A Machine Learning Approach for Identifying At-Risk Students in Learning Record Stores: A Case Study Using USALearning Experience API (xAPI)

Paul Jesukiewicz
Office of Personnel Management (OPM) USALearning
Washington, DC
paul.jesukiewicz@opm.gov

James Bilitski, Ph.D.
University of Pittsburgh
Johnstown, PA
bilitski@pitt.edu

Rob Chadwick
Veracity Technology Consultants, LLC
Gaithersburg, MD
rob@veracity.it

John DeCore
PowerTrain, Inc
Landover, MD
JDeCore@powertrain.com

Jonathan Poltrack
Veracity Technology Consultants, LLC
Johnstown, PA
jono@veracity.it

ABSTRACT

The use of machine learning technology in training and education has been increasing over the past few years, with the goal of enhancing the personalized learning experience of students. To meet this goal, there is a need for innovative solutions that can adapt to the changing needs of the education sector. One such solution is the Experience API (xAPI), which is a technology standard for tracking and storing learning data.

OPM USALearning designed a project to use machine learning (ML) in conjunction with significant amounts of xAPI data tracked to a learning record store (LRS). The goal of this research and development project is to explore the potential of ML to improve learning outcomes through the use of xAPI data by early prediction of at-risk students. This study aims to evaluate the feasibility of using machine learning algorithms to analyze xAPI data and provide personalized learning experiences for students.

This study constructed a model to assess student risk by analyzing the ratio between student engagement and assessment failures. Furthermore, a Long Short-Term Memory (LSTM) neural network for predictive analytics was used to analyze chronological student interactions provided by xAPI data. The results have been promising, with the LSTM model achieving 78% accuracy in determining student outcomes. This framework not only advances the identification and intervention measures for at-risk students but also equips instructors and administrators with the tools to offer timely, data-informed interventions, potentially increasing student success rates.

This paper presents the results of the analysis of a large xAPI dataset of approximately 18.8 million data points (xAPI statements). This paper describes the accuracy of the predictions, at-risk student identification, and the validity of recommended content. These features are critical in improving learning outcomes with the use of artificial intelligence in an xAPI-enabled environment.

ABOUT THE AUTHORS

Paul Jesukiewicz is a leader in the field of learning technologies with over 35 years of experience working in government, industry, and academia. He successfully led research, development, and implementation of a global program on advanced distributed learning. He was inducted into the Federal Government Distance Learning Association (FGDLA) Hall of Fame in 2012 as recognition for his significant career accomplishments in promoting and developing distance learning in the Federal Government.

James Bilitski, Ph.D. is an associate professor of computer science at the University of Pittsburgh at Johnstown and an industry consultant. He has worked for companies such as Motorola, Phillips, Ansaldo, Concurrent Technologies, Clair Global, and Problem Solutions. Dr. Bilitski has led technical projects in commercial and government spaces. He specializes in machine learning, artificial intelligence, education, real time systems, and audio and music software. He has several peer reviewed publications in the areas of machine learning.

Rob Chadwick, partner at Veracity Technology Consultants, is a web developer and data scientist who has been working in the digital media field for more than 15 years. In that time, he designed custom game engines, managed render farms, and written productivity tools. He has designed simulations for the FBI and the U.S. Navy. Rob worked 7 years with Advanced Distributed Learning (ADL) to foster interoperability of 3D assets, HTML5, and to bring immersive simulation to the web. His work at ADL includes standardization of the xAPI, automated LRS conformance testing, and development of various online digital content registries and repositories. Rob continues to leverage his development experience at Veracity, where he specializes in scalability of learning data analytics.

John DeCore has been developing and delivering training for 34 years. He began his career with the American Red Cross, has been a certified project management professional (PMP) since 2005, and joined PowerTrain, Inc. in 2006. In 2009, John earned a Master's of Adult Education and began building PowerTrain's Cloud Services focused on the use of open-source software to meet various federal government talent management challenges. In 2019, John was named Vice President of PowerTrain.

Jonathan Poltrack has worked in the learning industry for over 20 years, with large periods of time at the DoD's ADL Initiative. At the ADL Initiative, Jonathan was an early contributor to the Sharable Content Object Reference Model (SCORM), which became a de facto global e-learning specification. Later at ADL, Jonathan began leading efforts aimed at transitioning SCORM while specifying a new learning platform based on modern technologies and software architectures including the xAPI. Jonathan co-founded Veracity Technology Consultants, a company that focuses on standards-based learning technology services and products, with several learning technology expert partners. Jonathan is passionate about education, training, and performance support and their intersections with technology.

A Machine Learning Approach for Identifying At-Risk Students in Learning Record Stores: A Case Study Using USALearning Experience API (xAPI)

Paul Jesukiewicz
Office of Personnel Management (OPM) USALearning
Washington, DC
paul.jesukiewicz@opm.gov

James Bilitski, Ph.D.
University of Pittsburgh
Johnstown, PA
bilitski@pitt.edu

Rob Chadwick
Veracity Technology Consultants, LLC
Gaithersburg, MD
rob@veracity.it

John DeCore
PowerTrain, Inc
Landover, MD
JDeCore@powertrain.com

Jonathan Poltrack
Veracity Technology Consultants, LLC
Johnstown, PA
jono@veracity.it

INTRODUCTION

As online and technology-assisted education and training ecosystems grow in numbers of students and types of content and hardware devices supported, it is increasingly more important to have the means to manage a large student population. Some industry and government systems include hundreds of e-learning courses without a dedicated facilitator or instructor. This increases the need for technology-assisted tools for the management of students moving through curricula in online learning platforms.

