > H
	20SP-1	****	3E-06	.101	.070	L > H
	20SP-2	***	.0009	.229	.163	L > H
hw.submIncorr→ hw.submCorr	19SP-1	****	1.6E-05	.688	.755	L < H
	19SP-2	****	2.1E-05	.628	.696	L < H
	19FA-1	-	-	-	-	-
	19FA-2	**	.0020	.658	.706	L < H
	20SP-1	****	2E-06	.350	.404	L < H
	20SP-2	***	.0002	.599	.695	L < H
hw.submIncorr→ hw.submIncorr	19SP-1	****	1.4E-05	.310	.243	L > H
	19SP-2	****	2.7E-05	.366	.301	L > H
	19FA-1	*	.0121	.963	.948	L > H
	19FA-2	***	.0003	.336	.283	L > H
	20SP-1	***	.0001	.626	.581	L > H
	20SP-2	**	.0013	.377	.305	L > H

*Notes. 1. ****($p < 1e\text{-}04$), *** ($p < .001$), ** ($p < .01$), * ($p < .05$); 2. The 19FA-1 sample lacked 'hw.submCorr' in its key learning behaviours, so no t-test was performed on related transitions.*

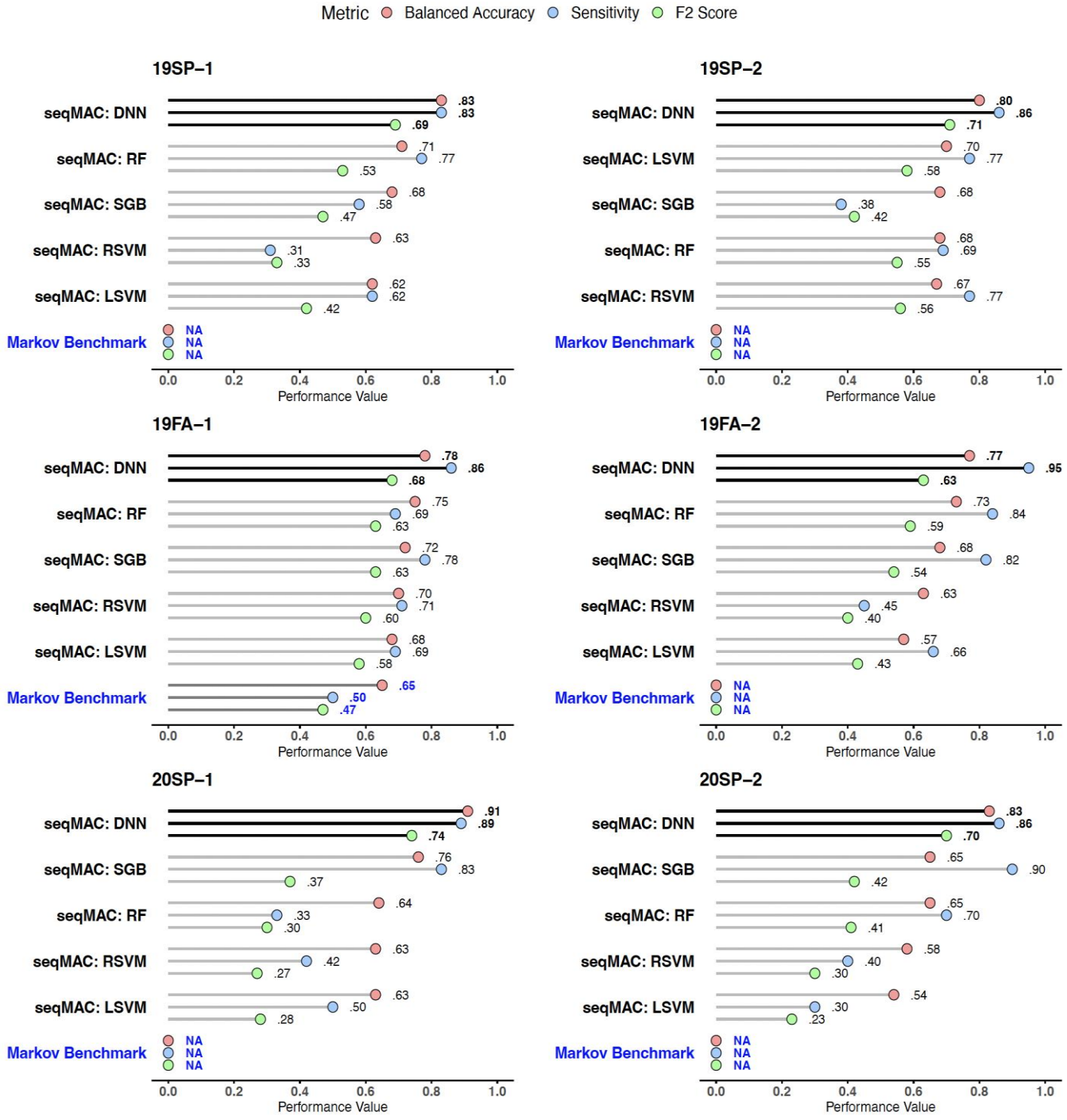
Supplementary Figure 1. Distributions of sequence length per sample



