applied sciences

MDPI

Article

MOOC Learner Profiling Using Relation-Aware Heterogeneous Graph Neural Networks

Bo Jiang [1,†], Xi Chen [1,†], Mingzheng Li [1,†], Zimeng Shan [1], Yuhan Liu [1] and Naidi Yang [2,*]

1 School of Educational Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing 210049, China; jiangbo@njupt.edu.cn (B.J.); 1023162811@njupt.edu.cn (X.C.); mingz3142875304@gmail.com (M.L.); 1024163103@njupt.edu.cn (Z.S.); b23170404@njupt.edu.cn (Y.L.)
2 Key Laboratory of Flexible Electronics (KLOFE), School of Flexible Electronics (Future Technologies), Institute of Advanced Materials (IAM), Nanjing Tech University (NanjingTech), Nanjing 211816, China
* Correspondence: iamndyang@njtech.edu.cn
† These authors contributed equally to this work.

Abstract

Research on learner profiling in education primarily focuses on utilizing students' personal characteristics and behavioral data to depict their learning status and traits. However, existing methods often face challenges such as incomplete data and difficulties in feature extraction, leading to incomplete and less accurate learner profiles. To address information gaps in learner profiles within educational datasets, this study proposes a profile completion technique based on relation-aware heterogeneous graph networks. Using the MOOCCube and MOOCCubeX datasets, we trained a relation-aware heterogeneous graph network model to predict students' age and gender. The model achieved significant advancements in gender prediction. While age prediction performance remains relatively low—a common challenge in the field due to the subtle and multifaceted nature of age-related behavioral signals—ablation studies confirm the model's robustness, demonstrating stable gender prediction accuracy even with significant data reduction. This work bridges information gaps in learner profiling within educational datasets, providing crucial support for personalized education and teaching quality improvement. It showcases the potential application of relation-aware heterogeneous graph networks in education and offers new ideas for research utilizing heterogeneous graph networks for learner profiling.

Keywords: learner profile; graph neural networks; heterogeneous information networks; MOOCCubeX

1. Introduction

The pursuit of personalized education in Massive Open Online Courses (MOOCs) relies on a deep understanding of diverse learners. Constructing comprehensive learner profiles is fundamental to this endeavor, as they integrate demographics, learning behaviors, and outcomes to enable tailored instructional support. However, the construction of these profiles is severely hampered by the inherent sparsity, heterogeneity, and incompleteness of educational datasets [1]. A significant challenge lies in the frequent absence of basic demographic information, such as age and gender, often due to privacy concerns or optional user reporting. This data gap limits the effectiveness of personalized learning systems.

Traditional imputation methods, including statistical techniques and conventional machine learning, often fail to capture the complex, relational nature of educational data [2]. They typically treat data instances in isolation, overlooking the rich contextual information

Check for updates

Academic Editor: Rui Araújo

Received: 8 November 2025
Revised: 22 December 2025
Accepted: 23 December 2025
Published: 26 December 2025

embedded in the relationships between learners, courses, and content. Recently, Graph Neural Networks (GNNs) have emerged as a powerful paradigm for modeling relational data [3]. In particular, Heterogeneous GNNs are adept at handling multi-typed nodes and edges, making them suitable for educational ecosystems [4]. However, directly applying existing heterogeneous GNNs for learner profile completion is suboptimal, as they often do not explicitly model the distinct semantic meanings of different relation types in the learning process.

To bridge this gap, we propose a novel approach for learner profile completion using a Relation-Aware Heterogeneous Graph Network (RA-HGN). Our core insight is that the multi-faceted interactions in a MOOC platform—such as a learner watching a video, enrolling in a course, or possessing an attribute—carry distinct relational semantics. By explicitly modeling these differing intents through a relation-aware attention mechanism, our method can more effectively aggregate information from a learner's neighborhood in the graph to infer missing profile attributes. The main contributions of this paper are as follows:

- We successfully applied a relational-aware heterogeneous graph network in the field of education, representing a cutting-edge approach which explicitly models distinct semantic relationships through a novel meta-relation aware attention mechanism.
- By predicting students' gender and age, we have filled in the missing learner profiles within the education dataset. This holds significant implications for personalized education and learner behavior analysis.
- We demonstrate the robustness of our approach, particularly for gender prediction, which maintains high accuracy even with significantly reduced data, thereby addressing data sparsity issues common in educational datasets.
- Our work provides a flexible graph-based framework that bridges information gaps in learner profiles and paves the way for predicting more pedagogically relevant attributes in the future.

2. Related Work

2.1. Graph Neural Networks

Graph Neural Networks (GNNs) [5] have gained significant attention in recent years as a machine learning technique, initially proposed by Gori et al. and Scarselli et al., aiming to address the challenges of modeling and analyzing graph data. Traditional neural network models, such as Convolutional Neural Networks [6] and Recurrent Neural Networks [7], are primarily suited for structured data like images and sequences, exhibiting limitations when dealing with irregular graph structures. In many real-world scenarios, data manifests as graphs, where entities and their connections are naturally represented as graphs, such as social networks [8], recommendation systems [9], and user profiles [10]. Traditional machine learning methods face challenges in directly processing such data, as they fail to account for the topological relationships and local structures among nodes. The emergence of Graph Neural Networks fills this gap, providing an effective solution for deep learning on graph data. Graph Neural Networks achieve representation learning for nodes and edges by defining and operating neural network layers on graph structures, thereby better capturing the topological information and node features within graphs. This makes Graph Neural Networks an ideal choice for handling various complex graph-structured data [2].

In recent years, GNNs have been widely applied to different tasks such as recommendation systems [11,12], image classification [13], session prediction [14,15], and physical simulations [16,17]. As a modification of GNN, Gated Graph Neural Networks [18] uses gated recurrent units and employs back-propagation through time (BPTT) to compute gradients. Graph Attention Networks (GAT) [19] applies the attention mechanism to learn

https://doi.org/10.3390/app16010263

the weight of nodes and neighbor nodes. M. Schlichtkrull et al. [20] introduced Relational Graph Convolutional Networks (R-GCNs) and applied them to two standard knowledge base completion tasks: Link prediction (recovery of missing facts, i.e., subject–predicate–object triples) and entity classification (recovery of missing entity attributes). They demonstrated the effectiveness of R-GCNs as a standalone model for entity classification. Afshin Rahimi, Trevor Cohn, and Timothy Baldwin [21] proposed a Graph Convolutional Network (GCN) that utilizes both text and network information to infer user location in a joint setting. Mengqi Zhang et al. [22] proposed A-PGNN, which integrates Personalized Graph Neural Networks (PGNN) with a Dot-Product Attention mechanism. This approach addresses the limitations of existing session-aware recommendation methods by effectively capturing complex item transitions and explicitly distinguishing the effects of different historical sessions. However, these works often focus on homogeneous graphs or specific prediction tasks, leaving the problem of holistic learner profile completion in rich, heterogeneous information networks relatively unexplored.

