Early Detection Prediction of Learning Outcomes in Online Short-Courses via Learning Behaviors

Weiyu Chen, *Member, IEEE,* Christopher G. Brinton, *Member, IEEE,* Da Cao, Amanda Mason-Singh, Charlton Lu, Mung Chiang, *Fellow, IEEE*

Abstract—We study learning outcome prediction for online courses. Whereas prior work has focused on semester-long courses with frequent student assessments, we focus on short-courses that have single outcomes assigned by instructors at the end. The lack of performance data and generally small enrollments makes the behavior of learners, captured as they interact with course content and with one another in Social Learning Networks (SLN), essential for prediction. Our method defines several (machine) learning features based on the processing of behaviors collected on the modes of (human) learning in a course, and uses them in appropriate classifiers. Through evaluation on data captured from three two-week courses hosted through our delivery platforms, we make three key observations: (i) behavioral data contains signals predictive of learning outcomes in short-courses (with classifiers achieving AUCs ≥ 0.8 after the two weeks), (ii) early detection is possible within the first week (AUCs ≥ 0.7 with the first week of data), and (iii) the content features have an "earliest" detection capability (with higher AUC in the first few days), while the SLN features become the more predictive set over time as the network matures. We also discuss how our method can generate behavioral analytics for instructors.

Index Terms—Clickstream Data, Data Mining, Predictive Learning Analytics, Learning Outcome Prediction, Social Learning Networks

1 Introduction

A multitude of online learning platforms have emerged over the past decade, offering services ranging from tutoring to professional development to higher education. For all its benefits, however, the quality of online learning has been criticized. In comparing it to traditional, face-to-face instruction, researchers have found lower engagement and knowledge transfer for learners, both in higher education [2] and corporate training [3]. These poorer outcomes have been attributed to factors such as the asynchronous nature of interaction online, which places limitations on social learning [4].

In free, open online courses, lower engagement and knowledge transfer may be acceptable, because learners have varying motivations for enrollment in the first place. Yet, in the case of corporate training, with over $50 billion has been spent on training by corporations in the US each year since 2009, engagement, retention, and knowledge transfer from courses to the workplace are reportedly not meeting the expectations of employers [5]. In this paper, we propose analytics derived from learner behaviors to improve learning outcomes.

- *W. Chen, D. Cao, C. Brinton, and A. Mason-Singh are with the Advanced Research team at Zoomi Inc. Email: {weiyu.chen, christopher.brinton, da.cao,amanda.mason-singh}@zoomiinc.com.*
- *C. Lu is with the Department of Computer Science at Duke University. Email: charlton.lu@duke.edu.*
- *M. Chiang is with the College of Engineering at Purdue University and the Department of Electrical Engineering at Princeton University. Email: chiangm@princeton.edu.*

- *A previous version of this paper appeared in IEEE INFOCOM 2017 [1]. This version includes a more comprehensive set of algorithms, evaluation, and corresponding discussion.*

1.1 Predictive Learning Analytics

Predictive Learning Analytics (PLA) is emerging as a research area with the promise of helping instructors improve course quality, particularly in online courses [4]. Prediction of student drop-off rates [6], quiz scores [7], exam performance [8], and beneficial collaboration groups [9] each detect scenarios for which instructor intervention has a high chance of positively impacting the learning experience.

Most PLA methods have been developed for and evaluated on semester-long courses, *e.g.,* in Massive Open Online Courses (MOOCs) [2], [10]. These course scenarios usually have two properties that are useful from a modeling perspective. First is frequent assessments to track student progress, which has been the most common data source for PLA methods to-date, *e.g.,* using matrix factorization to discover patterns across student scores [7]. Second is a large number of enrolled learners, which increases the samples available for the PLA model. But what about cases in which (i) assessments are *not* used frequently, if at all, and (ii) the number of learners in a course session is small? This is common in online corporate training, where courses may last only several days and may have considerably smaller enrollments [11]. These "short-courses" require the development of a PLA methodology that can work with the type of data that is available for modeling.

Today, online course platforms can collect behavioral measurements about learners, which includes how they interact in Social Learning Networks (SLN) [4] and with the course content. The resulting content clickstream [2] and SLN [9] data present novel opportunities to design PLA methods that model learner attributes based on behavioral data in short-courses. This paper presents and evaluates one

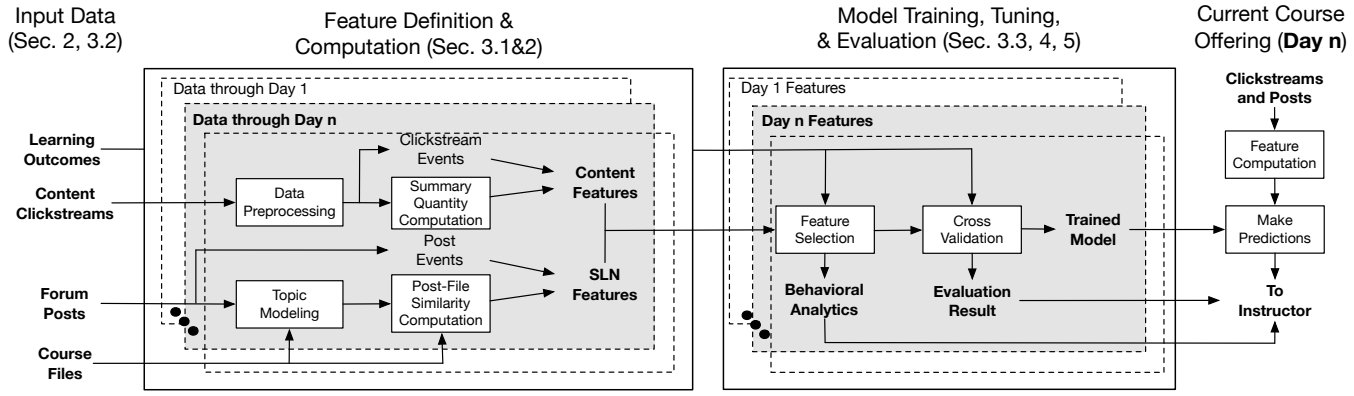


Fig. 1: Summary of the different components of the learning outcome prediction method we develop in this paper.

such method for learning outcome prediction, using data captured from short-courses hosted with our course Player, instructor Dashboard, and integrated discussion Forum.

1.2 Behavior-Based Outcome Prediction

In this work, we investigate the following research questions related to learning outcome prediction:

- *If pre-processed correctly, can behavior alone be used to predict learning outcomes in short-courses?*
- *How early into short-courses can these predictions be made with reasonable quality?*
- *Is the learning behavior associated with course content or with SLN more effective for prediction?*

Researchers have proposed algorithms for student performance prediction that augment assessment-based methods with behavior-based machine learning features [2], [7], [12]. Motivated by these schemes, in this work we consider the challenging case of short-courses with small enrollments and without intermediary assessments, thereby necessitating fully behavior-based, sparse PLA modeling.

Our methodology. Fig. 1 summarizes the methodology developed in this paper. To make predictions during the nth day of a course's current offering, we use the behavioral data collected from the first n days of prior offerings of this course as input. Using our system architecture for data capture (summarized in Sec. 2), one of the key challenges is to process this raw data into effective feature sets for modeling learning behavior, which we address in Sec. 3.1. In particular, we define two types of features:

(i) Content features: These features summarize learner behavior while interacting with course content in the Player. They include a novel definition of how to measure a learner's "engagement" on different content files.

(ii) SLN features: These features summarize learner discussions in the Forum. They include the similarity between learner's posts and the course content, determined through natural language processing models.

Prior works applying content features to prediction [6], [7] have relied on clickstream data from a single learning content type. Other works that have considered SLN features [12], [13] have neglected topic similarity component. Our subsequent feature selection (Sec. 3.3) shows that the engagement and topic similarity components are particularly useful in outcome prediction for short-courses.

With the objective of predicting whether a learner will ultimately pass or fail a course, our method uses the feature sets as input to different classifiers in training and evaluation, which is the focus of Sec. 4. The choice of classifier and parameters is made through cross validation accounting for the need for sparse modeling (Sec. 4.1). The evaluation result from this stage (Sec. 4.3), as well as behavioral analytics from the feature correlations (Sec. 5.1 & Sec. 5.2), can be shared with instructors to give them ways to assist learners, as pointed out in Fig. 1. Finally, the real-time predictions and corresponding early detections are made by applying the trained model to the features computed on the data collected thus far in the current offering, and the results are made available to the instructor too.

Evaluation and key results. To evaluate our outcome prediction method, we use datasets from three recent courses (described in Sec. 3.2) we delivered for a professional training course provider in the US. Each course session lasts two weeks and has a single binary outcome (pass/fail) at the end that is determined by the instructor. Through simulating the predictions for each course using our day-by-day modeling approach, we make three main observations:

- The highest performing algorithms that can model under sparse conditions reach ≥ 0.8 AUC by the end of the courses, exceeding 0.9 in some cases, with ≤ 0.1 Type II error.
- Using only the first week of data, the algorithms can still reach ≥ 0.7 AUC, which underscores the early detection capability of behavioral data.
- The content features exhibit an "earliest" detection capability in the first few days of a course, while the SLN features tend to bring superior quality after that.

2 Learning Technology Systems

Our system has four main parts, shown in Fig. 2: the course Player, the analytics Dashboard, the discussion Forum, and the Backend. We describe these parts in this section.

2.1 Player: Learner-facing

Learners obtain access to the Player through web browser. The data measurements collected through the Player are processed to compute the content-based features in Sec. 3.

Course architecture. Each course is organized into a set of modules, each module consisting of one or more units. A

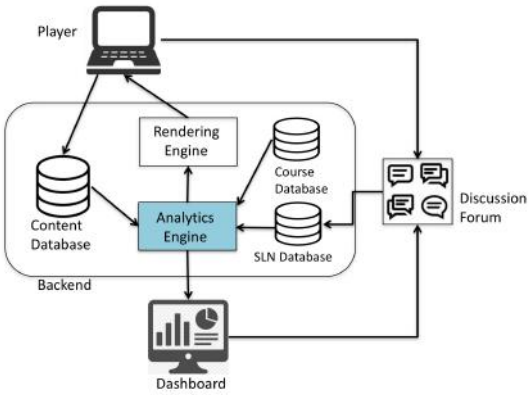


Fig. 2: System architecture overview.

unit is the most basic entity of a course, *i.e.,* the course is delivered to learners through the Player as a sequence of units. Each unit may contain one or more content files, where each file corresponds to a different learning mode (*i.e.,* content type). These files can include interactive slideshows, PDFs, text articles, and lecture videos. In the courses we consider in this paper, each unit is some combination of the first three of these content types; samples of slides and PDFs within the Player can be seen in Fig. 3.

User functions and data capture. In interactive slideshows, a learner can perform the following actions: `play` (`Pl`), `pause` (`Pa`), and `skip` forward (`Sf`) or backwards (`Sb`) on the current slide, and `advance` to the next slide. Within a PDF and an Article, learner can `scroll` up (`Su`) or down (`Sd`) on the pages. In any of these file types, learners can create a `note` or a `bookmark` at any location. Each time one of these actions occurs, a clickstream event with timestamp, user, and position identification information is sent to the Content Database in the Backend (Fig. 2).