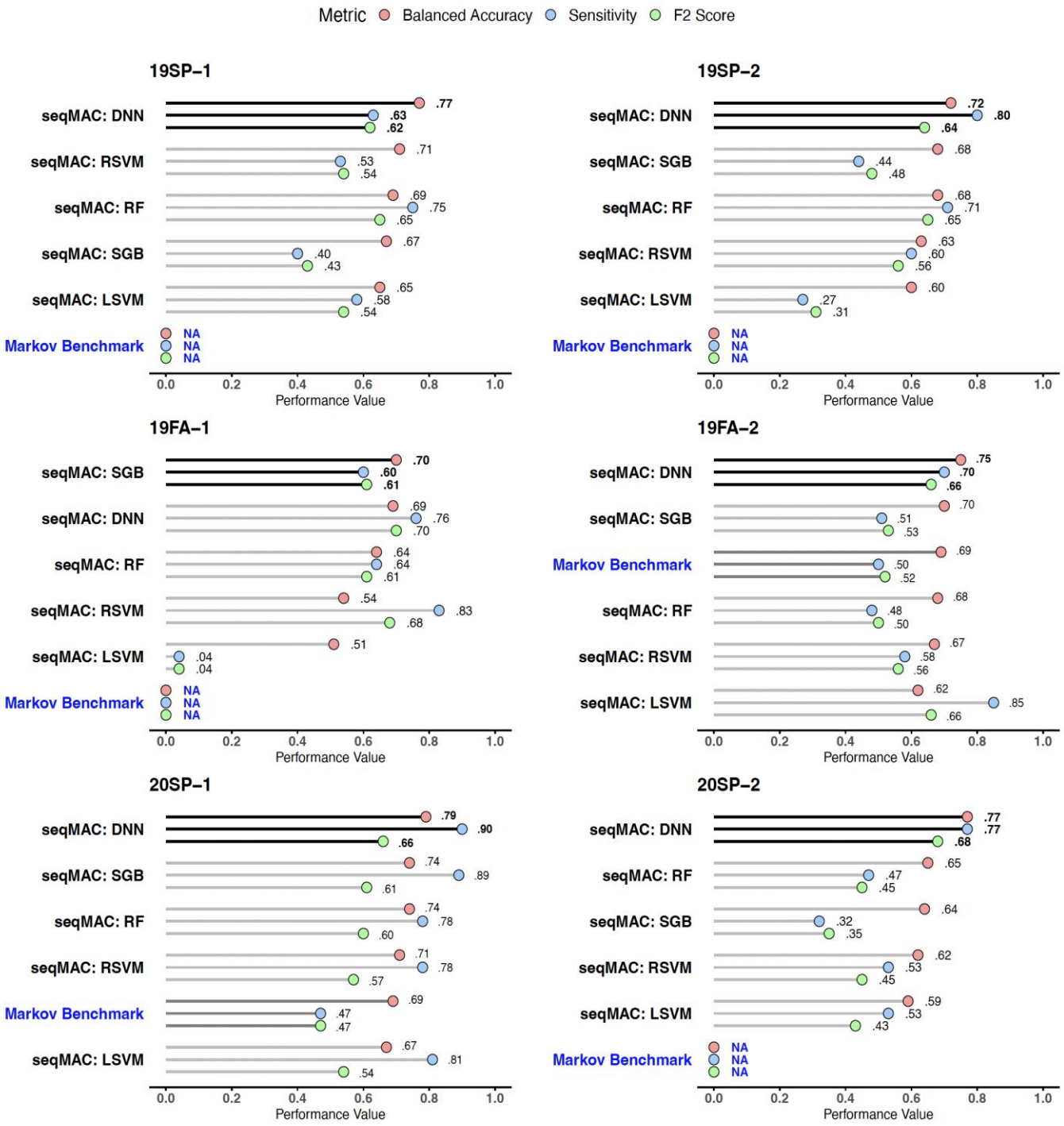
Note. The histograms show the diverse shapes of sequence length distributions across the samples. The x-axis indicates the length of an individual sequence. The y-axis means the count of sequences associated with the sequence length on the x-axis.

