Personalization and Contextualization of Large Language Models For Improving Early Forecasting of Student Performance

Ahatsham Hayat Mohammad Rashedul Hasan
Department of Electrical and Computer Engineering
University of Nebraska-Lincoln
aahatsham2@huskers.unl.edu, hasan@unl.edu

Abstract

Early forecasting of student performance in a course is a critical component of building effective intervention systems. However, when the available student data is limited, accurate early forecasting is challenging. We present a language generation transfer learning approach that leverages the general knowledge of pre-trained language models to address this challenge. We hypothesize that early forecasting can be significantly improved by fine-tuning large language models (LLMs) via personalization and contextualization using data on students' distal factors (academic and socioeconomic) and proximal non-cognitive factors (e.g., motivation and engagement), respectively. Results obtained from extensive experimentation validate this hypothesis and thereby demonstrate the prowess of personalization and contextualization for tapping into the general knowledge of pre-trained LLMs for solving the downstream task of early forecasting.

1 Introduction

Modern artificial intelligence (AI) methods, such as deep learning (DL), have been used as a cost-effective way to build early-warning systems for providing forecasting-based interventions in various domains, e.g., health (1; 2; 3; 4; 5; 6; 7; 8) and education (9; 10; 11; 12). Specifically, in education, cognitive data on students' assessment scores in a course are used to build AI-based forecasting interventions for changing behavior toward improving academic performance (13; 14; 15). The effectiveness of the interventions, however, depends on the accuracy of early forecasting, i.e., how early in the semester the overall performance of the course can be accurately forecasted (16; 17). This problem is particularly challenging when the amount of training data is limited. In such a case, training a neural network model (e.g., a recurrent, convolutional, or Transformer) from scratch does not yield satisfactory performance. On the other hand, transfer learning is challenging due to the unavailability of relevant pre-trained models or a large dataset on a similar task to pre-train a model (18).

In this paper, we design a transfer learning approach for early forecasting of learning outcomes in an undergraduate STEM (science, technology, engineering, and mathematics) course by using generative deep learning (19), specifically by leveraging a Transformer-based (20) pre-trained language model (LM). With rapid progress in developing general-purpose LMs (21; 22; 23; 24) capable of storing vast in-depth knowledge about the world (25; 26), and solving complex tasks via basic reasoning (22; 27; 28) and planning (29), LMs can be leveraged to scaffold AI for early-warning forecasting interventions. To this effect, we formulate performance forecasting as a natural language generation problem. Specifically, we adapt a pre-trained LM with text sequences of students' academic trajectory data for generating another text sequence containing their end-of-the-semester predicted performance. Since we are interested in early forecasting, we use data sequences of varying lengths from the

NeurIPS'23 Workshop on Generative AI for Education (GAIED).

beginning of the semester up to the middle of the semester (e.g., 2 weeks, 4 weeks, and 8 weeks) in a 16-week long semester for LM adaptation. In addition, we experiment with three types of features of students' academic trajectories, i.e., their (i) distal factors (academic meta-information and socioeconomic), (ii) proximal cognitive factors (formative and summative scores in cognitive tests), and (iii) proximal non-cognitive factors (repeated measures of non-cognitive attributes such as engagement). Our collected data is ordinal (numeric or real-valued), thus we verbalize it in natural language text sequences for adapting LMs and then augment it to balance the distribution of different performance types. Finally, our experimentation includes LMs of varying capacities (with respect to the number of parameters) with the aim of understanding the impact of large LMs (LLMs) on improving early forecasting.

Personalization and Contextualization. Distal features are employed to investigate the impact of personalization on the model's performance. Drawing on insights from Social Cognitive Career Theory (30), we propose our working hypothesis (Hypothesis 1): a student's academic trajectory and future performance in a course can be correlated with their background distal factors, thus serving as a valuable prior for the learning model to capture nuanced individualized patterns in their academic progression. On the other hand, we incorporate proximal non-cognitive features, measured concurrently with the proximal cognitive features, to offer contextual signals (31) regarding students' study-related behavior. Building upon previous research (32) that examines the influence of non-cognitive attributes, such as motivation and engagement, on students' learning outcomes, our working hypothesis (Hypothesis 2) posits that longitudinal assessments of these non-cognitive factors exhibit a stronger correlation with a student's evolving academic trajectory. Consequently, we propose that they should enhance a model's ability to discern subtle variations in academic performance that may not be captured solely from cognitive trajectory data. Hence, the fusion of distal and non-cognitive features with cognitive features should empower a learning model to predict a student's future academic performance early in the semester. We validate these hypotheses through experiments involving different combinations of these three feature types and address the following research questions.

- RQ1: Is the natural language generation approach more effective than numeric feature-based models for early forecasting of academic performance?
- RQ2: Do personalization and contextualization improve the LM's early forecasting efficacy?
- RQ3: How does the LM capacity (number of parameters) influence its forecasting performance?

Our main contributions comprise the development of a natural language generation approach for early forecasting of student performance by leveraging the general knowledge of pre-trained LMs. Most importantly, we demonstrate that the forecasting efficacy can be significantly enhanced by leveraging large models or LLMs through personalization and contextualization.

2 Method

We consider student end-of-the-semester performance prediction as a time-series learning problem and formulate it as a natural language generation problem. It has been shown that time-series forecasting can be effectively done by adapting pre-trained LMs (33; 34) for solving a natural language generation task. This approach achieved competitive performance compared to numeric-only Transformer-based time-series learning approaches (34). The language generation approach requires both the input and output data to be in a natural language format for adapting the LMs. Since our original dataset (X, Y) is numeric, first, we transform it into a natural language dataset (X_{text}, Y_{text}). Below we provide a formal description of the natural language generation problem, followed by a description of the natural language dataset development process.

