Behavior-Based Grade Prediction for MOOCs Via Time Series Neural Networks

Tsung-Yen Yang, Christopher G. Brinton, *Member, IEEE*, Carlee Joe-Wong, *Member, IEEE*, and Mung Chiang, *Fellow, IEEE*

***Abstract*—We present a novel method for predicting the evolution of a student's grade in massive open online courses (MOOCs). Performance prediction is particularly challenging in MOOC settings due to per-student assessment response sparsity and the need for personalized models. Our method overcomes these challenges by incorporating another, richer form of data collected from each student—lecture video-watching clickstreams—into the machine learning feature set, and using that to train a time series neural network that learns from both prior performance and clickstream data. Through evaluation on two MOOC datasets, we find that our algorithm outperforms a baseline of average past performance by more than 60% on average, and a lasso regression baseline by more than 15%. Moreover, the gains are higher when the student has answered fewer questions, underscoring their ability to provide instructors with early detection of struggling and/or advanced students. We also show that despite these gains, when taken alone, none of the behavioral features are particularly correlated with performance, emphasizing the need to consider their combined effect and nonlinear predictors. Finally, we discuss how course instructors can use these predictive learning analytics to stage student interventions.**

***Index Terms*—Clickstream data analysis, learning analytics, MOOC, student performance prediction.**

I. Introduction

MASSIVE Open Online Courses (MOOCs) have exploded in popularity over the past five years. MOOC delivery platforms such as Coursera, edX, and Udemy have partnered with content providers to deliver hundreds of thousands of courses to tens of millions of students around the world, either for free or at very cheap prices. An estimated 35 million people signed up for at least one MOOC in 2015, an increase of 50% from the year before [1]. Today, entire degree programs are offered through MOOC, with the eventual goal of providing global access to world class instruction [2].

For all their benefits, the quality of MOOCs has been the target of criticism. Research has pointed to their low completion rates–below 7% on average–as a property preventing more widespread adoption of these courses among instructors and institutions [3]. These high dropoff rates are often attributed to factors such as low teacher-to-student ratios, the asynchronous nature of interaction, and heterogeneous educational backgrounds and motivations, which make it difficult to scale the efficacy of traditional teaching methods with the size of the student body [4].

As a result, research on MOOCs is studying, and in turn attempting to enhance, student engagement and knowledge transfer in these online settings. The plethora of data that contemporary MOOC platforms (and eLearning platforms more generally) collect about users has ignited interest in data mining approaches, i.e., surfacing analytics to instructors that help them diagnose student needs. To see the value of this approach, consider the three dominant modes of learning in MOOCs: lecture videos, assessment questions, and social discussion forums. For video content, clickstream events are captured, with a record generated each time a student interacts with a video specifying the particular action, position, and time at which it occurred. For assessments, the specific responses to individual questions are recorded. For the discussion forums, all posts, comments, and votes made by learners and instructors are stored as well. This data has led to analytics both about learners and about content [5], such as the identification of Social Learning Networks (SLN) among students [6], relationships between engagement and performance levels [4], and segments of focus in lecture videos [7].

A. Predictive Learning Analytics

Within the field of MOOC analytics, predictive learning analytics (PLA)–methods that *predict* MOOC learning outcomes at different points in a course, so that appropriate actions can be taken in advance–is a relatively new area of exploration [8]. A student's course grade would be a particularly useful quantity to forecast, because it is indicative of how well the course is matched to the student's needs: a student who performs poorly needs attention from an instructor, while a student who performs exceedingly well may not be challenged enough by the material. It has been observed that both of these extreme cases will

Manuscript received October 15, 2016; revised March 1, 2017; accepted March 29, 2017. Date of publication May 2, 2017; date of current version July 18, 2017. This work was supported in part by Zoomi, Inc., under Grants NSF CNS-1347234 and ARO W911 NF-14-1-0190. The guest editor coordinating the review of this paper and approving it for publication was Prof. Mihaela van der Schaar. *(Corresponding author: Christopher G. Brinton.)*

T. Y. Yang is with the Department of Electrical Engineering and Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: tsungyenyang.eecs02@nctu.edu.tw).

C. Brinton is with the Department of Advanced Research, Zoomi, Inc., Malvern, PA 19355 USA (e-mail: christopher.brinton@zoomiinc.com).

C. Joe-Wong is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Mountain View, CA 94035 USA (e-mail: cjoewong@andrew.cmu.edu).

M. Chiang is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: chiangm@princeton.edu).

Digital Object Identifier 10.1109/JSTSP.2017.2700227

cause dropoffs [9]. If instructors were given an indication early on about which learners were likely to perform poorly before course completion, and at which points these falloffs were likely to occur, they could e.g., stage interventions or change content as preventative actions. The fact that students begin dropping off even during the first week underscores the utility of algorithms that can provide *early detection* of poor or exceptional user performance [4].

Grade prediction for MOOC has two unique challenges. The first is assessment response sparsity [10]: many students choose to only answer a few assessment questions, making it difficult to learn from this data alone. Second, our prediction models need to be personalized to different students, since learners have different motivations for taking MOOCs, which affects their behavior [11]. In this paper, we present and evaluate a time series neural network method that overcomes these challenges. Our algorithm predicts a MOOC student's overall course grade as he/she progresses through the course material, taking as input his/her prior (i) assessment performance and (ii) video-watching behavior. For the video-watching aspect, certain behavioral quantities (e.g., number of rewinds, average playback rate, fraction completed) that have been found to be correlated with quiz success are computed from the student's clickstream measurements [10].

We evaluate the quality of two algorithms, one learning from quiz (i.e., assessment) features only (FTSNN) and one from both behavioral features and quiz features (IFTSNN), against two baselines, one based on averaged past performance and one based on lasso regression, on two MOOC datasets. Overall, we find that:

1) Both algorithms consistently outperform both baselines, with average RMSE improvements of $> 61\%$ for IFTSNN and $>49\%$ for FTSNN over the naive baseline.
2) IFTSNN outperforms FTSNN overall as well, with an average improvement of $>10\%$, underscoring the importance of clickstream data to MOOC grade prediction.
3) In the case where only a few assessment results are available, however, FTSNN has the highest performance, indicating that performance-only algorithms may be sufficient for earliest detection.
4) Taken alone, none of the video-watching behavior quantities are particularly predictive of average grades, demonstrating the importance of considering their combined effect to predict student performance.
5) Personalized prediction models are exceedingly important, as applying parameters tuned to other students is less accurate than even the naive baseline algorithm.

We note that the overall purpose of our work is to assess the feasibility of a neural network-based algorithm for MOOC performance prediction. We make no claim that either the IFTSNN or FTSNN models developed here are the "optimal" predictors, i.e., higher quality may be possible with alternate network configurations tuned to specific courses. The above insights should instead be taken as lower bounds on the potential for behavior-based grade prediction via the family of neural network algorithms.

B. Related Work

The proliferation of MOOCs has led to several analytical studies on their datasets. Some research has focused on understanding student motivation and engagement across learning modes, e.g., [12], [13]. Other works have analyzed student behavior on specific modes, e.g., [6], [14] quantified participation on MOOC forums and [15], [16] studied interaction patterns in lecture videos. There has also been work on identifying taxonomies of student motivation for enrolling in MOOCs [11]. Our work is fundamentally different from these in that it focuses on algorithms for predictive analytics.

Methods for student performance prediction have been proposed and evaluated, mainly for traditional online and brick-and-mortar education settings. These include predicting how students will perform on assessments they have not yet taken [2], [17], [18] and what their final grades will be [19], [20], typically using their past assessment scores. Most recently, [19] proposed an algorithm to optimize the timing of grade predictions, and [2] proposed a deep learning version of student knowledge tracing. We instead consider performance prediction for MOOC settings, in which per-student performance data is sparse, necessitating the use of behavioral modeling.

In this regard, there have been a few recent works on predictive analytics for MOOC, proposing algorithms to predict dropoff rates [21], [22] and assessment scores [4], [10], [22], [23]. Among these, [4], [10] studied the relationship between video-watching behavior and in-video quiz performance and used the results as features for prediction; unlike these works, we consider the time series aspect of assessment responses and develop a personalized model for each student. Some works have studied prediction of average grades over time. [23] proposes a linear multi-regression model for assessment performance, using video, assessment, and time-related features; we apply neural networks on a similar set of features to increase prediction quality (with RMSEs as low as 0.06, compared to 0.16 to 0.23 in [23]). Finally, [22] uses demographic, discussion forum, and chapter access data as features in a probabilistic graphical model framework; our work focuses on a more specific set of video-watching features.

C. Our Methodology

Fig. 1 summarizes the main components of the grade prediction methodology we develop in this paper. At a given point in time, each student's video-watching clickstream data and assessment grades are processed to compute a set of prediction features for that student (Sections II-A and III-A). These features are subsequently used to train time series neural networks that account for the sparsity of the data (Section II-C), after partitioning the data for training and testing accordingly (Section II-B). Model quality is determined through RMSE, comparing against two baselines, one of averaged historical performance and one of linear regression, to give a relative gain (Section IV).