Homogeneous Graph and Heterogeneous Graph are two fundamental types of graphs, primarily distinguished by the types of nodes and edges in the graph. In a Heterogeneous Graph, nodes and edges can belong to multiple different types, implying diverse relationships between nodes and varied node attributes. Heterogeneous graphs often better capture the diversity and complexity of relationships between different entities in the real world. The Relation-aware Heterogeneous Graph Network [4] is a neural network model designed for processing Heterogeneous Graph data. It efficiently captures complex relationships between nodes in a Heterogeneous Graph, extracting features and representations of nodes. This technology combines Graph Neural Networks and attention mechanisms, capable of handling Heterogeneous Graph data with multiple node and edge types. It dynamically calculates node importance based on the weights between nodes, extracting crucial features and representations to generate more accurate learner profiles. The closest related work is by Yan et al. [23], who employed a relation-aware heterogeneous graph for user profiling in e-commerce. They demonstrated the superiority of modeling diverse user-item interactions for predicting user attributes. Our study is distinguished by its adaptation and extension of this concept to the educational domain, where the relational semantics and the end goal of supporting learning present unique challenges and opportunities.

2.2. Learner Profile

Learner profiles [24] are abstract, labeled learner models derived from personal attributes and real learning data, products of the era of educational informatization, big data, and artificial intelligence. They are abstract, labeled learner models derived from the personal attributes of learners and a series of real learning data. Using a tagging format, learner profiles precisely depict and visually present various dimensional features such as basic information, knowledge levels, learning engagement, and style preferences based on learners' personal attributes and authentic learning data. Learner profiles play a critical role in education, enabling teachers to adjust teaching strategies and provide targeted instructional content and activities to meet diverse student learning needs through the analysis of these profiles. The study by Andino Maseleno et al. [25] concludes that learning analytics, when tailored to individual needs, enhances student learning by providing personalized support and recommendations. It emphasizes the significance of integrating personalized learning into the framework of learning analytics. During instruction, dynamically updated learner profiles offer real-time data references for teachers. Teachers can adapt teaching methods and content based on students' performance in the classroom, thereby offering timely personalized guidance and support to enhance teaching effectiveness. Ganesan Kavitha and Lawrance Raj [26] utilized decision tree techniques within data mining classification to categorize student assessment result databases. This approach identified which students require support to achieve good grades and pass the course's final exam. Additionally, it aids teachers in assessing whether their current instructional strategies are suitable for their students' current learning abilities. Moreover, learner profiles can serve as an early warning system for monitoring students' learning progress.

Common dimensions of learner profiles include basic attributes, psychological attributes, behavioral attributes, skill attributes, and outcome attributes, among others (as shown in Figure 1). In simple terms, basic attributes involve identifying who the subject of the profile is, and relevant information about learner basic attributes can be obtained through the personal center of schools or online learning platforms. As for data related to the psychological attributes of learners, real-time facial and motion recognition, emotion analysis through artificial intelligence technology, and more complex data obtained through methods like electroencephalography (EEG) are possible. However, in the current scenario where such technological conditions are not universally widespread, surveys remain a good method for obtaining relevant psychological data. Behavioral attributes, skill attributes, or competency attributes mainly rely on various digital traces left by learners during the learning process to reflect and characterize them. Outcome attributes correspond to task completion and various stage test scores. These data can be integrated and presented through intelligent learning platforms like smart classrooms, or rapidly fetched from learning platform web pages using Python 3.14.0 web scraping techniques.

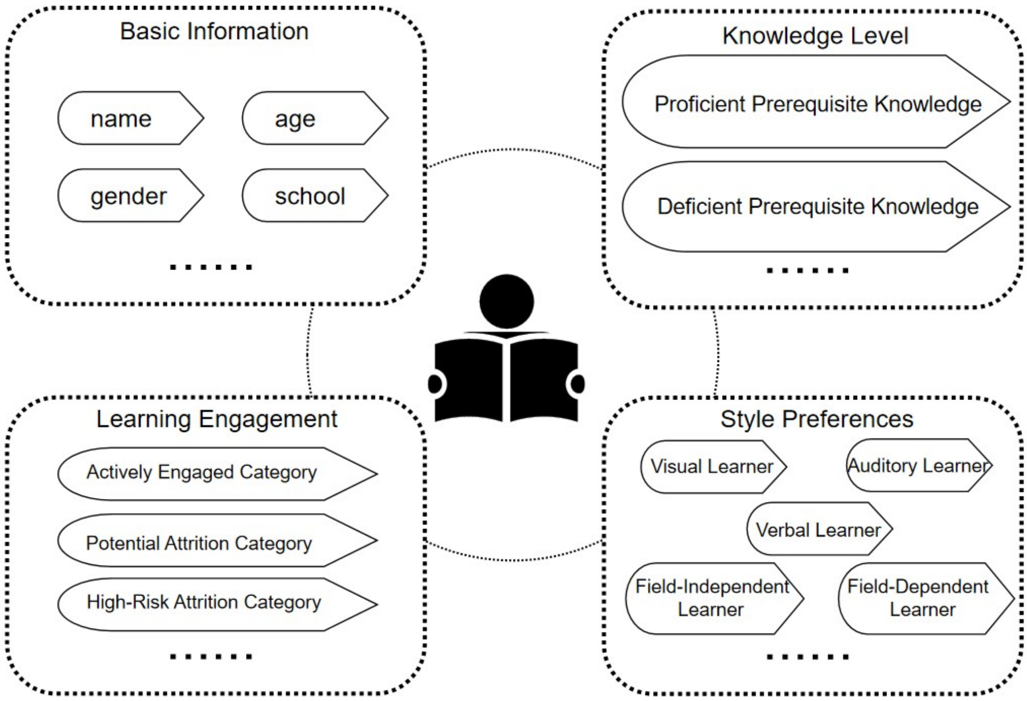

Figure 1. Learner profiles depict the different dimensional characteristics of learners in the form of labels.

