Deep Attentive Study Session Dropout Prediction in Mobile Learning Environment

Youngnam Lee[1], Dongmin Shin[1], HyunBin Loh[1], Jaemin Lee[1], Piljae Chae[1], Junghyun Cho[1], Seoyon Park[1], Jinhwan Lee[1], Jineon Baek[1,3], Byungsoo Kim[1] and Youngduck Choi[1,2]

[1] *Riiid! AI Research, Korea*

[2] *Yale University, U.S.A.*

[3] *University of Michigan, U.S.A.*

Keywords: Education, Artificial Intelligence, Transformer.

Abstract: Student dropout prediction provides an opportunity to improve student engagement, which maximizes the overall effectiveness of learning experiences. However, researches on student dropout were mainly conducted on school dropout or course dropout, and study session dropout in a mobile learning environment has not been considered thoroughly. In this paper, we investigate the study session dropout prediction problem in a mobile learning environment. First, we define the concept of the study session, study session dropout and study session dropout prediction task in a mobile learning environment. Based on the definitions, we propose a novel Transformer based model for predicting study session dropout, DAS: **D**eep **A**ttentive Study **S**ession Dropout Prediction in Mobile Learning Environment. DAS has an encoder-decoder structure which is composed of stacked multi-head attention and point-wise feed-forward networks. The deep attentive computations in DAS are capable of capturing complex relations among dynamic student interactions. To the best of our knowledge, this is the first attempt to investigate study session dropout in a mobile learning environment. Empirical evaluations on a large-scale dataset show that DAS achieves the best performance with a significant improvement in area under the receiver operating characteristic curve compared to baseline models.

1 INTRODUCTION

Maximizing the learning effect for each individual student is the primary problem in the field of Artificial Intelligence in Education (AIEd). Prevalent approaches for the problem mainly focus on generating optimal learning path, where an Intelligent Tutoring System (ITS) recommends learning items, such as questions or lectures, with the best efficiency based on student's learning activity records (Reddy et al., 2017; Zhou et al., 2018; Liu et al., 2019). However, one should consider not only the efficiency of learning items but also student engagement too, to maximize the overall effectiveness of learning experiences. Even though the ITS determines the optimal learning path with best efficiency, the educational goal is not achievable if a student drops out of a study session at an early stage. By predicting student dropout from a study session, an ITS can dynamically modify service strategy to encourage student engagement.

Previously, student dropout research has mainly studied on school dropout (Archambault et al., 2009; Márquez-Vera et al., 2016) and course dropout (Liang et al., 2016a; Márquez-Vera et al., 2016; Liang et al., 2016b; Whitehill et al., 2017), where traditional machine learning techniques, such as support vector machines, decision tree, logistic regression, and naive bayes, were commonly used. With the development of Massive Online Open Courses (MOOC) and the availability of massive user activity data, more complex models based on neural networks were proposed to predict course dropout in the MOOC environment (Hansen et al., 2019; Béres et al., 2019; Feng et al., 2019).

Unfortunately, despite the active studies conducted on student dropouts, mobile learning environments were not considered thoroughly in the AIEd research community. Students in a mobile learning environment are prone to be distracted by many external variables, such as phone rings, texting and social applications (Harman and Sato, 2011; Junco, 2012; Chen and Yan, 2016). As a result, unlike school dropout and course dropout, study session dropout in a mobile learning environment occurs more fre-

Lee, Y., Shin, D., Loh, H., Lee, J., Chae, P., Cho, J., Park, S., Lee, J., Baek, J., Kim, B. and Choi, Y.
Deep Attentive Study Session Dropout Prediction in Mobile Learning Environment.
DOI: 10.5220/0009347700260035
In *Proceedings of the 12th International Conference on Computer Supported Education (CSEDU 2020) - Volume 1*, pages 26-35
ISBN: 978-989-758-417-6

quently, which causes shorter study session length. Therefore, directly applying previous approaches for predicting student dropout to study session dropout prediction results in poor performance since it fails to capture the relations of student actions in shorter time frame.

In this paper, we investigate the study session dropout prediction problem in a mobile learning environment. First, we define the concept of a study session, study session dropout and study session dropout prediction task in a mobile learning environment. Based on the observation of student interaction data and following the work of (Halfaker et al., 2015), we define a study session as a sequence of learning activities where the time interval between adjacent activities is less than 1 hour. Accordingly, if a student is inactive for 1 hour, then we define it as a study session dropout.