2.1 Problem Formulation

The numeric (ordinal real-valued) input data $X = \{x_1, x_2, ..., x_n\}$ from n students contains three types of features, i.e., (i) one-time measure of distal factors $D = \{d_1, d_2, ..., d_p\}$ collected at the beginning of the semester, (ii) repeated measures of proximal cognitive factors starting from the beginning of the semester up to time t, formalized by $C^t = \{C^1, C^2, ..., C^t\}$, where C^t measures q features at time t, i.e., $\{c_1^t, c_2^t, ..., c_q^t\}$, and (iii) repeated measures of proximal non-cognitive factors starting from

the beginning of the semester up to time t, formalized by $NC^t = \{NC^1, NC^2, \ldots, NC^t\}$, where NC^t measures r features at time t, i.e., $\{nc_1^t, nc_2^t, \ldots, nc_r^t\}$. The distal and proximal measures (from the beginning of the semester up to time t) are concatenated to create each sequence (D, C^t, NC^t) in X. The output data $Y = \{y_1, y_2, \ldots, y_n\}$ contains end-of-semester performance which is grouped into four types: at-risk (grade C or below), prone-to-risk (grade above C but below B), average (grade above B but below A), and outstanding (grade A or above).

For fine-tuning the LM, the numeric dataset (X, Y) is verbalized into a natural language dataset (X_{text}, Y_{text}) (described further below). Specifically, each sequence in X_{text} and Y_{text} contains standard lexical literals used in English (e.g., words and phrases). The input text sequence, which is used as a prompt for the LM, includes suitable instructions for solving a task (e.g., predicting future performance) for instruction fine-tuning (35). The output text sequence captures a full expression of the prediction.

For the LM, an encoder-decoder architecture is used. The encoder $f_E(.)$ maps the input sequence $(x_{text_1}, x_{text_2}, \ldots, x_{text_l})$ to an intermediate latent embedding sequence $(z_1, z_2, \ldots, z_l)$.

$$z = f_E(x_{text_1}, x_{text_2}, \ldots, x_{text_l}; \theta_E)$$

where θ_E are the weights of the encoder.

The decoder $f_D(.)$ takes the latent embeddings $(z_1, z_2, \ldots, z_l)$ to generate an output sequence $(\hat{y}_{text_1}, \hat{y}_{text_2}, \ldots, \hat{y}_{text_m})$ in an auto-regressive fashion, i.e., at each step the decoder $f_D(.)$ uses previously generated symbols $\hat{y}_{text_{<m}}$ as additional input for generating the next token $\hat{y}_{text_m}$. The probability of generating the m-th token $\hat{y}_{text_m}$ is given by

$$p(\hat{y}_{text_m} | \hat{y}_{text_{<m}}; z_1, z_2, \ldots, z_l) = softmax(f_D(\hat{y}_{text_{<m}}; z_1, z_2, \ldots, z_l; \theta_D))$$

where θ_D are the weights of the decoder. For fine-tuning the encoder-decoder LM, the multi-class cross-entropy loss function is used. The number of classes in the loss function is set by the total number of tokens in the vocabulary. For a batch size B, the loss function is:

$$L = -\sum_{b=1}^{B}\sum_{m=1}^{M} y^b_{text_m} \log \hat{y}^b_{text_m}$$

2.2 Language Dataset Development

Creating the Numeric Dataset. We collected data on numeric measures of distal factors and time-varying proximal factors (cognitive and non-cognitive) of N = 48 first-year college students enrolled in an introductory programming course on MATLAB at a large public university in the USA. The distal data is 9-dimensional and was collected at the beginning of the semester. It includes students' course-related meta-information (class standing and major) and socioeconomic status (gender, race, international or native student, parents' education background, highest education level of a single parent, highest education level of another parent, family yearly income, science identity, and reflected science identity). The proximal cognitive data is 41-dimensional and includes students' assessment scores (formative and summative) in the 16-week course (12 homework assignments, 12 labs, 6 diaries, 8 quizzes, 2 projects, and 1 final exam). This data was obtained from the course's learning management system, namely Canvas. The proximal non-cognitive data is 28-dimensional and includes repeated measures of students' motivation (intrinsic and extrinsic) and engagement (behavioral, emotional, and cognitive) factors throughout the duration of the 16-week semester. The non-cognitive data were collected through a smartphone-based application that triggered contextually appropriate, study-specific daily questions based on rules specified by researchers. Participants' answers were aggregated on secure, cloud-based servers for analysis.

Due to the high dimensionality of the data (78-dimensional feature space) across three types of features (9-dimensional distal, 41-dimensional cognitive proximal, and 28-dimensional non-cognitive proximal), measures beyond the initial weeks result in lengthy sequences for each student. After verbalization, the sequences become even longer. Due to the limited size of the input context window of the LMs (512 tokens) we used in this research, we needed to shorten the length of the sequences by utilizing only a subset of the three types of features. Specifically, we manually selected the following features: 5-dimensional distal factors (class standing, major, gender, race, and family yearly income), 21-dimensional proximal cognitive factors spanning over the first 8 weeks of the semester (first 4 Diaries, 6 Labs, 4 Quizzes, 6 Homework Assignment, and 1 Project), and 2-dimensional proximal non-cognitive factors (i.e., two measures of emotional engagement). These three types of features

were used to create the numeric sequences in input data X. We interspersed the three features to maintain their temporal order.

