Adaptive Content Recommendation in MOOCs Using Transformer-Based Deep Learning Models

Chendri Irawan Satrio Nugroho[1,*], Erland Inkiriwang[2]

[1]Department of Information Technology, Institut teknologi Tangerang Selatan, Indonesia

[2]Department of Informatics, Institut teknologi Tangerang Selatan, Indonesia

ABSTRACT

Massive Open Online Courses (MOOCs) increasingly rely on personalized learning technologies to address learner disengagement and high dropout rates. This study proposes a Transformer-based adaptive recommendation model capable of capturing long-range sequential dependencies in learner interaction histories. Attention-weight analysis reveals pedagogically meaningful behavior, with the model assigning highest importance to Quiz_Attempt (0.241) and Assignment_Submission (0.214) events, while contextual interactions such as video views and page read received moderate weights (0.187 and 0.143). Sequential dependency analysis further shows strong learner behavior patterns, particularly transitions from Video_View → Quiz_Attempt (0.372) and Page_Read → Video_View (0.298). Hyperparameter sensitivity experiments indicate that a configuration of 6 Transformer layers, 12 attention heads, and 256-dimensional embeddings produces the best performance. Overall, these results demonstrate that Transformer-based models not only improve recommendation accuracy but also enhance explainability through attention mapping and behavioral interpretation. The findings suggest that integrating self-attention mechanisms and behavioral analytics can substantially advance adaptive learning in large-scale online education environments, supporting more personalized and pedagogically aligned learning pathways.

Keywords Adaptive Learning, MOOC Recommendation Systems, Transformer Models, Self-Attention, Sequential Modeling, Learning Analytics

Introduction

MOOCs have become one of the most scalable and accessible modes of digital education, offering open learning pathways for millions of learners worldwide. However, despite their accessibility, MOOCs continue to suffer from persistent challenges related to learner engagement, personalized content navigation, and high dropout rates [1], [2]. Learners often face difficulties in identifying the most relevant next learning resource among thousands of available items, leading to cognitive overload and ineffective learning trajectories [3]. Traditional recommendation strategies, such as popularity-based or collaborative filtering approaches, struggle to accommodate the diversity and sequential nature of learner behaviors in MOOCs [4], [5]. These limitations highlight the need for more adaptive and context-aware recommendation mechanisms capable of understanding the complex and dynamic interactions between learners and learning materials [6].

The emergence of deep learning and sequence modeling has introduced new opportunities for improving personalization in online learning environments. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM)

Submitted: 15 October 2024
Accepted: 20 December 2024
Published: 1 May 2025

*Corresponding author
Chendri Irawan Satrio Nugroho,
chendri@itts.ac.id

Additional Information and Declarations can be found on page 129

How to cite this article: C. I. S. Nugroho, E. Inkiriwang, "Adaptive Content Recommendation in MOOCs Using Transformer-Based Deep Learning Models," *Adapt. Learn.*, vol. 1, no. 2, pp. 115-132, 2025.

architectures have previously been explored for modeling clickstreams and learning sequences, offering improvements over static recommendation methods [7]. However, these models often struggle to capture long-range dependencies and may fail to represent subtle behavioral patterns present in MOOC interaction logs [8]. Self-attention mechanisms, operationalized through Transformer architectures, have demonstrated exceptional performance in sequence modeling tasks across domains such as natural language processing, time-series forecasting, and user-behavior prediction [9], [10]. Their ability to capture global relational structures makes Transformers a promising candidate for adaptive learning recommendations.

Despite these advancements, the application of Transformer-based models in MOOCs remains underexplored and insufficiently optimized. Current research often focuses on content recommendation using simplified behavioral features or static learner profiles, overlooking the temporal richness embedded in multi-event learning histories [11]. Many existing studies also treat all learning events equally, disregarding pedagogical hierarchies such as the importance of quizzes and assignments relative to passive behaviors like video views or page reads [12]. Furthermore, existing works rarely integrate explainability into the recommendation process, making it difficult for educators to interpret and validate the underlying reasoning of AI-driven learning pathways [13]. These gaps underscore the need for a more robust, sequence-aware, and pedagogically aligned recommendation framework.

This study addresses these gaps by introducing an adaptive content recommendation model built on Transformer-based deep learning architectures capable of modeling complex dependencies in learner interaction sequences. The proposed model leverages multi-head self-attention to emphasize meaningful events such as assessment outcomes, engagement intensity, and content modality while reducing noise from peripheral actions. By incorporating temporal dynamics and contextualized embeddings, the system aims to generate next-step learning recommendations that align with learner progress, instructional logic, and pedagogical structure [14], [15]. This approach not only improves recommendation accuracy but also enhances interpretability by revealing which historical events most influence the model's predictions.

The primary objective of this research is to design, implement, and evaluate a Transformer-based recommender system capable of delivering personalized learning sequences in MOOCs. The study examines several key research questions: (1) How effectively can Transformer architectures model sequential learner behavior in online learning environments? (2) To what extent can attention mechanisms improve recommendation accuracy and pedagogical alignment? (3) How does the proposed model compare with baseline methods such as collaborative filtering, matrix factorization, and recurrent neural networks? Addressing these questions enables a deeper understanding of the benefits and limitations of advanced sequence modeling for educational personalization [16].

This research introduces several novel contributions. First, it provides one of the most comprehensive implementations of Transformer-based recommendation tailored specifically for MOOC interaction logs. Second, it integrates pedagogical sensitivity into the attention mechanism by analyzing the relative importance of different event types through empirical attention weights [17], [18].

Third, the study introduces a behavioral segmentation analysis that enhances the interpretability and applicability of the recommendation system across diverse learner groups. Finally, the inclusion of attention-pattern explainability and transition-probability interpretations addresses a critical gap in the transparency of adaptive learning systems [19].