These personalized models are then used to predict how the student's grade will evolve as he/she progresses through more material. Fig. 2 summarizes the sequence of online predictions

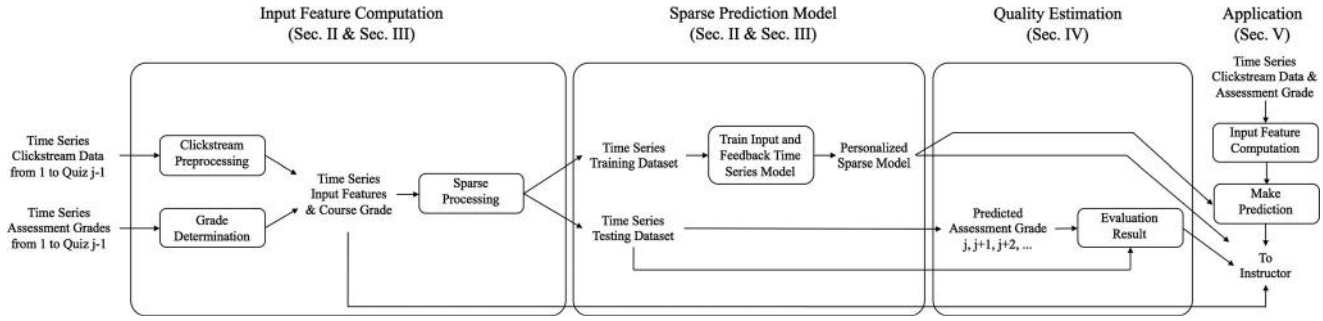


Fig. 1. Summary of the different components of the learning outcome prediction method we develop in this paper.

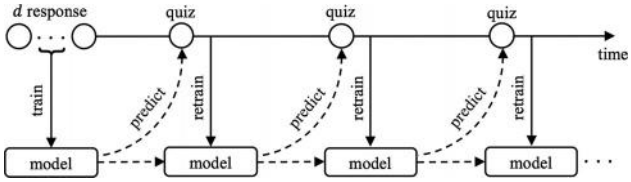


Fig. 2. Sequence of average CFA predictions made online as a student moves through the course. Each train, retrain, and prediction step involves the components in Fig. 1.

as the student moves through the course. After the student takes a quiz $j-1 > d$, where d is the memory of the time series, we split the student's past video watching behavior and quiz performance into training and testing datasets and retrain our prediction model. We then use the retrained model to predict the average CFA after the student takes quiz j, based on quizzes $1, 2, ..., j-1$. Each time a student takes another quiz, new data is used to refine the model parameters, and the predictions are updated accordingly. Finally, the predictions, model quality, and feature distributions will be provided to the instructor through an appropriate dashboard interface so that the instructor can take necessary action (Section V).

Contribution: The key contributions of this paper are summarized as follows:

1) We propose a method for predicting course grades from behavioral data in MOOCs using a novel set of features in a time-series neural network, overcoming the challenge of assessment data sparsity.
2) We show that personalized prediction models are essential for predictive analytics in MOOCs, since different students' behavior differs significantly.
3) We demonstrate the benefit that different forms of data–prior grades and prior clickstream behavior–offer for grade prediction in MOOCs.

II. Grade Prediction Algorithm

In this section, we first introduce the input and output variables of our algorithms, and then describe our algorithm design and evaluation method.

A. Input Features and Course Grade

Fig. 3 shows the general structure of a MOOC with lecture videos and quizzes. The course is delivered as a sequence of videos, with in-video quizzes interspersed at points designated by the instructor. With quizzes indexed sequentially as $j = 1, 2, \ldots$, the K_j videos occurring between quizzes $j-1$ and j are denoted $(j, 1), ..., (j, k), ..., (j, K_j)$.

The datasets we use in this paper come from two of our MOOCs on Coursera. The first one is called "Networks: Friends, Money, and Bytes" (NFMB) [24], and the second one is called "Networks Illustrated: Principles without Calculus" (NI) [25]. Both are networking courses that cover similar topics, but the material in NFMB is more advanced than that in NI. NFMB has 92 videos with exactly one quiz after each video (i.e., $K_j = 1\ \forall j$), while NI has 69 quizzes, some of which are preceded by multiple videos. We obtained two types of data from each MOOC: clickstream data and quiz answers.

Clickstream data for video (j, k): When a student watches a video, he/she may play, pause, slow down or speed up, or jump to another place in the video. MOOC providers store these events along with their video positions, UNIX timestamp of occurrence, and student/video identifiers. Analyzing them gives insight into learning behavior [4]: for example, when the contents of the video confuse a student, he/she may pause and re-watch the confusing part of the video. On the other hand, when a student is familiar with the concepts in a video, he/she may skip the video or watch only selected portions and quickly move to the next video. These clickstream data thus reflect the learning behavior of each specific student, creating a unique, personalized *learning pattern*.

Answer to quiz j: In both the NFMB and NI datasets, each quiz consists of a single multiple choice question with exactly one correct answer. We gauge success on a quiz as whether the student successfully answers the question Correctly on his/her First Attempt (CFA) or not (non-CFA) [4].

Our prediction algorithm uses both clickstream data and quiz responses to forecast students' course performance. In order to do so, we first transform the raw clickstream data to several algorithm input features, and use the students' quiz responses to define a performance measure.

Input clickstream features: Following the clickstream preprocessing methods outlined in [10], we compute eight input features from each video for each student:

1) Fraction completed (`F-Co`): The percentage of the video that the student played, not counting repeated intervals more than once; hence, it must be between 0 and 1.

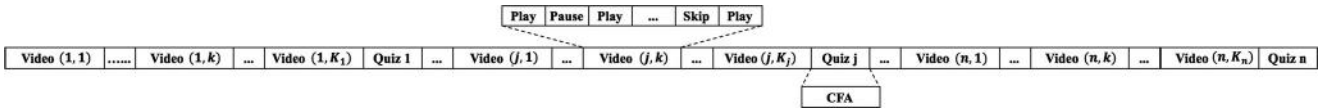


Fig. 3. General sequence of lecture videos and in-video quizzes in a MOOC.

2) Fraction spent (`F-Sp`): The amount of (real) time the student spent on the video (i.e., while playing or paused) divided by its total playback time.[1]
3) Fraction played (`F-Pl`): The amount of the video that the student played, including repetitions, divided by its total playback time.
4) Fraction paused (`F-Pa`): The amount of time the student spent paused on the video, divided by its total playback time.
5) Number of pauses (`N-Pa`): The number of times the student paused the video.
6) Average playback rate (`A-PR`): The time-average of the playback rates selected by the student while in the playing state. The player on Coursera allows rates between $0.75\times$ and $2.0\times$ the default speed.
7) Standard deviation of playback rates (`S-PR`): The standard deviation of the playback rates selected over time.
8) Number of rewinds (`N-R`): The number of times the student skipped backward in the video.[2]

In order to enforce a one-to-one correspondence between videos and quizzes, we average each of the eight features over all videos between consecutive quizzes for NI. Since each quiz j covered material in videos $(j,1),...,(j,K_j)$ between quizzes $j-1$ and j, our averaging ensures that we have a comprehensive picture of students' relevant video watching behavior. For ease of exposition, we refer to these averaged features as corresponding to "video j," an aggregation of videos $(j,1),\ldots,(j,K_j)$.

Average CFA grade: We define a student's performance in the course at a given point in time as his or her average quiz grade, i.e., the average number of CFA responses [23]. Since students answer quizzes throughout the course, we are able to track and predict their grades after each quiz answered. For a given student i, we define $c_i(t)$ as the student's response to quiz t; $c_i(t)=1$ if the student was CFA, and 0 otherwise (i.e., if the student answered incorrectly or did not answer at all). We let $(t_i(1),t_i(2),\ldots,t_i(n_i))$ denote the sequence of quiz indices that student i answers; importantly, students need not answer any questions and the order in which they are answered need not be sequential (we may have $t_i(j)>t_i(j+1)$). Each student's average CFA after answering j questions is then:

$$y_i(j)=\frac{\sum_{s=1}^{j}c_i(t_i(s))}{j}.$$

Fig. 4 shows the evolution of average CFA grades for several students from the NFMB and NI courses who answered all questions in the course. Each student's CFA score oscillates at the beginning of the course but eventually stabilizes after around 10 or 20 responses; after a student has answered several questions, a single quiz response will not significantly change his or her average CFA grade. Thus, we would expect the average CFA prediction to become easier as students answer more questions.

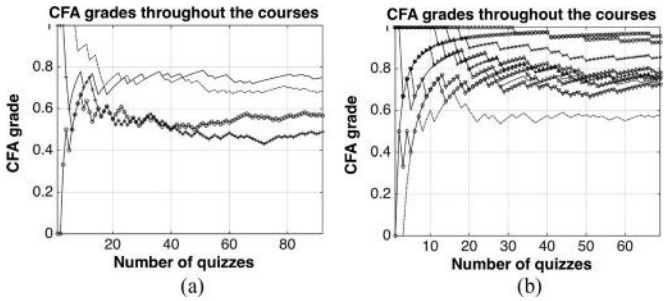


Fig. 4. Examples of students' average CFA grades throughout the courses. (a) Four students take all of NFMB's 92 quizzes. (b) Twenty-nine students take all of NI's 69 quizzes; for simplicity, we only plot the first 10 students.