Supplementary Figure 2. Comparing prediction performance with different cutoffs

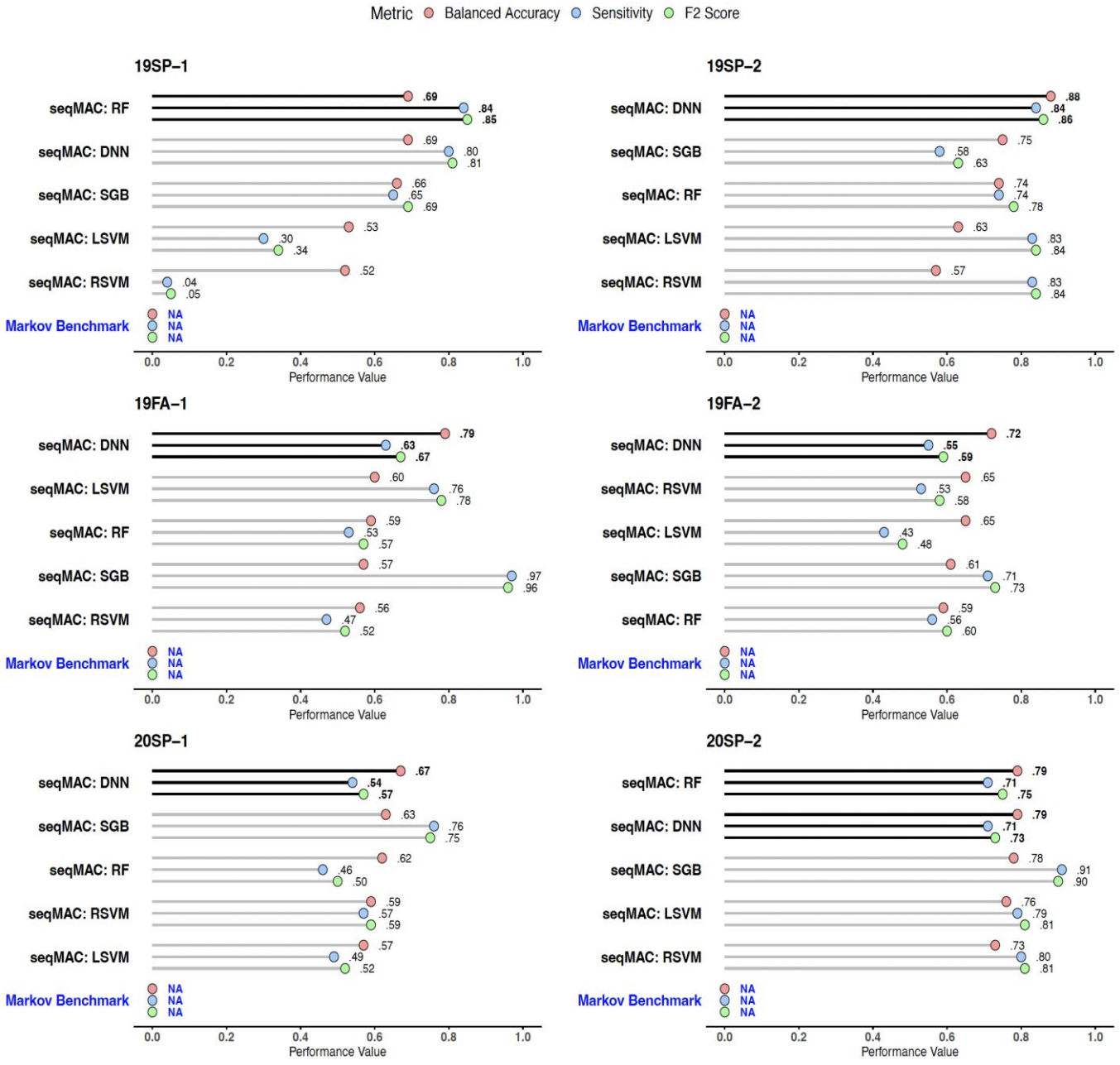
(1) Cutoff = score 60



(2) Cutoff = score 70