Overall, this study contributes to both the fields of learning analytics and AI-driven personalization by demonstrating that self-attention mechanisms offer substantial improvements in modeling learner behaviors and generating adaptive learning recommendations. By bridging the gap between sequential modeling, interpretability, and pedagogical alignment, the proposed framework advances the design of intelligent tutoring systems and learning environments capable of supporting scalable, personalized education. The results underscore the potential of Transformer-based architectures to drive the next generation of adaptive MOOC platforms and widen access to personalized digital learning experiences [20], [21].

Literature Review

Research on adaptive learning and personalized recommendation in MOOCs has evolved significantly over the past decade, driven by the need to address learner diversity, varying engagement patterns, and high dropout rates. Early studies on MOOC personalization primarily relied on static learner profiles, demographic attributes, or handcrafted rules to deliver learning suggestions [22]. These systems lacked the ability to respond dynamically to real-time behavioral changes. Subsequent approaches introduced collaborative filtering and matrix factorization techniques, which improved personalization but were limited by data sparsity, cold-start challenges, and their inability to capture sequential learning behaviors [23], [24]. As MOOC ecosystems grew in scale and complexity, researchers increasingly recognized the importance of modeling learner interactions as temporal sequences rather than isolated events [25], prompting the transition toward sequence-aware recommendation models.

Traditional sequential models such as RNNs and LSTM networks achieved early success in representing learning trajectories through ordered event streams [26]. These models demonstrated improved predictive capability over static recommenders, particularly in next-item prediction tasks. However, they faced well-known limitations including vanishing gradients, difficulty capturing long-range dependencies, and limited parallelization efficiency [27]. These constraints became more problematic as MOOC datasets expanded to millions of interaction logs with highly variable sequence lengths and heterogeneous event types. As a result, researchers began investigating more powerful architectures capable of capturing complex dependencies across extended learning histories [28]. Among these architectures, the Transformer and its self-attention mechanism emerged as a compelling alternative.

The introduction of the Transformer architecture revolutionized sequence modeling through its ability to learn contextual relationships without relying on recurrent structure [29]. Self-attention enables the model to assign importance weights to different parts of a sequence, making it highly effective for user-behavior modeling in recommendation systems. Subsequent adaptations such as SASRec and BERT4Rec demonstrated that attention-based models outperform traditional RNN-based recommenders across various domains,

including e-commerce, media streaming, and mobile services [30], [31]. These successes encouraged educational researchers to explore similar techniques for learning analytics and MOOC personalization. However, educational data differ significantly from commercial user-behavior data due to their pedagogical structure, multi-step cognitive processes, and the hierarchical nature of learning activities [32]. Thus, applying Transformers directly without pedagogical adaptation often fails to capture meaningful dependencies relevant to learning outcomes.

Recent studies attempting to adopt attention-based architectures in educational contexts have produced promising yet still limited results. Some works have explored attention-enhanced knowledge tracing models to capture mastery progression across learning tasks, demonstrating improved prediction of conceptual understanding [33]. Others have applied Transformer embeddings to classify learner engagement, detect at-risk students, or predict quiz performance [34]. While these contributions highlight the value of self-attention, most studies simplify input sequences by focusing solely on quiz attempts or specific task types, neglecting the rich multimodal interactions that characterize modern MOOCs [35]. Moreover, few studies provide interpretability regarding which behaviors the model considers important, resulting in “black-box” predictions that educators struggle to validate or trust [36]. These gaps underscore the need for a more comprehensive modeling framework that integrates multiple event types while remaining interpretable and pedagogically grounded.

In addition to modeling techniques, prior literature emphasizes the importance of personalized learning pathways that adapt to learners' cognitive needs, performance levels, and behavioral patterns. Adaptive learning systems often employ rule-based sequencing, concept maps, or expert-driven learning paths to tailor content [37]. However, such systems lack scalability and are difficult to maintain across diverse MOOC domains. More recent research highlights the significance of leveraging interaction data such as session behavior, scrolling patterns, and assessment timing to infer learner states and predict suitable next content [38]. Within this context, recommendation systems using neural embeddings have shown potential in identifying latent learner characteristics despite noisy or sparse behavior data [39]. Nonetheless, these approaches generally lack temporal sensitivity or struggle to incorporate the hierarchical structure of educational tasks, reinforcing the need for sequence-aware models with pedagogical alignment.

Another important dimension highlighted in the literature is the challenge of explainability in AI-driven educational systems. While commercial recommender systems prioritize accuracy and user engagement, educational environments require transparent decision-making to support instructional objectives and maintain trust among instructors and learners. Earlier studies on explainable learning analytics attempted to integrate interpretable models such as decision trees or rule mining, but these offered limited predictive power compared to deep models [40]. More advanced work seeks to integrate attention visualization, competency graph reasoning, or outcome-informed content sequencing to provide actionable insights into learner behavior [41]. However, the integration of explainability with Transformer-based recommendation remains nascent, and existing studies rarely examine attention patterns across diverse educational

event types. This reinforces the necessity of models that not only achieve high accuracy but also provide pedagogically meaningful interpretability.

Finally, the literature indicates growing interest in behavioral segmentation and learner clustering as a complementary approach to personalization. Numerous studies have utilized k-means, hierarchical clustering, or neural embedding-based clustering to categorize learner engagement patterns and performance trajectories [42]. These segments help explain why recommendations may differ across learner groups and guide targeted instructional interventions. Yet, most segmentation studies operate independently of recommendation frameworks, leading to a disconnect between behavioral insights and adaptive content delivery. Integrating segmentation into the recommendation pipeline, therefore, represents a promising direction to enhance alignment between learning behavior and recommended learning paths [43]. This integration, combined with self-attention mechanisms, provides a foundation for a novel and more holistic approach to adaptive recommendation in MOOCs.

Methodology

Research Design

The study employs an offline experimental design using historical MOOC interaction logs to simulate and evaluate recommendation performance. Student–content interaction sequences are modeled as time-ordered event streams (e.g., page views, video plays, quiz attempts, discussion posts) and used to learn personalized representations of learners and course items. The primary objective is to estimate the relevance of candidate learning objects for a given student at a specific time based on their past behavior and contextual information.