B. Algorithm Setup

Training: Our algorithm uses each student j's video-watching clickstream features and the previous average CFA grades as inputs to predict each average CFA grade $y_i(j)$ for j up to n_i, the number of questions that student i answers. We train the algorithm separately on each individual student's data; thus, letting $\vec{y}_i=\left[\,y_i(1)\ \ldots\ y_i(n_i)\,\right]$ denote the vector of student i's average CFA grades throughout the course, we choose a subset $\vec{y}_i^{train}$ of $\vec{y}_i$ on which to train the algorithm. The algorithm training is validated on a separate subset $\vec{y}_i^{valid}$ and then tested on yet another subset of student i's average CFA grades, $\vec{y}_i^{test}$, which does not intersect with $\vec{y}_i^{train}$ or $\vec{y}_i^{valid}$.

Evaluation: We use the Root Mean Square Error (RMSE) to evaluate the performance of our algorithm, which is developed in Section II-C, on each student's data. We exclude the training and validation data points, and instead calculate the RMSE for each student i over that student's testing data $\vec{y}_i^{test}$. Letting $z_i(n)$ denote the predicted value of student i's nth average CFA grade $y_i(n)$,

$$RMSE_i=\sqrt{\frac{1}{|\vec{y}_i^{test}|}\sum\nolimits_{y_i(n)\in\vec{y}_i^{test}}\left(y_i(n)-z_i(n)\right)^2}$$

where $|y|$ denotes the length of the vector y. We can then average different students' RMSEs to evaluate the algorithms' performance over a given set of students.

Naive Baseline: We compare our algorithm's performance to a naive baseline of simply averaging a given student's previous CFA grades:

$$z_i(j)=\frac{\sum_{s=1}^{j-1}c_i(t_i(s))}{j-1}.$$

[1]We define the *playback time* as the time it takes to play a video at the default speed, e.g., a 3:30 video has a playback time of 210 seconds.

[2]We do not consider the number of fast forwards because it was found to not be significantly correlated with CFA in [10].

with $z_i(j)$, $j > 1$, again denoting student i's estimated average CFA grade after answering j quizzes. Note that as j increases, i.e., the student answers more quiz questions, the naive baseline will likely perform better; the jth CFA response will not significantly change the student's average CFA grade.

Linear Regression Baseline: We also compare our algorithm's performance with linear regression, in which we optimize the coefficients of our linear predictor. To enhance the prediction accuracy, we use the lasso method to perform variable selection and regularization [26]. Comparing these results to those of the IFTSNN and FTSNN algorithms thus provides an idea of the additional accuracy achieved by including nonlinearity in the prediction algorithm at the expense of model interpretability [27].

We note that both the naive and lasso regression baselines are *linear* predictors, while the algorithm we develop in Section II-C is nonlinear. Thus, a comparison of the baseline to our algorithm also serves to compare (non-optimized) linear prediction algorithms, as used in [23], with a nonlinear predictor for average CFA grades.

C. Our Prediction Algorithm

Using the data processing from Section II-A, we define a features-CFA grade pair as follows:

1) $x_i\,(j)$: The vector of clickstream input features for student i in video $t_i(j)$.
2) $y_i\,(j)$: Student i's average CFA grade after video $t_i(j)$ (i.e., answering j quizzes).

We use the input features x_i and the previous average CFA grade y_i to predict each student i's average CFA grades y_i.

While many different algorithms can be used for this prediction (including the naive baseline in Section II-B), we use a time series neural network predictor due to their popularity in many research fields [28], including student knowledge tracing [2]. Time series neural networks are recurrent neural networks, with feedback connections enclosing several layers of the network. Long Short Term Memory (LSTM) [29] and Gated Recurrent Unit (GRU) [30] networks are two examples of recurrent neural networks. They are good at solving problems that require learning long-term temporal dependencies. However, most of the students in our dataset do not generate a long time series of data, as they skip many quizzes in the course. We also find little dependence between the behavior features of different quizzes. Therefore, standard recurrent neural networks are sufficient for our prediction. Moreover, neural networks are more robust to data sparsity than other nonlinear predictors, e.g., collaborative filtering methods rely on performance comparisons with similar students, and performance data in MOOC is too sparse to accurately assess student-to-student similarity [10]. While they may not be the *optimal* type of predictor for MOOC performance, our results demonstrate the feasibility of using time series neural network predictors on MOOC data.

Dealing with sparsity: Before introducing our neural network models, we first discuss our method for addressing data sparsity, as dealing with sparse data is one of the challenges of doing predictions in MOOCs [4]. As discussed previously, most students do not answer all of the quiz questions in a given MOOC, leading to a sparse set of quiz responses for any individual student. To handle this missing data, we simply "skip" the missing quiz data and consider the previous d quizzes that the student answered, instead of the previous d quizzes in the course. This logic is reflected in our definition of $y_i(j)$ in Section II-A.

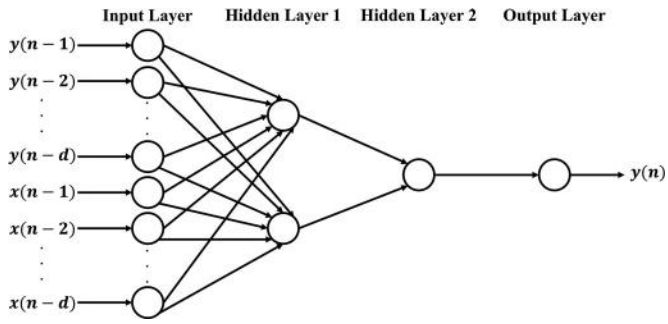


Fig. 5. Graphical representation of IFTSNN.

To validate this approach, we randomly shuffle the time instances of the CFA inputs to our IFTSNN and FTSNN prediction algorithms and find that there is no obvious performance degradation. Thus, the particular relationship between the topics covered by different quizzes has no bearing on the predictive power of prior video watching behavior and quiz responses. Since our goal is to predict the overall grade at any point in the course, this grade depends not on the topic of the next question but also on all the previous questions. Our approach is thus general enough to study how behavior and prior performance will impact future performance in a way that is independent of the particular topics covered by each quiz.

Neural network model: We use two hidden layers in each network that we train, which can be seen as a Deep Neural Network (DNN); thus, we have both a hidden layer and an output layer. The overall neural network model can be described as follows:

$$z_i(n) = f_i\big(y_i(n-1), y_i(n-2), ..., y_i(n-d), \\ x_i(n-1), x_i(n-2), \ldots, x_i(n-d)\big),$$

where $z_i(n)$ is again the predicted average CFA grade for student i after answering n quizzes, and d indicates the feedback delay, or the number of previous quiz responses considered. d in our model can also be understood as the minimum number of questions a student must answer before predictions on future average CFA will be made.[3] We use f_i to denote the model to emphasize that we train the model separately for each student i; thus, each student's neural network will have different parameters. We discuss the importance of model personalization in Section III-B. Fig. 5 summarizes the overall network structure of this model.

We will additionally use another type of neural network to evaluate the value of including the clickstream features x_i in

[3]If a student has answered $d_0 < d$ questions at time n and predictions at this time are desired, it is certainly possible for the neural network to use just these d_0 for model training, as long as $d_0 > 1$.

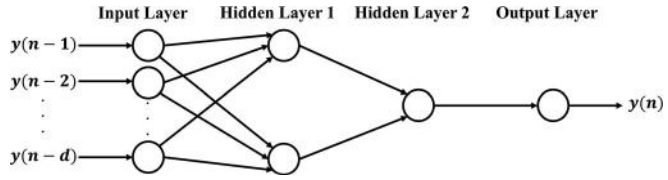


Fig. 6. Graphical representation of FTSNN.

N	IFTSNN	FTSNN
[2 1]	0.0561	0.0675
[5 5]	0.0597	0.0618
[5 2]	0.0557	0.0652
[10 5]	0.0553	0.0593
[20 10]	0.0580	0.0553
[15 5]	0.0586	0.0568
[20 5]	0.0573	0.0583

d	2	4	5	6
IFTSNN	0.0763	0.0630	0.0553	0.0539
FTSNN	0.0795	0.0629	0.0593	0.0524

Fig. 7. Average RMSE obtained (a) for different network configurations (N) and (b) for different input lengths (d) on the NFMB dataset.

our predictions. We call this type of network a Feedback Time Series Neural Network (FTSNN) model; compared to the previous model–which we call Input FTSNN (IFTSNN) since it has the clickstream input x_i–FTSNN does not use the clickstream features. Thus, it relies only on *feedback* data, i.e., past average CFA grades from student i:

$$z_i(n) = g_i\left(y_i(n-1), y_i(n-2), ..., y_i(n-d)\right).$$

Fig. 6 shows the overall structure of the FTSNN model.