Note that we adapted the LMs using data sequences of varying lengths, e.g., sequences spanning over the first 2 weeks, 4 weeks, and 8 weeks. Thus, the number of cognitive features varies based on the length of a sequence. For example, for the 8-week long sequence, we used 21 cognitive features; for the 4-week long sequence, we used 10 cognitive features; and for the 2-week long sequence, we used 4 cognitive features. The numeric output data Y was created by using the end-of-semester final letter grade. We categorized the output into four performance groups: at-risk, prone-to-risk, average, and outstanding. The groups were coded by integers 0 to 3, respectively. Finally, the input and output data were combined to create a numeric dataset (X, Y). We created three numeric datasets based on 8-week-long, 4-week-long, and 2-week-long input sequences.

Pre-processing the Numeric Dataset. The non-cognitive longitudinal data contained missing values caused by participants' skipping questions or temporarily uninstalling the app. We identified two types of missing values, (i) responses to all questions on a day were missing, and (ii) responses to a fraction of the questions were missing. For the first case, we used the Last Observation Carried Forward (LOCF) imputation method (36). However, in some cases we could not find a previous day with all questions answered, so we used a matching future day. Addressing the second case was challenging due to the presence of missed follow-up questions. When the response to the trigger question on the previous day differed, copying the response for the follow-up question using LOCF would be unreliable (37). To remedy this, we searched for a previous day in which the participant responded to both the trigger question and the follow-up question, and the trigger question's response was the same as the missing day's trigger question's response. In such a case, we applied the LOCF method on the matched previous day. If no matching previous days were found, we used a matching future day for imputation.

Creating the Language Dataset. We created a language generation dataset (X_{text}, Y_{text}) by transforming the numeric dataset (X, Y) into natural language text. For this transformation, we designed a template (see the Appendix) comprising two components: the verbalized input sequence and the verbalized output sequence. In addition, for making the adaptation of the LM amenable to instruction fine-tuning (35), the input sequences include suitable additional information, such as we prepended the following message to the student's proximal cognitive information: "A student obtained the following assessment scores in an introductory programming course ...", and the following message to the student's distal information: "Some background information about the student: ...".

Augmenting the Language Dataset. The verbalized data (X_{text}, Y_{text}) contains an unbalanced distribution of 48 input-output text sequences (24 outstanding, 12 average, 6 prone-to-risk, 6 at risk). Because of the small size of the dataset and its skewed distribution, it is challenging to effectively fine-tune an LM and reliably evaluate it. To address this challenge, we augmented the verbalized data by using the oversampling method that was based on the random sampling technique (38; 39). The oversampling method involves duplicating the samples from each class by introducing token variations using the synonym replacement method (40). We employed a straightforward heuristic to increase the number of samples in each class. Specifically, we duplicated the existing samples by an integer multiple, using smaller multiples for the larger classes and vice versa. For example, we doubled the instances in the largest class while increasing the samples of the minority classes by a factor of 5. The resulting 144 samples have a near-balance distribution of classes, i.e., 48 outstanding, 36 average, 30 prone-to-risk, and 30 at risk. For creating the test datasets, we sample ≈ 30% instances from the augmented datasets by maintaining a balanced class distribution.

3 Experiments

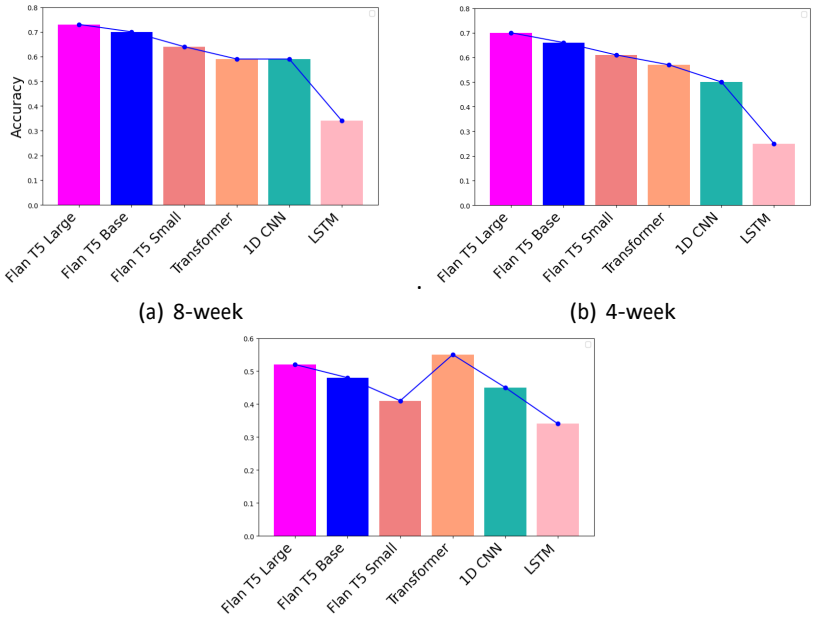
To address the three research questions given in Section 1, we conducted a set of four experiments, i.e., Experiment 1 (C + NC + D), Experiment 2 (C + NC), Experiment 3 (C + D), and Experiment 4 (C). These experiments are based on different combinations of the three types of features, i.e., distal (D), proximal cognitive (C), and proximal non-cognitive (NC). In each experiment, we fine-tuned the LMs using three language datasets created from proximal data of varying lengths: 8-week, 4-week, and 2-week. Furthermore, in the LM-based experiments, we utilized LMs with varying capacities in terms of parameters (small, medium, and large). The LMs were evaluated by searching for matching keywords from their predicted output sequences for the performance types.