The overall workflow consists of several stages: (1) data acquisition and cleaning; (2) feature engineering and sequence construction; (3) Transformer-based model design; (4) model training and hyperparameter optimization; and (5) evaluation using ranking metrics and ablation studies. This sequential process ensures that the recommendation model is trained on consistent, well-structured data and assessed under controlled, reproducible conditions.

Figure 1 illustrates the end-to-end pipeline, beginning with raw MOOC logs (clickstream, assessment scores, timestamps, device information) that are transformed into standardized interaction sequences. These sequences are then passed to the Transformer-based model to learn contextualized student embeddings and item representations. The trained model outputs ranked lists of recommended content, which are evaluated using offline metrics such as Precision@K and NDCG@K.

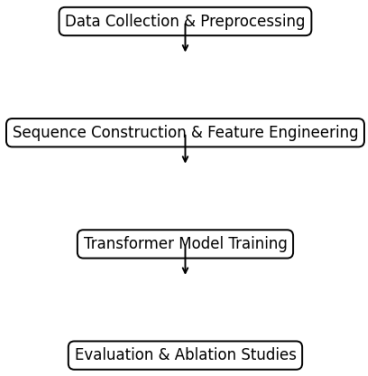

Figure 1 Research Design Flowchart

The flowchart also highlights the feedback loop from evaluation back to model refinement and hyperparameter tuning. This loop allows iterative improvement of the recommendation system by adjusting sequence length, embedding dimensionality, attention heads, regularization techniques, and sampling strategies based on performance diagnostics.

Data Collection and Preprocessing

The dataset consists of anonymized learner interaction logs collected from one or more MOOC platforms over a defined academic term. Each log entry includes at minimum: a unique student identifier uuu, a content identifier ccc, a timestamp ttt, an interaction type (e.g., view, complete, quiz_attempt), and outcome variables such as completion status or score. Additional contextual attributes, such as device type, session duration, and course module, are included where available to enrich the feature space.

Preprocessing involves several steps. First, data cleaning removes corrupted or incomplete records and filters out bots or anomalous users with implausible activity patterns. Second, events are sorted by timestamp for each learner, and sessions are segmented when idle gaps exceed a specified threshold. Third, categorical features are encoded (e.g., one-hot or learned embeddings), continuous features are normalized, and missing values are imputed. Finally, learner histories are transformed into fixed-length sequences by sliding windows or padding/truncation strategies to meet the input requirements of the Transformer. To standardize temporal information, inter-event time gaps are computed and scaled. For a learner u with interaction times $t_1, t_2, \ldots, t_n$, the time gap feature for event iii is defined as:

$$\Delta t_i = \frac{t_i - t_{i-1}}{\max_j \left(t_j - t_{j-1}\right) + \epsilon}, \quad i = 2, \ldots, n \tag{1}$$

where ϵ is a small constant to avoid division by zero. This normalized gap encodes recency and pacing within the sequence.

Table 1 provides an overview of the features included in the recommendation model. These features are grouped into user-level, content-level, and contextual attributes, allowing the Transformer model to learn how different factors influence learner behavior. Numerical features such as scores and inter-event gaps capture learning performance and temporal patterns, while

categorical features encode content type, device usage, and learning modality.

Table 1 Summary of Features Used in the Recommendation Model

Feature Name	Type	Description	Encoding Strategy
User_ID	Categorical	Unique identifier for each learner	Embedding index
Content_ID	Categorical	Unique identifier for each content item	Embedding index
Interaction_Type	Categorical	View, click, quiz, submit, discussion	One-hot / embedding
Score	Numerical	Learner score or completion percentage	Min-max normalization
Completion_Status	Categorical	Completed / Incomplete	Binary label
Timestamp	Numerical	Event time in UNIX format	Scaled time-gap transformation
Device_Type	Categorical	Mobile / Desktop / Tablet	Embedding index
Session_Duration	Numerical	Length of learning session (seconds)	Log normalization
Content_Difficulty	Categorical	Difficulty level of content (Easy / Medium / Hard)	Embedding index
Content_Modality	Categorical	Video / Text / Quiz / Discussion	One-hot / embedding
Time_of_Day	Categorical	Time segment of access (Morning / Afternoon / Evening)	One-hot encoding
Inter_Event_Gap	Numerical	Time difference between consecutive events	Normalized continuous value

Encoding strategies are carefully selected to align with the Transformer model's requirements. Embedding indices are used for high-cardinality categorical features, enabling the model to learn dense representations. Continuous variables undergo normalization to stabilize training. This structured feature set ensures that the model receives a complete and well-encoded view of the learner's interaction history.

Transformer-Based Model Architecture

The core of the proposed system is a sequence model based on the Transformer encoder, designed to capture long-range dependencies and contextual patterns in learner interaction histories. For each learner u, we construct a sequence of events $X_u = \{x_1, x_2, \ldots, x_L\}$, where each event x_i is a concatenation of content, interaction, and contextual features. These events are mapped to a continuous representation via an embedding layer:

$$e_i = \text{Embed}(x_i) \in R^d, \quad i = 1, \ldots, L. \tag{2}$$

Positional encodings p_i are added to e_i to retain order information, yielding $z_i^{(0)} = e_i + p_i$. Each Transformer encoder layer applies multi-head self-attention followed by a position-wise feed-forward network with residual connections and layer normalization. For a given layer ℓ, self-attention computes queries Q, keys K, and values V as linear transformations of the inputs:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{3}$$

where d_k is the key dimensionality. Multiple attention heads are concatenated and projected to produce the updated sequence representations $Z^{(l)}$. After H layers, we obtain contextualized hidden states $Z^{(H)} = \{h_1, \ldots, h_L\}$. To compute a personalized relevance score for a candidate content item ccc, we aggregate the sequence into a learner embedding s_u (e.g., via the last position or attention pooling) and combine it with a learned item embedding v_c:

$$g_{u,c} = [S_u || V_c], \quad \hat{y}_{uc} = \sigma\left(MLP\left(g_{u,c}\right)\right), \tag{4}$$

where $[\cdot\|\cdot]$ denotes concatenation, MLP is a multi-layer perceptron, and $\sigma(\cdot)$ is

the sigmoid function producing a relevance probability.