The primary objective of this project is to 1) establish a model for measuring risk of students, and 2) apply machine learning (ML) techniques to determine a method to identify at-risk students to discover student activity patterns that lead to passing and failing assessments. Such early detection mechanisms can facilitate timely interventions. Identifying and addressing the needs of potential at-risk students early on can increase students' chances of success (Pek et al., 2023). The culmination of this project will be a suite of tools within the USALearning (USAL) Ecosystem. These tools will enable instructors, mentors, and administrators to retrieve a list of potential at-risk students, gauge their likelihood of failing assessments, and understand the factors placing them on the at-risk spectrum. Consequently, a focused group of administrative users will be better equipped to assist the most vulnerable students, leveraging the insights provided by ML.

USALearning Ecosystem

The Office of Personnel Management (OPM) USAL Program is an ideal location for research into the identification and intervention of at-risk students. USAL provides learning products and services to hundreds of US government clients and serves over ten million end users. Many USAL systems are standards-based so data formats are consistent and broadly adopted. USAL provides Learning and Talent Management Systems (LMS/TMS), Learning Record Stores (LRS), Learning Content Management Systems (LCMS), Student Information Systems (SIS), and more. Most of the training is professional training for government agencies including large amounts of mandatory online training courses.

USAL began xAPI LRS efforts in 2017 and soon after began to store large amounts of learners' experiential data in USAL LRS instances. xAPI data is granular information about a learner's interaction with learning content. This can include events like a learner completing a course, a learner answering a question on an assessment, an instructor evaluating a learner's performance, or a user interacting with a virtual reality (VR) experience. xAPI enables a large set of data tracking requirements not available in previous learning standard technologies. Since there is a significant amount of xAPI data, the data is accessible, and the data represents many kinds of training content and associated interactions, this data was chosen as the source for this project.

The data selected is a real USAL learning dataset, collected from September 1, 2019 to August 31, 2022, that was preprocessed prior to use in this project. The data includes over 18 million xAPI statements about learner's interactions in online courses, virtual instructor led training (VILT) courses, and online assessments. Although we cannot release specific course or content titles, the course subjects span a broad set of government professional training topics including job-specific instruction, information systems, cybersecurity, trafficking in persons awareness, ethics, and more. The assessments include a range of content including inline knowledge checks, pre-tests, and post-tests. General interactions with online system components, represented as "views" in the data, are also included in the xAPI data set. Learner information and any associated sensitive data were anonymized during the preprocessing of the data prior to our analysis and machine learning processes. Data was then transformed into a tabular format that can be used by ML tools.

The remainder of this paper includes the project methodology and results to implementing early detection of at-risk students.

LITERATURE REVIEW

ML has been used for prediction in a wide variety of fields. One such field is the prediction of student risk, exam scores, and early identification of unsuccessful students. The identification of at-risk students and providing intervention has gained attention among the research community (Pek, et. al., 2023; Al Breiki, et. al., 2019, Chui, et. al., 2019; Er, et. al., 2012; Jang, 2022; Lakkaraju, 2015; Livieris, 2018; Macarini, 2019; Pilotti, et. al., 2022). There are a variety of ML prediction algorithms including Deep Networks, Decision Trees, Random Forests, Regressions, Naïve Bayes, Support Vector Machines, and dozens more. ML techniques are important in extracting information and knowledge for data sets. For example, Adnan (2021) developed a system to predict at-risk students by analyzing performance during different modules in a course. Pek (2022) performed a study to determine if initial student performance is informative in the early detection of at-risk students.

The process of identifying at-risk students hinges on the provision of customized interventions. While LMS, LRS, and BI tools can generate insightful reports, they often fall short in detecting and addressing the specific needs of at-risk students. Most contemporary LMS, LRS, and BI tools produce descriptive statistics, which are not ideal for the automated detection of students at-risk.

When the inherent capabilities of e-learning tools are not enough, external ML predictive tools step in. Recent research has shifted towards using ML techniques for predicting academic outcomes. Diving deeper into the recent studies, there is a gap in leveraging LRS data for student risk prediction. Much of the current literature on risk detection focuses on specific LRS data points, such as GPA and test scores, which might not align with the operational needs of industry and government e-learning systems that use PASS/FAIL metrics. Kondo et al. (2017 & 2018) employed various ML strategies to predict GPA and identify at-risk students. Meanwhile, Aljaloud et al. (2022) utilized a convolutional neural network (CNN) and an LSTM to predict learning outcomes within the Blackboard LMS. Though traditional metrics like grades may not be included in all xAPI datasets, LRS data typically provides a holistic view of student interactions. In this context, USAL emphasizes xAPI data, showcasing its event-driven nature over a stateful data model.

We present a novel approach that uses xAPI data to create a risk measurement model, further integrating an LSTM for predicting student outcomes. Our initial findings validate this method, with the LSTM model registering a notable 78% accuracy in predicting student outcomes using a historical sequence of 100 student xAPI verbs.

DATA ANALYSIS

Most ML and data science libraries do not support ingesting JavaScript Object Notation (JSON) data, the data interchange format used by xAPI. The xAPI JSON standardized data format was converted into flat comma-separated value (CSV) files that were supported by data science Python libraries. Our team pulled a subset of the existing data to leave additional data sets for later validation of the ML predictions. In summary, the data set used as part of this study:

- included over **18 million xAPI statements**,
- used about **65GB of storage**,
- was **collected from September 1, 2019 to August 31, 2022**, and
- included data tracked from **130,561 students**.