The construction of learner profiles involves gathering, integrating, and analyzing various types of data from educational datasets. Previous research indicates that traditional methods for building learner profiles primarily rely on statistical analysis and machine learning techniques, such as rule-based imputation, clustering analysis, and classification models. Soft clustering techniques, such as LCA, are a sophisticated form of unsupervised machine learning that assigns data points to each cluster based on probability [27–29]. LCA has been used in learning analytics to create learner profiles in various educational environments [30–33]. A. Zamecnik et al. [34] utilized learning analytics techniques to study non-traditional learners in a school in Australia. They employed these techniques to reveal the characteristics of non-traditional learners in an online environment and identified a range of learner profiles. Through questionnaire surveys and interviews, the learner profiles constructed by Maaike Koopman and Douwe Beijaard [35] indicate differences in learning strategies among different types of learners, highlighting the need for teachers to provide adequate support to students during the instructional process.

With the advent of educational data mining, methods have shifted towards leveraging behavioral trace data. Meghji et al. [36] developed a classification model using real student data to predict academic performance early in the semester. They tested various algorithms and found that the Random Forest classifier achieved the highest accuracy of 93.40%, providing valuable insights for educators to monitor and support student progress, potentially mitigating academic failure and encouraging higher performance. Chen-Hsuan Liao and Jiun-Yu Wu [37] investigated the click records of students' video viewing behaviors. After conducting K-means clustering analysis on the collected data, the research findings indicate that persistent individuals often pause videos frequently. A significant limitation of many existing methods is their inability to leverage the complex relational structure of educational data effectively. They often process feature vectors in isolation, ignoring the network effects where a learner's profile can be informed by the courses they take and the behaviors of similar learners. Our work directly addresses this by employing a GNN-based approach that intrinsically models these relational structures, offering a more powerful and integrated solution for profile completion and enrichment.

2.3. MOOCCube and MOOCCubeX

MOOCCube is a data repository that integrates courses,concepts, student behaviors, relationships, and externalresources. A data cell of MOOCCube is in terms of concepts,courses, and students, which represents a learning fact, a student s learns concept k in course c. Through different queries, MOOCCube can provide various combinations of these data cells to support existing research [38]. MOOCCube contains over 700 MOOC courses, 38 K videos, 200 K students, and 100 K concepts with 300 K relation instances, which provide sufficient resources for models that require large-scale data.

The pursuit of tailored educational experiences for individual learners has been a longstanding objective for educators across generations. Adaptive learning, also referred to as adaptive teaching, aims to fulfill this goal by delivering personalized learning experiences that cater to the unique needs of each learner. This is achieved by delivering just-in-time feedback, customized pathways, and tailored resources, thereby departing from the traditional one-size-fits-all approach.

However, despite the growing interest in adaptive learning, researchers face significant challenges due to limitations in existing educational datasets. These challenges include insufficient data coverage, coarse concept granularity, and limited data curation. These constraints hinder systematic research on adaptive learning and necessitate innovative solutions.

In response to these challenges, researchers, licensed by XuetangX (https://www.xuetangx.com), one of China's largest MOOC websites, have developed an extensive solution known as MOOCCubeX [39]. Spearheaded by Yu et al. in 2021 [39], MOOCCubeX stands as a pioneering, knowledge-centered repository. It comprises 4216 courses, 230,263 videos, 358,265 exercises, 637,572 fine-grained concepts, and a wealth of behavioral data from 3,330,294 students, totaling over 296 million data points.

The primary aim of MOOCCubeX is to support research endeavors related to adaptive learning in MOOCs. Through a meticulous curation process, Yu et al. [39]. leveraged their partnership with XuetangX to acquire abundant and diverse course resources along with comprehensive student behavioral data. Notably, the repository's emphasis on fine-grained concepts allowed for a knowledge-centric organization of data, further augmented by the integration of external learning resources from the web. The scale, richness, and public availability of MOOCCubeX make it an ideal testbed for our research. Its heterogeneous nature—containing multiple entity and relation types—allows us to construct a comprehensive graph that fully leverages the RA-HGN's capacity to model complex educational interactions for the task of learner profile completion. Meanwhile, we incorporate control experiments on the MOOCCube dataset to further demonstrate the stability and transferability of the model we proposed.

3. Model Design

In this section, we will introduce the overall architecture of the relation-aware heterogeneous graph network. As depicted in Figure 2, the core of this model consists of three components: neighborhood information propagation, multi-relation attention, and target state updating.

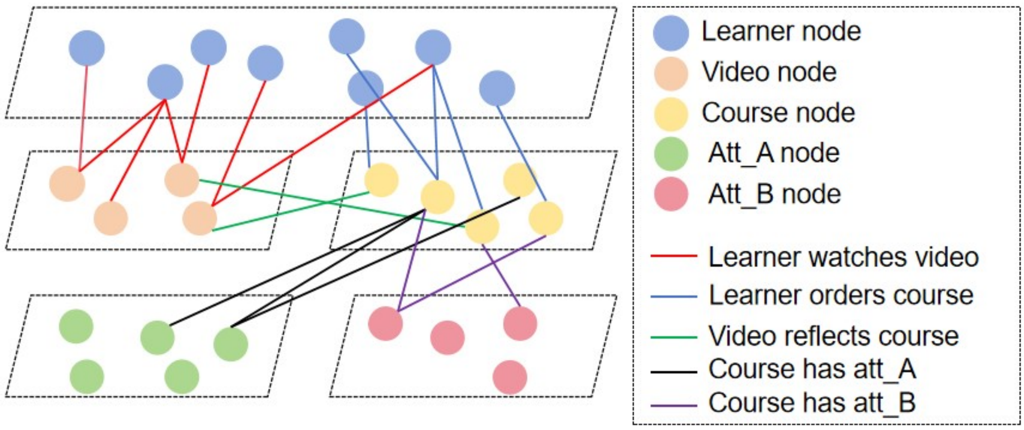

Figure 2. The heterogenerous graph with multiple types of entities and relations for learner profiling.

3.1. Input Layer

After obtaining a collection of user behaviors and attributes, the purpose of learner profile generation is to predict their labels, which in this study are age and gender. This study presents these entities and relations as a directed heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R})$, where nodes $v \in \mathcal{V}$ and edges $e \in \mathcal{E}$ are mapped to their types via functions $\tau(v) : \mathcal{V} \rightarrow \mathcal{A}$ and $\phi(e) : \mathcal{E} \rightarrow \mathcal{R}$ respectively. In the MOOCCubeX dataset, $\tau(v)$ typically includes types such as "Learner", "Video", "Course", and "Attribute", while $\phi(e)$ includes types such as "watch", "order", and "has attribute", as depicted in the Figure 2. Assuming an edge e connects a source node s and a target node t, its meta relation is represented as $\langle \tau(s), \phi(e), \tau(t) \rangle$. Meta relation generally reflect different interaction intents between entities.