We use Bayesian regularization with back-propagation to train both types of model. Bayesian regularization minimizes a linear combination of squared errors and weights. The training algorithm first finds the parameters that minimize a weighed sum of errors, and then adjusts the weights and trained parameters to minimize a different weighted sum of errors, in order to make sure that all errors are minimized. This Bayesian regularization takes place within the Levenberg-Marquardt algorithm [31].

In addition to the neural network parameters, there are several configuration parameters that we can tune for a time series neural network:

1) The number of feedback delays d: How much feedback and clickstream history should be used in the prediction.
2) The number of hidden layers H.
3) The number of neurons in each hidden layer N.

To select parameter values, we tested several configurations of the network a priori. In the end, we chose $d = 5, H = 2$, and $N = [2\ 1]$ since these values tended to yield consistently high quality results across both datasets; we will use these sets of configuration parameters for every model that we train. For completeness, Fig. 7 show the RMSEs achieved on the NFMB dataset by (a) different configurations N of a two-layer network and (b) different feedback delays d. We see that each setting of N yields qualitatively similar results for both algorithms, and the performance improvement in d becomes marginal after $d = 5$, constituting a reasonable tradeoff between model complexity and quality enhancement.

We do expect, however, that a more extensive search for the optimal choices of N and H (through e.g., cross validation) will improve our prediction quality further. However, the results for our chosen parameters are sufficient to demonstrate the feasibility of using neural networks to predict MOOC students' performance. A simpler two-layer, three neuron network has added advantages of efficient re-training in an online manner (discussed in Section IV-E) and less overfitting in the presence of sparse data.

Feature	Mean	S.D.	Feature	Mean	S.D.
F-Co	0.772	0.336	F-Co	0.756	0.362
	0.759	0.350		0.737	0.544
F-Sp	21.912	264.260	F-Sp	18.041	231.280
	28.360	380.510		17.855	244.750
F-Pl	1.022	4.563	F-Pl	0.846	0.552
	0.915	0.413		0.878	5.167
F-Pa	37.263	393.070	F-Pa	63.620	529.870
	34.562	339.320		63.410	459.420
N-Pa	3.113	72.504	N-Pa	1.997	5.410
	2.261	4.570		2.239	18.283
A-PR	1.112	0.313	A-PR	1.051	0.301
	1.088	0.319		1.036	0.318
S-PR	0.016	0.052	S-PR	0.002	0.0153
	0.012	0.046		0.002	0.0135
N-R	2.350	9.125	N-R	1.684	16.063
	2.018	23.576		1.772	17.137

Fig. 8. Tabulated statistics–mean and standard deviation (S.D.)–for the clickstream features corresponding to videos for different quiz responses. The top row for each feature corresponds to CFA responses, and the bottom to non-CFA responses. (a) NFMB. (b) NI.

III. Datasets and Analysis

A. Feature Distributions and Performance

We perform some statistical analysis on the relationship between the input features and CFA scores for each dataset, in order to provide some intuition for the prediction results in Section IV. These insights can be useful to instructors in devising interventions to assist students as well. Many features have large standard deviations, indicating that the data are not only sparse but also noisy.

CFA vs. non-CFA: Fig. 8 shows the means and standard deviations (S.D.) of all eight clickstream input features for both courses, considering the CFA and non-CFA responses separately. Here, the clickstream features $x_i(j)$ for student i on video $t_i(j)$ are tied to the binary CFA score $c_i\left(t_i(j)\right)$ on quiz $t_i(j)$. There are 19,432 CFA and 9,861 non-CFA responses in NFMB, while there are 34,886 CFAs and 11,762 non-CFAs in NI. We make some general observations for each feature:

Fraction completed (`F-Co`)*:* CFA responses in both courses have higher means than non-CFA responses. In other words, students who completed more of a video are more likely to be successful on the corresponding quiz.

Fraction spent (`F-Sp`)*:* The mean for CFA responses is 18.041, compared to 17.855 for non-CFA responses, in NI.

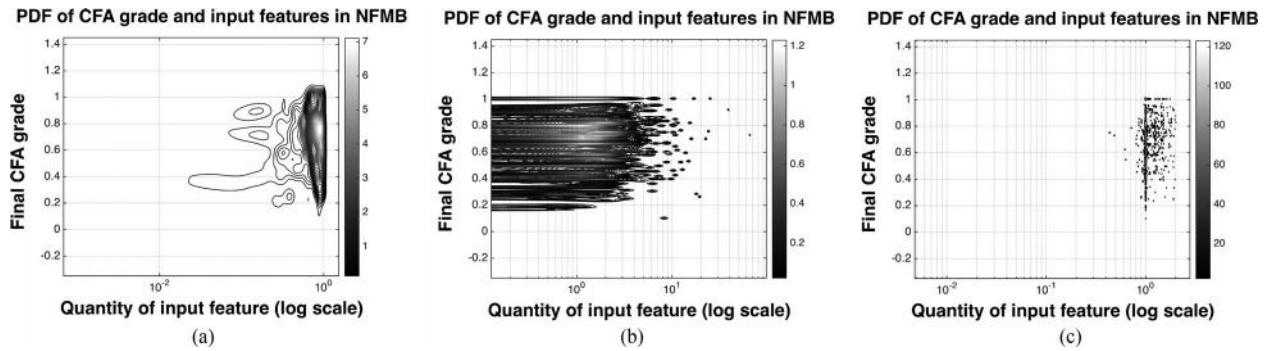


Fig. 9. Two dimensional probability density distributions of NFMB students' clickstream features and final CFA grades. (a) F–Co. (b) N–Pa. (c) A–PR

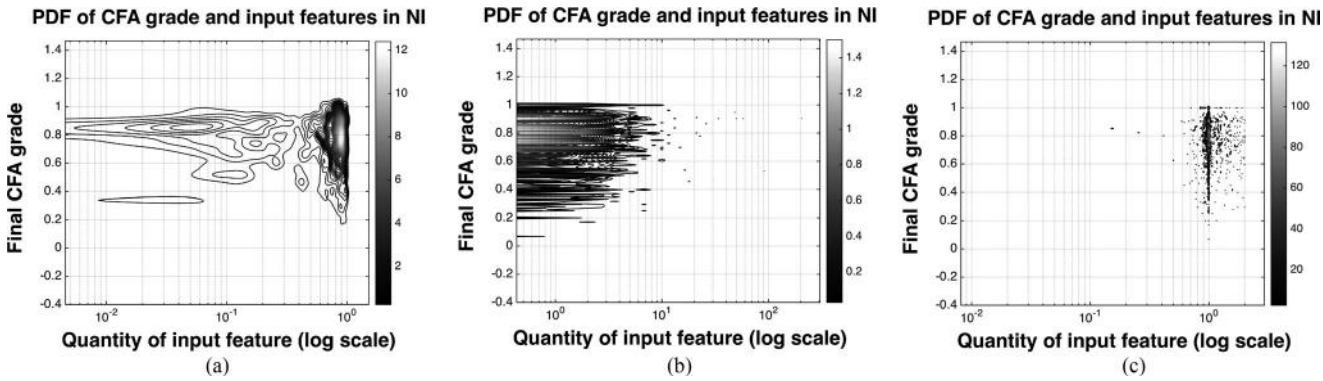


Fig. 10. Probability density distributions of the features in Fig. 9 for NI. (a) F–Co. (b) N–Pa. (c) A–PR.

Students who spend more time with the video may be more likely to answer questions correctly, as we would intuitively expect. However, we note that the standard deviations for `F-Sp` are quite large for both courses and CFA/non-CFA responses, indicating that the difference in means may not be significant. In fact, for the responses in NFMB, the mean for CFA responses is 21.912 compared to 28.360 for non-CFA, indicating a more complex relationship than one would expect from the NFMB results.

Fraction played (`F-Pl`)*:* Like `F-Sp`, the two courses show different results. The CFA responses have a higher mean for the NFMB course, but lower mean for the NI course. CFA students may tend to watch more of the video, increasing the amount played (including repetitions), but they may also repeat fewer parts of the video, leading to a lower `F-Sp`.

Fraction paused (`F-Pa`)*:* There is only a slight difference in the means for CFA and non-CFA responses for either course, but the CFA responses have slightly larger means. However, the standard deviations are also large, indicating that these differences are likely not significant, as for `F-Sp`.

Number of pauses (`N-Pa`)*:* The mean for the CFA responses in NFMB is higher than that for non-CFA responses; however, the opposite is true for the NI responses. Students who pause the video frequently may reflect more on the material covered, making them more likely to be CFA, or they may be more confused by the video, making them less likely to be CFA. The difference in significance between `N-Pa` and `F-Pa` indicates that it is more useful to consider pausing independent of video playback length.

Average playback rate (`A-PR`)*:* The means for the CFA responses in both courses are higher than the non-CFA means, but the differences are extremely small.

Standard deviation of playback rate (`S-PR`)*:* In NI, the CFA and non-CFA responses have the same means, but the mean for NFMB CFA responses is higher than the mean for non-CFA responses. The small overall means in both cases indicate that students tend to keep the default playback speeds.

Number of rewinds (`N-R`)*:* Like `F-Pl`, the mean for CFA responses is higher than the mean for non-CFA responses in FMB, but the mean for CFA responses is slightly lower than the mean for non-CFA responses in NI.