Experimental Setting. For the encoder-decoder LM, we used FLAN-T5 (41), which is a variant of the T5 model (25). The FLAN-T5 model is instruction fine-tuned, making it suitable for our purposes. We employed FLAN-T5 with three different capacities, determined by the number of parameters: FLAN-T5-Small (80M), FLAN-T5-Base (250M), and FLAN-T5-Large (770M). These LMs have a context window limited to 512 tokens. For the numeric feature-based baseline models, we used three types of neural models, i.e., the sequential long short-term memory (LSTM) (42), non-sequential convolutional neural network (CNN) with a one-dimensional convolutional kernel (43), and a Transformer (20)-based encoder. Additional information on the baseline models is provided in the Appendix.

The baseline models were trained using 3 variably-length numeric datasets containing only the cognitive features. Exploring baseline models with all three feature types is planned as future work. To ensure compatibility with the LM-based experiments, the numeric datasets were created from the augmented verbalized datasets by decoding the cognitive feature part of text sequences into numeric values. We used the same test sets to evaluate both model types, employing the following metrics: accuracy, precision, recall, and F1 score.

3.1 Results

RQ1: Is the natural language generation approach more effective than numeric feature-based models for early forecasting of academic performance? Figure 1 presents a performance comparison between three LMs and three baseline numeric feature-based neural models using three datasets of varying lengths. For the 8-week and 4-week data, LMs of all sizes outperform the baseline models. For the 2-week data, the numeric Transformer model (accuracy=55%) shows a slight improvement over the large LM (with an accuracy of 52%). However, it's important to note that, unlike the LMs, the numeric Transformer model does not exhibit improvement with more data; its accuracies for the 4-week and 8-week datasets are 57% and 59%, respectively. In contrast, the large LM achieves an accuracy of ≥ 70% with the 4-week and 8-week datasets. A detailed comparison of the performance of the baseline models is provided in the Appendix.



(a) 8-week (b) 4-week

(c) 2-week

Figure 1: Comparison of the models (cognitive feature-based).

RQ2: Do personalization and contextualization improve the LM's early forecasting efficacy? To address RQ2, we use the evaluation statistics (see Table 1) of the best-performing large LM (i.e., FLAN-T5-Large). The models were fine-tuned with four combinations of the 3 feature types using the 8-week, 4-week, and 2-week language datasets. We see that personalized and contextualized LMs (i.e., fine-tuned with cognitive (C), non-cognitive (NC), and distal (D) features) exhibit the best performance. With the use of these three features, the LM can forecast student performance with an accuracy of 77% as early as the end of the 2nd week of the semester. To enable effective early

intervention, achieving high recall for the at-risk (AR) and prone-to-risk (PR) groups is imperative. The recall for the AR group by the 2-week model is 100%. The 4-week LM achieves a recall of 100% for the at-risk (AR) group and 80% for the prone-to-risk (PR) group, both of which are critical. Finally, with more data, the 8-week model archives 89% accuracy.

Table 1: Evaluation of the large LM (FLAN-T5-Large) fine-tuned with four combinations of the 3 feature types using the 8-week, 4-week, and 2-week datasets. The best results are in bold.
Legends: C=Cognitive, NC=Non-Cognitive, D=Distal, AR=At-Risk, PR=Prone-To-Risk, AV=Average, OU=Outstanding, P=Precision, R=Recall, F1=F1 Score, A=Accuracy

Features	Class	8-week				4-week				2-week			
		P	R	F1	A	P	R	F1	A	P	R	F1	A
C + NC + D	AR	0.78	1.00	0.88	0.89	1.00	1.00	1.00	0.84	0.64	1.00	0.78	0.77
	PR	0.89	0.80	0.84		0.89	0.80	0.84		1.00	0.50	0.67	
	AV	0.92	1.00	0.96		0.71	0.91	0.80		0.73	1.00	0.85	
	OU	0.93	0.81	0.87		0.86	0.75	0.80		0.85	0.69	0.76	
C + NC	AR	0.70	1.00	0.82	0.82	0.70	1.00	0.82	0.77	0.62	0.71	0.67	0.68
	PR	1.00	0.60	0.75		0.86	0.60	0.71		0.71	0.50	0.59	
	AV	0.73	1.00	0.85		0.69	1.00	0.81		0.62	0.91	0.74	
	OU	0.92	0.75	0.83		0.91	0.62	0.74		0.77	0.62	0.69	
C + D	AR	0.78	1.00	0.88	0.77	0.88	1.00	0.93	0.77	0.60	0.86	0.71	0.64
	PR	0.89	0.80	0.84		0.71	1.00	0.83		0.71	0.50	0.59	
	AV	0.67	0.73	0.70		0.69	0.82	0.75		0.70	0.64	0.67	
	OU	0.79	0.69	0.73		0.89	0.50	0.64		0.59	0.62	0.61	
C	AR	0.60	0.86	0.71	0.73	0.62	0.71	0.67	0.70	0.36	0.57	0.44	0.52
	PR	0.86	0.60	0.71		0.67	0.60	0.63		0.88	0.70	0.78	
	AV	0.60	0.82	0.69		0.67	0.91	0.77		0.54	0.64	0.58	
	OU	0.92	0.69	0.79		0.83	0.62	0.71		0.42	0.31	0.36	

On the other hand, LMs fine-tuned with only cognitive data (no personalization or contextualization) perform worst on all datasets exhibiting as low as 52% accuracy by the 2-week model. The recall for the at-risk group in this early model is low (57%), making it unreliable. Even with the 8-week data, the LM can achieve only 86% recall for the at-risk group and 60% recall for the prone-to-risk group.