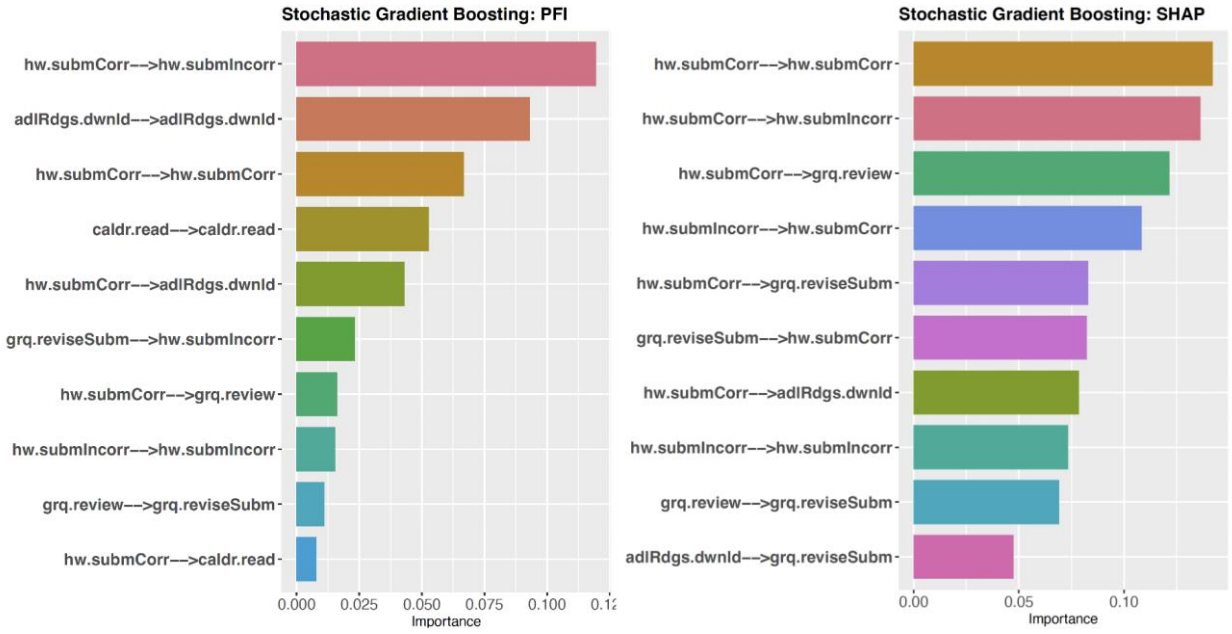
(3) Cutoff = score 90



Supplementary Figure 3. Top 10 important features of the sample representative classifiers