In general, we observe that the two different courses exhibit somewhat different means for CFA and non-CFA responses. This observation may indicate that the difficulty of the course affects students' learning behaviors.

Average CFA grade: Figs. 9 and 10 plot students' average CFA grades against selected features to see whether clear correlations exist. Each student i appears as one datapoint in each plot, as his/her average feature value and average CFA grade $y_i(n_i)$ over all n_i quizzes the student took.[4]

Intuitively, one would expect each of these features to be strongly correlated with quiz performance, e.g., as students complete larger portions of the videos (higher `F-Co`), we would expect them to have higher average quiz grades. As the

[4]The selection of `F-Co`, `N-Pa`, and `A-PR` to show in the paper is arbitrary; all clickstream features show a similarly nonlinear relationship with the average CFA grades.

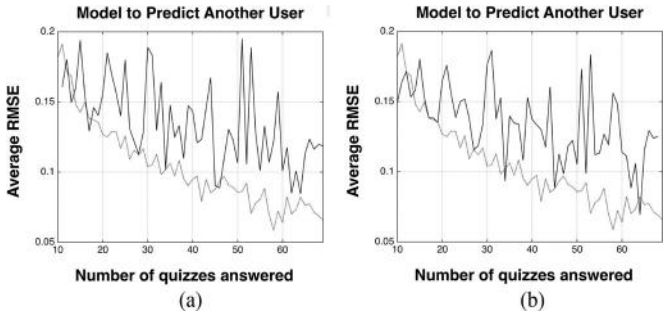


Fig. 11. Using IFTSNN models trained on NI students who answered N quizzes to predict average CFA grades for NI students who answered a different number of quizzes. The dotted line is the naive baseline. The x-axis shows the number of quizzes answered by the students whose scores we predict, and the y-axis is the avg. RMSE. (a) N = 10. (b) N = 69.

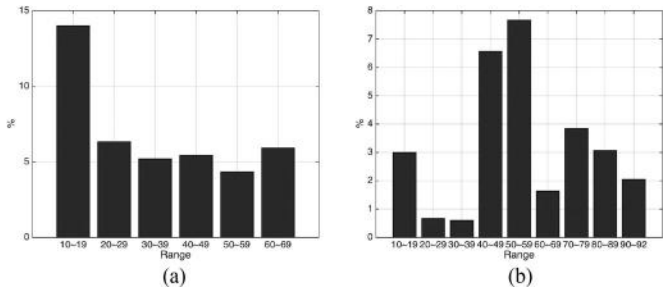


Fig. 12. The naive baseline nearly always performs worse than predicting students' average CFA score with an IFTSNN trained on another student's data. The y-axis shows the percentage of students for whom the naive baseline yields a larger RMSE. (a) NI. (b) NFMB.

figures show, however, the correlations between CFA grades and clickstream features are relatively weak. Our prediction results in Section IV will demonstrate that there is indeed a relationship when all features are considered together, but it is highly non-linear. Neural networks can discover such relationships, as they automatically learn their own internal representations of the different input features, and can decide dynamically which features to count and how effective they are at predicting the output [32].

B. Model Personalization

In order to motivate training individual models for each student, we consider the effect of using algorithms trained on one NI student to predict another NI student's average CFA scores. In particular, we test two IFTSNN models trained on students who answered 10 quizzes [Fig. 11(a)] and 69 quizzes [Fig. 11(b)] on data from other students, and compare the result with the naive baseline. The baseline algorithm performs better in most cases, particularly as the number of quizzes answered increases. As students answer more quizzes, we would expect the baseline algorithm to perform better (cf. Section II), which is consistent with these results.

Fig. 12 shows the percentage of students for whom the baseline algorithm's average RMSE is larger than the RMSE achieved by our IFTSNN algorithm trained on another student's data, grouped by the number of quiz questions that the student answered. The baseline algorithm rarely performs worse for either course. Thus, in order to measurably improve on the naive baseline, it is necessary to train our algorithms on individual students' data.

Note also that the results for the naive baseline in Fig. 11 are on the same order of magnitude as those for our individually trained algorithms. This indicates that our algorithms do not overfit to each individual student. Instead, they tend to reflect individual students' characteristics, which will allow them to outperform the baseline substantially in Section IV.

C. Dataset Groupings

We divide our datasets into different partitions for evaluation in Section IV. Let $\Omega_A^{s_0,e_0} \in \Omega$ denote the set of students in the dataset Ω_A who answered at least u_0 and at most e_0 questions, and let $\Omega_A^{s_0} \equiv \Omega^{s_0,s_0}$ be those who answer exactly s_0 questions. We take the subscript $A = F$ for NFMB students and $A = I$ for NI students; thus, Ω_F denotes all data from NFMB students, and Ω_I all data from NI students. We then spilt the students of both courses into four groupings:

1) Grouping A: NFMB students who answer exactly 10, 11, ..., 92 quizzes, i.e., $\Omega_A = \{\Omega_F^{s_0} | s_0 = 10, 11, ..., 92\}$.
2) Grouping B: NFMB students who answer between $10 \sim 10$, $10 \sim 11$, $10 \sim 12$, ..., $10 \sim 92$ quizzes, i.e., $\Omega_B = \{\Omega_F^{s_0,e_0} | s_0 = 10; e_0 = 10, 11, ..., 92\}$.
3) Grouping C: NI students who answer exactly 10, 11, ..., 69 quizzes, i.e., $\Omega_C = \{\Omega_I^{s_0} | s_0 = 10, 11, ..., 69\}$.
4) Grouping D: NI students who answer between $10 \sim 10$, $10 \sim 11$, ..., $10 \sim 69$ quizzes, i.e., $\Omega_D = \{\Omega_I^{s_0,e_0} | s_0 = 10; e_0 = 10, 11, ..., 69\}$.

For example, in grouping A, Ω_F^{11} is the subset of students in FMB who answer exactly 11 questions. In grouping B, $\Omega_F^{10,12}$ is those who answer between 10 and 12 questions.

Fig. 13 shows the distribution of the number of students in each subset of groupings A, B, C and D; groupings B and D are cumulative versions of A and C. We see that most students answer fewer than 20 quiz questions, leading to a sparse dataset.

IV. Grade Prediction Evaluation

In this section, we evaluate the performance of the model presented in Section II on our course data. In Section V, we propose some student interventions that use our prediction methods to help vulnerable or struggling students.

A. Algorithm Implementation

As described in Section II-B, we train our neural network prediction models separately on each individual student's data. For each student in both courses, we train two different models: one FTSNN (i.e., a neural network with only feedback data), and one IFTSNN. To ensure that we have enough data to train and test a reliable model, we only consider students who answered at least 10 quizzes. For each student, we randomly select 70% of their quiz responses as training data; 10% is used as validation data, and 20% of the data is used as testing data. Throughout

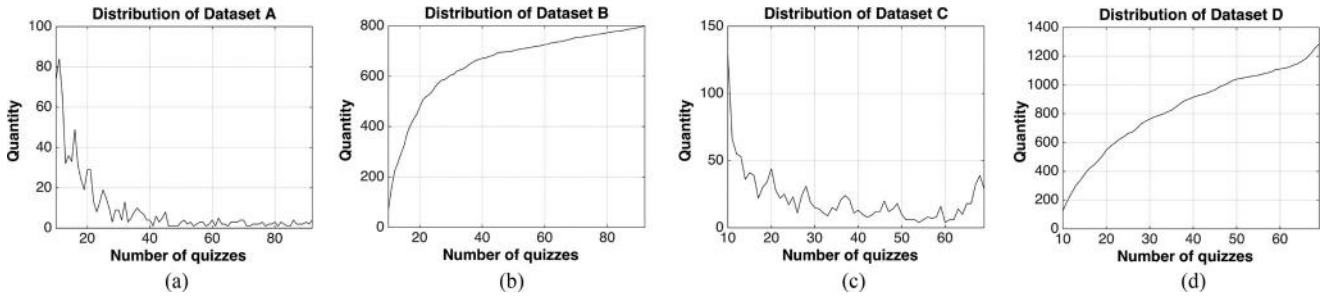


Fig. 13. Numbers of students in each subset of the groupings A, B, C and D. (a) Grouping A, (b) Grouping B, (c) Grouping C, (d) Grouping D.

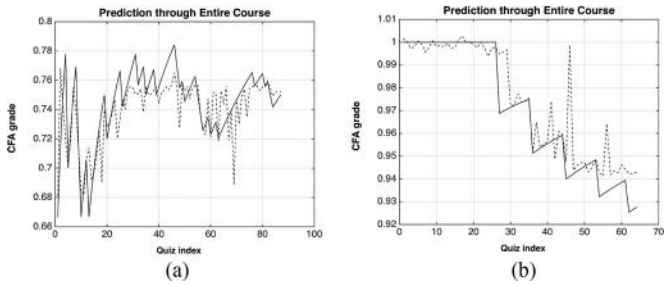


Fig. 14. Sample result of predictions for one student in each course. The solid line denotes the actual average CFA grade while the dashed line is the predicted average grade. (a) NFMB, (b) NI.