A comparison between the influence of personalization (cognitive + distal) and contextualization (cognitive + non-cognitive) shows that contextualization is slightly more effective in increasing the model's forecasting efficacy. For example, the accuracy of the 8-week contextualized model is 82%, while that of the 8-week personalized model is 77%. It's important to note that both the personalized-only and contextualized-only models perform equally well for the at-risk group, and the difference in their performance is minimal. More importantly, for achieving optimal forecasting power, both personalization and contextualization are essential but not sufficient, as we demonstrate below when addressing RQ3.

RQ3: How does the LM capacity (number of parameters) influence its forecasting performance? A comparison of the test accuracies among the three variably-size LMs based on cognitive features (see Figure 1) shows that as the model size gets bigger, the LM acquires more capacity for improved forecasting. Even after personalization and contextualization, the at-risk group recall for both the small and medium models is 86%, whereas the large model obtains 100% recall (see Appendix for a detailed comparison of the variably-size LMs). Thus, evidently, optimal early forecasting via personalization and contextualization of LMs is achievable when we utilize large LMs (LLMs).

Discussion. Our experimental results validate the two hypotheses. Specifically, we show that personalization and contextualization facilitate the early forecasting capability of the LM. However, tapping into the intricacies of personal and contextual signals from students' academic trajectories is contingent on the quality of the prior, i.e., the quality of LM's general knowledge. In other words, the general knowledge prior is effective only when we utilize LLMs. Thus, our research emphasizes the significance of personalization and contextualization to unleash the potential of pre-trained LLMs toward early forecasting. Apart from small data, the key constraint that narrowed the scope of our investigation is the limited memory of available GPUs. Due to this limitation, we could not utilize the full dimension of the distal and proximal non-cognitive features. Also, the small memory of the GPUs prevented us from using rich and expressive instructions in the prompts. Finally, we could not leverage LLMs with $\geq$ 1 billion parameters. In the future, our effort will be to circumvent these limitations for harnessing the full potential of LLMs.

Acknowledgments and Disclosure of Funding

This project was supported by a grant from the U.S. National Science Foundation (NSF DUE 2142558).

References

[1] D. A. Adler, F. Wang, D. C. Mohr, and T. Choudhury, "Machine learning for passive mental health symptom prediction: Generalization across different longitudinal mobile sensing studies," PLOS ONE, vol. 17, p. e0266516, Apr. 2022. Publisher: Public Library of Science.

[2] A. Mamun, K. S. Leonard, M. P. Buman, and H. Ghasemzadeh, "Multimodal Time-Series Activity Forecasting for Adaptive Lifestyle Intervention Design," in 2022 IEEE-EMBS International Conference on Wearable and Implantable Body Sensor Networks (BSN), pp. 1–4, Sept. 2022. ISSN: 2376-8894.

[3] J. Zhao, Q. Feng, P. Wu, R. A. Lupu, R. A. Wilke, Q. S. Wells, J. C. Denny, and W.-Q. Wei, "Learning from Longitudinal Data in Electronic Health Record and Genetic Data to Improve Cardiovascular Event Prediction," Scientific Reports, vol. 9, p. 717, Jan. 2019.

[4] A. G. Horwitz, S. D. Kentopp, J. Cleary, K. Ross, Z. Wu, S. Sen, and E. K. Czyz, "Using machine learning with intensive longitudinal data to predict depression and suicidal ideation among medical interns over time," Psychological Medicine, pp. 1–8, Sept. 2022.

[5] H. Liu, X. Zhang, H. Liu, and S. T. Chong, "Using Machine Learning to Predict Cognitive Impairment Among Middle-Aged and Older Chinese: A Longitudinal Study," International Journal of Public Health, vol. 68, p. 1605322, Jan. 2023.

[6] A. C. Collins, D. Lekkas, M. D. Nemesure, T. Z. Griffin, G. Price, A. Pillai, S. Nepal, M. V. Heinz, A. T. Campbell, and N. C. Jacobson, "Semantic signals in self-reference: The detection and prediction of depressive symptoms from the daily diary entries of a sample with major depressive disorder," Mar. 2023.

[7] X. Xu, X. Liu, H. Zhang, W. Wang, S. Nepal, Y. Sefidgar, W. Seo, K. S. Kuehn, J. F. Huckins, M. E. Morris, P. S. Nurius, E. A. Riskin, S. Patel, T. Althoff, A. Campbell, A. K. Dey, and J. Mankoff, "GLOBEM: Cross-Dataset Generalization of Longitudinal Human Behavior Modeling," Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 6, pp. 190:1–190:34, Jan. 2023.

[8] D. A. Adler, D. Ben-Zeev, V. W.-S. Tseng, J. M. Kane, R. Brian, A. T. Campbell, M. Hauser, E. A. Scherer, and T. Choudhury, "Predicting Early Warning Signs of Psychotic Relapse From Passive Sensing Data: An Approach Using Encoder-Decoder Neural Networks," JMIR mHealth and uHealth, vol. 8, p. e19962, Aug. 2020.

[9] R. Wang, P. Hao, X. Zhou, A. T. Campbell, and G. Harari, "SmartGPA: How Smartphones Can Assess and Predict Academic Performance of College Students," GetMobile: Mobile Computing and Communications, vol. 19, pp. 13–17, Mar. 2016.

[10] R. Wang, F. Chen, Z. Chen, T. Li, G. Harari, S. Tignor, X. Zhou, D. Ben-Zeev, and A. T. Campbell, "StudentLife: assessing mental health, academic performance and behavioral trends of college students using smartphones," in Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '14, (New York, NY, USA), pp. 3–14, Association for Computing Machinery, Sept. 2014.