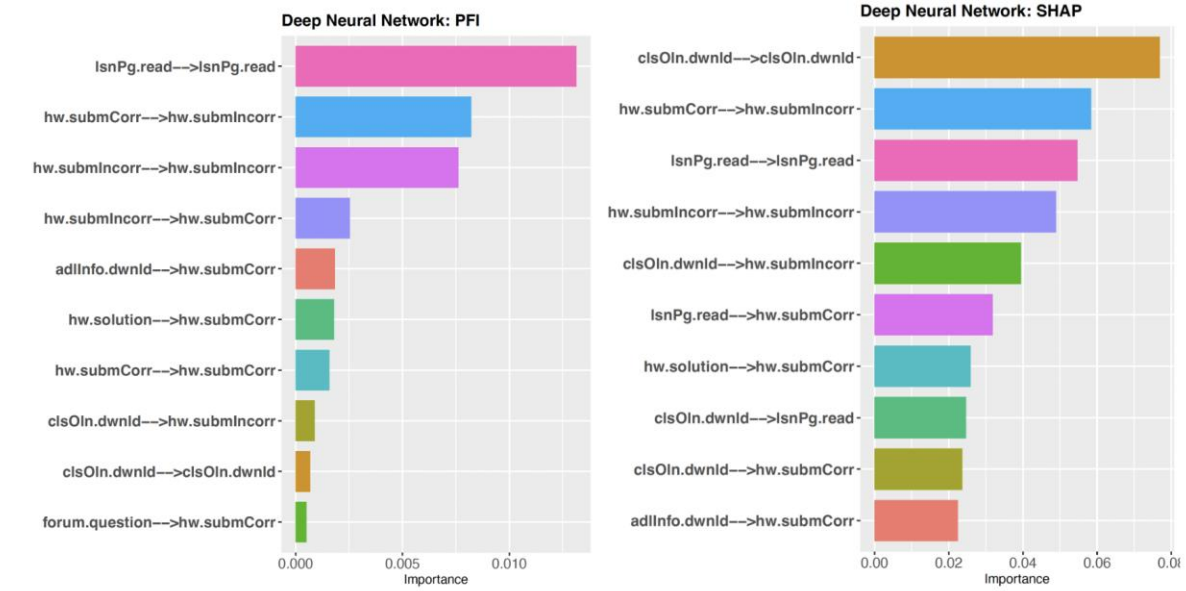
Sample 1: 19SP-1

Stochastic gradient boosting (74% sensitivity, 73% F2 score, 72% balanced accuracy)



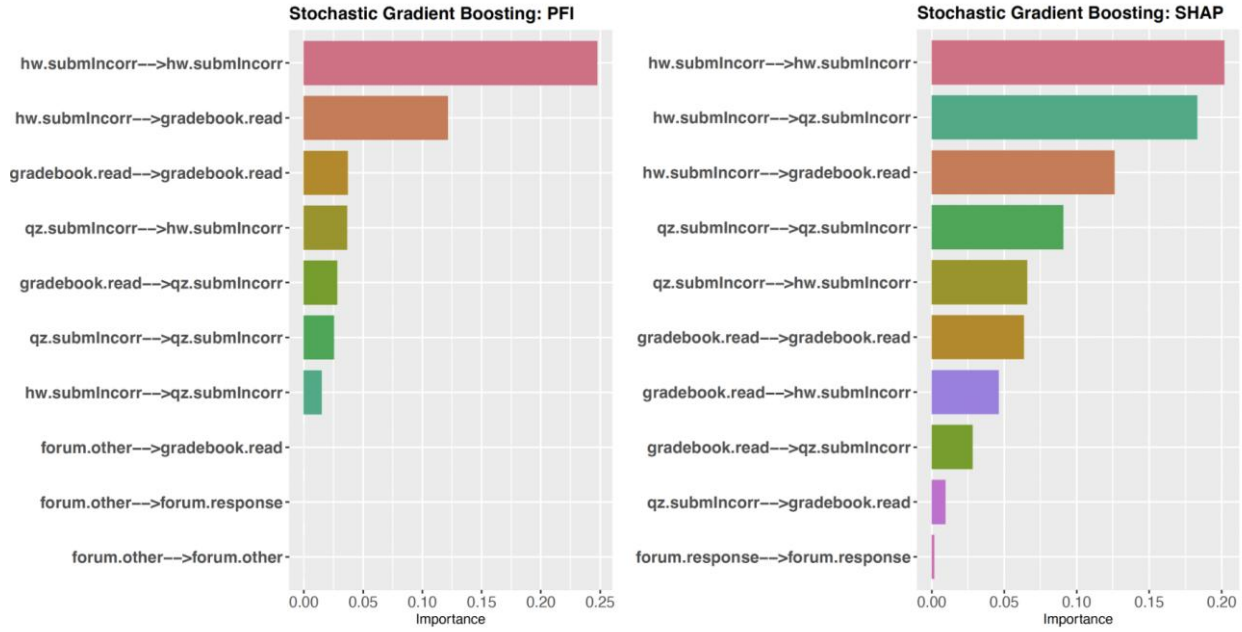
Sample 2: 19SP-2

Deep feedforward neural network (84% sensitivity, 80% F2 score, 73% balanced accuracy)