3.2. Neighbourhood Message Passing

To capture the essence of user representations H, it's crucial to gather their neighborhood messages, which encompass the items they've engaged with. These interactions often reveal a user's interests and priorities. Similarly, items benefit from learning through interactions with their neighboring users and attributes. Formally, considering a triplet consisting of a target node, its neighbor node $s \in N(t)$, and an edge relation $e = (s, t)$, the message passed in the l-th layer is computed as follows:

$$\begin{aligned} M(s,e,t) &= \underset{i\in[1,h]}{||} M - \text{head}^i(s,e,t) \\ M - \text{head}^i(s,e,t) &= F_M^i(H^l[s])W_{\phi(e)}^{\text{MSG}} \end{aligned} \tag{1}$$

Here , $||$ denotes concatenation, F^iM represents the i-th multihead linear function, h is the number of heads, and $W_{\phi(e)}^{\text{MSG}}$ is a matrix projecting messages into a relation-dependent space. Employing distinct $W_{\phi(e)}^{\text{MSG}}$ for each meta relation $\langle \tau(s), \phi(e), \tau(t) \rangle$ allows our model to discern different intents across various interaction types, such as orders and watches, thereby enhancing its ability to capture diverse user-course interactions.

3.3. Multi-Relation Attention

Numerous studies across various domains have indicated that not all neighboring information is essential for target nodes. For instance, a model may need to focus more on which courses users have enrolled in rather than which advertisements they have viewed [40]. Therefore, this study employs an attention mechanism to reevaluate the significance of each piece of information. Formally, we project the source and target nodes into Key and Query vectors, respectively. The similarity between them is then computed using the following formula:

$$\begin{aligned} \alpha(s,e,t) &= \underset{\forall s\in N(t)}{\text{softmax}}\Big(\underset{i\in[1,h]}{||} \alpha - \text{head}^i(s,e,t)\Big) \\ \alpha - \text{head}^i(s,e,t) &= \left(K^i(s)W_{\phi(e)}^{\text{ATT}}Q^i(t)^T\right) \cdot \frac{1}{\sqrt{d}} \\ K^i(s) &= F_K^i(H^l[s]) \\ Q^i(t) &= F_Q^i(H^l[t]) \end{aligned} \tag{2}$$

In the formulas, F_K^i and F_Q^i respectively represent the i-th multi-head linear functions for the Key and Query vectors. $W_{\phi(e)}^{ATT}$ denotes a projection matrix, where d is the dimension of the vectors. This architecture is akin to the Transformer [41], and $\sqrt{d}$ is utilized to smooth the dot product of Key vectors and Query vectors. However, traditional Transformers use the same set of parameters to compute the dot product for all inputs, thereby neglecting the impact of various relations. Additional weights $W_{\phi(e)}^{ATT}$ can assist the model in dynamically reallocating attention based different elemental relations.

3.4. Target State Updating

After propagating messages and their attentions to the target node, this study aggregates them to update the embedding of the target node. The updated representation is defined as follows:

$$\begin{aligned} H^{l+1}[t] &= F_{map}(\sigma(\hat{H}^l[t])) + H^l[t] \\ \hat{H}^l[t] &= \underset{\forall s\in N(t)}{\bigoplus} (\alpha(s,e,t) \cdot M(s,e,t)) \end{aligned} \tag{3}$$

where $Fmap$ is a linear function that maps information back to the target feature distribution, and σ denotes the activation function. The attention $\alpha(s,e,t)$, normalized through a softmax operation ($\sum_{\forall s\in N(t)} \alpha(s,e,t) = 1_{h\times 1}$), can be directly applied to the information $M(s,e,t)$ without affecting its distribution. Through stacked layers and residual connections, each node can incorporate information from up to L layers of neighbors. For example, a student can receive messages from other students with similar interests and behaviors, even if they are not directly connected.

3.5. Output Layer

The final step involves classifying the user representations in the last layer into profile labels. To accomplish this, we employ a single linear classifier and optimize the model using cross-entropy as follows:

$$\begin{gathered}\mathcal{L} = -\sum_{t \in 'V'} \sum_{p=1}^{P} Y_{tp} \log\left(Z_{tp}\right) \\ Z = \mathrm{softmax}(F_{out}(H^L[t]))\end{gathered} \tag{4}$$

where $'V'$ denotes the labelled user node set, P denotes the total number of profile labels, and Y denotes the ground truth.

4. Experiments

4.1. Dataset and Preprocessing

We utilized data from XuetangX, one of the largest MOOC platforms in China, including student information, course enrollment data, and course details. After initial preprocessing, certain outliers were removed to mitigate potential impacts on algorithm accuracy. Rizvi et al. (2019) [42] recently discovered that using just demographic indicators such as age, gender, region and SES are strongly linked with academic performance in online-based learning, so gender and age were used as labels for user profiles in this study. To mitigate potential biases in the analysis resulting from the substantial number of users with incomplete age data in the original dataset, student entries with missing age information were excluded to ensure experimental accuracy. Regarding course information processing, this study selected course ID, course title, field of study, and prerequisite courses as key course attributes. Student enrollment data was stored in list format in the original dataset; thus, this study established a one-to-one correspondence between each student's ID and their enrolled courses.

We use the FastText LID 2.0 tool to generate word vectors from Chinese natural language data and pass them to the neural network. FastText [43], an open-source Python library by Microsoft, operates on standard, general-purpose hardware and is primarily used for word vector computation and text classification [44]. It provides a simple and efficient approach to text classification and representation learning. The FastText model consists of three layers: input layer, hidden layer, and output layer (Hierarchical Softmax). The inputs are multiple word vectors representing words, and the outputs are specific targets. The hidden layer involves the average aggregation of multiple word vectors.

In NLP tasks [45], natural language must be mathematically formalized for machine processing. Vectors abstract elements from the natural world to facilitate this. Vectors serve as abstractions employed by humans to facilitate machine processing of elements from the natural world. Word vectors represent a method of mathematizing words in language, transforming a word into a vector. Before feeding words into neural network training, they need to be encoded into numerical variables. Two common encoding methods are One-Hot Representation and Distributed Representation.