[11] X. Li, X. Zhu, X. Zhu, Y. Ji, and X. Tang, "Student Academic Performance Prediction Using Deep Multi-source Behavior Sequential Network," in Advances in Knowledge Discovery and Data Mining (H. W. Lauw, R. C.-W. Wong, A. Ntoulas, E.-P. Lim, S.-K. Ng, and S. J. Pan, eds.), Lecture Notes in Computer Science, (Cham), pp. 567–579, Springer International Publishing, 2020.

[12] W. Xu and F. Ouyang, "The application of AI technologies in STEM education: a systematic review from 2011 to 2021," International Journal of STEM Education, vol. 9, p. 59, Sept. 2022.

[13] N. Greenstein, G. Crider-Phillips, C. Matese, and S.-W. Cho, "Predicting Student Outcomes to Drive Proactive Support: An Exploration of Machine Learning to Advance Student Equity & Success," tech. rep., University of Oregon, 2021.

[14] K. E. Arnold and M. D. Pistilli, "Course Signals at Purdue: Using Learning Analytics to Increase Student Success," Proceedings of the 2nd International Conference on Learning Analytics and Knowledge, pp. 267–270, 2012.

[15] L. T. Liu, S. Wang, T. Britton, and R. Abebe, "Reimagining the machine learning life cycle to improve educational outcomes of students," Proceedings of the National Academy of Sciences, vol. 120, p. e2204781120, Feb. 2023. Publisher: Proceedings of the National Academy of Sciences.

[16] M. R. Hasan and M. Aly, "Get More From Less: A Hybrid Machine Learning Framework for Improving Early Predictions in STEM Education," in The 6th Annual Conf. on Computational Science and Computational Intelligence, CSCI 2019, 2019. event-place: Las Vegas, Nevada.

[17] M. Hasan and B. Khan, "A Trajectory-Clustering Framework for Assessing AI-Based Adaptive Interventions in Undergraduate STEM Learning American Society for Engineering Education," 2023.

[18] M. Tsiakmaki, G. Kostopoulos, S. Kotsiantis, and O. Ragos, "Transfer learning from deep neural networks for predicting student performance," Applied Sciences, vol. 10, no. 6, 2020.

[19] D. Foster, Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play. USA: O'Reilly Media, Inc., 2nd ed., 2023.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," Dec. 2017. arXiv:1706.03762 [cs].

[21] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Lan-guage Models are Few-Shot Learners," in Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901, Curran Associates, Inc., 2020.

[22] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel, "PaLM: Scaling Language Modeling with Pathways," Oct. 2022. arXiv:2204.02311 [cs].

[23] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "LLaMA: Open and Efficient Foundation Language Models," Feb. 2023. arXiv:2302.13971 [cs].

[24] OpenAI, "GPT-4 Technical Report," Mar. 2023. arXiv:2303.08774 [cs].

[25] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," The Journal of Machine Learning Research, vol. 21, pp. 140:5485–140:5551, Jan. 2020.

[26] A. Roberts, C. Raffel, and N. Shazeer, "How Much Knowledge Can You Pack Into the Parameters of a Language Model?," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), (Online), pp. 5418–5426, Association for Computational Linguistics, Nov. 2020.

[27] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," Jan. 2023. arXiv:2201.11903 [cs].

[28] K. Bhatia, A. Narayan, C. De Sa, and C. Ré, "TART: A plug-and-play Transformer module for task-agnostic reasoning," June 2023. arXiv:2306.07536 [cs].

[29] S. Huang, L. Dong, W. Wang, Y. Hao, S. Singhal, S. Ma, T. Lv, L. Cui, O. K. Mohammed, B. Patra, Q. Liu, K. Aggarwal, Z. Chi, J. Bjorck, V. Chaudhary, S. Som, X. Song, and F. Wei, "Language Is Not All You Need: Aligning Perception with Language Models," Mar. 2023. arXiv:2302.14045 [cs].

[30] A. Bandura, "Social cognitive theory of mass communication," Media Psychology, vol. 3, pp. 265–299, 2001.

[31] B. Fogg, "A behavior model for persuasive design," in Proceedings of the 4th International Conference on Persuasive Technology, Persuasive '09, (New York, NY, USA), pp. 1–7, Association for Computing Machinery, Apr. 2009.

[32] J. Fredricks, Eight Myths of Student Disengagement: Creating Classrooms of Deep Learning. Thousand Oaks, California: Corwin Press, 2014.

[33] H. Xue, F. D. Salim, Y. Ren, and C. L. A. Clarke, "Translating Human Mobility Forecasting through Natural Language Generation," Dec. 2021. arXiv:2112.11481 [cs].

[34] H. Xue and F. D. Salim, "PromptCast: A New Prompt-based Learning Paradigm for Time Series Forecasting," June 2023. arXiv:2210.08964 [cs, math, stat].

[35] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned Language Models Are Zero-Shot Learners," Feb. 2022. arXiv:2109.01652 [cs].

[36] X. Liu, "Methods for handling missing data," in Methods and Applications of Longitudinal Data Analysis (X. Liu, ed.), ch. 14, pp. 441–473, Academic Press, 2016.

[37] J. M. Lachin, "Fallacies of last observation carried forward analyses," Clinical trials, vol. 13, no. 2, pp. 161–168, 2016.

[38] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," Expert systems with applications, vol. 73, pp. 220–239, 2017.

[39] J. N. Hernandez, J. A. Carrasco-Ochoa, and J. F. M. Trinidad, "An empirical study of oversampling and undersampling for instance selection methods on imbalance datasets," in Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications - 18th Iberoamerican Congress, CIARP 2013, Havana, Cuba, November 20-23, 2013, Proceedings, Part I (J. Ruiz-Shulcloper and G. S. di Baja, eds.), vol. 8258 of Lecture Notes in Computer Science, pp. 262–269, Springer, 2013.