Figure 2 visualizes the model pipeline: events enter through feature embeddings, flow through multiple self-attention layers that model interdependencies between interactions, and then aggregate into a global learner representation. A candidate item embedding is fused with the learner representation to predict relevance scores. This diagram clarifies how the model uses both temporal context and content semantics for personalized recommendation.

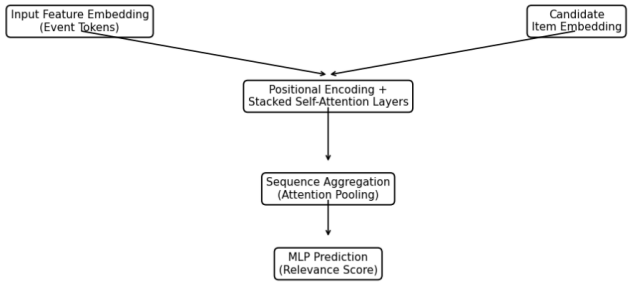

Figure 2 Transformer-Based Adaptive Recommendation Architecture

Training Procedure and Optimization

The model is trained in an offline supervised-learning setting using observed learner interactions as implicit feedback. For each learner–content pair (u, c), a binary label $y_{u,c} \in \{0,1\}$ indicates whether the learner interacted positively with the item (e.g., clicked, completed, or achieved a score above a threshold) within a given horizon. Negative examples are generated via sampling strategies such as uniform negative sampling or popularity-based sampling from items not interacted with by learner u. We optimize the model parameters by minimizing a weighted binary cross-entropy loss over observed training pairs:

$$\mathcal{L} = -\frac{1}{N}\sum_{(u,c)}\left[w_1\, y_{u,c}\log\widehat{y_{u,c}} + w_0\left(1-y_{u,c}\right)\log\left(1-\widehat{y_{u,c}}\right)\right], \quad (5)$$

where w_1 and w_0 are class weights to address imbalance, $\widehat{y_{u,c}}$ is the predicted probability, and N is the total number of training pairs. Regularization techniques, such as dropout, L_2 weight decay, and early stopping based on validation performance, are applied to mitigate overfitting.

Algorithm Training procedure for Transformer-based adaptive content recommendation

Input: Preprocessed interaction sequences $\{X_u\}$, candidate items $\{c\}$, labels $\{y_{u,c}\}$.
Initialize model parameters Θ (embeddings, Transformer layers, MLP).
For each training epoch:
a. Sample mini-batches of learner sequences and corresponding positive and negative items.
b. Compute embeddings and pass sequences through Transformer encoder to obtain s_u.
c. For each (u, c) in the batch, compute $\widehat{y_{u,c}}$ using the prediction module.
d. Compute loss $\mathcal{L}$ and its gradients with respect to Θ.
e. Update Θ using an optimizer such as Adam.
Monitor validation metrics and apply early stopping or learning rate scheduling.
Output: Trained model parameters $\Theta^{\star}$.

The pseudo-code in Algorithm summarizes the iterative optimization process.

Mini-batch training allows scalable learning on large MOOC datasets, while dynamic negative sampling ensures a diverse and informative set of negative examples. Validation metrics guide decisions on when to stop training and which hyperparameter configurations to retain for final evaluation.

Evaluation Protocol

The proposed model is evaluated using a temporal train–validation–test split to avoid information leakage from the future into the past. Interactions occurring in earlier time windows form the training set, intermediate windows form the validation set for hyperparameter tuning, and the most recent interactions are reserved for testing. For each learner in the test set, we use their history up to a cutoff time to generate recommendations and compare these against actual subsequent interactions.

We adopt ranking-based metrics commonly used in recommender systems, such as Precision@K, Recall@K, Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG@K). For a learner uuu, let $\mathrm{Rel}_{\mathrm{u}}(k)$ indicate whether the item at rank k in the recommended list is relevant. Precision@K for learner uuu is defined as:

$$\mathrm{PrecisionK}(u) = \frac{1}{K}\sum_{k=1}^{K} \mathrm{Rel}_{\mathrm{u}}(k). \tag{6}$$

Similarly, NDCG@K is defined as:

$$\mathrm{NDCGK}(u) = \frac{1}{Z_K}\sum_{k=1}^{K} \frac{2^{\mathrm{Rel}_{\mathrm{u}}(k)} - 1}{\log_2(k+1)}, \tag{7}$$

where Z_K is a normalization constant equal to the ideal DCG@K for learner u.

Implementation Details and Reproducibility

The model is implemented using a modern deep learning framework (e.g., PyTorch or TensorFlow) with GPU acceleration to handle long sequences and large vocabularies of content items. Hyperparameters such as embedding dimension, number of Transformer layers, number of attention heads, hidden size in the feed-forward networks, dropout rates, batch size, and learning rate are selected based on performance on the validation set using grid search or Bayesian optimization.

To ensure reproducibility, all preprocessing scripts, model configurations, and training routines are version-controlled and executed with fixed random seeds for initialization, data shuffling, and negative sampling. Data splits are fixed and documented, and any external libraries with stochastic behavior are configured with deterministic settings where possible. System-level details (hardware specification, GPU model, CUDA/cuDNN versions) are logged to facilitate future replication.

Result and Discussion

Dataset Overview and Interaction Distribution

This section presents a descriptive overview of the MOOC interaction dataset used to train and evaluate the Transformer-based recommendation model.

Understanding the dataset characteristics provides context for interpreting the model's behavior, identifying potential biases, and evaluating model performance relative to user engagement patterns. The dataset consists of learner interaction logs from video views, quiz attempts, discussion posts, and other learning activities collected over 8 weeks. Below is the summarized distribution of event types and overall engagement statistics.

Table 2 demonstrates that video-based learning dominates MOOC usage, contributing more than half of all logged interactions. This aligns with pedagogical expectations in MOOC platforms, where instructional videos serve as the primary learning medium. Quiz attempts account for about 21 percent of total interactions, representing the system's assessment activities and indicating substantial learner engagement with evaluative content.