From the definitions above, we propose a novel Transformer based model for predicting study session dropout, DAS: **D**eep **A**ttentive Study **S**ession Dropout Prediction in Mobile Learning Environment. DAS consists of an encoder and a decoder that are composed of stacked multi-head attention and point-wise feed-forward networks. The encoder applies repeated self-attention to the sequential input of question embedding vectors which serve as queries, keys, and values. The decoder computes self-attention to the sequence of response embedding vectors which are queries, keys and values, and attention with the output of the encoder alternately. Unlike the original Transformer architecture (Vaswani et al., 2017), DAS uses a subsequent mask to all multi-head attention networks to ensure that the computation of current dropout probability depends only on the previous questions and responses. By considering inter-dependencies among entries, and giving more weights to relevant entries for prediction target, the deep attentive computations in DAS are capable of capturing complex relations of student interactions.

We conduct experimental studies on a large-scale dataset collected by an active mobile education application, Santa, which has 21K users, 13M response data points as well as a set of 15K questions gathered since 2016. We compare DAS with several baseline models and show that it outperforms all other competitors and achieves the best performance with the significant improvement in area under the receiver operating characteristic curve (AUC).

In short, our contributions can be summarized as follows:

- We define the problem of study session dropout prediction in a mobile learning environment.
- We propose DAS, a novel Transformer based encoder-decoder model for predicting study session dropout, where deep attentive computations effectively capture complex relations among dynamic student interactions.
- Empirical studies on a large-scale dataset show that DAS achieves the best performance with a significant improvement in AUC compared to the baseline models.

2 RELATED WORKS

Dropout Prediction is an important problem studied in multiple areas, such as online games (Kawale et al., 2009), telecommunication (Huang et al., 2012), and streaming services (Chen et al., 2018). Predicting dropout in short-term enables dynamic updates of service strategy, which results in longer session lengths of users. In long-term, it enables the examination of favorable features of the services, from the relations of service features and dropout rates (Halawa et al., 2014).

In the field of education, student dropout prediction has been studied in mainly two areas: school dropout (Márquez-Vera et al., 2016), and course dropout (Liang et al., 2016a). The research in (Sara et al., 2015) was the first large scale study on high-school dropout. This research examined 36,299 students, where the authors state that previous studies were based on a few hundred students.

Compared to previous works on school dropout prediction which are based on relatively small (hundreds of students), massively generated MOOC log data is actively used in the research on course dropout. These log data include user actions such as user responses, page accesses, registrations, and click-streams. For instance, the dataset described in (Reich and Ruipérez-Valiente, 2019) includes data of 12.67 million course registrations from 5.63 million learners. Using the large data from MOOC services, machine learning models such as random forest, SVM (Lykourentzou et al., 2009), and neural networks (Feng et al., 2019) are applied to course dropout predictions in education. The model in (Liu et al., 2018) based on Long Short Term Memory Networks (LSTM) predicts course dropout in MOOC. However, there is no research on the prediction of study session dropout in MOOC which happens more frequently than a course dropout.

There are works on session dropout in other fields, such as streaming services, medical monitoring (Pappada et al., 2011), and recommendation services (Song et al., 2008). A famous problem similar to this topic is the *Spotify Sequential Skip Prediction Chal-*

lenge, which is a problem to predict if a user will skip a song given the previous playlist. The data schema of this problem is similar to the case of predicting study session dropout based on student-question responses. A major portion of suggested models in the Spotify Challenge are based on neural networks (Hansen et al., 2019).

The research in (Daróczy et al., 2015) suggests a model predict session dropout in the LTE network for network optimization purposes. The research above is based on sequential models such as LSTM, but recently the Transformer is showing higher performance on similar tasks.

The transformer was first introduced in (Vaswani et al., 2017). It replaced the recurrent layers in the encoder-decoder architecture with multi-head self-attention. The architecture is widely used in natural language processing tasks since the training process can be parallelized. In AIEd, Transformer based model (Pandey and Karypis, 2019) shows higher performance than existing seq2seq models (Lee et al., 2019).

3 STUDY SESSION DROPOUT IN MOBILE LEARNING

In this section, we define the concept of a study session, study session dropout, and study session dropout prediction task in mobile learning. A *study session* is a learning process in which the user contiguously participates in learning activities while retaining the educational context of his previous activities. During each study session, a *study session dropout* happens when the user is inactive in learning for a sufficiently long time, losing the context of his most recent learning process. Following the work of (Halfaker et al., 2015), which examined user activity data in various fields to conclude that the 1 hour inactivity time gives the best results in clustering user behaviors, we considered inactivity of 1 hour as the threshold for study session dropout. Note, however, that the criterion for determining study session dropout is flexible and can be chosen according to the particular needs and properties of each ITS. An example of user activities, sessions, and session dropouts is illustrated in Figure 1.

Identifying study sessions of a mobile ITS user helps the tutoring system to understand student behaviors in coherent units. For example, one may analyze a student's knowledge state per each activity in a single session to study the short-term learning effect in a session, and per each session to study the long-term learning effects across different sessions. Along this line, the prediction of study session dropouts may be utilized to guide a student's learning path. For instance, students with very short session time may not absorb enough learning materials to maintain what he has learned in his session. To guide them, an ITS could push pop-up messages that provide educational feedback or encourage additional learning activities to lengthen the study sessions of students. For students with very long study sessions, the ITS could suggest the students take a break for more effective learning.