In xAPI, events that represent a user's typical student's interaction with learning content, are tracked using a data structure called a *"statement"*. A *statement* can include different types of information about the event, but at a minimum must include an actor (typically the student), a verb (the event that was tracked), and an object (typically the activity that was interacted with by the actor) (*Experience API (xAPI) Specification*, 2014). The verb in xAPI provides context about the actor's interaction with the object. For example, a *scored* verb indicates that an event occurred resulting in the actor getting a score, like the completion of a quiz with a scaled score of 0.85.

The dataset used in the study consists of xAPI verbs representing a student's interaction with an LMS including:

- *viewed* – the student viewed a course, assessment (quiz), or other resource or activity in the system.
- *responded* – the student answered a question on an assessment or feedback survey.
- *completed* – the student completed a course, module, SCO, or other content.
- *initialized* – the student launched a Sharable Content Object (SCO) in a Sharable Content Object Reference Model (SCORM) package.
- *logged into* – the student logged in to the LMS.
- *scored* – the user received a score on an assessment, quiz, SCORM package, or other scored content.
- *terminated* – the user completed an attempt on a SCO in a SCORM package.
- *passed* – the user passed the objective of a SCORM package, course, assessment, or other content.
- *received* – the user received a grade from an instructor.
- *failed* – the user received a failing score on a SCORM package or assessment.

The data in this form describes only raw student activity and lacks data characteristics that directly identify a student as being at-risk. There are numerous data attributes that can help identify an at-risk student such as poor grades, lack of participation, lack of completion of course, etc. The data used in this study had to be broadly applicable to all the different types of content (e.g., instructor-led, SCORM packages, etc.) in the learning environment. Some verbs evaluated were selected because they are generally applicable to all types of content in the environment. For example, Moodle courses that contain other learning activities, SCORM packages built in external authoring tools, and LMS-built assessments all include *viewed* verbs and the associated event. However, other verbs, like *terminated*, are not used beyond SCORM packages so their utility was limited in our study.

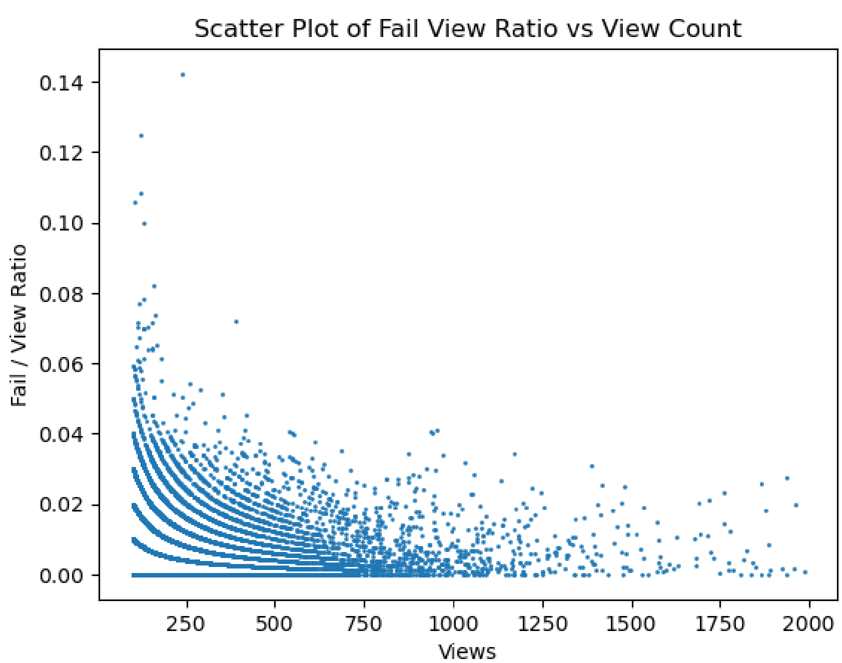

Figure 1. All students fail / view ratio vs number of views per each student.

A Model for Measuring Risk Level

The risk model in this study considered the verbs *viewed* and *failed*. The *viewed* verb indicated that a student is interacting with the system such as viewing a content item in a course. It is essentially a measure of overall usage of the learning environment. The *failed* verb indicates that a student attempted an assessment and *failed*. It is evident that a student should not have an excessive number of failures respective to the number of views. A student with excessive amounts of failed events is considered to be at-risk. An analysis was conducted observing the number of fails and views of each student. To normalize data relative to student total activity within the system, the ratio between fails and views was calculated for each student. Figure 1 shows ratio of fails / views with respect to the total number of views. The densely clustered points show a decaying exponential pattern.

$$f(x) = a \cdot e^{(-b \cdot x)} + c \qquad (1)$$

$$a = \frac{y_1 - y_2}{e^{-b \cdot x_1} - e^{-b \cdot x_2}} \qquad (2)$$

$$b = \frac{\ln(y_1 - y_2)}{x_2 - x_1} \qquad (3)$$

$$c = y_1 - a \cdot e^{-b \cdot x_1} \qquad (4)$$

It was observed that there were some students outside the clustered points indicating possible at-risk students because their fail/view ratio was high in comparison to the entire data set. Since there are no labels in the raw data that identify a student as at-risk, some unsupervised approaches were considered. Ideally, ML can establish a threshold separating at-risk students. Preliminary experimentation using K-Means clustering failed to separate students into reliable distinct groups, likely due to the exponential distribution of data. Several other techniques were considered but would face challenges with the nonlinear data. A simpler approach was used fitting a custom non-linear function representing the decay model. A decaying exponential function was developed so that it passes through the approximate points of (250, 0.8) and (1500, 0,1). It was observed that these points are approximately where the points start to become more densely clustered. Equation 1 shows the general exponential decay functions. Equations 2, 3, and 4 show the parameters used in Equation 1.