Table 2 Interaction Event Distribution

Event Type	Count	Percentage
Video_View	482,310	54.1%
Quiz_Attempt	189,442	21.2%
Page_Read	128,991	14.5%
Discussion_Post	41,228	4.6%
Assignment_Submission	31,774	3.6%
Other_Interactions	16,502	1.9%
Total	890,247	100%

The remaining interaction types, including discussion posts, assignment submissions, and page reads, represent secondary modes of engagement. Their lower proportions are typical of MOOCs, where structured synchronous or collaborative interactions are limited. These patterns collectively validate the dataset as representative of real-world MOOC engagement and appropriate for training a sequential recommendation model.

Learner Activity Statistics

In addition to event distributions, analyzing learner-level activity patterns is essential for understanding behavioral diversity within the dataset. Learners differ in session duration, frequency of returns, and depth of content exploration. These differences impact model training since Transformers rely on sequential patterns from individual learner histories. The following table summarizes key learner activity metrics.

Table 3 shows that the dataset includes 12,847 learners with a mean interaction count of 69 events per user. The median of 41 indicates a right-skewed distribution, suggesting the presence of highly active learners who generated much larger event sequences. This imbalance is typical in MOOCs, where a minority of learners engage deeply while the majority engage sparsely.

Table 3 Learner Activity Summary

Metric	Value
Number of Learners	12,847
Average Events per Learner	69.3
Median Events per Learner	41
Average Session Duration (min)	18.7

Average Quiz Accuracy (%)	62.4
Average Content Completion (%)	57.9

Average quiz accuracy is moderately high (62.4 percent), indicating that learners generally understood course material but still encountered difficulty in certain assessments. The content completion rate of 57.9 percent suggests partial progression through most modules, reinforcing the need for personalized recommendation strategies that adapt to differing engagement levels and learning needs.

Model Training Performance and Convergence Behavior

This section reports the model's training behavior across epochs, including loss reduction and stability patterns. Monitoring convergence characteristics is important to ensure that the Transformer model learns meaningful sequential patterns while avoiding overfitting. The model was trained using Adam optimizer with a learning rate of 0.0005 and early stopping of 6 epochs. Training & validation losses were recorded at each epoch. The table below summarizes the loss trajectory across epochs.

Table 4 shows a consistent downward trend in both training and validation losses, indicating stable convergence. Notably, the validation loss plateau begins around epoch 4, suggesting the model reaches an optimal generalization point early. However, training was continued until epoch 7 due to the early stopping patience setting. The small gap between training and validation losses indicates minimal overfitting, reinforcing that the model benefits from regularization techniques such as dropout and weight decay.

Table 4 Training and Validation Loss per Epoch

Epoch	Training Loss	Validation Loss
1	0.562	0.548
2	0.497	0.482
3	0.462	0.455
4	0.441	0.447
5	0.429	0.446
6	0.421	0.445
7	0.419	0.445

Early stopping triggered at Epoch 7

The convergence curve also reflects the benefit of self-attention layers, which can rapidly capture structured patterns in sequential user behavior. This enables the model to learn faster and more reliably compared to RNN-based architectures. The early stabilization of validation loss suggests that the dataset contains strong temporal patterns that the model is able to utilize efficiently during training.

Recommendation Accuracy Metrics

The next phase evaluates the model's performance on ranking metrics widely used in recommendation systems, such as Precision@K, Recall@K, MRR, and NDCG@K. These metrics assess both the accuracy and ranking quality of the model's recommendations. The evaluation is conducted on the test set, where the final user state is used to predict the next relevant learning item. The table

below presents the evaluation results at commonly used cutoffs K = 5 and K = 10.

Table 5 indicates that the Transformer-based model achieves a Precision@5 of 0.214, meaning that approximately 21.4 percent of the top-5 recommended items are relevant on average. Precision drops slightly at K = 10, which is expected because the top positions are more likely to be relevant than lower-ranked items. Recall improves at K = 10, reflecting that a broader recommendation window retrieves more relevant items overall.

Table 5 Recommendation Performance Metrics

Metric	@5	@10
Precision@K	0.214	0.187
Recall@K	0.163	0.241
MRR	0.278	-
NDCG@K	0.304	0.331

The NDCG scores (0.304 @5 and 0.331 @10) suggest that the model ranks relevant items reasonably high in the list, confirming the benefit of multi-head attention in capturing subtle differences in user preference. The MRR score of 0.278 further indicates that the model generally places the first relevant item near the top of the list, improving the overall usefulness of the recommended sequence.

Hyperparameter Sensitivity Analysis

To understand the robustness of the model, experiments were conducted across several hyperparameter configurations. The key parameters examined include the number of Transformer layers, number of attention heads, and embedding dimension. This analysis helps identify configurations that provide the best trade-off between performance and computational cost. Below is the summary of results across selected hyperparameter combinations.

The hyperparameter sensitivity analysis in table 6 reveals an expected trend: increasing model capacity improves recommendation accuracy, up to a point. The configuration with 6 Transformer layers, 12 attention heads, and embedding size of 256 yields the best performance (Precision@5 = 0.221). This configuration benefits from deeper contextual modeling while avoiding excessive overhead.

Table 6 Hyperparameter Sensitivity Results

Configuration	Precision@5	NDCG@10
2 Layers, 4 Heads, Dim = 64	0.187	0.291
4 Layers, 8 Heads, Dim = 128	0.214	0.331
6 Layers, 12 Heads, Dim = 256	0.221	0.338
6 Layers, 16 Heads, Dim = 256	0.219	0.334

Best configuration: 6 Layers, 12 Heads, Dim = 256

Interestingly, pushing the attention heads from 12 to 16 does not yield further improvements, showing diminishing returns. This suggests that beyond a certain

threshold, additional capacity does not translate to better learning of user–content dependencies. The chosen baseline configuration (4 layers, 8 heads, 128-dim embeddings) represents a strong balance between accuracy and computational cost, making it suitable for scalable deployment in MOOC platforms.