Model	Grouping A		Grouping B	
IFTSNN	0.0601	61.1%	0.0579	105.1%
FTSNN	0.0664	49.9%	0.0606	98.2%
Lasso	0.0724	26.8%	0.0832	42.3%
Naive	0.0918	–	0.1184	–

Model	Grouping C		Grouping D	
IFTSNN	0.0702	92.5%	0.0683	144.1%
FTSNN	0.0754	79.6%	0.0724	138.1%
Lasso	0.0804	28.9%	0.0791	66.1 %
Naive	0.1036	–	0.1314	–

Fig. 15. Overall average RMSEs for the different algorithms, with the percent improvement relative to the naive baseline indicated. (a) NFMB students, (b) NI students.

this section, we use RMSE on the testing data to evaluate each prediction algorithm's accuracy. Unless stated otherwise, figures show the average RMSE, taken over the specified set of students.

B. Overall Quality

Fig. 14 shows a sample result of our IFTSNN predictions for two students, one in each class. We can observe that the predicted CFA grades track students' realized average CFA grades well throughout the course. While we would expect the average CFA grades to level off as the student answers more questions–each individual CFA grade affects the average less as we collect more student responses–the average CFA grades for the students in Fig. 14 show some oscillation as the number of questions increases. Our prediction algorithms track these oscillations, particularly those for the NFMB student, as the NFMB course included more quizzes than the NI course.

Fig. 15 shows the overall performance of our algorithms (i.e., the percentage improvement in RMSE), averaged over all students. We see that both the IFTSNN and FTSNN predictions significantly outperform both the naive baseline and the linear regression baseline for both courses, and that including input data (i.e., IFTSNN vs. FTSNN) further improves the prediction. Also, the lasso regression algorithm performs better than the naive baseline in each case, as expected. We note that, since our clickstream features included a vector of eight inputs at each timeslot, including clickstream data in the prediction algorithm greatly increases the size of the input data and thus the potential for model overfitting; however, the modest performance gains indicate that our training algorithm avoided overfitting for IFTSNN compared to FTSNN.

We next investigate how the two algorithms' performance on different students depends on the number of quizzes the students answered, allowing us to evaluate the early detection capability and compare the two courses in more detail. We then consider the impact of individual clickstream features.

C. Quality by Number of Questions Answered

Figs. 16 and 17 show the average RMSE improvement when grouping students as in groupings A and C, i.e., by number of questions answered. Analyzing Fig. 16(a) and (b), we observe that as the number of quizzes increases, the RMSE improvement compared to the naive baseline decreases, yielding the 61.1% and 49.9% overall RMSE improvements respectively for NFMB (Fig. 15). The IFTSNN's and FTSNN's improvement in RMSE gets better with a smaller number of quizzes answered, before the average CFA begins to stabilize. This *early detection* capability, to work with data as it becomes available at the beginning of the course, is one of the advantages of our system. However, this decrease in improvement does not imply that the IFTSNN and FTSNN algorithms perform worse for students who answer many quiz questions–it simply reflects the fact that the naive baseline performs better. When students have answered only a few quizzes, we expect the naive baseline to perform poorly: at this point, each quiz answer will dramatically change the student's average CFA grade. Thus, the IFTSNN and FTSNN algorithms realize a smaller improvement for students who answer many quizzes: though the baseline algorithms may realize high errors early in the course, they will likely exhibit smaller errors in predicting these students' performance later after they have answered many quiz questions.

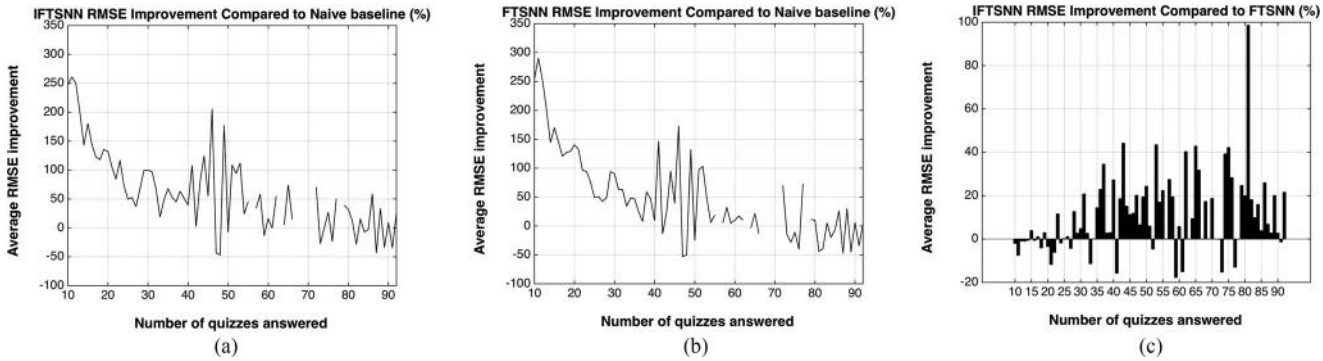


Fig. 16. Performance of grouping A (NFMB students, grouped by the exact number of questions answered). The (a) IFTSNN and (b) FTSNN algorithms improve the average RMSE more compared to the naive baseline for students who answered few questions, while (c) the IFTSNN algorithm improved the average RMSE more compared to the FTSNN algorithm for students who answered more questions. Break points in the lines at $N = 56, 63, 67, 71$ and 76 mean that no data were available for that number of questions, i.e., that no student answered exactly that number of questions.

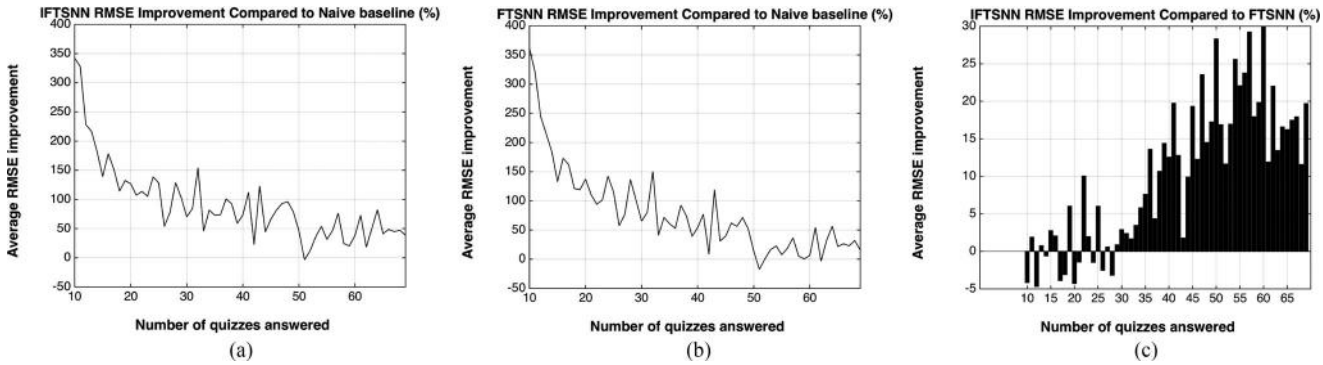


Fig. 17. Performance of grouping C (NI students, grouped by the exact number of questions answered). As for the NFMB students in Figs. 16, the (a) IFTSNN and (b) FTSNN algorithms improve the average RMSE more compared to the naive baseline for students who answered few questions, while (c) the IFTSNN algorithm improved the average RMSE more compared to the FTSNN algorithm for students who answered more questions.

On students who answered fewer than 10 quizzes, the IFTSNN and FTSNN algorithms achieve an average RMSE of 0.0505, indicating that these algorithms perform well on students with extremely small numbers of questions. However, given that we need to 5 initial states to train the model, the testing and training data for these students is very small, leading to a large risk of overfitting. Incorporating data from other students reduces this risk but significantly reduces the model's performance (cf. Fig. 12): we find that if we train the model with 92 quizzes and apply it to students with answering from 10 to 15 quizzes, the average RMSE is much higher, at 0.1866. Students in the NI course exhibit similar results, as shown in Fig. 17(a) and (b).

Comparing the quality of IFTSNN and FTSNN allows us to assess the value of including clickstream data in our prediction algorithms. We find that the clickstream-based input features of IFTSNNs help predict the CFA grade, with an average improvement of 11.5% and 10.1% respectively on groupings A and C (Fig. 15). We might expect that as students answer more quiz questions, the quality of the feedback-only model will improve [10], as the algorithm can be trained on more student data. However, in practice, the IFTSNN model also improves as students answer more questions; Figs. 16(c) and 17(c) show that as the student answers more questions, the IFTSNN model generally realizes a greater improvement. Algorithms trained on these groupings can take advantage of more quiz responses, preventing them from overfitting to a small sample of student clickstream data and accompanying quiz scores.