However, existing dropout prediction methods cannot be applied to mobile ITSs due to the large difference between traditional and mobile education sessions in length. In traditional education, study sessions like school lectures, exercise sessions or timed exams are usually held in environments regulated by instructors. This enables students to focus solely on given activity in specific space and time limit which results in a longer time span of learning sessions. In contrast, study sessions in mobile learning do not impose any condition on students' behavior and surrounding environments, allowing students to diverge to other activities. According to (Chen and Yan, 2016), tasks that demand multi-tasking to students like phone rings, texting, and social applications are the main sources of distraction that affect learning activity in a mobile environment. Another factor for shorter learning time in mobile education is the limitation of the hardware. Unlike offline tools like books and blackboards, a mobile device has a smaller screen with constant light emission which is harder to focus for a prolonged time interval, which results in a shorter attention time span.

To this end, we propose the *study session dropout prediction* problem for mobile ITSs. The task is the prediction of probability that a user drops out from his ongoing study session. Unlike offline learning, mobile ITSs can take advantage of automatically collected student behavior data to complete the task in real-time. For example, a dropout prediction model may utilize the question-response log data of a student as input. Under this setting, we formalize the problem as the following. Note that utilizing student data aside from question-response logs are also possible, which we defer to future works.

Let

$$I_1, I_2, \cdots, I_i = (e_i, l_i), \cdots, I_T$$

be the sequence of question-response pairs I_i of a student. Here, e_i denotes the meta-data of the i-th question asked to the student such as question ID or the relative position of the question in the ongoing session. Likewise, l_i denotes the metadata of the student's response to the i-th question, which includes the actual response of the student and the time he has spent on the given question. The *study session*

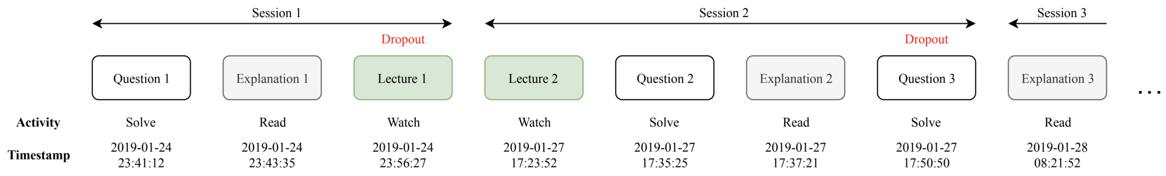

Figure 1: Session illustration.

dropout prediction is the estimation of the probability

$$P(d_i = 1 | I_1, \ldots, I_{i-1}, e_i)$$

that the student leaves his session after solving the i-th question, where d_i is equal to 1 if the student leaves and 0 otherwise.

4 PROPOSED METHOD

4.1 Input Representation

The proposed model predicts student dropout probability based on two feature collections of each interaction I_i: the set e_i of question meta-data features and the set l_i of response meta-data features. The members of e_i and l_i are summarized in Table 1. In total, there are a total of nine features constituting e_i and l_i:

- Question ID id_i: The unique ID of each question.
- Category c_i: Part of the TOEIC exam the question belongs to.
- Starting time st_i: The time the student first encounters the given problem.
- Position in input sequence p_i: The relative position of the interaction in the input sequence of our model. Note that this positional embedding was used by (Gehring et al., 2017) to replace the sinusoidal positional encoding introduced in (Vaswani et al., 2017).

Table 1: Features of e_i and l_i.

Input	Description	e_i	l_i
id	Question ID	✓	
c	Category	✓	
st	Starting time	✓	✓
p	Position in input sequence	✓	✓
sp	Position in session	✓	✓
r	Response correctness		✓
et	Elapsed time		✓
iot	Is on time		✓
d	Dropout		✓

- Position in session sp_i: The relative position of the interaction in the session it belongs to. The number increments by each problem and resets to 1 whenever a new session starts.
- Response correctness r_i: Whether the user's response is correct or not. The value is 1 if correct and 0 otherwise.
- Elapsed time et_i: The time the user took to respond to given question.
- Is on time iot_i: Whether the user responded in the time limit suggested by domain experts. The value is 1 if true and 0 otherwise.
- Is dropout d_i : Whether the user dropped out after this interaction. The value is 1 if true and 0 otherwise.

The vector representations of e_i and l_i are computed by summing up the embeddings of aforementioned features. Put formally, we have:

$$e_i = \text{emb}_e(id_i, c_i, st_i, p_i, sp_i)$$
$$l_i = \text{emb}_l(r_i, et_i, st_i, iot_i, d_i, p_i, sp_i)$$

where $\text{emb}_{-}(.)$ denotes the summation of the corresponding features as distributional vectors. Note that this is equivalent to projecting the concatenation of all features by linearity. Separate embeddings are used for features shared across e_i and l_i. For example, the same positional number sp have different distributional vectors in question embedding emb_e and response embedding emb_l.