This exponential decay function was established to identify a threshold dividing at-risk students from those not at-risk. It was assumed that most students are not at-risk and thus a conservative value threshold function was chosen to include less than 1% of students in the at-risk group. Figure 2 shows the established threshold along with colored points. There were 248 students in the at-risk group representing about 0.2% of the population of students.

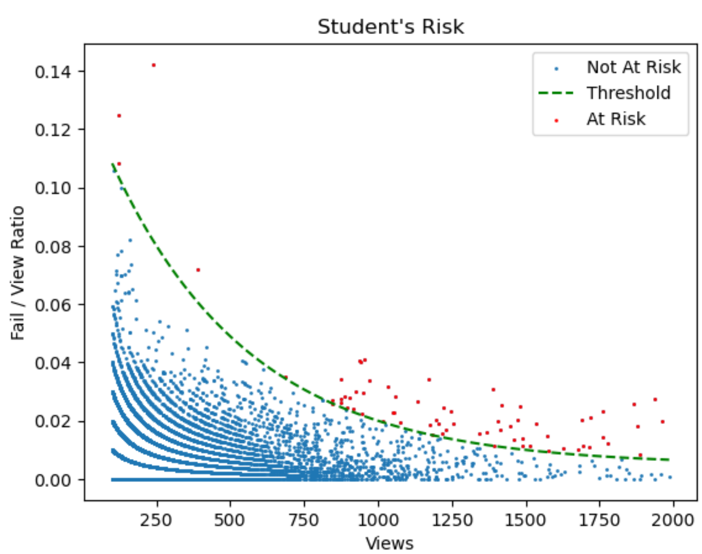

Figure 2. Threshold dividing at-risk and not at-risk students.

The red points above the threshold establish that a student is at-risk at a specific point in time. The two groups establish an identifying classification label of at-risk and not at-risk. Note that Figure 2 shows the at-risk students as of August 31, 2022 04:00Z which was the timestamp at the end of the study. Furthermore, the degree of at-risk students is established as the vertical distance from the threshold line. Those students significantly above the threshold line are potentially at higher risk. Similarly, those students far below the threshold are safely not at-risk. This establishment of categorical labels of at-risk students is beneficial to potentially identify the students at-risk. Observing a student after thousands of interactions with an LMS is informative but does not provide an early indicator for risk. Early indicators of students becoming at-risk can allow for intervention to improve student performance. To provide an early indicator, the risk levels were assessed continuously to analyze a student's journey of risk level as the system is used.

The risk level of each student was plotted showing the level of risk during a student's entire use of the system as of August 31, 2022. The amount of data available for each student varies. Some students who have been in the system for a while have several thousand xAPI statements, while others have much less. Each point in Figure 2 represents one single student at the point in time of August 31, 2022. However, the plot does not show changes to a student's risk level over time as they utilize the LMS, e-learning courses, virtual instructor-led courses, and assessments.

Figure 3 shows the plots of students who were classified into the at-risk category after the time period.