Actions outside content are also captured. An `enter` (`En`) / `exit` (`Ex`) event is created whenever a learner enters / exits a unit, as is a `login` / `logout` event whenever a learner logs in / out of the course. Also, learners can customize the window layout in their browser. A `window` event is created each time a learner maximizes (`Wx`) or minimizes (`Wn`) a window.

2.2 Dashboard: Instructor-facing

The Dashboard is divided into tabs, each with charts on a different learning aspect. The instructors for the courses considered in this paper had access to the three following tabs, the latter two of which are shown in Fig. 4:

Overview. This provides high-level summaries of learners' activity and progress.

Engagement. Each learner is given an engagement score in each unit and module, and for the whole course. This tab visualizes these scores for an instructor to draw comparisons.

Content. This shows time spent, number of views, and completion rate on each content file (formalized as prediction features in Sec. 3). A progress bar is shown for the average completion rate. Instructors can access plots of time spent and view count across each partition of a file.

2.3 Discussion Forum

Our system integrates with NodeBB[1], an open-source discussion forum platform. Each course's forum is divided into a set of threads, with the first post in each thread being made by the instructor.

User functions and data export. Within a thread, a user can create a `post` (consisting of some text), `reply` to a post, and `up-vote` or `down-vote` a post. At the end of a course, the NodeBB API provides the details of each thread to the SLN Database in the Backend (Fig. 2). For each post, it indicates the user ID, timestamp, text, net votes (up-votes minus down-votes), replies, and whether each reply was an instructor or a learner.

The interaction between learners in the forum is an important part of the SLN. In Fig. 5, we illustrate interaction graphs for three course sessions considered in this paper (see Sec. 3). Each node is a learner, and the weight $w_{i,j}$ from learner i to j is proportional to the number of times i posted and/or responded to j. We see that the structure in these courses is rather dense (with $\geq \%34$ of the links non-zero, including learners who do not post that are not depicted), contrary to the case of MOOCs [9]. This foreshadows an observation we will make in Sec. 4 that differences in outcomes are more readily detected from the contextual rather than the structural aspects of the discussions.

Table 1 summarizes the main event types from the Player and the Forum considered in this paper.

2.4 Backend: Storage and Processing

The Backend in (Fig. 2) is divided into five main parts.

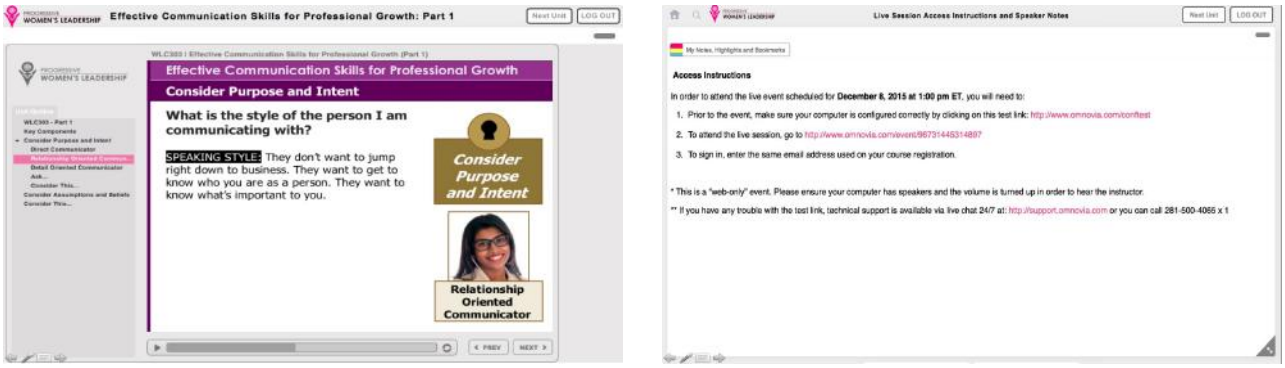
Databases. The Course Database is where the learning resources for a course are stored. The Content Database (resp. SLN Database) is where the measurements described in Sec. 2.1 (Sec. 2.3) on learners' activity in the Player (discussion forum) are stored. The clickstream events are stored in JSON format, each with an associated timestamp, learner ID, course ID, and session ID (unique to each log-in). The discussion information are also stored in JSON format.

Engines. The Rendering Engine fetches information from the Content Database about learners' current state to determine what from the Course Database they are shown next. In these courses, this is done in a unit-by-unit fashion, *i.e.,* at the beginning of the course, only the first unit is available in the table of contents, and every time a learner finishes the current unit, a new one is loaded. The Analytics Engine performs computations on the Content and SLN Databases, feeding the results to the Dashboard for visualization (and to the Rendering Engine if the course is adaptive). The computations made by this engine, and the demonstration of prediction algorithms that will be incorporated into it, is what we will focus on in Sec. 3, 4, and 5.

3 ML Features and Datasets

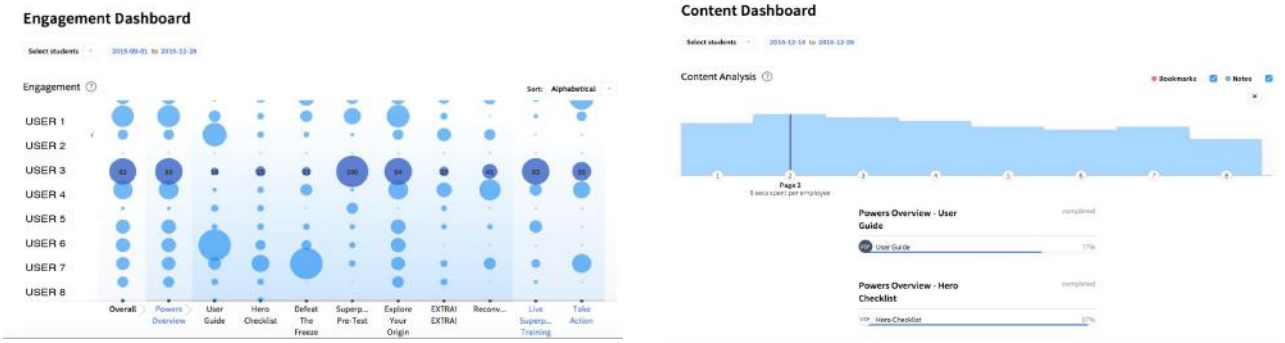
In this section, we present our behavior-based machine learning features. We will first specify the feature matrix that we compute for each dataset (Sec. 3.1), then give descriptive statistics of datasets in terms of these features (Sec. 3.2), and finally describe the feature selection process (Sec. 3.3).

1. www.nodebb.org.



(a) Interactive slide (b) Article

Fig. 3: Snapshots of two of the file types as they were delivered to learners in the Player for these courses. The Player interaction features are also shown here, *e.g.*, the slide scrubber in (a) and the next button in (b).



(a) Engagement (b) Content

Fig. 4: Snapshot of two of the tabs on the Dashboard used by instructors in these courses.

3.1 Our Machine Learning Features

Let $\mathbf{A} = [a_{v,f}]$ be the learner-feature matrix for a course, where $a_{v,f}$ is the value that feature $f \in \mathcal{F}$ takes for each learner v. We write $\mathbf{A} = [\mathbf{A}_c \ \mathbf{A}_s]$, where $\mathbf{A}_c$ and $\mathbf{A}_s$ are the matrices of content features and SLN features, respectively. In what follows, we define the quantities that comprise the corresponding feature subsets $\mathcal{F}_c$ and $\mathcal{F}_s$.

3.1.1 Content Features ($\mathcal{F}_c$)

$\mathcal{F}_c$ summarizes the interactions a learner has with the Player. Event interactions consist of the eight different types summarized in Table 1: each type appears in $\mathcal{F}_c$ one time for each learning content file (*e.g.*, slide, PDF) that they apply to. We use the frequency of events rather than indicator variables to account for how often learners use different behaviors. Additionally, $\mathcal{F}_c$ includes summative quantities given in the Dashboard; in particular, time spent, completion rate, and engagement. In what follows, we divide each file o into a set of smaller partitions $\mathcal{P}(o)$, where $p \in \mathcal{P}(o)$ refers to the pth partition. For article and PDF, $\mathcal{P}(o)$ is the set of pages, and for interactive slides, $\mathcal{P}(o)$ is the set of one-minute video segments making up the full set of slides.

Time Spent. For each content file, this is the amount of (real) time that a learner spent on that file. Letting $t_{v,o}$ be the time spent by learner v on file o, we sort v's clickstream events on v by timestamp and aggregate the time elapsed between each pair of events. In doing so, we filter two cases of clear off-task behavior. First is if more than $2\bar{t}_o$ [2] has elapsed between a pair of measurements, where $\bar{t}_o$ is the expected time spent on o (defined later); in this case, the learner will be given $2\bar{t}_o$ for this pair. Second is if the first event in the pair is a `Wn` event, in which case the learner has minimized the Player; in this case, no time will be awarded. We then calculate time spent by summing over files in the unit, for each module by summing over units, and for the full course.

Completion Rate. The completion rate $r_{v,o} \in [0, 1]$ is the fraction of file o that learner v viewed [7]. This is determined by finding the fraction of content partitions $\mathcal{P}(o)$ that the learner generated at least one clickstream measurement on. For example, if the Player recorded `scroll` events for two pages of a 10-page PDF o, then $|\mathcal{P}(o)| = 10$ and $r_{v,o} = 0.2$. We calculate the completion rate for each unit by averaging over the files in the unit, for each module by averaging over the files in the module, and for the full course.

Engagement. We define engagement as a model for the amount of effort a learner is putting into studying a piece of content. As with time spent and completion rate, engagement appears in $\mathcal{F}_c$ once per content file, once per unit, once per module, and once more as overall for the course.