4.2 Model Description

Our model is based on Transformer, which consists of the encoder and decoder part. First, we give a brief description of both part. The encoder consists of n stacked encoder blocks, where each encoder block has two sub-layers, a self-attention layer, and a fully connected feed-forward network layer, where both sub-layers are followed by residual connection and layer normalization. The encoder takes the sequence of question information embedding $e_1, \ldots, e_n$ as the input, where each e_i has dimension d_{model}. We define E^i as the input of the i'th block, and the output of the $i-1$'th block for $i = 2, \ldots, n$. Here, E^1 is the input

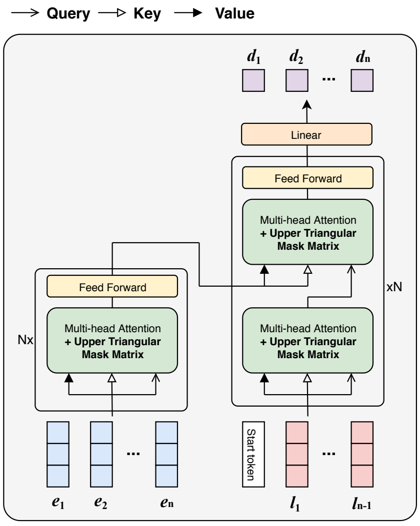

Figure 2: The model architecture.

for the first encoder block, which is the output of the embedding layer. The final output of the encoder is denoted as $h_1, \dots, h_n$

The decoder also consists of n stacked decoder blocks and one linear layer, where each block has a self-attention layer, an attention layer, and a fully connected feed-forward network layer. All three sub-layers are followed by residual connection and layer normalization.The decoder takes the user log information embedding $S, l_1, \dots, l_{n-1}$ as input, where S is a constant starting token. We define D^i as the input of the i'th decoder block, and the output of the $i-1$'th block for $i = 2, \dots, n$. The additional attention layer takes outputs of self-attention layers as query, and outputs of encoder as key and value. At the last, we use a learned linear transformation to predict the dropout probabilities $\hat{d}^1 \dots, \hat{d}^n$.

Now, we describe each sub-layer in detail. The self-attention layer of the i'th encoder block takes query, key, value matrices as inputs. These matrices are given by multiplying E^i to the parameter matrices W^Q, W^K, W^V.

$$Q = E^i W^Q = [q_1, \cdots, q_n]^T$$
$$K = E^i W^K = [k_1, \cdots, k_n]^T$$
$$V = E^i W^V = [v_1, \cdots, v_n]^T$$

The i'th rows q_i, k_i, v_i of Q, K, V are the representations of query vector, key vector, and value vector of the i'th input e_i. Let the dimensions of q_i, k_i, v_i be d_k, d_k, d_v. Our model uses multi-head attention, instead of single self-attention layers, to capture various aspects of attention. Multi-head attention is the method to split the embedding vector into the number of heads h and perform h self-attentions on each part of the split vector.

Then, the attention layer output is computed by multiplying v_j on the Softmax of normalized $q_i \cdot k_j$ for all $j = 1, \dots, n$. This is written as:

$$\text{Multihead}(E^i) = \text{Concat}(\text{head}_1, \cdots, \text{head}_h) W^O$$

Here, each

$$\text{head}_j = \text{Attention}_j(E^i) = \text{Softmax}\left(\frac{Q_j K_j^T}{\sqrt{d_k}}\right) V_j,$$

where $Q_j = E^i W_j^Q$, $K_j = E^i W_j^K$ and $V_j = E^i W_j^V$ respectively. We mask the matrix QK^T, where the masking details are described in the following subsection.

As $M = \text{Multihead}(E^i)$ of an attention is a linear combination of values, a position-wise feed forward network is applied to add non-linearity to the model. The formula is given by:

$$\begin{aligned} F = (F_1, \cdots, F_n) &= \text{FFN}(M) \\ &= \text{ReLU}\left(MW^{(1)} + b^{(1)}\right) W^{(2)} + b^{(2)} \end{aligned}$$

where $W^{(1)}$, $W^{(2)}$, $b^{(1)}$ and $b^{(2)}$ are weight matrices, and bias vectors.