4.2. Model Construction

This study utilized the Python third-party library, PyTorch 2.6.0 [46], for constructing neural network models. As illustrated in Figure 3, the model consists of three segments: Neighbourhood Message Passing, Multi-Relation Attention, and Target State Update. PyTorch 2.6.0 and TensorFlow 2.19.0 [47] are commonly used Python libraries for building neural network models and are currently the most popular deep learning frameworks. They are both based on the computational graph framework, supporting automatic differentiation, enabling the construction and training of multi-layer neural networks by defining computational graphs. In PyTorch, this study employed a custom neural network model defined as a subclass of nn.Module to facilitate neural network construction. Previous works have mainly focused on single types of entities and relations, yet more information needs to be explored through different types. However, in practical scenarios, user labels are often limited. Hence, it is necessary to utilize a large amount of unlabeled data for semi-supervised learning. In this experiment, course names, course IDs, and prerequisite courses were selected as neighbor information for predicting students' elective courses. In the specific code implementation, this study incorporated the three preprocessed feature vectors as inter-neighbor information, and these were read during the initialization of the model.

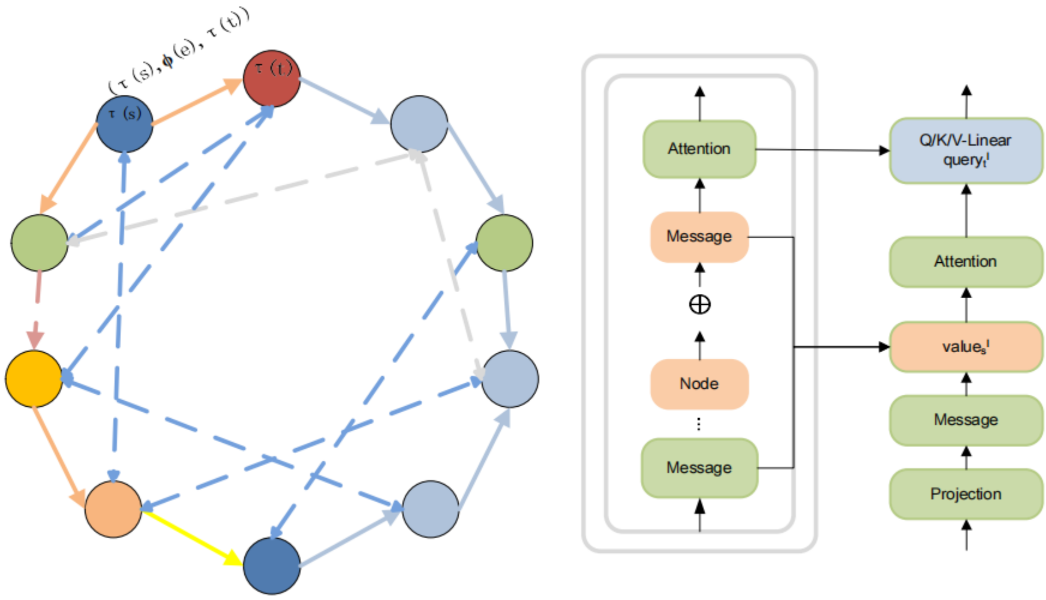

Figure 3. The overall architecture of the relation-aware heterogeneous graph network.The diagram illustrates the information flow from a source node s to a target node t via an edge e. **Node colors:**Nodes of different colors represent different entity types, and **Arrow types:** Solid arrows indicate the primary flow of *message propagation*. Dashed arrows indicate the computation and application of *relation-aware attention weights*.

Upon obtaining the feature vectors, this study employed the Embedding method provided by PyTorch to encode the feature vectors. In neural networks, it is common to encode input data, with one approach being the conversion of a word or a character into a vector, known as word embedding. nn. Embedding is a function used to implement such vector conversion. Given a one-hot vector, nn.Embedding transforms it into the corresponding word vector, also referred to as an embedding vector.

This study employed the Attention Mechanism provided by PyTorch to implement the attention mechanism. Simultaneously, the model defined excitation to determine which parts of the model should be focused on within the attention mechanism. In the neural network, three core variables required for the attention mechanism were defined: key, value, and query. These variables are used to calculate attention scores in the attention mechanism, employing dot product operations to compute weights. This process aids in selecting nodes that are most relevant to the current features, thereby modeling the input data features. The attention mechanism is represented by the formula,

$$\text{Attention}(Q, K, V) = \sum_{i}^{L} \langle Q, K_i \rangle V_i \tag{5}$$

where the target Q (query) from this study is dot-multiplied with other K(key) elements to calculate attention scores. These scores are then used to weight the corresponding V (value) for information aggregation.

The excitation in this experiment is a Sequential container comprised of multiple convolutional layers, batch normalization layers, and ReLU activation layers concatenated together, with each layer's hyperparameters passed during instantiation. The nn.ReLU module primarily implements the ReLU (Rectified Linear Unit) function in activation functions, compressing negative values to zero while retaining the positive portion.

4.3. Model Training and Optimization

Based on the feature engineering, this study will select appropriate machine learning algorithms for model training and prediction. The definition of neural network models and forward propagation will be implemented using the foundational framework available in PyTorch. To assess the model performance for the specific problem and select the optimal model, this study will employ Parameter Optimization Method and performance evaluation methods from the Scikit-learn library [48].

In the model definition, this study incorporates the AdamW optimizer [49] and the OneCycle learning rate scheduling strategy [50] to enhance the performance of the proposed model. The AdamW optimizer is an improved iteration of the Adam optimizer, specifically tailored for training neural network models. We selected AdamW due to its effective decoupling of the weight decay term from the gradient-based update step. In standard Adam, L2 regularization (weight decay) is entangled with the adaptive gradient scaling, which can lead to suboptimal regularization, especially when training models with large embedding tables—a common feature in graph neural networks dealing with many entity types (e.g., learners, courses, videos). AdamW addresses this by explicitly and separately applying weight decay after controlling the parameter-wise step size, leading to more effective regularization, better convergence, and often improved generalization performance [49]. This is crucial for our RA-HGN model to prevent overfitting on the sparse and heterogeneous MOOC graph data.

The OneCycle strategy [50] is a learning rate scheduling technique employed to dynamically adjust the learning rate throughout the training process (as shown in Figure 4). Grounded in a cyclic learning rate schedule, it involves a gradual increase in the learning rate during the initial phase of training, followed by a gradual decrease. The primary objective of implementing the OneCycle strategy is to expedite the training process, enhancing both the convergence speed and generalization capability of the model. These meticulously defined optimization techniques play a crucial role in the model design, contributing to the overall improvement of the research model's performance.