Finally, we can compare the results of groupings A and C to observe the difference in quality between the predictions in NFMB and NI. The NI students tend to exhibit more consistent improvement than the NFMB students over the naive baseline as the number of quizzes answered increases (Fig. 16 vs. Fig. 17). This is likely due to the larger number of NI students: the percentage improvement for NFMB even dips below zero for some numbers of quizzes answered, due to a small number of students who answered that number of questions. The IFTSNN models for the NI students also demonstrate more consistent improvements over the FTSNN models, compared to NFMB. This result could reflect the fact that the NI course covered material at a more introductory level than the NFMB course, so the NI students were likely less familiar with the background material and may have exhibited less consistent performance, leading the naive baseline algorithm to perform worse and yielding better improvement in quality for our IFTSNN and FTSNN algorithms. These students may also have relied more on the videos to learn the material presented, due to their inexperience; thus, the clickstream input features could yield more insights

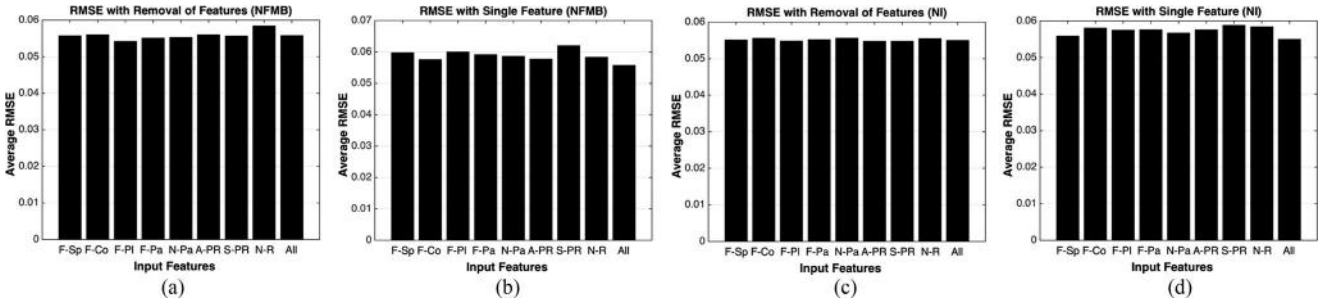


Fig. 18. RMSE averaged over all NFMB students with (a) individual features removed, and (b) a single feature included, compared to the IFTSNN algorithm (far right); RMSE averaged over all NI students with (c) individual features removed, and (d) a single feature included, compared to the IFTSNN algorithm (far right). No single feature dominates the algorithms' RMSE. Feature names indicate the clickstream input feature (a) and (c) removed or (b) and (d) included, with abbreviations defined in Section II-A.

into student performance than for the NFMB course, resulting in more consistent improvement in the IFTSNN compared to the FTSNN algorithms.

D. Feature Importance

While Figs. 16(c) and 17(c) show that including the clickstream input data does improve prediction quality as students answer more quizzes, they do not show the effect of any individual feature. To measure this, we retrain our algorithms with individual features excluded and compare the retrained algorithms' performance to the IFTSNN algorithm (i.e., with all features included). Other feature selection methods can yield similar insights [33], but excluding particular features directly shows the impact of each feature on the network performance. Fig. 18(a) and (c) show the average RMSE when each clickstream input feature is removed; there are no significant changes, particularly for the NI students. The largest decline across the two courses occurred when removing the `N-R` (number of rewinds) feature for the NFMB students, yielding a near 4.5% decline. While this decline is relatively small, it indicates that `N-R` plays an important role in predicting the CFA grade: this feature indicates how frequently students re-watch content, so it may reflect how well they understand the material, and thus their CFA grade. From Fig. 8, `N-R` is not clearly higher for CFA or non-CFA students, indicating a significant but nonlinear relationship between this feature and average CFA grades.

Fig. 18(b) and (d) show the average RMSE over all students in each course with a single clickstream input feature (combined with feedback). We see that the RMSE in both courses visibly increases with only one feature compared to the IFTSNN algorithm with all clickstream features, indicating that each feature does contain information useful for predicting the average CFA grades. Again, no single feature overly contributes to the improved performance, but a combination yields measurably lower RMSE.

E. Online Prediction

In practice, our prediction algorithms will be run in an online manner, with retraining as new student data is recorded. Specifically, each time a student takes another quiz, the student behavior features for that quiz and its associated video are computed, and the neural network parameters are updated accordingly. We can then use the updated neural network to predict future average CFA grades for that student.

quizzes	12	22	32
IFTSNN	0.0627 49.2%	0.0677 16.5%	0.0627 20.2%
quizzes	42	52	62
IFTSNN	0.0732 27.7%	0.0532 69.7%	0.0570 19.7%
quizzes	72	82	92
IFTSNN	0.0460 5.5%	0.0739 9.4%	0.0512 40.8%

Fig. 19. RMSE of online prediction obtained for NFMB students. The right column indicates the performance improvement compared to the lasso regression.

Again, this "early detection" capability, to work with data as it is available, is one of the advantages of our system. Fig. 19 shows the results of our online prediction for NFMB students. Here, students are divided into groups according to the number of quizzes they answered, i.e., $\Omega_A^{s_0}$ according to the notation in Section III-C: Dataset Groupings. For each student, predictions are made on his/her average CFA score after the jth quiz response training on $1, ..., j-1$ for each $j = 6, 7, ..., s_0$; the model is re-trained for each j, and the RMSE is computed for each student and averaged across the group. We see that the achieved RMSEs are consistently low, though they are somewhat smaller for students who answer more quizzes. We would intuitively expect this result, since the average CFA grade stabilizes after students answer several quizzes. Thus, our neural network models can be used for online as well as offline prediction.

V. Conclusion, Discussion, and Future Work

In this paper, we used time series neural networks for personalized prediction of students' average CFA grade in two MOOCs. We considered neural network prediction models that use as inputs only past quiz performance or a combination of past quiz performance and clickstream input data. We showed that video-watching clickstream events can be used as learning features to improve our prediction accuracy. In implementing these prediction algorithms, we employed sophisticated pre-processing to handle the sparsity of available data on student quiz performance. We trained personalized algorithms for

individual students in order to capture unique characteristics of each student's learning patterns. We found that both neural network-based algorithms consistently outperform a naive baseline of simply averaging historical CFA data for each student. We also found that each clickstream input feature is equally important to the algorithms' accuracy, with no single feature contributing the most.

Discussion: From Figs. 16 and 17, we see that our IFTSNN and FTSNN algorithms are especially useful for predicting the performance of students who answer relatively few quizzes, for whom the naive baseline algorithm does worse. Thus, our algorithms can be used to detect students with low average CFA grades early in the course, allowing instructors to automatically target these potentially struggling students with course interventions. Note that the FTSNN algorithms tend to perform slightly better than the IFTSNN algorithms when there are few student quiz answers available, indicating that feedback-only algorithms may be sufficient for designing early-course interventions.

Identifying struggling students early in a course allows instructors to stage a variety of possible interventions to improve these students' performance. Even simply alerting the instructor to students who are predicted to have low average CFA grades can prompt them to give these students more individual attention. In another possible intervention, when our algorithms forecast that a student's average CFA grade will fall below an instructor-specified threshold, the course software could automatically present students with additional, possibly personalized study material for the next course topic [34] before the next video lecture. Instructors could prepare this additional study material in advance based on the topics covered in the course, and perhaps historical information on which topics students generally struggled with. Thus, an important step for future work would be to implement an algorithm in a technology platform that flags students with low predicted average CFAs and presents them with intervention course material.

Future work: Due to the low correlation of the input features and CFA grade and the sparsity of the available time series data, we choose neural networks for our prediction algorithms. However, other time series prediction methods may also be effective compared to the naive baseline; our paper demonstrates the feasibility of using historical quiz performance and clickstream data to predict performance, rather than definitively establishing the "best" type of algorithm to perform these predictions. A promising direction of future work would be to comprehensively compare our results to the accuracy of other types of algorithms, e.g., nearest-neighbor and other neural network approaches, including other network configurations.

In this work, we were primarily concerned with relating users' video watching behavior to their quiz performance, independent of the specific course topics each quiz covers. Future work could augment our neural network method to be topic specific. One possibility would be to use behavioral data to train these recurrent neural networks based on the topics of the particular videos a student has watched. This could be done e.g., by applying topic extraction to the textual component (audio track) of the video and weighing the inputs to the network based on the similarity of these videos to upcoming quiz questions.

Our model can easily be extended to real-world (offline) classroom scenarios. Instead of using clickstream data inputs, we could use in-course data such as the number of times that students ask instructors questions, how much time they spend studying, etc. to predict students' average grades throughout the course. While many traditional courses do not include a single quiz question after each module, we could instead predict students' average test scores or homework grades based on these input features. Even in a MOOC context, we could use social learning networks (SLNs) [6], [8] to enhance prediction performance by incorporating features like the number of questions that students post in online course forums. Another direction of future work would be to investigate whether the students who are predicted to have low course grades perform better after different types of instructor interventions, which may indicate not only the efficacy of different intervention methods but also our algorithms' effectiveness at identifying truly struggling students.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their valuable comments; and also would like to thank the Coursera students from the NFMB and NI courses, whose behavior they analyze in this paper.

REFERENCES

[1] D. Shah, "By the numbers: MOOCs in 2015," 2015. [Online]. Available: https://www.class-central.com/report/moocs-2015-stats

[2] C. Piech *et al.*, "Deep knowledge tracing," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 505–513.