[40] B. Li, Y. Hou, and W. Che, "Data augmentation approaches in natural language processing: A survey," AI Open, vol. 3, pp. 71–90, 1 2022.

[41] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, and J. Wei, "Scaling Instruction-Finetuned Language Models," Dec. 2022. arXiv:2210.11416 [cs].

[42] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[43] Y. Kim, "Convolutional neural networks for sentence classification," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), (Doha, Qatar), pp. 1746–1751, Association for Computational Linguistics, oct 2014.

[44] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in International Conference on Learning Representations, 2019.

4 Appendix

In this section, we provide performance statistics of the language models (medium and small), baseline models, the template used for verbalizing numeric data into a language generation dataset, and the experimental setting.

4.1 Performance of the Language Models (Medium & Small) and Baseline Models

Table 2: Evaluation of the medium LM (FLAN-T5-base) fine-tuned with four combinations of the 3 feature types using the 8-week, 4-week, and 2-week datasets. The best results are in bold.
Legends: C=Cognitive, NC=Non-Cognitive, D=Distal, AR=At-Risk, PR=Prone-To-Risk, AV=Average, OU=Outstanding, P=Precision, R=Recall, F1=F1 Score, A=Accuracy

Features	Class	8-week				4-week				2-week			
		P	R	F1	A	P	R	F1	A	P	R	F1	A
C + NC + D	AR	1.00	0.86	0.92	0.86	0.78	1.00	0.88	0.84	0.55	0.86	0.67	0.68
	PR	0.78	0.70	0.74		0.89	0.80	0.84		0.71	0.50	0.59	
	AV	0.91	0.91	0.91		0.79	1.00	0.88		0.71	0.91	0.80	
	OU	0.83	0.94	0.88		0.92	0.69	0.79		0.75	0.56	0.64	
C + NC	0.88	1.00	1.00	1.00	0.80	0.71	0.71	0.71	0.73	0.46	0.86	0.60	0.61
	0.58	0.70	0.70	0.70		0.75	1.00	0.67		0.64	0.70	0.67	
	0.82	0.71	0.91	0.80		0.69	1.00	0.81		0.67	0.73	0.70	
	0.77	0.85	0.69	0.76		0.77	0.62	0.69		0.75	0.38	0.50	
C + D	0.60	0.78	1.00	0.88	0.75	1.00	1.00	1.00	0.73	0.67	0.86	0.75	0.64
	0.83	0.78	0.70	0.74		0.69	0.90	0.78		0.56	0.50	0.53	
	0.73	0.73	0.73	0.73		0.64	0.82	0.72		0.86	0.55	0.67	
	0.71	0.73	0.69	0.71		0.70	0.44	0.54		0.58	0.69	0.63	
C	0.64	0.64	1.00	0.78	0.70	0.86	0.86	0.86	0.66	0.50	0.57	0.53	0.48
	0.67	0.75	0.60	0.67		0.57	0.40	0.47		0.83	0.50	0.62	
	0.73	0.82	0.82	0.82		0.53	0.82	0.64		0.35	0.64	0.45	
	0.54	0.64	0.56	0.60		0.77	0.62	0.69		0.50	0.31	0.38	

Table 3: Evaluation of the small LM (FLAN-T5-small) fine-tuned with four combinations of the 3 feature types using the 8-week, 4-week, and 2-week datasets. The best results are in bold.
Legends: C=Cognitive, NC=Non-Cognitive, D=Distal, AR=At-Risk, PR=Prone-To-Risk, AV=Average, OU=Outstanding, P=Precision, R=Recall, F1=F1 Score, A=Accuracy

Features	Class	8-week				4-week				2-week			
		P	R	F1	A	P	R	F1	A	P	R	F1	A
C + NC + D	AR	1.00	0.86	0.92	0.82	0.60	0.43	0.50	0.75	0.60	0.86	0.71	0.64
	PR	1.00	0.40	0.57		0.89	0.80	0.84		0.62	0.50	0.56	
	AV	0.77	0.91	0.83		0.77	0.91	0.83		0.67	0.73	0.70	
	OU	0.76	1.00	0.86		0.71	0.75	0.73		0.64	0.56	0.60	
C + NC	0.88	1.00	0.93	0.82	0.75	0.50	0.71	0.59	0.66	0.42	0.71	0.53	0.59
	0.58	0.70	0.64	0.75		0.67	0.60	0.63		0.80	0.40	0.53	
	0.82	0.82	0.82	0.85		0.67	0.73	0.70		0.50	0.73	0.59	
	0.77	0.62	0.69	0.83		0.77	0.62	0.69		0.82	0.56	0.67	
C + D	0.60	0.86	0.71	0.88	0.70	0.86	0.86	0.86	0.64	0.67	0.86	0.75	0.59
	0.83	0.50	0.62	0.84		0.75	0.90	0.82		0.60	0.60	0.60	
	0.73	0.73	0.73	0.70		0.44	0.64	0.52		0.56	0.82	0.67	
	0.71	0.75	0.73	0.73		0.67	0.38	0.48		0.56	0.31	0.40	
C	0.64	1.00	0.78	0.71	0.64	0.67	0.57	0.62	0.61	0.25	0.43	0.32	0.41
	0.67	0.60	0.63	0.71		0.50	0.30	0.37		0.40	0.20	0.27	
	0.73	0.73	0.73	0.69		0.64	0.82	0.72		0.45	0.45	0.45	
	0.54	0.44	0.48	0.79		0.61	0.69	0.65		0.50	0.50	0.50	