In each epoch, the model first enters the training mode and then iterates through the training data batches. For each batch, the model performs forward propagation, computes the loss, carries out backpropagation, and optimizes parameters. Simultaneously, it tracks and prints the training loss and accuracy. The OneCycleLR learning rate scheduler employed in this study adjusts the learning rate during the training process based on given parameters. The scheduler calculates the appropriate learning rate based on the number of training epochs and the current training step and updates the learning rate after each training step. The parameters of the scheduler include the total number of epochs, the number of steps (i.e., batch size) per epoch, and the maximum learning rate. We adopt the grid-search strategy to fnd the optimal parameter combination for our model. The parameters used in this study are as follows: 50 epochs, a batch size of 512, with the maximum learning rate defaulting to 1×10^{-2}. After the completion of training in each epoch, the model enters the evaluation mode. Firstly, it evaluates the validation set, calculating validation loss and accuracy. Then, it evaluates the test set, computing test loss and accuracy. Additionally, F1 scores for both the validation and test sets are calculated to assess the model's performance.

The experiment consists of two classification tasks: gender prediction (binary classification task) and age prediction (multi-class classification task). In this study, Accuracy and Macro-F1 are utilized to evaluate the models, which are widely employed metrics for assessing the accuracy of classification models.

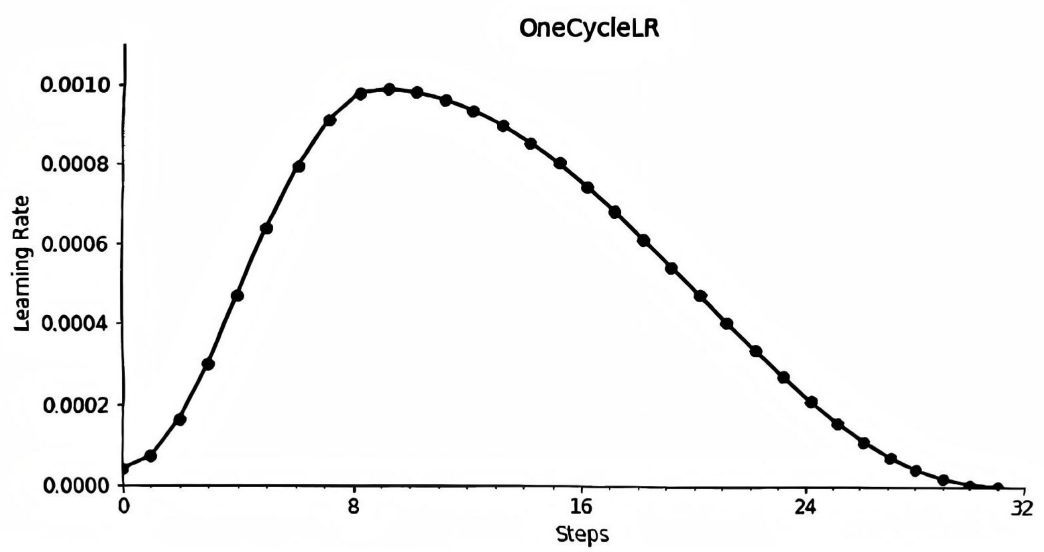

Figure 4. The OneCycle strategy is a learning rate scheduling technique that dynamically adjusts the learning rate during the training process. It enables better exploration and utilization of the training space within a shorter duration, thereby expediting the training process and enhancing model performance.

5. Results and Analysis

This study utilizes the two datasets: MOOCCube and MOOCCubeX, and then employs Relation-aware Heterogeneous graph network to learn node representations and predict gender and age from user profiles. The final model, developed after multiple rounds of training and tuning, was evaluated on a test set using metrics such as accuracy and F1-score. Please see Tables 1 and 2.

Table 1. Results on MOOCCubeX.

	Test Acc	Val Acc	Test F1	Val F1	Best Test	Best Val
gender	75.55%	75.51%	61.92%	61.94%	75.55%	75.53%
age	32.96%	32.69%	17.01%	16.80%	32.96%	32.69%

Table 2. Results on MOOCCube.

	Test Acc	Val Acc	Test F1	Val F1	Best Test	Best Val
gender	80.13%	80.29%	67.38%	67.69%	80.13%	80.29%
age	41.25%	42.05%	28.17%	29.01%	41.25%	42.05%

In this study, the performance on MOOCCubeX is relatively lower than that on MOOCCube dataset. Even on MOOCCubeX dataset, the performance of the model in gender prediction demonstrates strong results, with a test accuracy of 75.55% and a validation accuracy of 75.51%. The corresponding F1 scores are also notable, reaching 61.92% for the test set and 61.94% for the validation set. These findings indicate the model's capability to effectively learn from user features and make accurate gender predictions. However, age prediction performance is comparatively weaker, It achieves a test accuracy of 32.96% and a validation accuracy of 32.69%, along with lower F1 scores of 17.01% and 16.80% for the test and validation sets, respectively. This suggests that age prediction may be influenced by more complex factors, posing challenges for the model's predictive ability. Further refinement and enhancement of the model may be necessary to improve its performance in age prediction.

6. Ablation Study

In our ablation experiments, we aimed to assess the robustness of the model to variations in data quantity (see Table 3). Initially, our model exhibited satisfactory accuracy in gender prediction, reaching 75.55%, while achieving 32.96% accuracy in age prediction. Subsequently, we conducted three ablation experiments, retaining 70%, 50%, and 30% of the original data, respectively. Under these experimental settings, as shown in Table, we observed that the model maintained relatively stable performance in gender prediction, with accuracies of 74.49%, 73.37%, and 72.14%, respectively. Despite the reduction in data volume, the accuracy of gender prediction remained at a high level. However, there was a slight decline in accuracy in age prediction, with accuracies of 31.69%, 31.23%, and 30.59% respectively, indicating that the model's performance in age prediction was influenced by changes in data volume.

Table 3. Ablation study: Model performance under varying data volumes.

Data Volume	Gender Accuracy	Gender F1-Score	Age Accuracy	Age F1-Score
100% (Original)	75.55%	61.92%	32.96%	17.01%
70%	74.49%	60.87%	31.69%	16.45%
50%	73.37%	59.68%	31.23%	16.12%
30%	72.14%	58.21%	30.59%	15.73%

Overall, despite the decrease in data volume, the model retained a certain level of predictive capability, further validating its robustness and applicability across different data scales.