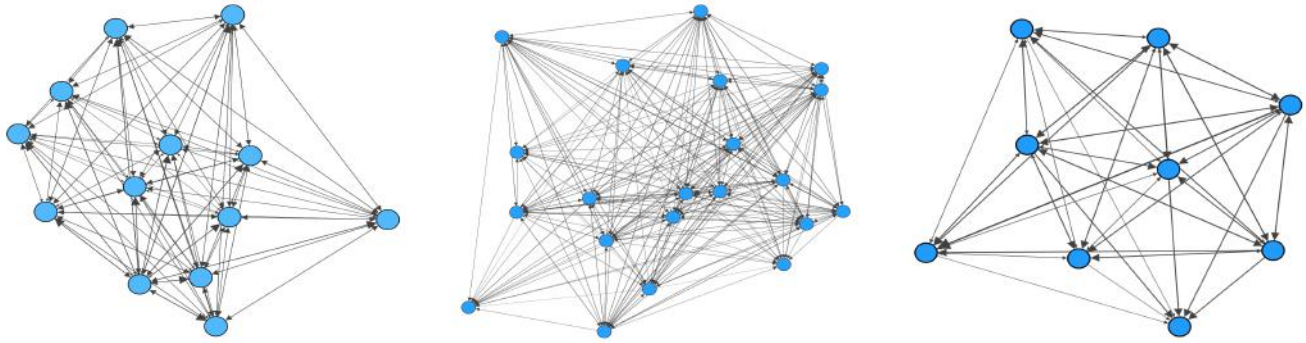
File-level: Let $t_{v,p}$ be the time spent by user v on partition p, and $\bar{t}_p$ be the "expected" time spent on p for normalization (defined below). Similarly, let $b_{v,p}$ be the number of notes and bookmarks (referred together as annotations) created on p, and $\bar{b}_p$ be the expected value of this quantity. Engagement on o, $e_{v,o}$, is defined as:

$$e_{v,o}(r,t) = \min(\gamma \times r_{v,o} \times \prod_{p \in \mathcal{P}_o} \left(\frac{1 + t_{v,p}/\bar{t_p}}{2}\right)^{\alpha_t} \left(\frac{1 + b_{v,p}/\bar{b_p}}{2}\right)^{\alpha_b}, 1) \quad (1)$$

Here, $\alpha_t, \alpha_b \geq 0$ are parameters that model the diminishing marginal returns property of the time spent and note creation components, respectively. Through this, a learner's

Event	Description	File type(s)
`play` (`Pl`)	A `play` event begins when a click event changes a content file to the `playing` state.	Slides
`pause` (`Pa`)	A `pause` is recorded when a click event changes a content file to the `paused` state.	Slides
`skip` (`Sb`, `Sf`)	A `skip back` (`forward`) occurs when a scrubber is brought to an earlier (later) position.	Slides
`scroll` (`Su`, `Sd`)	A `scroll up` (`down`) occurs when a scroll bar is brought to an earlier (later) position.	PDF, Article
`note`	A `note` is recorded when a note is created.	PDF, Article, Slides
`bookmark`	A `bookmark` is recorded when a bookmark is created.	PDF, Article, Slides
`window` (`Wx`, `Wn`)	A `window max` (`min`) event occurs when a content file is maximized (minimized).	PDF, Article, Slides
`enter` (`En`, `Ex`)	An `enter` (`exit`) event occurs when a a learner enters (exits) a unit in the Player.	–
`post`	A `post` event happens when a user creates a post in a thread.	Forum
`reply`	A `reply` event occurs when a user creates a reply to a post.	Forum
`vote`	An `up-vote` (`down-vote`) occurs when a post receives an up-vote (down-vote).	Forum

TABLE 1: Summary of the behavioral events analyzed in this paper, captured by the Player (content events) and Forum (SLN events).



(a) VE (b) ES (c) LS

Fig. 5: Graph of learner interaction on the discussion forums, for one session of each course analyzed in this paper (see Table 2.)

time spent and annotations on each specific p counts incrementally less towards her engagement, *i.e.,* a learner is rewarded more for distributing her time spent across more partitions, and similarly for a learner's note creation behavior, *i.e.,* a learner is rewarded more for distributing notes across more partitions. The division by 2 makes the computation for each partition relative to a learner that registers the expected time spent $t_{v,p} = \bar{t}_p$ and annotations made $b_{v,p} = \bar{b}_p$. $\gamma \in (0, 1]$ is an instructor-specified constant that controls the spread of the overall engagement distribution; note that if completion on the file $r_{v,o} = 1$ and the learner has $t_{v,p} = \bar{t}_p$ and $b_{v,p} = \bar{b}_p$ on each p, then $e_{v,o} = \gamma$. We discuss the selection of γ, α_t, and α_b in Sec. 3.2.

Unit, module, and course-level: A weighted average is taken across the files $\mathcal{O}(u)$ in a unit u to come up with the unit-level engagement: $e_{v,u} = \sum_{o \in \mathcal{O}(u)} \bar{t}_o e_{v,o} / \sum_o \bar{t}_o$ from (1) for each learner, where $\bar{t}_o$ is the expected length of o (defined below). Similarly, a weighted average is taken across units to come up with module and course-level engagements.

Normalization values: To calculate $\bar{t}_p$ and $\bar{t}_o$ for PDF and article, we first use Optical Character Recognition to obtain text transcripts, and correct any inconsistencies in the output. The reference time spent $\bar{t}_p$ on p is the expected time a learner will take to read the transcript of this partition, assuming a standard average reading speed of 6.6 characters per second. $\bar{t}_o$ is then $\sum_p \bar{t}_p$. For slides, $\bar{t}_p = 60$ sec $\forall p$, and $\bar{t}_o = 60|\mathcal{P}(o)|$ sec is the total length of the videos that comprise the interactive presentation. Since learners do not frequently use the note/bookmark creation functions, we set $\bar{b}_p = 0.05 \ll 1$, i.e., less than one note is expected across the 15-or-so files in each course. Note that these quantities are defined in terms of average viewing and reading speeds for video and text content, respectively, which may be different from the speed at which a learner can comprehend the material [14]. We will see in Sec. 3.3 that the definitions we use lead to engagement features that are among the most correlated with learning outcomes, which gives justification to our choices for the purposes of prediction.

3.1.2 SLN Features ($\mathcal{F}_s$)

$\mathcal{F}_s$ contains quantities that summarize a learner's interaction within the SLN. This includes the frequency of the Forum events from Table 2: the number of `posts` (and replies) a learner made, the number of `replies` the learner received, and the net `votes` the learner received on her posts/replies. It also includes the total number of words contained in said posts/replies. Finally, it includes the time period that a leaner stayed active in the forum, defined as the time elapsed between the learner's first and last post.

Content similarity. $\mathcal{F}_s$ also contains features describing the contextual/topical aspect of a learner's posts. To measure the relevance of a learner's discussion to the course content, we define a content similarity measure $s_{v,u}$ between unit $u \in \mathcal{U}$ and learner $v \in \mathcal{V}$. The $s_{v,u}$ are included as features in $\mathcal{F}_s$ for each course unit. They are obtained as follows:

Topic distributions: We first extract the set of topics $\mathcal{K}$ in the course, and represent u's content and v's posts as probability distributions $\mathbf{d}_i = (d_{i,1}, ..., d_{i,|\mathcal{K}|})$ over the topics, where $i \in \mathcal{I} = \{1, ..., |\mathcal{U}| + |\mathcal{V}|\}$ indexes unit $u(i) = i$ if $i \leq |\mathcal{U}|$ and learner $v(i) = i - |\mathcal{U}|$ otherwise. To do this, we represent each i as a word frequency vector $\mathbf{w}_i = (w_{i,1}, ..., w_{i,|\mathcal{X}|})$ over the full dictionary $\mathcal{X}$ of words. For $i > |\mathcal{U}|$, $w_{i,x}$ is the number of times learner $v(i)$ wrote the xth word in $\mathcal{X}$ across all her posts, and otherwise $w_{i,x}$ is how many times the xth

Course Name		Days	Units	Slide	Articles	PDF	Enrolled	Pass	Fail	Click	Post	Features
Vanquishing Toxic Employees	VE	14	11	1	6	6	79	15	64	20,126	73	145
Effective Communication Skills	ES	14	11	2	4	8	94	45	49	45,380	104	154
Developing Leadership Styles	LS	14	11	2	6	5	96	44	52	48,449	116	192

TABLE 2: Summary information of the short-course datasets used in this paper.

word appears in the text transcripts of $u(i)$.[2] In collecting words for $\mathcal{X}$ across the posts and content, we also apply appropriate stopword filtering, as in [9]. Then, with $\mathbf{W} = [\mathbf{w}_i] \in \mathbb{Z}^{|\mathcal{I}|\times|\mathcal{K}|}$ as the document-word matrix, we apply the popular Latent Dirichlet Allocation topic model [9], which results in the $\mathbf{d}_i$.

Similarity measure: With the topic distributions in hand, we define the similarity via total variation distance: $s_{v,u} = 1 - 0.5\|\mathbf{d}_{i(v)} - \mathbf{d}_{i(u)}\|_1$.[3] In this way, $s_{v,u} \in [0,1]$ captures the variation between the two topic-word distributions (over a finite alphabet).

3.1.3 Time-varying features

For each course, we define $\mathbf{A}(n)$, and its subsets $\mathbf{A}_c(n)$ and $\mathbf{A}_s(n)$, to be the feature matrices using the behavior available from the launch of the course through day n. Evaluating using day-by-day data allows us to assess how the quality of our predictions is expected to vary at different points along the course timelines. Note that prior works on student performance prediction [2], [7] have used the equivalent of a unit-by-unit approach for early detection (*i.e.,* using data collected in the first few units). The day-by-day approach allows us to account for the fact that learners tend to re-visit units at different times throughout a course.

3.2 Datasets and Computed Features

3.2.1 Course Design and Datasets

The datasets used to evaluate our method came from three short-courses that we hosted for a corporate training provider: "Vanquish Toxic Employees" (VE), "Effective Communication Skills" (ES), and "Techniques for Developing Your Leadership Styles" (LS). Summary information on the datasets is shown in Table 2 and is discussed further below. As the titles suggest, these courses emphasize business operations and leadership for professional development; most enrolled learners were company employees whose managers had required them to take this training.

Learning activities/tasks: Each of the three courses were two weeks long, and consisted of three tasks that learners were required to complete: (i) course content, (ii) forum discussions, and (iii) live events. The content consisted of 11 units, with each unit being delivered in slide, article, and/or PDF form. The Forum was available throughout the two weeks for learners to exchange questions/comments about the course with one another as well as with the instructors. The live events consisted of 2-3 real-time sessions moderated by the instructors, facilitated through the Forum.

For the live events, the first session was typically held one week in, aiming to, in the words of the instructors, "exchange thoughts and learning experiences." This is where learners would introduce themselves and describe events in the workplace relevant to the course content topics that they had individually experienced (*e.g.,* a toxic employee on their team). The latter two sessions were meant as a discussion of methods taught in the course, and how learners applied them to their individual situations in the workplace (*e.g.,* mitigating the impact of the toxic employee on others). In Sec. 4.3, we will see that effective outcome predictions can be made starting around the time of the first live session.

Evaluation: At the end of a course, each learner was given a single grade (pass, fail, extend, or expired). As there are no quizzes or exams in these courses, this outcome was based on the instructor's impression of the learner's knowledge transfer, informed by the information in our Forum and Dashboard systems; as described in Section 2, these systems track learner activity and participation throughout the course. There is no exact formula for how this evaluation is made, but our prediction results in Section 4 confirm that behavior is indicative of evaluation. In our analysis, we group fail, extend, and expired into a single group (denoted fail), because the instructors view these as undesirable.

Dataset summary: From Table 2, the courses contain between 20K and 50K clickstream events each. The number of each type of content file (interactive slideshow, article, and PDF) is also given here. LS and ES are well balanced in their ratio of Pass to Fail, but in VE, most of the learners (81%) fail. After computing the features $\mathbf{A}(15)$ for each course and removing any that were 0 for every learner, we are left with 140 to 200 features in each case.