In summary, the whole process of the encoder and decoder can be written by:

$$h_1, \cdots, h_n = \text{Encoder}(e_1, \cdots, e_n)$$
$$\hat{d}^1, \cdots, \hat{d}^n = \text{Decoder}(S, l_1, \cdots, l_{n-1}, h_1, \cdots, h_n)$$

where $\hat{d}^i$ is the predicted value of

$$P(\text{Dropout when given } e_i \mid e_1, \cdots, e_{i-1}, l_1, \cdots, l_{i-1}).$$

4.3 Subsequent Masking

Transformer models use offset and subsequent masks to handle causality issues. Appropriate subsequent masks in the sequential data case can let each row represent a time step, so that different time steps can be fed to the network for training. Here, the details of masking should depend on the context of problem. A Transformer model for Machine Translation (MT) uses the whole input sentence, and the first $i-1$ words of the partially translated sentence $v_1, \dots, v_{i-1}$ to translate the i'th word of the input sentence. To respect this causality, the sequence $S, V_1, \cdots, V_{i-1}$ with offset by a starting token S enters the decoder from the encoder at step i. We apply this structure of Transformer to session dropout prediction, but with modifications on masking to fit our problem details.

Compared to the original transformer model, our model uses a subsequent mask on all multi-head attention layers (encoder multi-head attention, decoder

Table 2: User activity log data.

timestamp	question_id	user_answer	correctness	elapsed_time	part	session_id	dropout
2019-02-12 09:40:21	5279	c	1	33	5	1	0
2019-02-12 09:40:51	5629	b	0	26	5	1	0
2019-02-12 09:41:10	6048	a	1	16	5	1	0
2019-02-12 09:41:54	6158	b	0	41	2	1	1
2019-02-14 19:32:27	5022	d	1	30	2	2	0

multi-head attention, encoder-decoder multi-head attention) to prevent invalid attending. We mask all attention layers to ensure that the computation of $\hat{d^i}$ depends only on the information from the previous questions $e_1, \cdots, e_i$ and responses $l_1, \cdots, l_{i-1}$ on the i'th step. In the MT example, it is natural for the decoder to translate a word by attending all the words before and after the word in the source sentence from the encoder. But in our case, attending to future questions $e_{i+1}, \cdots, e_n$ to predict d_i is invalid since further problem suggestions depend on l_i. Let $I_1, \ldots, I_n$ be a sequence of question-response pairs with a student ending the session after solving e_n, where the session length is n. Directly applying the MT model to this situation will be predicting each d_i given $e_1, \ldots, e_n$ and $l_1, \ldots, l_{n-1}$. However, as described above, our case differs from the MT case e_j for $j > i$ is not given at the point i. Therefore, we apply subsequent masks on future question information to both the input sequence and the in mid-processes.

5 DATASET

We use the *Santa* dataset for training our model, which is released in 2019 by mobile AI tutor *Santa* for English education (Choi et al., 2019). Specifically, *Santa* aims to prepare students for TOEIC®(Test Of English for International Communication®). The test consists of seven parts divided into listening and reading sessions, with 100 questions assigned for each session. The final score is subjective and ranges from 0 to 990 in a score gap of 5. At the time of writing, the application is available via Android and iOS applications with 1,047,747 users signed up for the service. In *Santa*, users are provided multiple-choice questions recommended by the *Santa* AI tutor. After a user responds to a given question, he receives corresponding educational feedback by reading an expert's commentary or watching relevant lecture videos. Every user responses from 2016 to 2019 are recorded in the *Santa* dataset with the following columns of our interest (see Table 2).

- `user_id`: A unique ID assigned to each user. We group the rows having the same `user_id` to recover each user's activity log data.
- `timestamp`: The time the user received a question.
- `question_id`: The ID of the question the user received.
- `user_answer`: User's response, which is one of the four possible choices A, B, C, or D.
- `correctness`: Correctness of the user's response, which is 1 if correct and 0 otherwise.
- `elapsed_time`: The time the user took to respond in milliseconds.
- `part`: Part of the TOEIC® exam the question belongs to, from Part 1 to Part 7.

The dataset involves a total of 13,840,169 interactions between 216,575 users and 14,900 questions. For training and testing our model, the dataset is split into train set (151,602 users, 9,643,191 responses), validation set (21,658 users, 1,409,323 responses) and test set (43,315 users, 2,787,655 responses) per user basis by the 7:1:2 user count ratio.

Following our criterion for identifying session dropout, we divide a user's interactions into different sessions by every inactive time intervals of ≥ 1 hour. Figure 4 shows that the value of 1 hour separates the time differences of consecutive actions into a large bump with a peak around 30 seconds, and a small bump with a peak around a day. This suggests

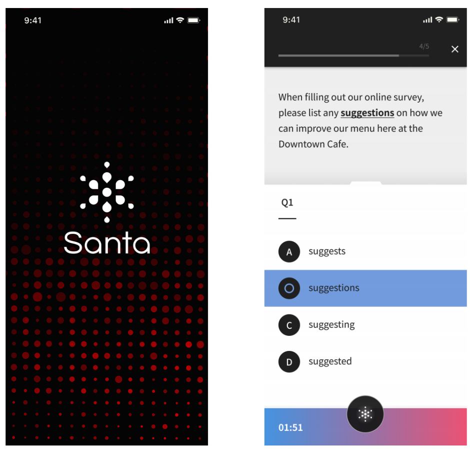

Figure 3: User interface of *Santa*.