7. Discussion

7.1. Interpretation of Results and Error Analysis

The significant disparity between gender and age prediction performance warrants discussion. Our error analysis reveals two primary factors for the age prediction challenge:

- Semantic Similarity of Adjacent Groups: The model's confusion matrix shows highest error rates between adjacent age groups. This indicates that the behavioral signals captured from MOOC interactions are not sufficiently discriminative to sharply delineate these age brackets, suggesting a fundamental limitation of the behavioral data source for precise age inference, as opposed to model architecture.
- Data Sparsity and Label Imbalance: Age-labeled data is sparser and more unevenly distributed than gender data. Learners with very sparse interaction histories provided insufficient contextual signals for reliable age inference. We considered treating age prediction as an ordinal regression task. However, preliminary review of related work in demographic inference from behavioral data [3,17] indicates that age prediction accuracy often remains relatively low even with advanced models, reflecting the inherent noisiness and weak correlation of digital footprints with chronological age. Therefore, while ordinal regression is a valid future direction, we posit that a major leap in accuracy would likely require incorporating different data modalities rather

than a different output layer formulation alone. The primary strength of our current RA-HGN model is validated by its robust and stable performance on gender prediction and its relational reasoning capability.

7.2. Personalized Intervention Framework

To demonstrate how our learner profiling model can translate predictions into actionable educational interventions, we present a decision function that leverages the predicted demographic attributes along with behavioral data to recommend personalized learning strategies.

This decision function (Algorithm 1) illustrates how even basic demographic predictions can inform personalized learning strategies. By combining demographic insights with behavioral metrics, educators and platform designers can create more targeted interventions that account for both who the learner is and how they engage with the material.

Algorithm 1 Demographic-Aware Learning Intervention Decision

Require: Predicted gender g, predicted age a, behavioral metrics $B = \{v_{\text{watch}}, c_{\text{complete}}, f_{\text{forum}}\}$ (video watch time, course completion rate, forum activity)
Ensure: Recommended intervention set I
Initialize $I \leftarrow \emptyset$
Step 1: Age-based content adaptation
if $a < 20$ **then**
  $I \leftarrow I \cup \{$More interactive content, gamified elements$\}$
else if $20 \leq a < 40$ **then**
  $I \leftarrow I \cup \{$Balanced theory-practice mix, career-relevant examples$\}$
else
  $I \leftarrow I \cup \{$Self-paced learning, practical applications$\}$
end if
Step 2: Gender-aware communication style
if $g =$ female **then**
  $I \leftarrow I \cup \{$Encourage collaborative learning, provide detailed feedback$\}$
else
  $I \leftarrow I \cup \{$Competitive elements, concise instructions$\}$
end if
Step 3: Behavioral reinforcement
if $v_{\text{watch}} < \text{threshold}_1$ **then**
  $I \leftarrow I \cup \{$Send video completion reminders, highlight key video segments$\}$
end if
if $c_{\text{complete}} < \text{threshold}_2$ **then**
  $I \leftarrow I \cup \{$Suggest prerequisite review, break course into smaller units$\}$
end if
if $f_{\text{forum}} = 0$ **then**
  $I \leftarrow I \cup \{$Invite to introductory forum discussions$\}$
end if
return I

7.3. Limitations and Future Work

We openly acknowledge several limitations of this study, which also chart the course for our future work.

Beyond Demographics;Predicting Pedagogically Relevant Attributes: We fully concur with that the ultimate goal of learner profiling is to support learning. Predicting demographics is a foundational step that serves two purposes: (1) it allows for the privacy-preserving enrichment of datasets where this information is missing, and (2) it validates our graph-based framework on a well-defined task. We clarify that while these demographic attributes are not direct measures of learning capability, they have been shown to correlate with learning behaviors and preferences [42]. The true potential of our model lies in its extensibility. The same RA-HGN architecture can be directly applied to predict more directly pedagogically relevant attributes, such as learning styles, risk of dropout, or knowledge mastery levels. This is a primary direction for our immediate future research.

Model Interpretability: While the attention mechanism provides some insight into which neighbors are important, a more thorough analysis is needed to understand why. Future work will incorporate model interpretation techniques to uncover the specific behavioral patterns that drive the predictions, making the model more transparent and actionable for educators.

Exploring a Multiplexed RAMHN extension: The current RA-HGN model processes different relation types independently. Future work could explore a Multiplexed RAMHN extension that explicitly models interactions between multiple relation types occurring between the same entity pairs. A multiplexed approach could capture synergistic effects between coexisting relations, potentially improving prediction accuracy for complex attributes.

Granularity based contrastive learning: Another promising direction for improving our learner profiling framework is the integration of contrastive learning techniques that operate at multiple granularity levels. Current graph neural networks, including our RA-HGN, often learn node representations at a single, fixed granularity, which may not capture the hierarchical nature of learning behaviors. We may extending our model with a granularity-aware contrastive learning. Such granularity-aware contrastive learning aligns with recent advances in self-supervised graph representation learning and could make our model more robust to sparse or noisy educational data.

8. Conclusions

This study utilized the MOOCCube and MOOCCubeX dataset to analyze user demographics, specifically gender and age, in Massive Open Online Courses (MOOCs) through a relation-aware heterogeneous graph network. By constructing a robust heterogeneous graph model and integrating user behavioral data, course attributes, and social network information, the proposed approach achieved accurate predictions of user gender and age. Furthermore, ablation experiments were conducted, confirming the model's stability and consistent performance across varying data volumes. This indicates robustness to data sparsity.

The use of a relation-aware heterogeneous graph network, coupled with diverse data integration, proved instrumental in predicting demographics and facilitating comprehensive learner profile development. Beyond its immediate application, this work makes a broader contribution by providing a flexible and powerful graph-based framework for learner modeling. The methodology established here is not limited to demographics but serves as a foundational step towards the automated inference of a wide range of complex learning attributes. By bridging a critical data gap, this research contributes to the development of more complete, dynamic, and actionable learner profiles, ultimately supporting the advancement of personalized and adaptive online education.

Author Contributions: Conceptualization, B.J.; Methodology, B.J.; Software, X.C.; Validation, X.C. and Z.S.; Formal analysis, B.J.; Investigation, N.Y.; Resources, N.Y.; Data curation, M.L. and Y.L.; Writing—original draft, X.C. and M.L.; Writing—review & editing, Z.S.; Visualization, Y.L.; Supervision, N.Y.; Project administration, N.Y.; Funding acquisition, B.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant No. 61907025).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://github.com/THU-KEG/MOOCCubeX (accessed on 24 February 2025).