3.2.2 Statistics of Content Features

Fig. 6 gives distributions of several of the features in $\mathcal{F}_c$ for each dataset.[4] Each point in each plot corresponds to one learner. The events in (a)-(c) are aggregated over all files, and then normalized by the number of units for comparative purposes.[5] The distributions in (d)-(f) are of course-level engagement, time spent, and completion rate, respectively. The distributions in (g)-(i) are of unit-level engagement, time spent, and completion rate across all units.

In comparing the distributions, we employ (i) a Wilcoxon Rank Sum test for the null hypothesis that there was no difference between the distributions overall, to compare in terms of shift, and (ii) Levene's test for the null hypothesis that there was no difference between the variances of the population, to compare in terms of standard deviation.[6] We consider the p-values (p_w and p_l, respectively) and the following are the main findings:

(i) `Pa` *is most common*: This is especially true in VE, where the median number of pauses per unit is 9, and the effect is significant in comparing to other events ($p_w \leq$ 1E-3).

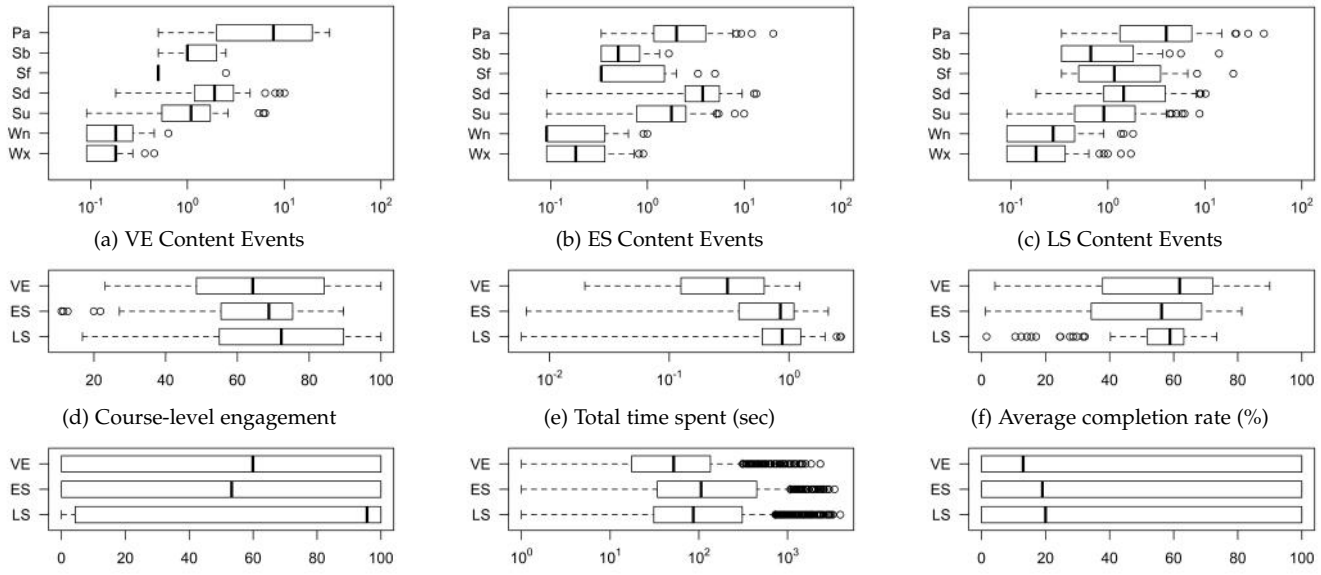
2. Since text transcripts are for PDF and article file types only, this does not explicitly include slides in a unit. However, for our datasets, we notice that the text is usually a repetition of the slide content.

3. $i(u)$ maps from u to its index in $\mathcal{I}$, and likewise $i(v)$ maps from v to $\mathcal{I}$.

4. For the log-scale plots, we only consider the non-zero values.

5. By doing this, we are implicitly assuming that the number of files is representative of the "length" of that type of content.

6. Note that Shapiro-Wilk tests detected significant departures from normality [7], making these tests appropriate.



(a) VE Content Events (b) ES Content Events (c) LS Content Events

(d) Course-level engagement (e) Total time spent (sec) (f) Average completion rate (%)

(g) Unit-level engagement (h) Unit-level time spent (hr) (i) Unit-level completion rate (%)

Fig. 6: Boxplots of select content features $f \in \mathcal{F}_c$ computed for each dataset. (a)-(c) are for the event features, (d)-(f) are for the analytic features that appear on the Dashboard at the course level, and (g)-(h) are for the analytics features at the unit level. In (a)-(c), we see that `Pa` tends to be the most common type of event while the window events `Wn` and `Wx` are least common. (d) and (g) show that the engagement settings in formula 1 lead to healthy distributions across learners. In (e) and (f), we see that VE has the highest completion but the lowest time spent, which could be a reason for the outcomes being skewed towards fail.

(ii) `Sd` *occurs more often than* `Su`: The shift is significant in ES and VE (the medians increase from 1.1 to 1.7 and from 1.9 to 3.7, with $p_w \leq$ 7E-15), though it is less significant in LS ($p_w \leq$ 9E-2). This is intuitive since learners must scroll down to move forward in the articles and PDFs.
(iii) Engagement distributions are useful: We set $\gamma = 1$, $\alpha_t = 0.1$, and $\alpha_b = 0.01$ in (1) to generate engagement distributions with large and relatively uniform spreads across the ranges. With this setting in each course, learner engagement varies from low values ($\leq$ 23) to 100, with medians between 60 and 70, which makes it a useful metric for instructors to compare learners.[7] Note that $\alpha_b = 0.01$ combined with $\bar{b}_p \ll 1$ means learners are not penalized substantially for lack of annotations, but still are rewarded if they do create them. The fact that engagement is one of the most correlated content features for prediction in Sec. 3.3 also validates these choices.
(v) LS has more consistent completion rate: The completion rate for LS in (f) has a smaller standard deviation compared with the other courses ($p_l \leq$ 1.8E-4).
(vi) `window` *events and annotations are consistently uncommon*: `Wx` and `Wn` both occur significantly less than all other events plotted in each course ($p_w \leq$ 1.1E-4). This means learners rarely change the window layout. Also, only 5 `note` and `bookmark` events were created across the three datasets.

3.2.3 Statistics of SLN Features

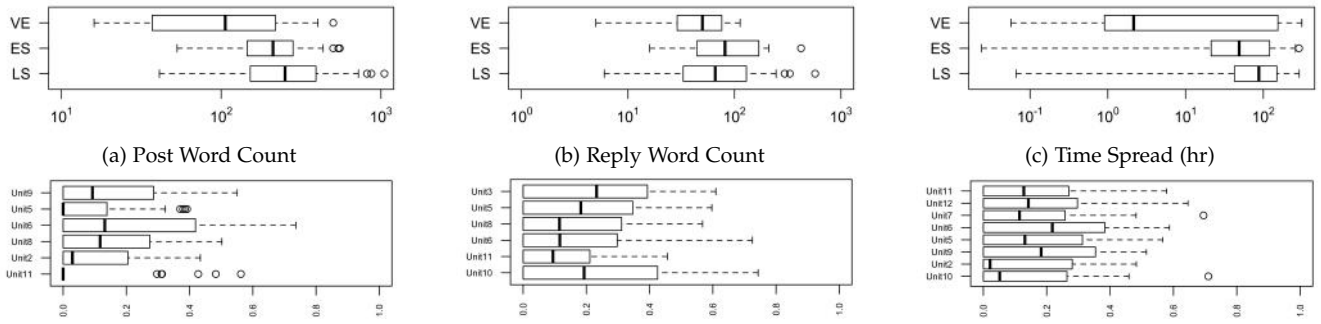
Fig. 7 gives the distributions of several of the features in $\mathcal{F}_s$. In (a)-(c), word count in a learner's posts, word count in replies to a learner's posts, and posting time spread are given across courses. (d)-(f) shows the learner-unit discussion similarities each course is within each of the three courses. Table 4 summarizes the five topics with highest support extracted from the posts and text content in each course. We make a few observations:
(i) SLN activity is lower in VE: Each of the three features (posts, replies, and time spread) are lower in VE than in other courses (though only significant for word count, $p_w \leq$ 7.3E-5). Given that the course outcome in VE is heavily skewed towards fail (81%), this foreshadows our point in Sec. 3.3 that SLN features are correlated with outcomes.
(ii) Topic words are relevant and supports are consistent: From the titles of the courses, we see that the topics are representative of likely discussions for each course (*e.g.*, $k = 1$ in VE is about "toxic employees", $k = 2$ in ES is about "communicating effectively"), and are reasonably non-overlapping in the top words they include.
(iii) Similarities vary unit-to-unit: We can see that content in certain units is more heavily discussed by learners than others; in particular, unit 6 in VE, units 3 and 10 in ES, and units 6 and 9 in LS. These insights can be useful to instructors to see which content is the focus and whether that is in line with success. However, the statistical significance in shift only holds consistently across the course for unit 6 in VE ($p_w \leq 0.014$ compared with all other units).

3.3 Feature Selection

Recall from Table. 2 that the full feature matrix $\mathbf{A}(n)$ for each course has approximately 140-200 feature columns. In order to reduce overfitting and improve model interpretability, we perform feature selection prior to training on each $\mathbf{A}(n)$, $\mathbf{A}_s(n)$, and $\mathbf{A}_c(n)$. We implemented three standard methods: correlation analysis, information gain, and random forest importance [15].

Comparing the features selected from these methods in terms of their eventual predictive quality, we found that

7. The distribution of engagement in VE is approximately normal distributed, with a Shapiro-Wilk $p > 0.03$.



(a) Post Word Count (b) Reply Word Count (c) Time Spread (hr)

(d) VE Similarity (e) ES Similarity (f) LS Similarity

Fig. 7: Boxplots of the main SLN features $f \in \mathcal{F}_s$ collected for each dataset. (a)-(c) are comparing across courses, while (d)-(f) are comparing the discussion similarity values $s_{v,u}$ across several units u in each course. In (a)-(c), we see that SLN activity in VE is significantly lower than in the other courses. In (d)-(f), we see that content in certain units (e.g., unit 6 in VE) stands out as being the most correlated with learner discussions.

those selected by correlation analysis tended to yield the best results. In running correlation analysis on $\mathbf{A}(15)$ (*i.e.,* the full feature matrix built from all the course data), the selected behavioral features for each course are summarized in Table 3. We choose the top ten because prediction quality saturates beyond this point (for an analysis on the effect of varying the number of features, see Sec. 5.2). Noting that each of these ten have *positive* correlations with the course outcome, we make a few observations:

(i) Of the content features, the Dashboard quantities are more correlated: Engagement, time spent, and completion are more correlated with the outcome than the events in $\mathcal{F}_c$. With the exception of `enter`, events do not appear in the top-10.

(ii) The SLN features are more correlated than the content features: Features in $\mathcal{F}_s$ are more frequent in these lists than those in $\mathcal{F}_c$. The discussion post similarity features $s_{v,u}$ are notably important. The more relevant a learner's posts, the more familiar the learner is with the course content, which the instructors are likely to pick up on. This is consistent with the fact that the similarity values exhibit large variation in Fig. 7 (all except six of the ranges are ≥ 0.5).