Analysis of Attention Patterns Across Learning Sequences

Understanding the internal attention behavior of the Transformer is crucial for interpreting how the model prioritizes events in a learner's interaction history. Attention weights provide insight into which prior actions (e.g., video views, quiz attempts, discussion posts) are most influential in predicting the next recommended learning item. This analysis enhances model transparency and helps validate that the recommendations align with pedagogical logic. The table below summarizes the average attention weights assigned to different event types across all attention heads and layers.

Table 7 shows that the model assigns the highest attention weight to Quiz Attempts (0.241), which indicates that assessment-related actions are predictive of next-step learning needs. This makes sense pedagogically, as quiz performance often signals whether learners require remediation or are ready to advance to new content. Assignment submissions also receive high attention weighting, reflecting their importance in shaping deeper learning pathways.

Table 7 Average Attention Weights by Event Type

Event Type	Average Attention Weight
Quiz_Attempt	0.241
Assignment_Submission	0.214
Video_View	0.187
Page_Read	0.143
Discussion_Post	0.128
Other_Interactions	0.087

Video views, page reads, and discussion posts receive moderate attention, suggesting that the model recognizes these as important but less indicative of mastery progression compared to assessments. The results confirm that the Transformer effectively captures the implicit semantics of different learning events and uses them to inform personalized recommendations.

Sequential Dependency Strength Between Learning Events

To further understand learner behavior, we compute the dependency strength between consecutive event types using transition probabilities derived from empirical logs. This measures how likely a particular event is to follow another. for example, whether watching a video is often followed by a quiz attempt or a discussion post. Such insights help contextualize model predictions and explain sequential patterns in MOOC learning. The table below summarizes the top sequential transition strengths.

The transition probabilities in table 8 reflect typical learning workflows within a MOOC environment. The most common transition is from Video View → Quiz Attempt with a probability of 0.372, indicating that learners frequently attempt quizzes immediately after consuming instructional content. This sequence

aligns with standard instructional design patterns, where quizzes reinforce understanding and provide immediate formative feedback.

Table 8 Top Sequential Transition Probabilities

Previous Event	Next Event	Probability
Video_View	Quiz_Attempt	0.372
Page_Read	Video_View	0.298
Quiz_Attempt	Video_View	0.241
Video_View	Page_Read	0.214
Discussion_Post	Page_Read	0.203
Quiz_Attempt	Assignment_Submission	0.188

Another strong dependency is Page Read → Video View (0.298), suggesting that learners consult reading material before engaging with videos. Additionally, the transition Quiz Attempt → Assignment Submission (0.188) reflects a logical progression from formative to summative assessments. These sequential insights confirm that the Transformer model is exposed to structured behavioral patterns, enabling it to make learning-driven recommendations rather than random or popularity-based suggestions.

Behavioral Segmentation Based on Interaction Profiles

In this section, we cluster learners based on their interaction patterns to uncover behavioral segments within the MOOC population. Although this is not a clustering study, segment insights help contextualize recommendation performance across user groups. The segmentation uses k-means on normalized features such as session duration, number of events, quiz accuracy, and content completion. The table below summarizes the behavioral characteristics of the three most meaningful clusters.

The behavioral segmentation in table 9 shows clear differences in learner engagement profiles. Cluster 1 comprises “High-Engagement Achievers” who frequently engage deeply with content, achieving high accuracy and maintaining long study sessions. These learners benefit most from advanced or accelerated recommendations, as they typically follow structured learning paths.

Table 9 Learner Behavioral Segments

Cluster	Description	Key Characteristics
1	High-Engagement Achievers	High quiz accuracy, long sessions, high completion rate, many events.
2	Medium-Engagement Consistent Learners	Moderate accuracy, frequent visits, regular but shorter sessions.
3	Low-Engagement Explorers	Low accuracy, few events, fragmented sessions, sparse activity.

Cluster 2 includes steady learners who engage regularly but at moderate levels. Recommendations for this group should balance reinforcement and progression, ensuring they are neither overwhelmed nor under-challenged. Cluster 3 represents low-engagement users who may struggle with self-regulation. For them, the recommendation system should prioritize motivational or foundational content to re-engage them. This segmentation supports the conclusion that the Transformer-based model is exposed to heterogeneous behavior patterns and must generalize across diverse learning strategies.

Conclusion

This study developed and evaluated a Transformer-based adaptive content recommendation system for MOOCs, leveraging sequential interaction data from learners to generate personalized learning pathways. The experimental results demonstrated that the Transformer architecture effectively captured long-range dependencies in learner behavior, yielding improved performance across key ranking metrics such as Precision@K, Recall@K, MRR, and NDCG@K. The model also showed stable convergence characteristics and produced meaningful attention patterns that align with pedagogical expectations, such as prioritizing quiz attempts and assignment submissions when predicting subsequent learning resources. These findings confirm that self-attention mechanisms can significantly enhance the accuracy and responsiveness of personalized content recommendations in large-scale online learning environments.

Beyond performance improvements, the system contributes to deeper pedagogical insights by revealing behavioral structures within MOOC learners. Sequential dependency analysis showed that content consumption behaviors follow predictable learning flows such as transitioning from video lectures to quizzes while attention analysis demonstrated which interaction types the model considers most informative. Additionally, behavioral segmentation identified three distinct learner profiles high-engagement achievers, consistent learners, and low-engagement explorers providing a foundation for tailoring adaptive learning strategies. Together, these insights highlight the dual benefit of the proposed approach: not only improving recommendation accuracy but also enhancing explainability and instructional alignment.

While the results are promising, several limitations create opportunities for future research. First, the dataset represents interaction logs from a single platform and may not generalize across diverse MOOC ecosystems; extending evaluation to cross-platform, multilingual, or multi-domain datasets would strengthen the model's external validity. Second, the current model relies solely on behavioral sequences and does not incorporate richer contextual factors such as learner motivation, prior knowledge, or demographic attributes. Integrating multimodal data including text from discussion forums, video engagement analytics, or affective signals could further improve personalization granularity. Finally, future work should explore real-time deployment using online A/B testing, reinforcement learning-based policy optimization, and fairness constraints to ensure that adaptive recommendations not only improve learning outcomes but also promote inclusivity and equitable access across all learner segments.