4.2 Template for Natural Language Sequence Generation

Input Sequence:
A student obtained the following assessment scores in an introductory programming course on

Table 4: Evaluation of the three baseline models trained with cognitive features using the 8-week, 4-week, and 2-week datasets. The best results are in bold.
Legends: AR=At-Risk, PR=Prone-To-Risk, AV=Average, OU=Outstanding, P=Precision, R=Recall, F1=F1 Score, A=Accuracy

Model	Class	8-week				4-week				2-week			
		P	R	F1	A	P	R	F1	A	P	R	F1	A
CNN	AR	0.50	0.86	0.63	0.59	0.44	0.57	0.50	0.50	0.45	0.71	0.56	0.45
	PR	0.83	0.50	0.62		1.00	0.30	0.46		0.44	0.70	0.54	
	AV	1.00	0.09	0.17		0.33	0.55	0.43		0.22	0.18	0.20	
	OU	0.56	0.88	0.68		0.37	0.56	0.58		0.75	0.38	0.50	
LSTM	AR	1.00	0.14	0.25	0.34	0.00	0.00	0.00	0.25	0.15	0.29	0.20	0.34
	PR	0.27	0.40	0.32		0.00	0.00	0.00		0.00	0.00	0.00	
	AV	0.33	0.27	0.30		0.26	0.73	0.38		0.00	0.00	0.00	
	OU	0.37	0.44	0.40		0.33	0.19	0.24		0.42	0.81	0.55	
Transformer	AR	0.78	1.00	0.88	0.59	0.54	1.00	0.70	0.57	0.56	0.71	0.63	0.55
	PR	0.57	0.40	0.47		1.00	0.60	0.75		0.80	0.60	0.71	
	AV	0.41	0.64	0.50		0.40	0.18	0.25		0.00	0.00	0.00	
	OU	0.73	0.50	0.59		0.50	0.62	0.56		0.46	0.81	0.59	

[NAME OF LANGUAGE] in [SEMESTER] from week 1 to week [n] for [LIST OF GRADED COMPONENTS]: in week 1, scored [?] out of [?] in [NAME OF COGNITIVE TEST], ..., [MEASURE OF TWO EMOTIONAL ENGAGEMENT FEATURES:] student believes that student might get [X] grade and student is [Y] satisfied with performance; in week 2 [CONTINUE AS BEFORE] ... in week [n], scored [?] out of [?] in [NAME OF COGNITIVE TEST], ..., [MEASURE OF TWO EMOTIONAL ENGAGEMENT FEATURES:] student believes that student might get [X] grade and student is [Y] satisfied with performance. Some background information about the student: Student is a [RACE], [GENDER] in his/her class standing year with a major in [Z]. His/Her family income is [$].

Output Sequence:

- If the student's grade is (A+, A, A-), the output sequence would be: At the end of the semester, the student will exhibit an outstanding performance.
- If the student's grade is (B+, B, B-), the output sequence would be: At the end of the semester, the student will exhibit an average performance.
- If the student's grade is (C+, C, C-), the output sequence would be: At the end of the semester, the student will be prone to risk.
- If the student's grade is (below C-), the output sequence would be: At the end of the semester, the student will be at-risk.

Legends:

- X: A/B/C/D/not pass
- Y: very/somewhat/a little/not at all
- Z: Agriculture Engineering/Biological System Engineering/Construction Engineering/Mechanical Engineering/Prefer to self-describe
- $: Less than $10,000/$10,000 - $19,999/$20,000 - $49,999/$50,000 - $99,999/$100,000 - $149,999/More than $150,000.

4.3 Experimental Setting

For the encoder-decoder LM, we used FLAN-T5 (41), which is a variant of the T5 model (25). The FLAN-T5 model is instruction fine-tuned, which is suitable for our purpose. We utilized three varying sizes of FLAN-T5, i.e., FLAN-T5-Small (80M), FLAN-T5-Base (250M), and FLAN-T5-Large (770M). These LMs have a context window limited to 512 tokens.

For the numeric feature-based baseline models, we used three types of neural models, i.e., the sequential long short-term memory (LSTM) (42), non-sequential convolutional neural network

(CNN) with a one-dimensional convolutional kernel (43), and a Transformer (20). The LSTM processes numeric features as a sequence of observations for capturing long-term dependencies. We used a 3-layer LSTM network with 128 neurons, 64 neurons, and 32 neurons, respectively. After each LSTM layer, a Batch Normalization layer is applied to enhance training stability and accelerate convergence. Unlike the LSTM model, which can capture long-term dependencies, the 1D CNN model is good at capturing local patterns in the data. The CNN comprises three 1D convolutional layers, each featuring 64 filters, a 3-unit kernel size, and enhanced by batch normalization and ReLU activation functions, followed by a Global Average Pooling 1D layer before the output layer. The encoder-based Transformer includes residual connections, layer normalization, and dropout. In addition, the projection layers are implemented through 1D CNN. We used 4 encoder blocks. Finally, the dense layer with 128 neurons is followed by the classification layer. We initialized these models with random weights.

For all experiments, we used a batch size of 6, fine-tuned/trained for 50 epochs using an AdamW (44) optimizer. The choice of batch size was constrained by the limited memory available for fine-tuning the LMs.

The experiments with FLAN-T5 small and base models were conducted on Google Colab using a Tesla T4 GPU with 16 GB of memory. The FLAN-T5 large experiments were done on two Tesla V100 GPUs with 32 GB of memory using distributed training.