(iii) Word count is a correlated feature in all courses: A higher word count for a learner tends to imply a higher probability of successfully passing the course. Given that this feature is independent of any course content and/or structure, it may be useful for course-independent prediction algorithms.

4 Prediction and Analytics

We now apply the feature sets from Sec. 3 to prediction. We first describe the algorithms and procedures used for evaluation (Sec. 4.1), and then present and discuss our results (Sec. 4.2 and Sec. 4.3.) Subsequently, in Sec. 5 we will show examples of behavioral analytics (Sec. 5.1) and also study the effect of feature selection (Sec. 5.2).

4.1 Classifiers and Procedure

Prediction classifiers. We consider six classifiers: K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Linear Discriminant Analysis (LDA), Random Forest (RF), Forward Neural Network (ANN), and Gradient Boosting (XGB). We choose these for a few reasons. First, they have each demonstrated good performance in predicting student outcomes in other works, *e.g.,* KNN in [16], SVM in [2], [7], LDA in [17], [18], RF in [19] (though only in [2], [7] with behavioral features), ANN in [20], and XGB in [10], [21].

Second, given their optimization approaches, they are typically applied to different feature types, and we are using a combination of indicator, integer, and continuous features; each of these classifiers has different learning properties that make it better suited in different scenarios. For instance, RF is an ensemble tree method applied to any type of feature, whereas XGB is also an ensemble method that boosts weak data. SVM uses a kernel function to find the optimal hyperplane separation and is typically applied to non-indicator features. LDA has been seen to work better on continuous quantities given that it finds a linear combination of the features which best separates groups [18].

Third, these classifiers each have a relatively small number of parameters to train (particularly RF and XGB), which is useful in preventing overfitting on the small training sample sizes of short-courses. While neural networks do not have this property, we consider them for completeness; we chose ANN over other possibilities because it has been seen to perform well on high dimensional data. We did also investigate other approaches (Convolutional Neural Networks and Recurrent Neural Networks) but found suboptimal performance on our datasets.

Parameters: For SVM, we use the radial basis function (rbf) kernel. [8] The parameters for SVM (kernel standard deviation (η) and regularization penalty (C)), RF (number of tress (τ) and variables (δ)), KNN (number of neighbors (κ)), XGB (maximum depth of tree (ν) and learning rate (υ)), and ANN (learning rate (υ) and hidden layer size (ι)) are tuned during the cross validation procedure described below.

Metrics. We primarily consider AUC (*i.e.,* the area under the ROC curve) and Type II error (*i.e.,* fraction of fails that are incorrectly predicted as passes) as evaluation metrics. In practice, we are interested in identifying learners who are at risk of failing in advance so the instructor can be notified. Consequently, we seek a classifier that obtains a low Type II error while maintaining a high AUC, so that we would minimize the number of failing learners that we misclassify (low Type II error) while not generating too many false alarms (high AUC). We will discuss the exact selection criteria of the algorithms further in Section 4.3. For

8. We found the radial kernel to obtain the better results.

f	VE	ES	LS
1	Word Count	Post Similarity to Unit 2	Post Count
2	Post Similarity to Unit 5	Post Similarity to Unit 3	Post Similarity to Unit 3
3	Post Count	Post Similarity to Unit 8	Post Similarity to Unit 7
4	Time Spread	Unit 10 Article Engagement	Post Similarity to Unit 5
5	Session Count	Word Count	Post Similarity Between to Unit 2
6	Post Similarity to Unit 7	Unit 10 Engagement	Post Similarity to Unit 4
7	Post Similarity to Unit 6	Unit 11 Article Engagement	Word Count
8	Unit 5 Slideshow Completion	Unit 11 Engagement	Time Spread
9	Unit 5 Slideshow Engagement	Unit 10 `enter`	Unit 11 Article Engagement
10	Post Similarity to Unit 1	Unit 11 Article Completion Rate	Unit 11 Engagement

TABLE 3: List of the 10 behavioral features selected based on correlation analysis on the full matrix $\mathbf{A}(15)$ for each course. All of these features have positive correlations with outcome, and they are ordered from highest to lowest.

VE

k	z_k(%)	top five words
1	12.7	employee problem toxic manage hero
2	12.5	jacki liza work steven member
3	11.8	situation always difficult address handle
4	9.8	conversation team behavior solution learn
5	9.5	behavior step toxic culture consequence
6	9.4	villain question workplace complete post
7	9.1	action plan start discuss point
8	9.0	time person set expect thought
9	8.7	negative civil work avoid comment
10	7.5	general person position attitude supervisor

ES

k	z_k(%)	top five words
1	13.4	key direct email consider prefer
2	13.2	communicate skill effect talk point
3	11.4	inform question gather time option
4	10.6	person word hear assumption success
5	9.6	listen understand paraphrase focus train
6	9.1	relationship message impact intent build
7	9.0	differ face program improve colleague
8	8.6	express speaker person describe step
9	8.1	assert idea tip mission statement
10	6.9	content complete discuss open earn

LS

k	z_k(%)	top five words
1	14.9	leadership style kill train role
2	13.5	people style work direct approach
3	13.5	team motivates focus challenge task
4	12.3	time award emotion leader language
5	10.7	high place change decision result
6	8.6	leadership reality stress train role
7	7.2	inspire authentic opportunity task trust
8	6.8	discuss complete present post choice
9	6.7	something adopt goal adapt event
10	5.8	great true welcome angry annoy

TABLE 4: Summary of the topics with $|\mathcal{K}| = 10$. Given for each topic k are its support z_k and highest five constituent words.

completeness, we also report Accuracy (Acc, *i.e.,* fraction of all predictions that are correct).

Cross validation. For training and evaluation, we (i) divide the dataset into folds ($K = 5$) stratified such that each fold has the same proportion of passes and fails, (ii) train and tune the algorithms through cross-validation, choosing the set of parameters with highest average accuracy, [9] and (iv) evaluate on the holdout fold, similar to the procedure detailed in [2]. The metrics we report are averaged over several runs, to obtain a general estimate of quality.

4.2 End-of-Course Prediction

In Table 5, we show the prediction results for each algorithm on (a) the full feature set $\mathbf{A}(15)$, (b) the SLN-only $\mathbf{A}_s(15)$, and (c) the content-only $\mathbf{A}_c(15)$ for each course. We make a few observations:

Behavioral data contains signals for outcome prediction. Considering the full (combined) feature matrix $\mathbf{A}(15)$ in (a), we see that at least one of the algorithms is able to obtain a high quality prediction, which indicates that behavioral data can be used to make effective outcome predictions even when no assessment data is available and the sample size is limited. More specifically, for at least one algorithm in each course the AUC is larger than 0.85, while the Type II error is less than 0.11, meaning that less than 11% of the fails would be incorrectly identified. RF, in particular, is able to obtain consistently high quality on the combined feature set (a) for each of the courses (Acc > 0.81, AUC > 0.72, Type II < 0.18). The repeatability across independent datasets gives evidence that our methodology will work when applied to other short-courses as well, and and has the potential to obtain similar/better performance on larger datasets.

SLN features are more useful than content features by the end. Comparing the quality of predictions using SLN features ($\mathbf{A}_s(15)$) vs. content features ($\mathbf{A}_c(15)$) in Table 5, we see that while the AUCs are comparable across courses and algorithms (SLN being higher in $9/18$ cases), predictions on SLN features obtain substantially lower Type II errors (SLN is lower in $16/18$ cases); in particular, the best cases achieved in each course are less than 0.20 for clickstream as opposed to less than 0.11 for SLN. This implies that by the end of the course, classifiers using the SLN features are better able to avoid predicting those who fail as passing incorrectly.

Algorithm choice varies based on course and feature set. With the exception of ANN, there is at least one pair of course and feature set for which each algorithm performs best (or close to best). We speculate that the poor performance of neural networks is due to the small sample sizes of these courses, as opposed to other predictive learning analytics settings where they have obtained high quality, *e.g.,* in forecasting cumulative grades from clickstream data in Massive Open Online Courses (MOOCs) [22]. Another example is [10] which used an ensemble of methods including both NN and XGB to predict course dropouts in MOOCs; again, the complexity and hyperparameters of ANN may require substantially more data points to correctly train. Also note that SVM demonstrates particularly low quality on the content features (AUC ≤ 0.5 in two cases). Interestingly, this is in contrast to results in [2] which showed SVM to obtain high AUC (> 0.75) with similar features in predicting quiz performance in MOOCs. In that application, however, there are orders-of-magnitude more samples for training, and each learner appears in the dataset multiple times,

9. We test $\eta \in \{0, 1, ..., 10\}$, $C \in$ {1E-5, 1E-4, ..., 1E5}, $\kappa \in \{1, 2, ..., 10\}$, $\tau \in \{10, 11, ..., 300\}$, $\upsilon \in \{0.001, 0.002, ..., 0.02\}$, $\nu \in \{3, 4, ..., 7\}$, $\iota \in \{2, 3, ..., 30\}$, $\delta \in \{1, 2, ..., 10\}$.

Course	Algo	Acc	AUC	TypeII
VE	RF	0.835 ± 0.011	0.727 ± 0.011	0.101 ± 0.011
	LDA	0.858 ± 0.012	0.895 ± 0.012	0.092 ± 0.012
	SVM	0.810 ± 0.001	0.500 ± 0.001	0.196 ± 0.001
	KNN	0.865 ± 0.010	0.796 ± 0.010	0.068 ± 0.010
	ANN	0.788 ± 0	0.742 ± 0	0.135 ± 0
	XGB	$\mathbf{0.915 \pm 0.007}$	$\mathbf{0.907 \pm 0.005}$	$\mathbf{0.010 \pm 0}$
ES	RF	0.827 ± 0.012	0.824 ± 0.012	0.179 ± 0.012
	LDA	0.750 ± 0	0.843 ± 0	0.25 ± 0
	SVM	$\mathbf{0.826 \pm 0.011}$	0.829 ± 0.011	$\mathbf{0.086 \pm 0.011}$
	KNN	0.817 ± 0.001	0.821 ± 0.008	0.222 ± 0.009
	ANN	0.489 ± 0	0.451 ± 0	0.255 ± 0
	XGB	0.826 ± 0.018	$\mathbf{0.862 \pm 0.010}$	0.255 ± 0.017
LS	RF	0.813 ± 0.012	0.808 ± 0.012	0.179 ± 0.012
	LDA	0.781 ± 0.012	$\mathbf{0.851 \pm 0.012}$	0.216 ± 0.012
	SVM	0.817 ± 0.010	0.822 ± 0.010	$\mathbf{0.102 \pm 0.010}$
	KNN	$\mathbf{0.821 \pm 0.012}$	0.562 ± 0.012	0.190 ± 0.012
	ANN	0.800 ± 0	0.747 ± 0	0.306 ± 0
	XGB	0.785 ± 0.018	0.660 ± 0.028	0.306 ± 0