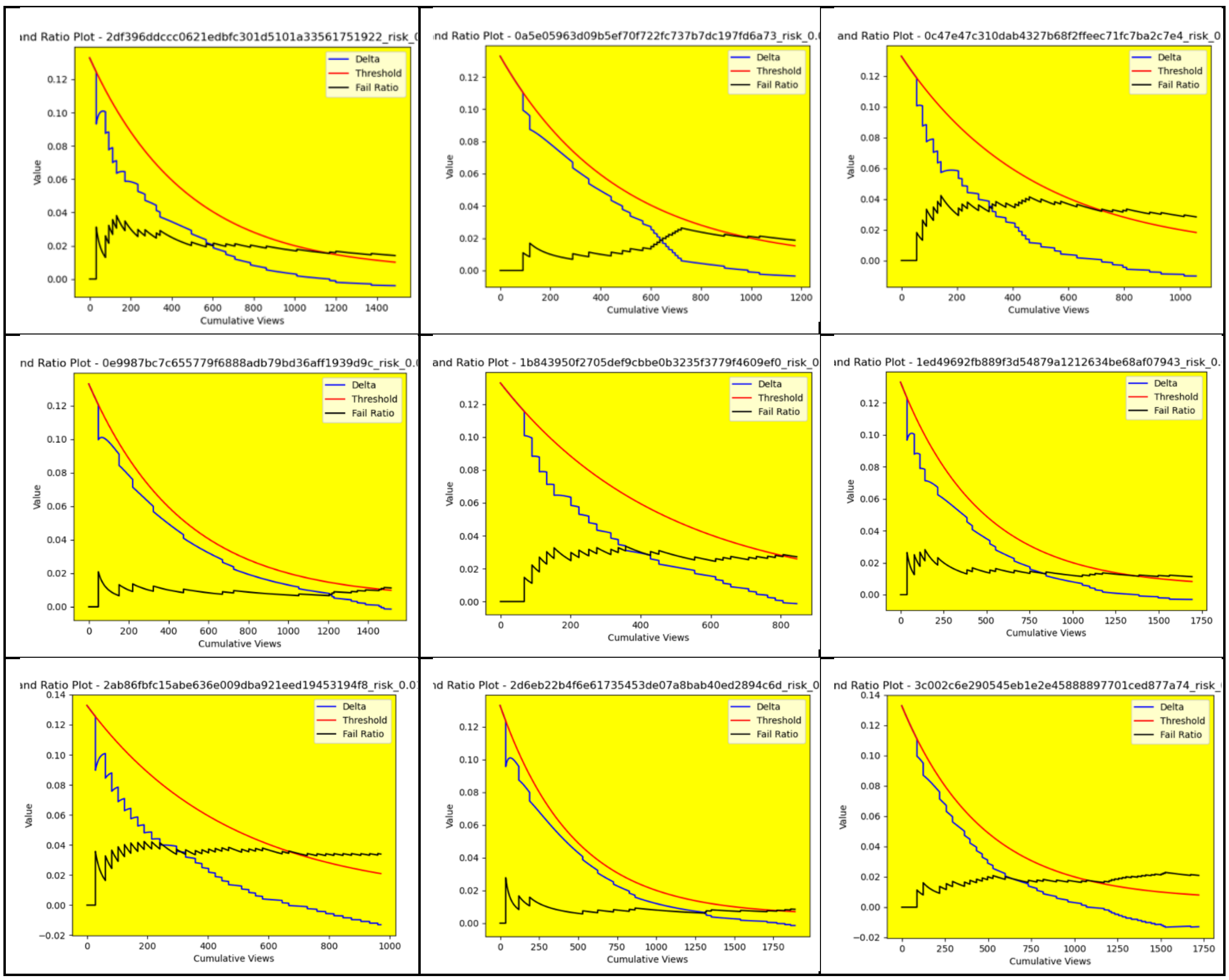

Figure 3. At Risk Students: Example of nine students who were in the at-risk category at the end of the time period.

The black line shows the fail/view ratio. The red line shows the threshold between at-risk and not at-risk. The blue line is the delta between fail ratio and the threshold. It was observed that the crossing of the black and blue lines resulted in a student eventually falling into the at-risk category. We refer to the crossing of the two lines as an indicator that a student is heading toward being at-risk. Similar indicators are used in time series to predict stock behavior and other time series data.

The students that were not at-risk were observed in a similar fashion. Figure 4 shows an example of nine students who were not at-risk.

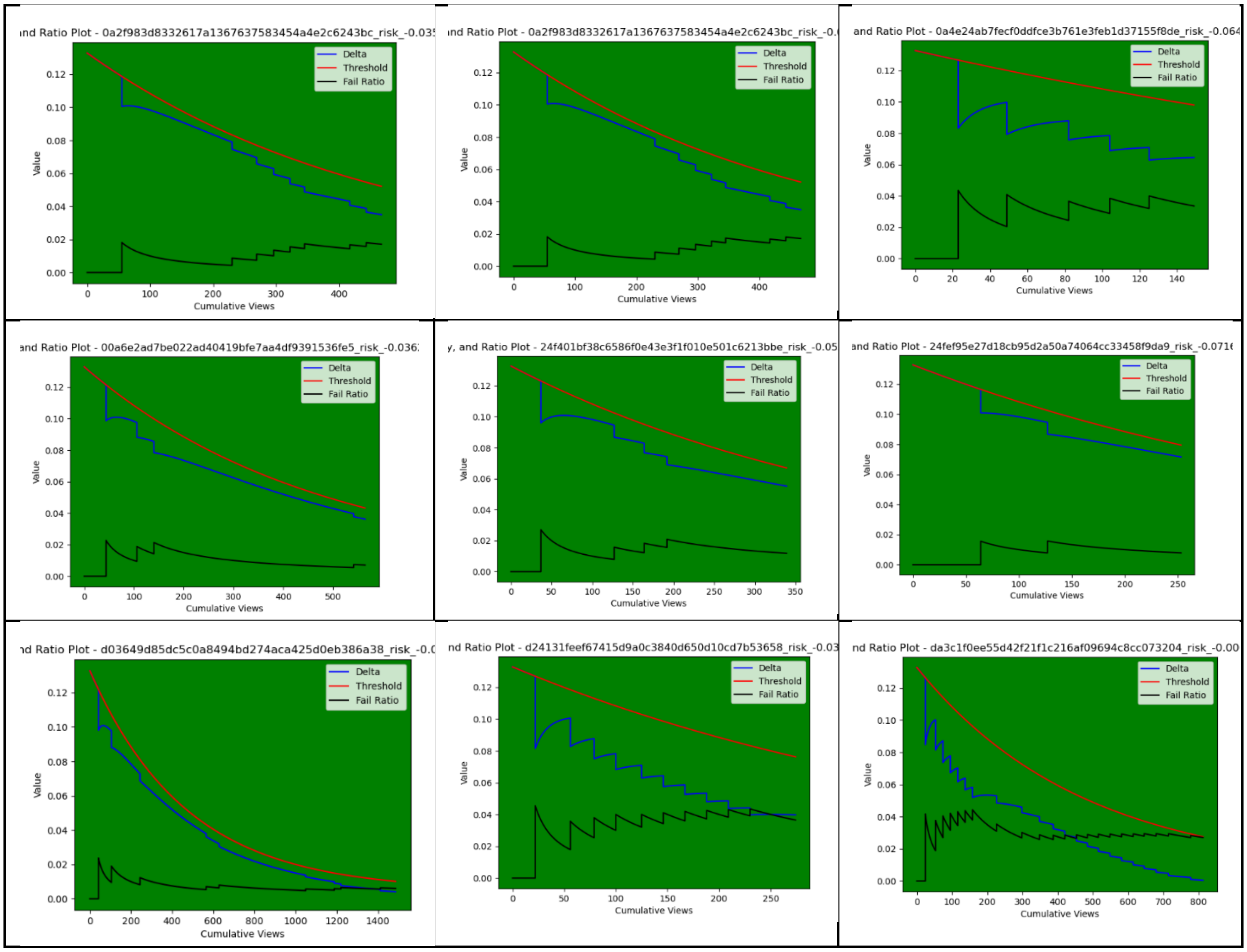

Figure 4. Not At-Risk Students: Example of nine students who were not in the at-risk category at the end of the time period.