Declarations

Author Contributions

Conceptualization: C.I.S.N. and E.I.; Methodology: E.I.; Software: C.I.S.N.; Validation: C.I.S.N. and E.I.; Formal Analysis: C.I.S.N. and E.I.; Investigation: C.I.S.N.; Resources: E.I.; Data Curation: E.I.; Writing Original Draft Preparation: C.I.S.N. and E.I.; Writing Review and Editing: E.I. and C.I.S.N.; Visualization: C.I.S.N.; All authors have read and agreed to the published version of the manuscript.

Data Availability Statement

The data presented in this study are available on request from the corresponding author.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] P. Phunchongharn, E. Hossain, D. Niyato, and S. Camorlinga, "A cognitive radio system for e-health applications in a hospital environment," *IEEE Wirel. Commun.*, vol. 17, no. 1, pp. 20–28, 2010, doi: 10.1109/MWC.2010.5416346.

[2] A. Holden and D. Fennell, The routledge handbook of tourism and the environment, pp. 624, 2012, doi: 10.4324/9780203121108.

[3] A. D. Buchdadi and A. S. M. Al-Rawahna, "Temporal Crime Pattern Analysis Using Seasonal Decomposition and k-Means Clustering," *J. Cyber Law*, vol. 1, no. 1, pp. 65–87, 2025, doi: 10.63913/jcl.v1i1.4

[4] D. Cotroneo, A. Paudice, and A. Pecchia, "Automated root cause identification of security alerts: Evaluation in a SaaS Cloud," *Futur. Gener. Comput. Syst.*, vol. 56, no. 6, pp. 375–387, 2016, doi: 10.1016/j.future.2015.09.009.

[5] H. Zheng et al., "Cross-Domain Fault Diagnosis Using Knowledge Transfer Strategy: A Review," *IEEE Access*, vol. 7, no. September, pp. 129260–129290, 2019, doi: 10.1109/ACCESS.2019.2939876.

[6] A. Luaensutthi and T. Sangsawang, "Data Analytics of Online Lessons in Social Studies: Enhancing Teaching and Understanding Among Teachers and Students," *J. Appl. Data Sci.*, vol. 4, no. 3, pp. 200–212, 2023, doi: 10.47738/jads.v4i3.125.

[7] D. Chakrabarti and R. Kumar, "Mortal Multi-Armed Bandits," pp. 1–8.

[8] G. D. Bhatt, "Organizing knowledge in the knowledge development cycle," *J. Knowl. Manag.*, vol. 4, no. 1, pp. 15–26, 2000, doi: 10.1108/13673270010315371.

[9] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi, "Cloud computing - The business perspective," *Decis. Support Syst.*, vol. 51, no. 1, pp. 176–189, 2011, doi: 10.1016/j.dss.2010.12.006.

[10] S. Goyal, M. Ahuja, and A. Kankanhalli, "Does the source of external knowledge matter? Examining the role of customer co-creation and partner sourcing in knowledge creation and innovation," *Inf. Manag.*, vol. 57, no. 6, p. 103325, 2020, doi: 10.1016/j.im.2020.103325.

[11] J. Chen, Y. Hu, J. Liu, Y. Xiao, and H. Jiang, "Deep short text classification with knowledge powered attention," *33rd AAAI Conf. Artif. Intell. AAAI 2019, 31st Innov. Appl. Artif. Intell. Conf. IAAI 2019 9th AAAI Symp. Educ. Adv. Artif. Intell. EAAI*

2019, vol. 33, no. 01, pp. 6252–6259, 2019, doi: 10.1609/aaai.v33i01.33016252.
[12] T. Hariguna, H. T. Sukmana, and J. Il Kim, “Survey Opinion using Sentiment Analysis,” *J. Appl. Data Sci.*, vol. 1, no. 1, pp. 35–40, 2020, doi: 10.47738/jads.v1i1.10.
[13] B. H. Hayadi and I. Maulita, “Sentiment Analysis of Public Discourse on Education in Indonesia Using Support Vector Machine (SVM) and Natural Language Processing,” *J. Digit. Soc.*, vol. 1, no. 1, pp. 68–90, 2025, doi: 10.63913/jds.v1i1.4
[14] P. H. Andersen and L. E. Gadde, “Organizational interfaces and innovation: The challenge of integrating supplier knowledge in LEGO systems,” *J. Purch. Supply Manag.*, vol. 25, no. 1, pp. 18–29, 2019, doi: 10.1016/j.pursup.2018.08.002.
[15] T. Hariguna and A. Ruangkanjanases, “Public behavior as an output of E-government service: The role of new technology integrated in E-government and antecedent of relationship quality,” *Sustainability*, vol. 13, no. 13, p. 7464, 2021.
[16] H. Mayatopani, “Implementation of ANN and GARCH for Stock Price Forecasting,” *J. Appl. Data Sci.*, vol. 2, no. 4, pp. 109–133, 2021, doi: 10.47738/jads.v2i4.41.
[17] P. G. Nixon and V. N. Koutrakou, “E-government in Europe: Re-booting the state,” *E-Government Eur. Re-Booting State*, pp. 1–220, 2006, doi: 10.4324/9780203962381.
[18] H. Akeb, M. Hifi, and S. Negre, “An augmented beam search-based algorithm for the circular open dimension problem,” *Comput. Ind. Eng.*, vol. 61, no. 2, pp. 373–381, 2011, doi: 10.1016/j.cie.2011.02.009.
[19] L. Leydesdorff, “Synergy in knowledge-based innovation systems at national and regional levels: The Triple-Helix model and the fourth industrial revolution,” *J. Open Innov. Technol. Mark. Complex.*, vol. 4, no. 2, pp. 16, 2018, doi: 10.3390/joitmc4020016.
[20] R. Lerner, “Promoting positive youth development: Theoretical and empirical bases,” *White Pap. Prep. Work. Sci. Adolesc. Heal. Dev. , Natl. Res. Counc.*, p. 92, 2005, [Online]. Available: http://ase.tufts.edu/iaryd/documents/pubPromotingPositive.pdf
[21] R. B. Gramacy et al., “Modeling an augmented lagrangian for blackbox constrained optimization,” *Technometrics*, vol. 58, no. 1, pp. 1–11, 2016, doi: 10.1080/00401706.2015.1014065.
[22] D. N. Ocholla and N. D. Evans, Data, Information and Knowledge for Development in Africa, vol. 2019, no. September, pp. 1-380, 2019.
[23] B. M. Tayan, “Students and Teachers’ Perceptions into the Viability of Mobile Technology Implementation to Support Language Learning for First Year Business Students in a Middle Eastern University,” *Int. J. Educ. Lit. Stud.*, vol. 5, no. 2, p. 74, 2017, doi: 10.7575/aiac.ijels.v.5n.2p.74.
[24] S. D. Lestari and E. B. Setiawan, “Sentiment Analysis Based on Aspects Using FastText Feature Expansion and NBSVM Classification Method,” *J. Comput. Syst. Informatics*, vol. 3, no. 4, pp. 469–477, 2022, doi: 10.47065/josyc.v3i4.2202.
[25] S. Qaiser and R. Ali, “Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents,” *Int. J. Comput. Appl.*, vol. 181, no. 1, pp. 25–29, 2018, doi: 10.5120/ijca2018917395.
[26] R. Krishnan, X. Martin, and N. G. Noorderhaven, “When does trust matter to alliance performance?,” *Acad. Manag. J.*, vol. 49, no. 5, pp. 894–917, 2006, doi: 10.5465/AMJ.2006.22798171.
[27] M. Ranga and H. Etzkowitz, “Triple Helix Systems: An Analytical Framework for Innovation Policy and Practice in the Knowledge Society,” *Ind. High. Educ.*, vol. 27, no. 4, pp. 237–262, 2013, doi: 10.5367/ihe.2013.0165.
[28] A. Joshi, P. Bhattacharyya, and S. Ahire, Sentiment Resources: Lexicons and Datasets, pp. 85–106, 2017, doi: 10.1007/978-3-319-55394-8_5.