(a) Combined

Course	Algo	Acc	AUC	TypeII
VE	RF	0.857 ± 0.009	0.749 ± 0.009	0.093 ± 0.009
	LDA	0.752 ± 0.048	0.731 ± 0.048	0.061 ± 0.048
	SVM	0.804 ± 0.007	0.503 ± 0.007	0.186 ± 0.007
	KNN	$\mathbf{0.873 \pm 0.011}$	$\mathbf{0.786 \pm 0.011}$	0.080 ± 0.011
	ANN	0.788 ± 0	0.474 ± 0	$\mathbf{0 \pm 0}$
	XGB	0.764 ± 0.021	0.349 ± 0.144	0 ± 0
ES	RF	0.789 ± 0.007	0.790 ± 0.007	0.243 ± 0.007
	LDA	0.800 ± 0	$\mathbf{0.828 \pm 0}$	0.273 ± 0
	SVM	$\mathbf{0.816 \pm 0.012}$	0.821 ± 0.012	$\mathbf{0.087 \pm 0.012}$
	KNN	0.753 ± 0.002	0.746 ± 0.002	0.250 ± 0.002
	ANN	0.755 ± 0	0.649 ± 0	0.353 ± 0
	XGN	0.747 ± 0.023	0.727 ± 0.002	0.302 ± 0.024
LS	RF	$\mathbf{0.850 \pm 0.011}$	0.849 ± 0.011	0.130 ± 0.011
	LDA	0.785 ± 0.0137	$\mathbf{0.853 \pm 0.014}$	0.192 ± 0.014
	SVM	0.828 ± 0.012	0.831 ± 0.012	$\mathbf{0.113 \pm 0.012}$
	KNN	0.828 ± 0.010	0.503 ± 0.010	0.186 ± 0.010
	ANN	0.800 ± 0	0.710 ± 0	0.306 ± 0
	XGB	0.775 ± 0.024	0.634 ± 0	0.306 ± 0

(b) SLN

Course	Algo	Acc	AUC	TypeII
VE	RF	0.806 ± 0.010	0.594 ± 0.010	0.156 ± 0.010
	LDA	0.830 ± 0.012	0.861 ± 0.012	$\mathbf{0.102 \pm 0.012}$
	SVM	0.809 ± 0.001	0.500 ± 0.001	0.191 ± 0.001
	KNN	0.802 ± 0.010	0.630 ± 0.010	0.141 ± 0.010
	ANN	0.606 ± 0	0.545 ± 0	0.308 ± 0
	XGB	$\mathbf{0.854 \pm 0.026}$	$\mathbf{0.894 \pm 0.008}$	0.115 ± 0.030
ES	RF	$\mathbf{0.812 \pm 0.005}$	0.820 ± 0.005	0.261 ± 0.005
	LDA	0.800 ± 0	$\mathbf{0.869 \pm 0}$	0.273 ± 0
	SVM	0.630 ± 0.009	0.609 ± 0.009	0.280 ± 0.009
	KNN	0.741 ± 0.005	0.755 ± 0.005	0.345 ± 0.005
	ANN	0.766 ± 0	0.754 ± 0	$\mathbf{0.039 \pm 0}$
	XGB	0.677 ± 0.023	0.716 ± 0.028	0.373 ± 0.030
LS	RF	$\mathbf{0.722 \pm 0.013}$	$\mathbf{0.726 \pm 0.013}$	$\mathbf{0.204 \pm 0.013}$
	LDA	0.679 ± 0.014	0.725 ± 0.014	0.266 ± 0.014
	SVM	0.515 ± 0.009	0.490 ± 0.009	0.468 ± 0.009
	KNN	0.644 ± 0.014	0.635 ± 0.014	0.302 ± 0.014
	ANN	0.705 ± 0	0.695 ± 0.002	0.265 ± 0
	XGB	0.577 ± 0.020	0.650 ± 0.013	0.359 ± 0.036

(c) Content

TABLE 5: Prediction quality of the algorithms on the content, SLN, and combined feature sets at the end of the course ($\mathbf{A}_c(15)$, $\mathbf{A}_s(15)$, and $\mathbf{A}(15)$). For each metric, we report the average and standard deviation across 50 cross validation trials. The algorithm obtaining the best value for each course-feature-metric triple is bold.

which allows the SVM to include learner/quiz indicator features. With SLN features, though, SVM's quality increases substantially, and for the combined case it is arguably the highest quality algorithm in two of the datasets (ES and LS).

4.3 Day-by-day Prediction

In Fig. 8, we evaluate the early detection capability of the full feature set for each course. To do this, we choose the algorithm with highest quality on $\mathbf{A}(15)$ for each course and each feature type from Table 5. Focusing on AUC and Type II error, the selection criteria is as follows: maximize AUC subject to Type II error being ≤ 0.1 rounded to one decimal place; referring to Table 2, this threshold means that only about 5 learners in each course who actually fail will be misclassified by the end of the course. With that algorithm identified, we perform training and evaluation over $\mathbf{A}(n)$ for $n \in \{1, ..., 15\}$. In order to evaluate the effect that each group of features has over time, we repeat this over $\mathbf{A}_s(n)$ and $\mathbf{A}_c(n)$, and show the resulting AUC by day in Fig. 9. From these plots, we make a few observations:

Behavioral data has an early detection capability. In Fig. 8 we can see that, as expected, the quality of the predictors tends to rise from the beginning to the end of the course, with AUC and Acc generally increasing and Type II error decreasing.[10] There is a tradeoff, then, between how early the predictions are applied and the expected quality. The following are two interesting points along the tradeoff in each course at which forecasts can be made in advance:

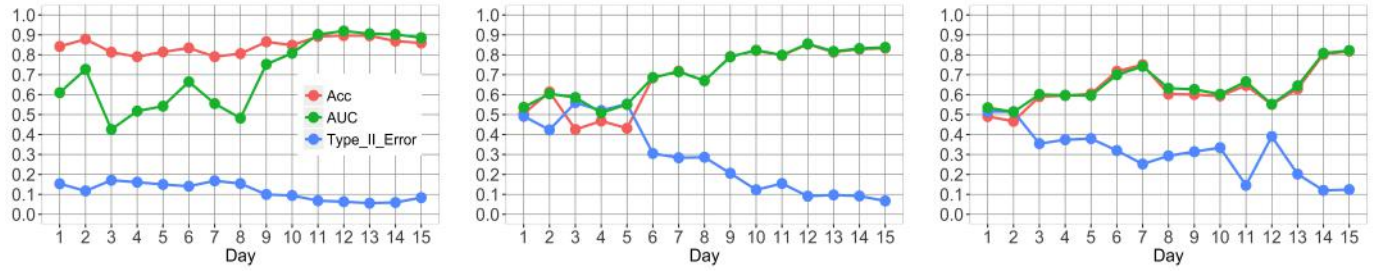
(i) Detection midway through: The AUC hits a local maximum around the midpoint of the courses (day 6 or 7), hitting approximately at 0.7 in each case. The corresponding Type II errors are 0.3 or lower, indicating that roughly 70% or more of the learners that will ultimately fail will be correctly identified as such at this point. This is right around the time of the first live event in the courses (see Sec. 3.2.1), which the instructors indicated would be a useful point for the information to be provided.

(ii) Detection three-quarters through: In VE and ES, the AUC saturates around three-fourths of the way through the course (day 10 or 11), at which point it is higher than 0.8 in both cases, exceeding 0.9 in VE. The Type II errors have also dropped to 0.1 or below, meaning that we can expect 90% or more of fails to be correctly identified. If the final stretch of the course provides sufficient time for instructor intervention, then this is an ideal point to apply the algorithms.

For "earliest" detection, content features have an advantage. After the first half or so of each course in Fig. 9, we see that SLN features obtain higher AUC than content features in ES and LS, consistent with the observation in Sec. 4.2. For all three courses, the content features provide higher quality early. This indicates that content data may be more useful for detections that must be provided at the earliest stages of a course, consistent with [2] for MOOCs. This phenomenon can be explained by the fact that a course's SLN develops and evolves over time. Interestingly, however, at least 10% of the social data comes in the last day, yet they are still important for prediction far prior to the end. We will investigate this through our feature analysis next in Sec. 5.

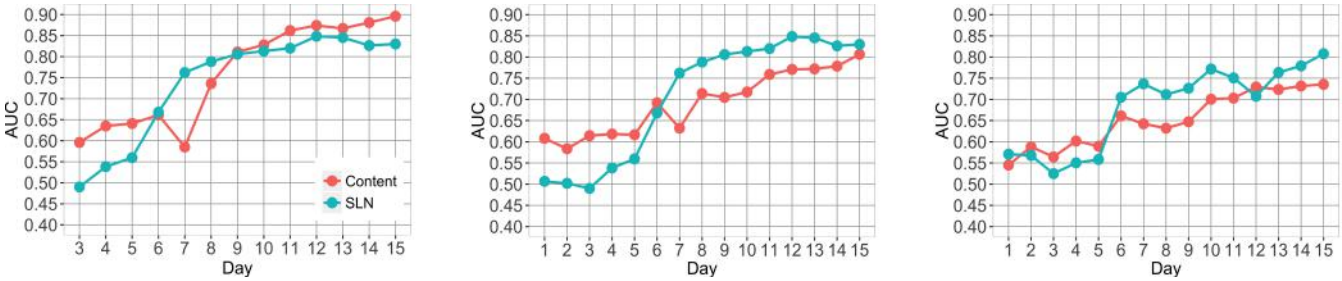
In practice, when an instructor is provided with an early detection of potentially failing learners, the next step would be to reach out to them. Type II error gives an expected fraction of failing learners that will not be detected. Type I error, on the other hand, would give the false alarm rate, *i.e.*, the fraction of learners reached out to that would have gone on to pass. We consider this to be less critical since a learner

10. The exception to this is in VE, where XGB quality oscillates until day 9. This is likely due to the joint effect of three factors that can cause overfitting here: (i) the small amount of data available towards the beginning, (ii) the imbalance of this particular dataset (which has a fail rate of 81%), and (iii) the negative binomial log-likelihood loss function of XGB not translating exactly to these classification metrics.



(a) XGB applied to VE (b) SVM applied to ES (c) SVM applied to LS

Fig. 8: Variation in prediction quality by day for each course, using the full feature set. At day n, the predictor is using $\mathbf{A}(n)$ for training. The AUCs reach 70% by day 7, which shows that behavioral features can be used for early detection in short-courses. Note that the line plots may overlap in some cases, *e.g.*, accuracy and AUC for later days in (b) and (c).



(a) VE (b) ES (c) LS

Fig. 9: Variation in prediction quality by day for each course, using the content and SLN features. The algorithm achieving the highest performance for is used in each case. Specifically, we use XGB for content in VE, KNN for SLN in VE, SVM for both types in ES, and RF for both types in LS. In ES and LS, the SLN features have higher quality beyond the first few days, while the content features are useful for earliest detection. In VE, the content features have higher quality except in the middle of the course.

predicted to fail would need to be exhibiting characteristics of the failing group (*e.g.*, someone procrastinating on key activities), and thus may still benefit from being reached out to. Still, for the selected algorithm in each course, the number of false alarms generated on $\mathbf{A}(15)$ was only 6.1, 12.1, and 12.3 for VE, ES, and LS respectively.