In all nine examples, the student did not enter the at-risk category. However, it should be noted that some of the students did have an early indictor where the black and blue line crossed as shown in the bottom three graphs. Even though these students did not make it to the at-risk category, they could potentially be on their journey to being at-risk. This is particularly noticeable in the bottom right graph as the black and red line are practically ready to cross the threshold. In addition to students currently at-risk, these students that are not currently at-risk but are illustrating early indications of heading to an at-risk categorization are targets for additional study, and potentially, eventual intervention.

Summarizing the risk model, 130,561 students were analyzed. Any student that did not have any failures was ignored. Of those analyzed, 248 resulted in students entering the fully at-risk category at the end of the study time period. There were 1,033 positive indicators of at-risk where the black line crossed the blue line but did not end up at the fully at-risk state.

A MODEL FOR PREDICTING STUDENT OUTCOMES

The risk measurement model introduced enables the application of a risk metric to individual students. This model provides a chronological narrative of a student's risk level, highlighting potential early indicators of vulnerability. It serves as an analytical tool to monitor a student's trajectory towards being at-risk. Since failures often signify a student's elevated risk, early detection can enable timely interventions. Students undertake various course actions, such as logging in, viewing content, and completing course tasks, eventually resulting in either passing or failing an assessment. We investigate these action sequences to discern if specific patterns correlate with student failures. The LSTM deep network, with its proficiency in predicting outcomes from action sequences, is a fitting choice for this analysis. The student data we examined comprises xAPI entries paired with timestamps, offering a sequential record of student activities, thus making it particularly suitable for LSTM models.

LSTM models are a type of recurrent neural network (RNN) architecture that excels at learning and remembering long sequences of information. An LSTM is comprised of various gates and a cell state, as explained below:

The Forget Gate decides what information should be thrown away from the cell state. It uses a sigmoid activation function as shown in equation 5:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (5)$$

where W_f is the weight matrix for the forget gate, b_f is the bias, h_{t-1} is the previous hidden state, and x_t is the current input.

The Input Gate updates the cell state with new information. It uses two equations (6 and 7):

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (6)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (7)$$

where W_i and b_i are the weight and bias for the input gate, and W_C and b_C are the weight and bias for creating a vector of new candidate values, $\tilde{C}_t$, for the state.

The Cell State C_t is updated using the forget gate and the input gate (shown in equation 8):

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (8)$$

where C_{t-1} is the previous cell state.

The Output Gate decides the next hidden state h_t, using the updated cell state and the input shown in equation 9 and 10:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o \quad (9)$$
$$h_t = o_t \cdot \tanh(C_t) \quad (10)$$

where W_o and b_o are the weight and bias for the output gate. The hidden state h_t can be used for predictions, while the cell state C_t helps the LSTM to keep track of the underlying contextual information in the sequence.

LTSM can work on time series data. However, the online courses that generated the xAPI data do not have hard start dates and daily course activities. Rather, the courses can be started at any time by a student and are generally self-paced. Some students may start and finish a course within a single day while others may finish the course months after starting. Thus, the xAPI data time stamps are utilized in this study to keep the student events chronologically sorted.

However, since the delta time between events can be vastly different for students, the timestamps were only used to ensure student events were analyzed in chronological order ignoring the duration between events. The xAPI statement data was read in and grouped by students chronologically.

Sequences were established by extracting the data between Passed and Failed events. Each sequence was comprised of a series of chronological xAPI events such as {Registered, Viewed, Viewed, Viewed, Logged In, Responded, Viewed, Responded, etc.}. Each sequence was labeled as Passed or Failed since a sequence is a series between a Pass or Fail. There were 126,431 sequences extracted from the xAPI data. The average sequence length was 56.4 events per sequence. LSTMs require that sequences all be the same size. A sequence size of 100 was chosen as it covers the average sequence length of 56.4. Sequences shorter than 100 are padded with an ignored value. Sequences longer than 100 are truncated preserving the latest 100 events.

Each model had a masking layer to ensure that padding values were ignored by the model. The hidden LSTM layers varied in the experimentation, but the model's final LSTM layer had *return_sequences* disabled allowing previous layers to pass sequences to the next layer. Each model also had a dense layer as the final layer with a sigmoid activation. Experimentation occurred with various LSTM models and hyper parameters varying the batch size, epochs, and hidden layers. Table 1 shows the configurations of the models tested. Model 1 was conservatively designed and had a simple architecture with only two hidden layers. Models 2 and 3 had a slightly deeper network adding in another hidden layer. Model 2 used a larger batch size and less epochs in comparison to Model 3. Model 4 focused on discovering the effectiveness of a significantly deeper network. Finally, Model 5 investigated a longer training period with 40 epochs.

Table 1 LSTM Hyperparameters

	Batch Size	Epochs	Hidden Layers
Model 1	512	20	100,100
Model 2	256	20	50,100,100
Model 3	512	10	50,100,100
Model 4	256	20	50,50,50,50,50,50,50,50
Model 5	128	40	50,100,100

The data was shuffled and divided into a training set (80%) and a validation set (20%). The code was written using standard machine learning tools such as TensorFlow, Keras, SKLearn, Pandas, Numpy, and MatPlotLib. The models were trained using a v100 GPU with 32GB RAM and took between 60 to 300 minutes to train each model depending on the hyperparameters.

PREDICTION RESULTS

The training results from Models 1, 2, and 3, are shown in Figures 5, 6, and 7 respectively. In all three of these cases, the validation data accuracy stayed relatively close to the validation accuracy with some minor oscillations. These three models all had similar results with a training accuracy of about 78%. Figure 8 shows Model 4 used a deeper network compared to the first three models. Model 4 showed fast training early on, likely due to the extra hidden layers. Consequently, it showed signs of overfitting because the validation data line is starting to trend away from the training line. Figure 9 shows that Model 5 has even more over fitting as the validation line is trending away from the training line.