[3] M. Keramida, "What is wrong with MOOCs? Key points to consider before launching your first MOOC," 2015. [Online]. Available: https://elearningindustry.com

[4] C. G. Brinton, M. Chiang, and H. V. Poor, "Mining MOOC clickstreams: Video-watching behavior vs. in-video quiz performance," *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3677–3692, Jul. 15, 2016.

[5] A. S. Lan, A. E. Waters, C. Studer, and R. G. Baraniuk, "Sparse factor analysis for learning and content analytics," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1959–2008, 2014.

[6] C. G. Brinton, S. Buccapatnam, F. M. F. Wong, M. Chiang, and H. V. Poor, "Social learning networks: Efficiency optimization for MOOC forums," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun.*, 2016, pp. 1–9.

[7] C. Qing, Y. Chen, D. Liu, C. Shi, Y. Wu, and H. Qu, "Peakvizor: Visual analytics of peaks in video clickstreams from massive open online courses," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 10, pp. 2315–2330, Oct. 2016.

[8] C. G. Brinton and M. Chiang, "Social learning networks: A brief survey," in *Proc. 2014 48th Annu. Conf. Inf. Sci. Syst.*, 2014, pp. 1–6.

[9] C. G. Brinton, R. Rill, S. Ha, M. Chiang, R. Smith, and W. Ju, "Individualization for education at scale: MIIC design and preliminary evaluation," *IEEE Trans. Learn. Technol.*, vol. 8, no. 1, pp. 136–148, Jan.–Mar. 2015.

[10] C. G. Brinton and M. Chiang, "MOOC performance prediction via clickstream data and social learning networks," in *Proc. 2015 IEEE Conf. Comput. Commun.*, IEEE, 2015, pp. 2299–2307.

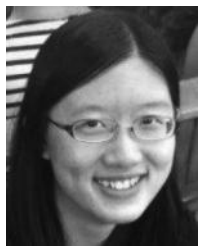
[11] S. Zheng, M. B. Rosson, P. C. Shih, and J. M. Carroll, "Understanding student motivation, behaviors and perceptions in MOOCs," in *Proc. 18th ACM Conf. Comput. Supported Cooperative Work Social Comput.*, ACM, 2015, pp. 1882–1895.

[12] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec, "Engaging with massive online courses," in *Proc. 23rd Int. Conf. World Wide Web*, ACM, 2014, pp. 687–698.

[13] R. F. Kizilcec and E. Schneider, "Motivation as a lens to understand online learners: Toward data-driven design with the OLEI scale," *ACM Trans. Comput.-Human Interact.*, vol. 22, no. 2, 2015, Art. no. 6.
[14] C. G. Brinton, M. Chiang, S. Jain, H. Lam, Z. Liu, and F. M. F. Wong, "Learning about social learning in MOOCs: From statistical analysis to generative model," *IEEE Trans. Learn. Technol.*, vol. 7, no. 4, pp. 346–359, Oct.–Dec. 2014.
[15] J. Kim, P. J. Guo, D. T. Seaton, P. Mitros, K. Z. Gajos, and R. C. Miller, "Understanding in-video dropouts and interaction peaks in online lecture videos," in *Proc. ACM Conf. Learn. @ Scale Conf.*, ACM, 2014, pp. 31–40.
[16] T. Sinha, P. Jermann, N. Li, and P. Dillenbourg, "Your click decides your fate: Inferring information processing and attrition behavior from MOOC video clickstream interactions," in *Proc. ACL Emperical Methods Natural Language Process.*, ACL, 2014, pp. 3–14.
[17] A. S. Lan, C. Studer, and R. G. Baraniuk, "Time-varying learning and content analytics via sparse factor analysis," in *ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, ACM, 2014, pp. 452–461.
[18] Y. Bergner, S. Droschler, G. Kortemeyer, S. Rayyan, D. Seaton, and D. E. Pritchard, "Model-based collaborative filtering analysis of student response data: Machine-learning item response theory," in *Proc. Int. Conf. Educ. Data Mining*. ERIC, 2012, pp. 95–102.
[19] Y. Meier, J. Xu, O. Atan, and M. van der Schaar, "Predicting grades," *IEEE Trans. Signal Process.*, vol. 64, no. 4, pp. 959–972, Feb. 2016.
[20] C. Romero, M.-I. López, J.-M. Luna, and S. Ventura, "Predicting students' final performance from participation in online discussion forums," *Comput. Educ.*, vol. 68, pp. 458–472, 2013.
[21] M. Kloft, F. Stiehler, Z. Zheng, and N. Pinkwart, "Predicting MOOC dropout over weeks using machine learning methods," in *Proc. 2014 Conf. Empirical Methods Natural Language Process Workshop Model. Large Scale Social Interact. Massively Open Online Courses*, 2014, pp. 60–65.
[22] J. Qiu *et al.*"Modeling and predicting learning behavior in MOOCs," in *Proc. 9th ACM Int. Conf. Web Search Data Mining*, ACM, 2016, pp. 93–102.
[23] Z. Ren, H. Rangwala, and A. Johri, "Predicting performance on MOOC assessments using multi-regression," in *Proc. 9th Int. Conf. Educational Data Mining*, 2016, pp. 484–489.
[24] M. Chiang, "Networks: Friends, money, and bytes." Sept. 2012. [Online]. Available: https://www.coursera.org/course/friendsmoneybytes
[25] C. G. Brinton and M. Chiang, "Networks illustrated: Principles without calculus." Jun. 2013. [Online]. Available: https://www.coursera.org/learn/networks-illustrated
[26] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc. Series B (Methodological)*, pp. 267–288, 1996.
[27] M. Khajah, R. V. Lindsey, and M. C. Mozer, "How deep is knowledge tracing?" in *Proc. 9th Int. Conf. Educational Data Mining*, 2016, pp. 94–101.
[28] E. M. Azoff, *Neural Network Time Series Forecasting of Financial Markets*. New York, NY, USA: Wiley, 1994.
[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[30] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. 2014 Conf. Empirical Methods in Natural Language Processing*, 2014, pp. 1724–1734.
[31] Bayesian regularization backpropagation, 2016. [Online]. Available: http://www.mathworks.com/help/nnet/ref/trainbr.html
[32] J. Henderson, "A neural network parser that handles sparse data," in *Proc. 6th Int. Workshop Parsing Technol.*, Citeseer, 2000, pp. 123–134.
[33] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.
[34] C. Tekin, J. Braun, and M. van der Schaar, "eTUTOR: Online learning for personalized education," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2015, pp. 5545–5549.

Tsung-Yen Yang is currently a Senior Undergraduate Student in the National Chiao Tung University, Hsinchu, Taiwan. His major is in electrical engineering and computer science. He is the cofounder of the IOT company Orzda, Inc., Hsinchu, Taiwan. He was a Research Scholar in the University of California, Los Angeles, CA, USA, for two months in 2015 and in Princeton University, Princeton, NJ, USA, for two months in 2016. His research interest focuses on using machine learning to solve specific problem and programmable logic circuit design.

Christopher G. Brinton (S'08–M'16) received the Ph.D. degree from Princeton University, Princeton, NJ, USA, in 2016, the Master's degree from Princeton in 2013, and the BSEE degree (*valedictorian* and *summa cum laude*) from The College of New Jersey, Ewing Township, NJ, USA, in 2011, all in electrical engineering. He is currently the Head of the Department of Advanced Research, Zoomi, Inc., Malvern, PA, USA, a learning technology company he cofounded in 2013. He coauthored the book *The Power of Networks: Six Principles That Connect our Lives* (Princeton Univ. Press, 2016), and has reached more than 250 000 students through MOOCs based on his book. His research interest focuses on developing systems and methods to improve the quality of student learning, through big learning data analytics, social learning networks, and individualization. He received the 2016 Bede Liu Best Dissertation Award in electrical engineering.



Carlee Joe-Wong (S'11–M'16) received the Ph.D. degree at Princeton University's Program in applied and computational mathematics, the A.B. degree in mathematics in 2011, and the M.A. degree in applied mathematics in 2013, from Princeton University, Princeton, NJ, USA. She is currently an Assistant Professor in the Department of Electrical and Computer Engineering, Carnegie Mellon University, Moffett Field, CA, USA. Her research interests include network economics and optimal control. In 2013, she was the Director of Advanced Research at DataMi, a startup she cofounded in 2012 that commercializes new ways of charging for mobile data. She received the INFORMS ISS Design Science Award in 2014 and the Best Paper Award at the IEEE INFOCOM 2012. In 2011, she received a National Defense Science and Engineering Graduate Fellowship.

Mung Chiang (S'00–M'03–SM'08–F'12) is currently the Arthur LeGrand Doty Professor of electrical engineering at Princeton University, Princeton, NJ, USA, 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. 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 received a TR35 Young Innovator Award, he created the Princeton EDGE Lab in 2009 to bridge the theory-practice divide in networking by spanning from proofs to prototypes, resulting in several technology transfers to industry and two startup companies. He also received the 2013 Frederick E. Terman Award from the American Society of Engineering Education. He was named a Guggenheim Fellow in 2014.