Table 3: Statistics of the *Santa* dataset.

Statistics	*Santa* dataset
total user count	216,575
train user count	151,602
validation user count	21,658
test user count	43,315
total response count	13,840,169
train response count	9,643,191
validation response count	1,409,323
test response count	2,787,655

that our criterion is reasonable for identifying long intervals between different sessions from short gaps in an ongoing session.

Each separated session is then assigned a unique `session_id`. The last interaction of each session is marked as dropout interaction by setting the value of the column `dropout` to 1 (the default value of `dropout` is 0). Table 2 illustrates a sample user's question response data with columns `session_id` and `dropout`. The processed data has 772,235 sessions, with an average of 3.57 sessions per user and 17.92 questions per each session. Among all responses, $1/17.92 = 5.58\%$ are dropout responses. Each session lasts for 26.00 minutes on average.

6 EXPERIMENTS

6.1 Training Details

During training, we maintain the ratio of positive and negative `dropout` labels to 1:1 by over-sampling dropout interactions. Model parameters that give the best AUC on validation set is chosen for testing. The best-performing model have $N = 4$ stacked encoder and decoder blocks. The latent space dimension d_{model} of query, key, value and the final output of each encoder/decoder block is equal to 512. Each multi-head attention layer consists of $h = 8$ heads. All model are trained from scratch with weights initialized by Xavier uniform distribution (Glorot and Bengio, 2010). We use the Adam optimizer (Kingma and Ba, 2014) with hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.98$, epsilon = 10^{-9}. The learning rate $lr = d_{\text{model}}^{-0.5} \cdot \min(step\_num^{-0.5}, step\_num \cdot ws^{-1.5})$ follows that of (Vaswani et al., 2017) with $ws = 6000$ warmup steps. A dropout ratio of 0.5 was applied during training.

6.2 Experiment Results

We compare DAS with current state-of-the-art session dropout models based on Long Short-Term Memory

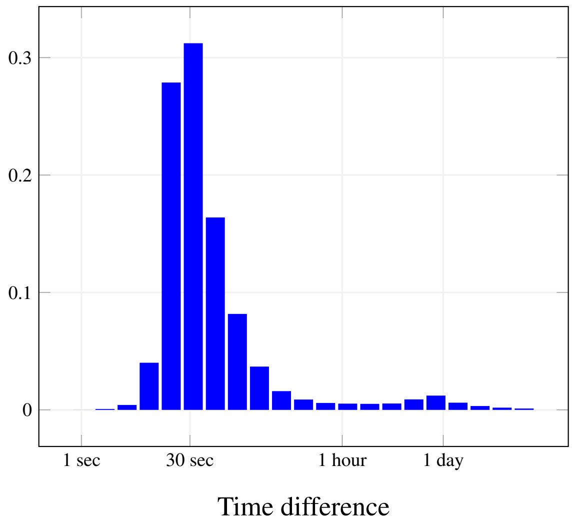

Figure 4: Histogram of time differences between two consecutive interactions of all users in log scale.

(LSTM) and Gated Recurrent Unit (GRU) (Hussain et al., 2019) by training and testing the models with *Santa* dataset. As LSTM and GRU models take only one input for each index, we set the i'th input of the models as $in_{i-1} + e_i$, where $in_{i-1} = e_{i-1} + l_{i-1}$ is the representation of the previous ($(i-1)$'th) question-response interaction. For the first input, we replace in_{i-1} with a starting token. Like DAS, optimal parameters for LSTM and GRU were found by maximizing the AUC over validation set. All models were trained and tested with sequence size 5 and 25 to find the best input length for session dropout prediction. The results in Table 4 shows that DAS with sequence size 5 outperforms best LSTM and GRU models by 12.2 points.

6.3 Ablation Study

In this section, we present our ablation studies on different sequence sizes and input feature combinations of DAS.

First, we run an ablation study on different input sequence sizes of DAS. The results in Table 5 show

Table 4: Model comparison.

Methods	AUC
LSTM-25	0.5786
LSTM-5	0.6830
GRU-25	0.5640
GRU-5	0.6830
DAS-25	0.6895
DAS-5	**0.7661**

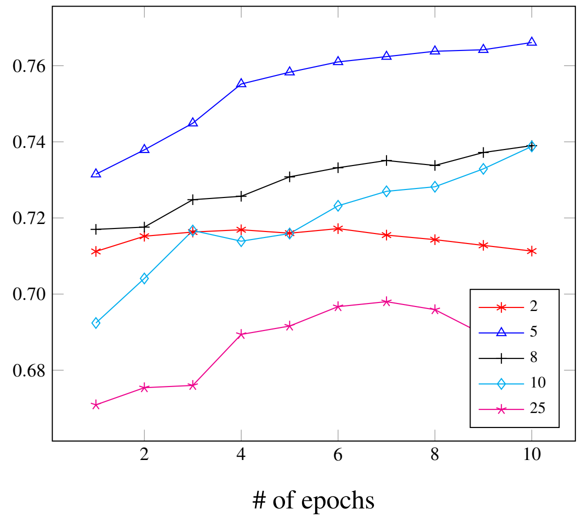

Figure 5: AUC of DAS models with different sequence sizes.