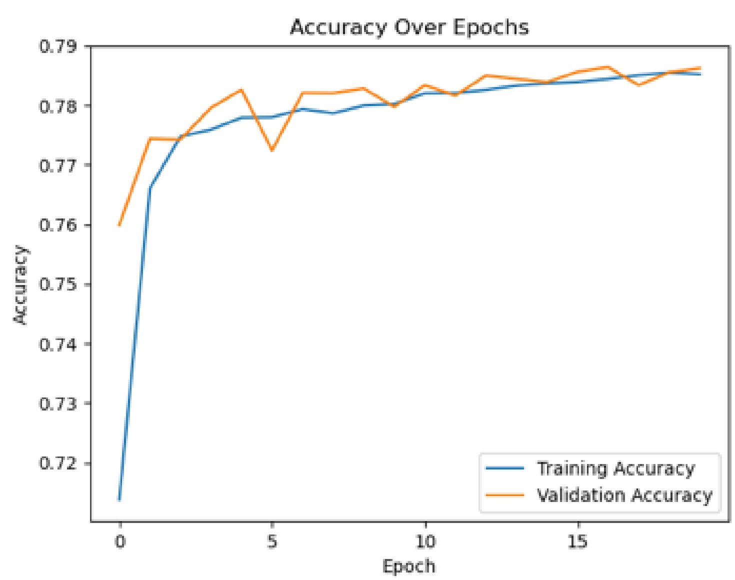

Figure 5. Model 1 (Batch Size: 512, Epochs: 20, Hidden Layer Sizes: 100,100).

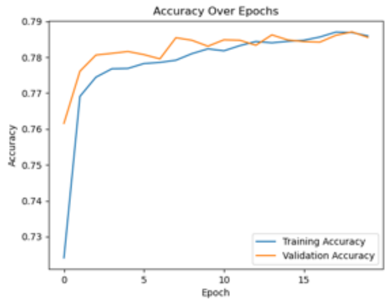

Figure 6: Model 2: (Batch Size: 256, Epochs: 20, Hidden Layer Sizes: 50, 100, 100).

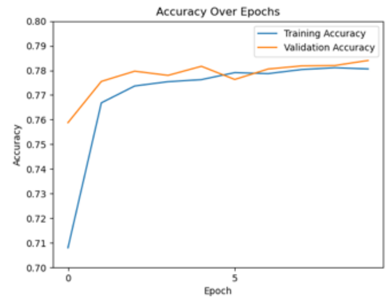

Figure 7: Model 3 (Batch Size: 512, Epochs: 10, Hidden Layer Sizes: 50, 100, 100).

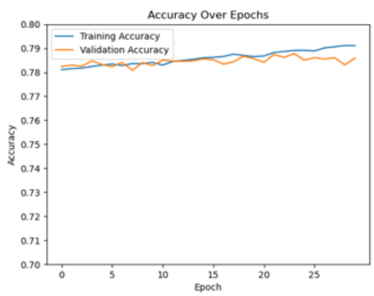

Figure 8: Model 4 (Batch Size: 256, Epochs: 30, Hidden Layer Sizes: 50, 50, 50, 50, 50, 50, 50, 50).

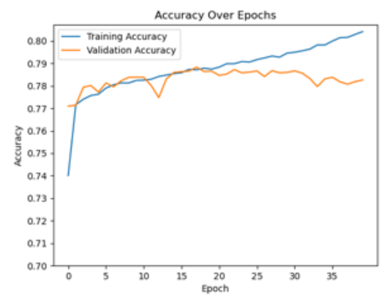

Figure 9: Model 5 (Batch Size: 128, Epochs: 40 Hidden Layer Sizes: 50, 100,100).

The LSTM models were generally able to be trained to around 78% accuracy. The smaller models (1 and 2) show the most potential as they seem to be more resistant to overfitting. After training completed, Model 1 was tested again for real world performance. Recall that 20% of the data was set as test data resulting in 25287 sequences. These sequences were predicted into Model 1 and the task completed predicting 25287 sequences in 24 seconds signifying that a trained model can be utilized in a production system without extensive hardware requirements.

DISCUSSION

Many LMS and LRS systems lack a metric to assess student risk and predict potential student failures. We presented 1) a novel methodology to model at-risk students using an exponential decaying function representing a threshold for at-risk students and 2) a method to predict assessment outcomes based on student activity. The at-risk model provides the current assessment of a student's risk level and quickly allows identification of students that are at-risk. The predictions help uncover which students are likely to fail their next assessment. With these indicators, intervention can take place for 1) at-risk students and 2) students who are likely to fail an assessment.

We presented a data analysis section above including observations that a student's fail/view ratio that trends toward the threshold will eventually cross the threshold. We anticipate that the delta line discussed can act as an early indicator. We plan on continuing this study to test this hypothesis. More experimentation will be conducted, testing the activity for each student to determine if a crossing of the delta line leads to crossing the at-risk threshold as more data becomes available.

The results can indicate the reliability of detecting at-risk students. The number of line crossings for each student will be counted then compared to the risk state at the end of the data collection period. There will likely be some students that have a positive indicator who have not reached the at-risk category but are very likely to eventually become at-risk. To provide more insightful information when the indicator is present, the closeness or relative distance to crossing the threshold will be examined. When discussing the results, groups of students can be discussed: at-risk students, students that are *close* to being at-risk, and students who are not at-risk. The reliability of the indicator can then be measured using the entire data set by the indicator's accuracy to detect at-risk and nearly at-risk students.