5 Feature Analytics

5.1 Feature Correlation Analysis

We also analyze how the correlations of the top features vary over time. In practice, this can give instructors insight into which specific behaviors are most related to eventual learning outcomes at different points and lead to recommendations to improve success.

The correlations of the top 5 features in Table 3 are plotted over time in Fig. 10. We make a few observations:

Rank convergence. The feature correlations generally become stronger over time with more data, as expected. The increases are not monotonic, however. There are points in time where learners who end up failing are participating in the discussions, so the instructors may attempt to further engage the learners during these periods.

Content discussion recommendations. As discussed in Sec. 3.3, the top features for each course include discussion post similarity to specific units. Analyzing the trends of these correlations leads to some interesting findings that can be turned into SLN discussion recommendations. In LS, notice that "post similarity to unit 7" has remarkably low correlation compared with the other features until day 9, even though by the end it becomes the third most correlated. This is likely because this unit is far down the syllabus, so learners are not focusing on this content until later; therefore, it may be beneficial to give advanced warning on the importance of this content.

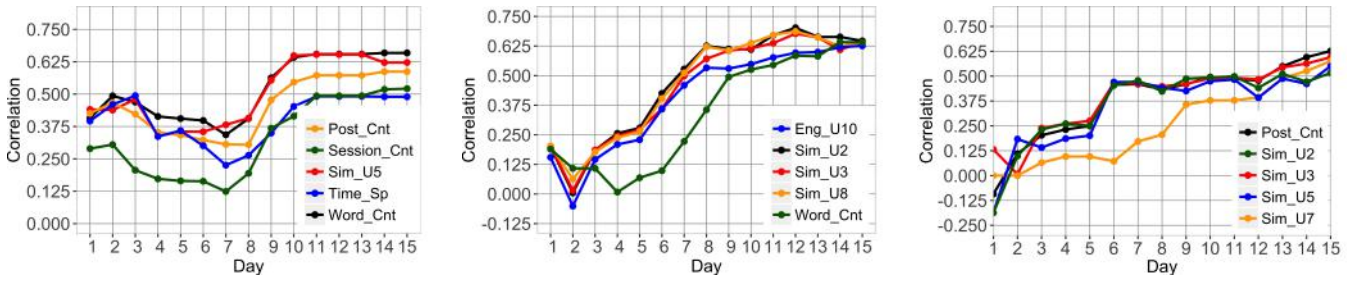
SLN developments. As discussed in Sec. 4.3, although the network matures over time, the SLN features are still indicative throughout most of the course period. Investigating the correlations the SLN features in Fig. 10, we see that for all courses, even part of the SLN data comes at the end of the course, the correlation of these features have remained relatively constant around course midpoint. This suggests that the maturing SLN has early signals that are predictive of course outcomes even when it is still developing.

5.2 Feature Selection Analysis

Recall from Sec. 3.3 that we have used the top ten features in training the predictors, with the specific subset determined from correlation analysis on the full feature set $\mathcal{F}$. Here, we investigate the impact of the number of the features selected on the results; while adding more features will increase the information available for training, it also reduces efficiency and makes the models more prone to overfitting. We train each algorithm over features $\mathbf{A}(7)$ from the LS dataset after the first week, since LS has the most users and most balanced Pass/Fail outcomes.

Fig. 11 shows the results. The values shown in the plots at feature f is using the f-most correlated features from $\mathcal{F}$ for training. We make the following observations:

Beyond $f \approx 10$, performance tends to vary more substantially. For LDA, SVM, and XGB, the metrics vary significantly (drop in quality with few exceptions) when the number of features increases beyond 10. This suggests overfitting, which is important to avoid in the our prediction.



(a) VE Features Correlation (b) ES Features Correlation (c) LS Features Correlation

Fig. 10: Variation in feature correlation by day for the top 5 correlated features for each course, corresponding to Table 3 ("Sim_UX" stands for "post similarity to unit X"). Correlation gradually increases through days $10-11$ in each case, where it stabilizes.

RF performance is reasonably independent of f. In Fig. 11(a), the metrics do not change substantially, especially compared to the variations in the other algorithms. This may be explained by the fact that RF works with information gain, which already considers a form of feature selection.

ANN performance remains constant for $f \in [5, 14]$. The metrics, in particular accuracy and Type II error, remain consistent regardless of the number of features in this range. This suggests that the activation function to the hidden layer of the ANN may be able to filter out additional information, similar to the information gain in RF. Nonetheless, when additional features are added to the network, the increasing dimension (*i.e.,* the increasing number of hyper-parameters to train) results in lower performance, similar to what is seen for in LDA and XGB.

In selecting the number of features, we aim to optimize quality and stability in generalizing to new data. Therefore, in hindsight, we conclude that 10 was a reasonable choice.

6 Related Work

This work is generally related to data mining and machine learning for education, which has been increasingly studied over the past decade (see [23], [24] for surveys). In this section, we discuss prior research on learning outcome prediction (Sec. 6.1), and specific works on content clickstream (Sec. 6.2) and social learning (Sec. 6.3) feature mining.

6.1 Learning Outcome Prediction

Predictive learning analytics methods have been developed to forecast different attributes of learners in advance, mainly for students in higher education and MOOC courses (see [25] for a survey). These include how learners will perform on assessments [7], [22], [26], [27], learners' risk of obtaining adverse outcomes [28], [29], learners' final grades [8], [12], [30], [31], and learner attrition [6], [10], [12]. The latter was also the focus of KDD Cup 2015 [10].

Different from these works, we focus on predicting binary (pass/fail) outcomes in short-courses, a type of course characterized by short timescales and a lack of intermediate quiz/assessment data. In the absence of assessment measurements, prior quiz-based outcome prediction models are not directly applicable; instead, our method turns learners' content clickstreams and social learning data into features for binary outcome classification. Additionally, the content file types we consider in this work – interactive slideshows, articles, and PDFs – are common in corporate training but different from standard lecture video formats, and our system enables collection of data on these file types too.

6.2 Content Feature Mining

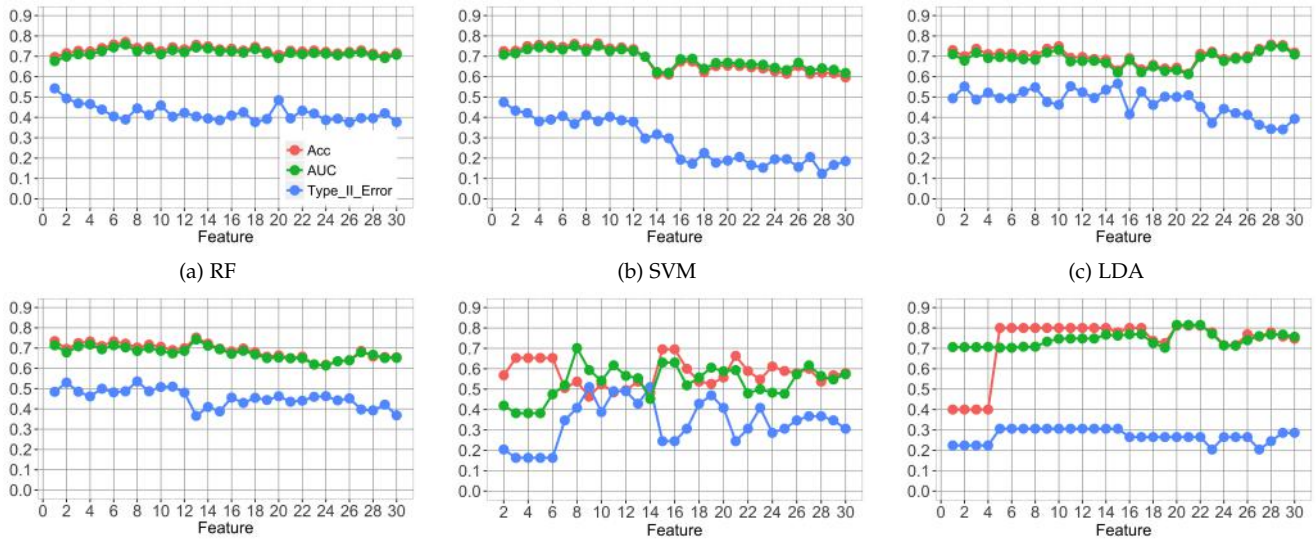
Researchers have mined behavioral features from student clickstream data. Some methods have focused on exploratory analysis of clickstreams [32], [33], [34]; [34] modeled raw numbers of clicks by students each day, while [33] used Markov modeling to learn transitions between learning activities. We are instead interested in developing features for outcome prediction. In this regard, some works have focused specifically on using video-watching behavior for grade prediction in MOOCs [2], [6], [7]; [7] defined aggregate quantities like fraction of time spent and number of rewinds, while [2], [6] searched for recurring subsequences of click actions in student behavior. Others have used clickstream features across multiple content types for outcome prediction [10], [28]; [10] mined learner activities across course content, forums, and wikis for drop-off and quiz prediction, while [28] employed mixture models to group students based on time spent for predicting certification.

Our work is perhaps most similar to [7], [28] in regards to using clickstream data for early detection. The short timescales of our courses pose additional modeling challenges that we overcome by defining features for each piece of content separately, and performing day-by-day predictions to capture learners re-visiting content, rather than the unit-by-unit scheme proposed in [7]. Further, while training an SVM on behavioral features was seen to work well in [7], it obtained low quality on our datasets. In [28], the authors also study early detection for a binary outcome, but we define several features beyond time spent (*e.g.,* engagement) that we find more predictive in short-courses.

6.3 Social Learning Features

Several recent studies have considered the Social Learning Networks (SLN) in different learning scenarios, *e.g.,* MOOCs [9], [12], [13], online courses [31], [35], [36], Q&A sites [4], and enterprise social networks [37]. Similar to the discussion forums in online courses [9], SLN emerge on Q&A sites through users asking and answering questions, so analysis methods naturally overlap between these scenarios.

Some of these prior works have focused on the SLN itself, such as exploratory analysis [4], [37] and optimization of interactions [9]. We are focused instead on using features



(a) RF (b) SVM (c) LDA

(d) KNN (e) XGB (f) ANN

Fig. 11: Effect of the number of features selected on the prediction quality of each algorithm at the critical halfway point $\mathbf{A}(7)$, for dataset LS. In hindsight, this shows that $|\mathcal{F}| = 10$ was a reasonable choice for feature selection in Sec. 3.3.

that can be extracted from the SLN for outcome prediction, similar to [12], [13], [31]. Considering these in particular, [12] built a probabilistic graphical model to predict grades and completion from learner post and reply frequencies, [31] built predictors of final performance from participation indicators in both quantitative, qualitative and social network forums, and [13] predicted whether instructor intervention will be needed from semantics and explicit references to course files. In addition to structural SLN attributes, our method defines other types of features not present in these prior works. These include the topic similarity between learner posts and the course content, as well as the content features measuring how learners interact with the course material. In our short-course scenarios, we find that the topic similarity features are particularly predictive, and that the content features are useful for earliest detection.