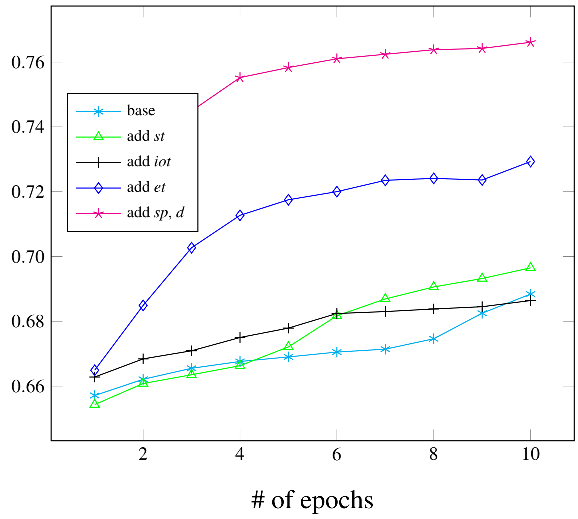

Figure 6: AUC of DAS models with different input feature combinations.

Table 5: Effects of sequence size on AUC.

Sequence Size	AUC
2	0.7172
5	**0.7661**
8	0.7390
10	0.7388
25	0.6980

Table 6: Effects of input features on AUC.

Name	Encoder Inputs	Decoder Inputs	AUC
Base	id, c, p	r, p	0.6884
add *st*	id, c, p, st	r, p, st	0.6965
add *iot*	id, c, p, st	r, p, st, iot	0.6864
add *et*	id, c, p, st	r, p, st, iot, et	0.7293
add sp, d	id, c, p, st, sp	r, p, st, iot, et, sp, d	**0.7661**

that a sequence size of 5 produces the best AUC. This trend is consistent across all epochs (see Figure 5), suggesting that session dropouts are more correlated with latest student interactions than earlier ones. However, as the sequence size of 2 overfits quickly with poor results, it is also observed that the model needs a sufficient amount of context for effective prediction.

Second, we run an ablation study on different combinations of input features for DAS. Table 6 shows that test AUC gradually increases as we include input features one by one, achieving the highest value with all the features in Table 1. From Figure 6, it can be seen that elapsed time *et*, session position *sp* and session dropout labels of previous interactions *d* improves the overall AUC prominently. The large improvement with elapsed time *et* suggests that the time a student spends on learning activity is highly correlated to his dropout probability. Also, as it is likely for students with sufficient amount of learning activities to drop his session, it is natural for the features *sp* and *d* to be effective for student dropout prediction task.

7 CONCLUSION

Student dropout prediction provides an opportunity to improve student engagement and maximize the overall effectiveness of learning experience. However, student dropout research has been mainly conducted on school dropout and course dropout, and study session dropouts in mobile learning environments were not considered thoroughly in literature. In this paper, we defined the problem of study session dropout prediction in a mobile learning environment. We proposed DAS, a novel Transformer based encoder-decoder model for predicting study session dropout, in which the deep attentive computations effectively capture the complex relations among dynamic student interactions. Empirical studies on a large-scale dataset showed that DAS achieves the best performance with a significant improvement in AUC compared to the baseline models.

REFERENCES

Archambault, I., Janosz, M., Fallu, J.-S., and Pagani, L. S. (2009). Student engagement and its relationship with early high school dropout. *Journal of adolescence*, 32(3):651–670.

Béres, F., Kelen, D. M., and Benczúr, A. A. (2019). Sequential skip prediction using deep learning and ensembles.

Chen, Q. and Yan, Z. (2016). Does multitasking with mobile phones affect learning? a review. *Computers in Human Behavior*, 54:34–42.

Chen, Y., Xie, X., Lin, S.-D., and Chiu, A. (2018). Wsdm cup 2018: Music recommendation and churn prediction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 8–9. ACM.

Choi, Y., Lee, Y., Shin, D., Cho, J., Park, S., Lee, S., Baek, J., Kim, B., and Jang, Y. (2019). Ednet: A large-scale hierarchical dataset in education. *arXiv preprint arXiv:1912.03072*.

Daróczy, B., Vaderna, P., and Benczúr, A. (2015). Machine learning based session drop prediction in lte networks and its son aspects. In *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, pages 1–5. IEEE.

Feng, W., Tang, J., and Liu, T. X. (2019). Understanding dropouts in moocs.