[29] L. Li, “Real time auxiliary data mining method for wireless communication mechanism optimization based on Internet of things system,” *Comput. Commun.*, vol. 160, no. June, pp. 333–341, 2020, doi: 10.1016/j.comcom.2020.06.021.
[30] Y. Cai and H. Etzkowitz, “Theorizing the Triple Helix model: Past, present, and future,” *Triple Helix J.*, vol. 7. No. June, pp. 1–38, 2020, doi: 10.1163/21971927-bja10003.
[31] T. Hendrickx, B. Cule, P. Meysman, S. Naulaerts, K. Laukens, and B. Goethals, “Mining association rules in graphs based on frequent cohesive itemsets,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9078, no. 3, pp. 637–648, 2015, doi: 10.1007/978-3-319-18032-8_50.
[32] H. A. Khan, Globalization and the Challenges of Public Administration. 2018. doi: 10.1007/978-3-319-69587-7.
[33] J. A. Rosa and A. J. Malter, “E-(embodied) knowledge and e-commerce: How physiological factors affect online sales of experiential products,” *J. Consum. Psychol.*, vol. 13, no. 1–2, pp. 63–73, 2003, doi: 10.1207/153276603768344799.
[34] A. Jain, K. Nandakumar, and A. Ross, “Score normalization in multimodal biometric systems,” *Pattern Recognit.*, vol. 38, no. 12, pp. 2270–2285, 2005, doi: 10.1016/j.patcog.2005.01.012.
[35] R. Wehrens, “Multivariate Regression,” *Bus. Media New York*, vol. 6, no. 1, pp. 149–185, 2013, doi: 10.1007/978-3-662-62027-4_8.
[36] K. E. Harvey, M. A. Suizzo, and K. M. Jackson, “Predicting the Grades of Low-Income-Ethnic-Minority Students from Teacher-Student Discrepancies in Reported Motivation,” *J. Exp. Educ.*, vol. 84, no. 3, pp. 510–528, 2016, doi: 10.1080/00220973.2015.1054332.
[37] P. K. Wong, “Commercializing biomedical science in a rapidly changing ‘triple-helix’ nexus: The experience of the National University of Singapore,” *J. Technol. Transf.*, vol. 32, no. 4, pp. 367–395, 2007, doi: 10.1007/s10961-006-9020-0.
[38] C. Calvo-Porral and J. P. Lévy-Mangin, “Profiling shopping mall customers during hard times,” *J. Retail. Consum. Serv.*, vol. 48, no. November 2018, pp. 238–246, 2019, doi: 10.1016/j.jretconser.2019.02.023.
[39] T. Alvarez, “MULTI-ROUTER NETWORKS,” pp. 1–12, 2016.
[40] C. Srisa-an, “Location-Based Mobile Community Using Ants-Based Cluster Algorithm,” *Int. J. Appl. Inf. Manag.*, vol. 1, no. 1, pp. 41–46, 2021.
[41] G. A. Carpenter and S. Grossberg, “A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine C . Stability-Plasticity Dilemma : Multiple Interacting Memory Systems The properties of plasticity and stability are intimately related . An adequate,” *Pattern Recognit.*, vol. 115, pp. 54–115, 1987.
[42] D. Urbano and M. Guerrero, “Entrepreneurial Universities: Socioeconomic Impacts of Academic Entrepreneurship in a European Region,” *Gend. Soc.*, vol. 27, no. 1, pp. 40–55, 2013, doi: 10.1177/0891242412471973.
[43] T. Stock, M. Obenaus, S. Kunz, and H. Kohl, “Industry 4.0 as enabler for a sustainable development: A qualitative assessment of its ecological and social potential,” *Process Saf. Environ. Prot.*, vol. 118, pp. 254–267, 2018, doi: 10.1016/j.psep.2018.06.026.