The LSTM shows great promise for predicting student outcomes. The model converts a probability value between to a binary category of pass or fail by simply comparing to a threshold value of 0.5. Any value over 0.5 is classified as a pass and the other values are classified as a fail. A student prediction of 0.85 would be classified as a pass and a student prediction of 0.52 would also be classified as a pass. Binary classification systems will inherently have errors for prediction values that fall in the midpoint between categories. While these middle-sitting values generally cause errors in prediction systems, the raw prediction should provide insight when making decisions about intervention. Middle prediction values simply mean that a student might fail, and an appropriate level of intervention should take place. Conversely, a student with a very low prediction value near 0.0 should get more aggressive intervention.

The results yielded a very promising 78% accuracy with the typical caveats of binary classification as described above. The LTSM network used a sequence size of 100 based on the average sequence length. The sequences used actions right up until a pass/fail event. In real world systems, prediction would need to be based on earlier sequence data so that intervention can be provided in a timely manner. This study should continue investigating various windows of sequences between pass/fail events. Overlapping windows of shorter sequences from between events can be tested to see if predictions can be successful ignoring later data in a sequence. Additionally, more research is needed investigating if shorter sequences are effective at prediction.

We have developed a promising prediction model for student outcomes. However, we have not yet discovered the root causes for student failures based solely on sequences. Further studies should take place to extract the specific attributes of a sequence that drives the prediction decision. The discovery of such features can aid in intervention decisions. One intervention recommendation may for a student to limit the number of simultaneous courses. Another intervention recommendation may be for a student to complete more course exercises before taking assessments. There are several techniques for discovering features such as visualization observations, permutation feature importance, Shapley Additive Explanations, Attention Mechanism, and logistical regression.

Finally, tools will be built into the USAL ecosystem to run reports on actual live data in systems so that instructors and administrative users can focus on the students that most need attention. As a result, this study aims to provide instructors with the tools to manage large populations of students that would be difficult to guide through a learning curriculum without the aid of ML techniques and resulting knowledge.

REFERENCES

Adnan, M., Habib, A., Ashraf, J., Mussadiq, S., Raza, A. A., Abid, M., Bashir, M., & Khan, S. U. (2021). Predicting at-risk students at different percentages of course length for early intervention using machine learning models. *IEEE Access*, *9*, 7519–7539. https://doi.org/10.1109/access.2021.3049446

Advanced Distributed Learning (ADL) Initiative (2016, September 21). *Experience API (xAPI) Specification Version 1.0.3*. GitHub xAPI Specification Markdown. Retrieved August 14, 2023 from https://github.com/adlnet/xAPI-Spec

Al-azazi, Fatima Ahmed, and Mossa Ghurab. “Ann-LSTM: A Deep Learning Model for Early Student Performance Prediction in MOOC.” Heliyon, vol. 9, no. 4, 2023, https://doi.org/10.1016/j.heliyon.2023.e15382.

Al Breiki, B., Zaki, N., & Mohamed, E. A. (2019). Using educational data mining techniques to predict student performance. *2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*. https://doi.org/10.1109/icecta48151.2019.8959676

Aljaloud, Abdulaziz Salamah, et al. “A Deep Learning Model to Predict Student Learning Outcomes in LMS Using CNN and LSTM.” IEEE Access, vol. 10, 2022, pp. 85255–85265, https://doi.org/10.1109/access.2022.3196784.

Buschetto Macarini, L. A., Cechinel, C., Batista Machado, M. F., Faria Culmant Ramos, V., & Munoz, R. (2019). Predicting students success in blended learning—evaluating different interactions inside learning management systems. *Applied Sciences*, *9*(24), 5523. https://doi.org/10.3390/app9245523

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

Er, E. (2012). Identifying at-risk students using machine learning techniques: A case study with is 100. *International Journal of Machine Learning and Computing*, 476–480. https://doi.org/10.7763/ijmlc.2012.v2.171

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

Kondo, N., Okubo, M., & Hatanaka, T. (2017). Early detection of at-risk students using machine learning based on LMS Log Data. *2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI).* https://doi.org/10.1109/iiai-aai.2017.51

Lakkaraju, H., Aguiar, E., Shan, C., Miller, D., Bhanpuri, N., Ghani, R., & Addison, K. L. (2015). A machine learning framework to identify students at risk of adverse academic outcomes. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. https://doi.org/10.1145/2783258.2788620

Livieris, I. E., Drakopoulou, K., Tampakas, V. T., Mikropoulos, T. A., & Pintelas, P. (2018). Predicting secondary school students' performance utilizing a semi-supervised Learning Approach. Journal of Educational Computing Research, 57(2), 448–470. https://doi.org/10.1177/0735633117752614

Pek, R. Z., Ozyer, S. T., Elhage, T., Ozyer, T., & Alhajj, R. (2023). The role of machine learning in identifying students at-risk and minimizing failure. *IEEE Access*, *11*, 1224–1243. https://doi.org/10.1109/access.2022.3232984

Pilotti, M. A., Nazeeruddin, E., Nazeeruddin, M., Daqqa, I., Abdelsalam, H., & Abdullah, M. (2022). Is initial performance in a course informative? machine learning algorithms as AIDS for the early detection of at-risk students. *Electronics, 11*(13), 2057. https://doi.org/10.3390/electronics11132057

Soobramoney, Ranjin. *Early Prediction of Students at Risk in a Virtual Learning Environment Using Ensemble Machine Learning Techniques*, https://doi.org/10.51415/10321/4072