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR. org.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.

Halawa, S., Greene, D., and Mitchell, J. (2014). Dropout prediction in moocs using learner activity features. *Proceedings of the second European MOOC stakeholder summit*, 37(1):58–65.

Halfaker, A., Keyes, O., Kluver, D., Thebault-Spieker, J., Nguyen, T., Shores, K., Uduwage, A., and Warncke-Wang, M. (2015). User session identification based on strong regularities in inter-activity time. In *Proceedings of the 24th International Conference on World Wide Web*, pages 410–418. International World Wide Web Conferences Steering Committee.

Hansen, C., Hansen, C., Alstrup, S., Simonsen, J. G., and Lioma, C. (2019). Modelling sequential music track skips using a multi-rnn approach. *arXiv preprint arXiv:1903.08408*.

Harman, B. A. and Sato, T. (2011). Cell phone use and grade point average among undergraduate university students. *College Student Journal*, 45(3):544–550.

Huang, B., Kechadi, M. T., and Buckley, B. (2012). Customer churn prediction in telecommunications. *Expert Systems with Applications*, 39(1):1414–1425.

Hussain, M., Zhu, W., Zhang, W., Abidi, S. M. R., and Ali, S. (2019). Using machine learning to predict student difficulties from learning session data. *Artificial Intelligence Review*, 52(1):381–407.

Junco, R. (2012). Too much face and not enough books: The relationship between multiple indices of facebook use and academic performance. *Computers in human behavior*, 28(1):187–198.

Kawale, J., Pal, A., and Srivastava, J. (2009). Churn prediction in mmorpgs: A social influence based approach. In *2009 International Conference on Computational Science and Engineering*, volume 4, pages 423–428. IEEE.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lee, Y., Choi, Y., Cho, J., Fabbri, A. R., Loh, H., Hwang, C., Lee, Y., Kim, S.-W., and Radev, D. (2019). Creating a neural pedagogical agent by jointly learning to review and assess.

Liang, J., Li, C., and Zheng, L. (2016a). Machine learning application in moocs: Dropout prediction. In *2016 11th International Conference on Computer Science & Education (ICCSE)*, pages 52–57. IEEE.

Liang, J., Yang, J., Wu, Y., Li, C., and Zheng, L. (2016b). Big data application in education: dropout prediction in edx moocs. In *2016 IEEE Second International Conference on Multimedia Big Data (BigMM)*, pages 440–443. IEEE.

Liu, Q., Tong, S., Liu, C., Zhao, H., Chen, E., Ma, H., and Wang, S. (2019). Exploiting cognitive structure for adaptive learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 627–635, New York, NY, USA. ACM.

Liu, Z., Xiong, F., Zou, K., and Wang, H. (2018). Predicting learning status in moocs using lstm. *arXiv preprint arXiv:1808.01616*.

Lykourentzou, I., Giannoukos, I., Nikolopoulos, V., Mpardis, G., and Loumos, V. (2009). Dropout prediction in e-learning courses through the combination of machine learning techniques. *Computers & Education*, 53(3):950–965.

Márquez-Vera, C., Cano, A., Romero, C., Noaman, A. Y. M., Mousa Fardoun, H., and Ventura, S. (2016). Early dropout prediction using data mining: a case study with high school students. *Expert Systems*, 33(1):107–124.

Pandey, S. and Karypis, G. (2019). A self-attentive model for knowledge tracing. *arXiv preprint arXiv:1907.06837*.

Pappada, S. M., Cameron, B. D., Rosman, P. M., Bourey, R. E., Papadimos, T. J., Olorunto, W., and Borst, M. J. (2011). Neural network-based real-time prediction of glucose in patients with insulin-dependent diabetes. *Diabetes technology & therapeutics*, 13(2):135–141.

Reddy, S., Levine, S., and Dragan, A. (2017). Accelerating human learning with deep reinforcement learning. In *NIPS'17 Workshop: Teaching Machines, Robots, and Humans*.

Reich, J. and Ruipérez-Valiente, J. A. (2019). The mooc pivot. *Science*, 363(6423):130–131.

Sara, N.-B., Halland, R., Igel, C., and Alstrup, S. (2015). High-school dropout prediction using machine learning: A danish large-scale study. In *ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence*, pages 319–24.

Song, Y., Zhuang, Z., Li, H., Zhao, Q., Li, J., Lee, W.-C., and Giles, C. L. (2008). Real-time automatic tag recommendation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 515–522. ACM.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Whitehill, J., Mohan, K., Seaton, D., Rosen, Y., and Tingley, D. (2017). Delving deeper into mooc student dropout prediction. *arXiv preprint arXiv:1702.06404*.

Zhou, Y., Huang, C., Hu, Q., Zhu, J., and Tang, Y. (2018). Personalized learning full-path recommendation model based on lstm neural networks. *Information Sciences*, 444:135–152.