7 Conclusion and Future Work

We developed a methodology for predicting learning outcomes from learner behavior in online short-courses. The lack of intermediate assessments coupled with the small enrollments in this type of course makes the development of predictive learning analytics particularly challenging. Our method relies solely on behavior-based machine learning features obtained by processing measurements collected during the learning process, including a learner's interaction with the content and with one another in Social Learning Networks. Evaluating on data collected from three short-courses and using models such as gradient boosting that can work under sparse conditions, we obtained high prediction quality by the middle stages of the courses, underscoring the capability of our method to provide early detection to instructors. We also observed that SLN attributes became the more useful set of behaviors for prediction over time, while the content attributes provided better quality for "earliest" detection in the first few days. Further, we found that our method can generate behavioral analytics for instructors.

Future work may investigate other content and SLN features, as well as other classifiers to further enhance performance. We will also incorporate these methods into our Dashboard, so that instructors can access the predictions in an online manner during future course sessions. This will allow us to collect feedback, and to measure changes in pass rates resulting from interventions made based on the predictions and analytics – the ultimate measure of efficacy.

References

[1] W. Chen, C. G. Brinton, M. Chiang, and D. Cao, "Behavior in Social Learning Networks: Early Detection for Online Short-Courses," in *IEEE INFOCOM*, 2017.

[2] C. G. Brinton, S. Buccapatnam, M. Chiang, and H. V. Poor, "Mining MOOC Clickstreams: Video-Watching Behavior vs. In-Video Quiz Performance," *IEEE TSP.*, vol. 64, no. 14, pp. 3677–3692, 2016.

[3] S. Kimmel. (2015) Training Evolution: The Current and Future State of Corporate Learning Modalities. http://www.workforce.com/articles.

[4] C. G. Brinton and M. Chiang, "Social Learning Networks: A Brief Survey," in *IEEE CISS*, 2014, pp. 1–6.

[5] J. Bersin. (2016) A Bold New World of Talent, Learning, Leadership, and HR Technology Ahead. http://marketing.bersin.com.

[6] T. Sinha, P. Jermann, N. Li, and P. Dillenbourg, "Your Click Decides Your Fate," in *ACL EMNLP*, 2014, pp. 3–14.

[7] C. G. Brinton and M. Chiang, "MOOC Performance Prediction via Clickstream Data and Social Learning Networks," in *IEEE INFOCOM*, 2015, pp. 2299–2307.

[8] Y. Meier, J. Xu, O. Atan, and M. van der Schaar, "Predicting Grades," *IEEE Trans. Signal Proc.*, vol. 64, no. 4, pp. 959–972, 2016.

[9] C. G. Brinton, S. Buccapatnam, F. M. F. Wong, M. Chiang, and H. V. Poor, "Social Learning Networks: Efficiency Optimization for MOOC Forums," in *IEEE INFOCOM*, 2016.

[10] J.-Y. Lee, A. Toescher, M. Jahrer, K. Ozaki, M. Bay, P. Yan, S. Chen, T. T. Nguyen, and X. Zhou, "Multi-stage ensemble and feature engineering for mooc dropout prediction," June 2016. [Online]. Available: http://www.conversionlogic.com/

[11] GP and Training Industry. (2010) Delivering Virtual Instructor-Led Training. http://www.salt.org/.

[12] J. Qiu, J. Tang, T. X. Liu, J. Gong, C. Zhang, Q. Zhang, and Y. Xue, "Modeling and Predicting Learning Behavior in MOOCs," in *ACM WSDM*, 2016, pp. 93–102.

[13] M. Kumar, M.-Y. Kan, B. C. Tan, and K. Ragupathi, "Learning Instructor Intervention from MOOC Forums: Early Results and Issues," *ERIC EDM*, pp. 218–225, 2015.

[14] T. Bell, "Extensive reading: Speed and comprehension," *The reading matrix*, vol. 1, no. 1, 2001.

[15] Y.-W. Chen and C.-J. Lin, "Combining SVMs with Various Feature Selection Strategies," in *Feature Extraction*. Springer, 2006, pp. 315–324.

[16] A. Toscher and M. Jahrer, "Collaborative Filtering Applied to Educational Data Mining," *KDD Cup*, 2010.

[17] L. V. Morris, S.-S. Wu, and C. L. Finnegan, "Predicting Retention in Online General Education Courses," *American Journal of Distance Education*, vol. 19, no. 1, pp. 23–36, 2005.

[18] S. Guo and W. Wu, "Modeling student learning outcomes in moocs," in *The 4th ICTALE*, 2015, pp. 1305–1313.

[19] Z. A. Pardos and N. T. Heffernan, "Using HMMs and Bagged Decision Trees to Leverage Rich Features of User and Skill from an Intelligent Tutoring System Dataset," *JMLR*, 2011.

[20] D. S. Chaplot, E. Rhim, and J. Kim, "Predicting student attrition in moocs using sentiment analysis and neural networks." in *AIED Workshops*, 2015.

[21] J. Liang, C. Li, and L. Zheng, "Machine learning application in moocs: Dropout prediction," in *ICCSE*. IEEE, 2016, pp. 52–57.

[22] T.-Y. Yang, C. G. Brinton, C. Joe-Wong, and M. Chiang, "Behavior-based grade prediction for moocs via time series neural networks," *IEEE JSTSP*, 2017.

[23] A. Peña-Ayala, "Educational data mining: A survey and a data mining-based analysis of recent works," *Expert systems with applications*, vol. 41, no. 4, pp. 1432–1462, 2014.

[24] R. S. Baker and P. S. Inventado, "Educational data mining and learning analytics," in *Learning analytics*. Springer, 2014, pp. 61–75.

[25] W. Hämäläinen and M. Vinni, "Classifiers for educational data mining," *Handbook of Educational Data Mining*, pp. 57–74, 2010.

[26] A. S. Lan, C. Studer, and R. G. Baraniuk, "Time-varying learning and content analytics via sparse factor analysis," in *SIGKDD*. ACM, 2014, pp. 452–461.

[27] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein, "Deep Knowledge Tracing," in *NIPS*, 2015, pp. 505–513.

[28] C. A. Coleman, D. T. Seaton, and I. Chuang, "Probabilistic use cases: Discovering behavioral patterns for predicting certification," in *L@S*. ACM, 2015, pp. 141–148.

[29] H. Lakkaraju, E. Aguiar, C. Shan, D. Miller, N. Bhanpuri, R. Ghani, and K. Addison, "A Machine Learning Framework to Identify Students at Risk of Adverse Academic Outcomes," in *ACM SIGKDD*, 2015, pp. 1909–1918.

[30] B. Minaei-Bidgoli, D. A. Kashy, G. Kortemeyer, and W. Punch, "Predicting student performance: an application of data mining methods with an educational web-based system," in *Frontiers in education, 2003.*, vol. 1. IEEE, 2003, pp. T2A–13.

[31] C. Romero, M. I. López, J. M. Luna, and S. Ventura, "Predicting Students' Final Performance from Participation in Online Discussion Forums," *Computers & Education*, vol. 68, pp. 458–472, 2013.

[32] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec, "Engaging With Massive Online Courses," in *WWW 2014*. ACM, 2014, pp. 687–698.

[33] C. Geigle and C. Zhai, "Modeling mooc student behavior with two-layer hidden markov models," in *L@S*. ACM, 2017, pp. 205–208.

[34] J. Park, K. Denaro, F. Rodriguez, P. Smyth, and M. Warschauer, "Detecting changes in student behavior from clickstream data." in *LAK*, 2017, pp. 21–30.

[35] H.-L. Yang and J.-H. Tang, "Effects of social network on students' performance: a web-based forum study in taiwan," *Journal of Asynchronous Learning Networks*, vol. 7, no. 3, pp. 93–107, 2003.

[36] M. Abdous, H. Wu, and C.-J. Yen, "Using data mining for predicting relationships between online question theme and final grade," *Journal of Educational Technology & Society*, vol. 15, no. 3, p. 77, 2012.

[37] J. Cao, H. Gao, L. E. Li, and B. Friedman, "Enterprise Social Network Analysis and Modeling: A Tale of Two Graphs," in *IEEE INFOCOM*, 2013, pp. 2382–2390.

Weiyu Chen (M'17) is a Lead Data Scientist at Zoomi Inc, a technology startup that delivers learning insight platform based on machine learning. At Zoomi, Weiyu is working on utilizing data mining and analytics to optimize e-learning experiences. Weiyu researches learning prediction algorithms, deep learning, real-time analytics, and NLP methods for understanding learner behaviors, identifying learners at risk, and assisting personalized e-learning. Weiyu earned her B.A. in mathematics from Vanderbilt University and her M.S.E. in Applied Mathematics at Johns Hopkins University.

Christopher G. Brinton (S'08, M'16) is the Head of Advanced Research at Zoomi Inc, a learning technology company he co-founded in 2013, and a Lecturer in Electrical Engineering at Princeton University. His research focus is developing systems and methods to improve the quality of student learning, through predictive learning analytics, social learning networks, and individualization. Chris co-authored the book *The Power of Networks: Six Principles that Connect our Lives*., and has reached over 250,000 students through MOOCs based on his book. A recipient of the 2016 Bede Liu Best Dissertation Award in Electrical Engineering, Chris received his PhD from Princeton in 2016, his Master's from Princeton in 2013, and his BSEE from The College of New Jersey (valedictorian and summa cum laude) in 2011, all in Electrical Engineering.

Da Cao is a Research Algorithm Engineer at Zoomi Inc, an innovative learning technology company. Da's work involves software development, data processing, feature extraction and exploring machine learning algorithms for behavioral analysis. He received his Bachelor of Science in Electrical Engineering from University of Wisconsin - Madison and his M.S. in Electrical Engineering from University of Pennsylvania.

Amanda Mason-Singh is a Lead Data Scientist at Zoomi Inc, an innovative learning technology company. Amanda's research focuses on the socialization of academic achievement motivation and using data to optimize student learning throughout the lifespan. Amanda earned her B.A. in psychology and sociology from Iowa Wesleyan College (summa cum laude), her M.S. in developmental psychology from Illinois State University, and her Ph.D. in human development and quantitative methodology from the University of Maryland, College Park.

Charlton Lu is a research intern at Zoomi Inc, a technology startup that delivers learning insight platform based on machine learning intelligence. His work involves research on optimizing different types of neural network classifiers to predict learning performance. Charlton is an undergraduate student at Duke University, pursuing degrees in Mathematics and Computer Science.

Mung Chiang (S'00, M'03, SM'08, F'12) is the John A. Edwardson Dean of the College of Engineering at Purdue University, West Lafayette, IN. Previously he was the Arthur LeGrand Doty Professor of Electrical Engineering at Princeton University, Princeton, NJ. His research on communication networks received the 2013 Alan T. Waterman Award from the U.S. National Science Foundation, the 2012 Kiyo Tomiyasu Award from IEEE, and various young investigator awards and paper prizes. He is the Chairman of the Princeton Entrepreneurship Advisory Committee and the Director of the Keller Center for Innovations in Engineering Education. His MOOC in social and technological networks reached about 200,000 students since 2012 and lead to two undergraduate textbooks and he received the 2013 Frederick E. Terman Award from the American Society of Engineering Education. He was named a Guggenheim Fellow in 2014.