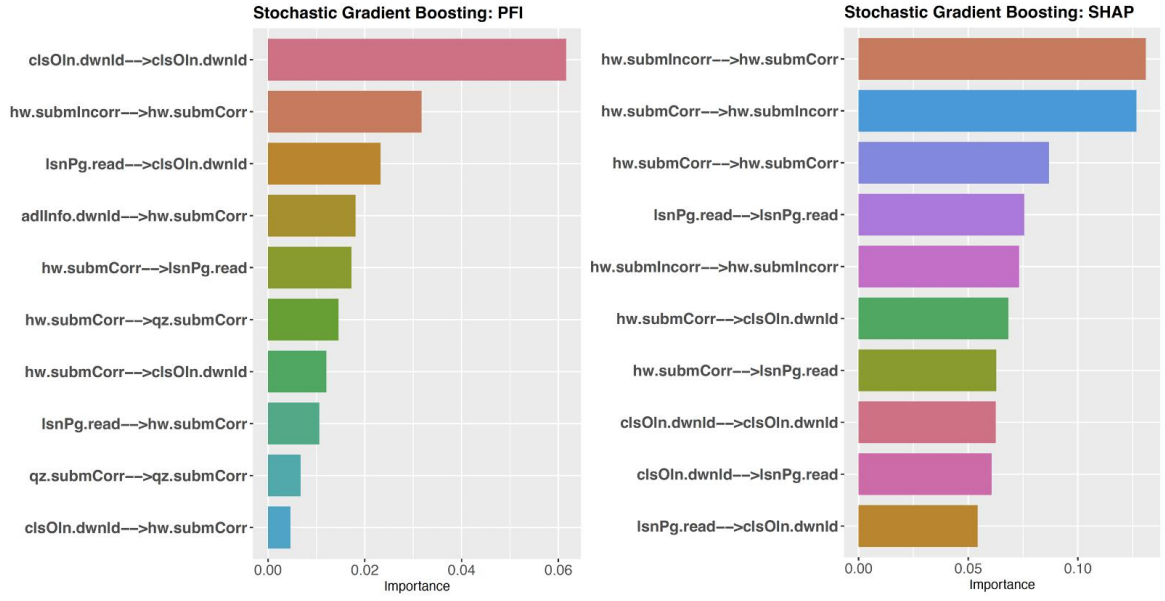
Sample 3: 19FA-1

Stochastic gradient boosting (82% sensitivity, 81% F2 score, 71% balanced accuracy)



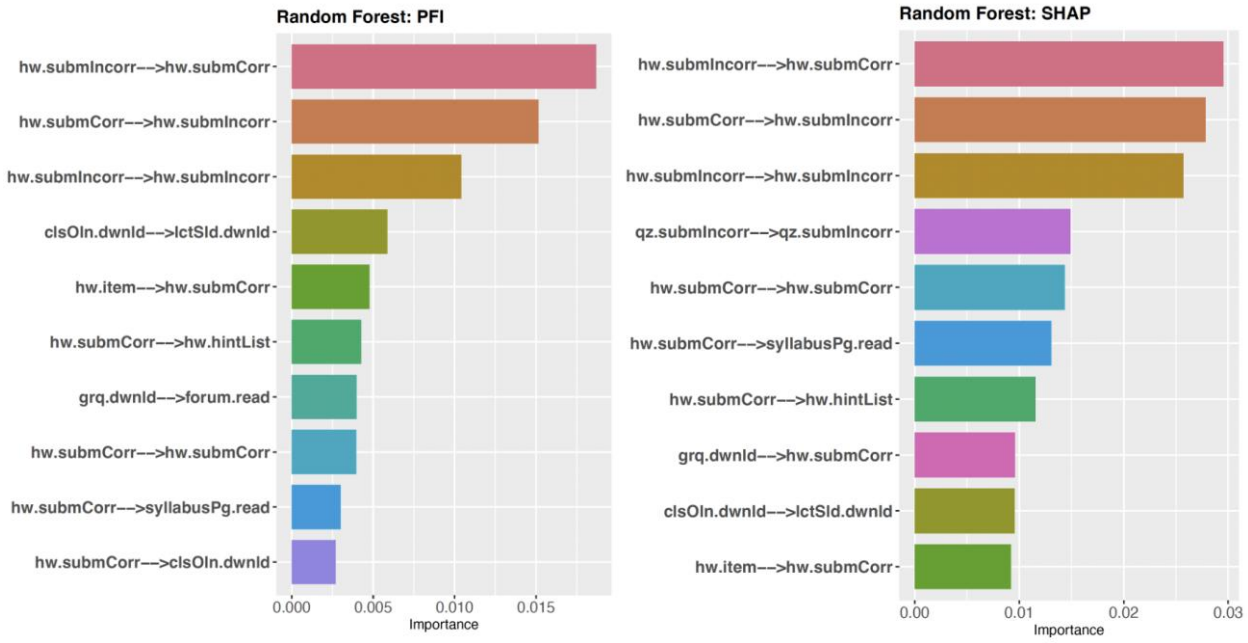
Sample 4: 19FA-2

Stochastic gradient boosting (69% sensitivity, 69% F2 score, 67% balanced accuracy)



Sample 5: 20SP-1

Random forest (73% sensitivity, 67% F2 score, 72% balanced accuracy)



Sample 6: 20SP-2

Stochastic gradient boosting (81% sensitivity, 80% F2 score, 82% balanced accuracy)