Acknowledgments: The authors would like to thank all the anonymous reviewers for their valuable suggestions to improve this work. We also thank Fangjie Sun for his valuable suggestions on processing the original dataset.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Xiong, Z.; Li, H.; Liu, Z.; Chen, Z.; Zhou, H.; Rong, W.; Ouyang, Y. A Review of Data Mining in Personalized Education: Current Trends and Future Prospects. *arXiv* **2024**, arXiv:2402.17236. [CrossRef]
2. Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; Yu, P.S. Heterogeneous graph attention network. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 2022–2032.
3. Wu, C.; Wu, F.; Liu, J.; He, S.; Huang, Y.; Xie, X. Neural demographic prediction using search query. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, Melbourne, Australia, 11–15 February 2019; pp. 654–662.
4. Chen, R.; Jankovic, F.; Marinsek, N.; Foschini, L.; Kourtis, L.; Signorini, A.; Pugh, M.; Shen, J.; Yaari, R.; Maljkovic, V. Developing measures of cognitive impairment in the real world from consumer-grade multimodal sensor streams. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD'19), Anchorage, AK, USA, 4–8 August 2019.
5. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The Graph Neural Network Model. *IEEE Trans. Neural Netw.* **2009**, *20*, 61–80. [CrossRef]
6. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
7. Werbos, P.J. Generalization of backpropagation with application to a recurrent gas market model. *Neural Netw.* **1988**, *1*, 339–356. [CrossRef]
8. Qiu, J.; Tang, J.; Ma, H.; Dong, Y.; Wang, K.; Tang, J. Deepinf: Social influence prediction with deep learning. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 2110–2119.
9. Ko, H.; Lee, S.; Park, Y.; Choi, A. A survey of recommendation systems: Recommendation models, techniques, and application fields. *Electronics* **2022**, *11*, 141. [CrossRef]
10. Raad, E.; Chbeir, R.; Dipanda, A. User profile matching in social networks. In Proceedings of the 2010 13th International Conference on Network-Based Information Systems, Gifu, Japan, 14–16 September 2010; IEEE: New York, NY, USA, 2010; pp. 297–304.
11. Zhang, J.; Zhu, Y.; Liu, Q.; Wu, S.; Wang, S.; Wang, L. Mining latent structures for multimedia recommendation. In Proceedings of the 29th ACM International Conference on Multimedia, Chengdu, China, 20–24 October 2021; pp. 3872–3880.
12. Liu, C.; Li, X.; Cai, G.; Dong, Z.; Zhu, H.; Shang, L. Noninvasive self-attention for side information fusion in sequential recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; Volume 35, pp. 4249–4256.
13. Fei, Z.; Guo, J.; Gong, H.; Ye, L.; Attahi, E.; Huang, B. A GNN Architecture with Local and Global-Attention Feature for Image Classification. *IEEE Access* **2023**, *11*, 110221–110233. [CrossRef]
14. Wu, S.; Tang, Y.; Zhu, Y.; Wang, L.; Xie, X.; Tan, T. Session-based recommendation with graph neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January 27–1 February 2019; Volume 33, pp. 346–353.
15. Yu, F.; Zhu, Y.; Liu, Q.; Wu, S.; Wang, L.; Tan, T. TAGNN: Target Attentive Graph Neural Networks for Session-based Recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, New York, NY, USA, 25–30 July 2020; ACM: New York, NY, USA, 2020. [CrossRef]
16. Yang, C.; Gao, W.; Wu, D.; Wang, C. Learning to simulate unseen physical systems with graph neural networks. *arXiv* **2022**, arXiv:2201.11976. [CrossRef]
17. Preoţiuc-Pietro, D.; Ungar, L. User-Level Race and Ethnicity Predictors from Twitter Text. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; Bender, E.M., Derczynski, L., Isabelle, P., Eds.; pp. 1534–1545.
18. Li, Y.; Tarlow, D.; Brockschmidt, M.; Zemel, R. Gated graph sequence neural networks. *arXiv* **2015**, arXiv:1511.05493.
19. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.

20. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; Van Den Berg, R.; Titov, I.; Welling, M. Modeling relational data with graph convolutional networks. In Proceedings of the Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, 3–7 June 2018; Proceedings 15; Springer: Berlin/Heidelberg, Germany, 2018; pp. 593–607.
21. Rahimi, A.; Cohn, T.; Baldwin, T. Semi-supervised user geolocation via graph convolutional networks. *arXiv* **2018**, arXiv:1804.08049. [CrossRef]
22. Zhang, M.; Wu, S.; Gao, M.; Jiang, X.; Xu, K.; Wang, L. Personalized Graph Neural Networks With Attention Mechanism for Session-Aware Recommendation. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 3946–3957. [CrossRef]
23. Yan, Q.; Zhang, Y.; Liu, Q.; Wu, S.; Wang, L. Relation-aware Heterogeneous Graph for User Profiling. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM '21), Virtual Event, 1–5 November 2021; ACM: New York, NY, USA, 2021; pp. 3573–3577. [CrossRef]
24. Premlatha, K.; Dharani, B.; Geetha, T. Dynamic learner profiling and automatic learner classification for adaptive e-learning environment. *Interact. Learn. Environ.* **2016**, *24*, 1054–1075. [CrossRef]
25. Maseleno, A.; Sabani, N.; Huda, M.; Ahmad, R.; Jasmi, K.A.; Basiron, B. Demystifying Learning Analytics in Personalised Learning. *Int. J. Eng. Technol.* **2018**, *7*, 1124–1129. [CrossRef]
26. Kavitha, M.G.; Raj, D.L. Educational data mining and learning analytics-educational assistance for teaching and learning. *arXiv* **2017**, arXiv:1706.03327.
27. Magidson, J.; Vermunt, J.; Madura, J., Latent class analysis. In *SAGE Research Methods Foundations*; Atkinson, P., Delamont, S., Cernat, A., Sakshaug, J., Williams, R., Eds.; Sage Publications: Thousand Oaks, CA, USA, 2020. [CrossRef]
28. McCutcheon, A.L. *Latent Class Analysis*; Sage Publications: Thousand Oaks, CA, USA, 1987; p. 64,
29. Oberski, D. Mixture Models: Latent Profile and Latent Class Analysis. In *Modern Statistical Methods for HCI*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 275–287. [CrossRef]
30. Kang, I.G. Heterogeneity of Learners' Behavioral Patterns of Watching Videos and Completing Assessments in Massive Open Online Courses (MOOCs): A Latent Class Analysis. *Int. Rev. Res. Open Distrib. Learn.* **2020**, *21*, 222–237. [CrossRef]
31. Kim, D.; Lee, I.H.; Park, J.H. Latent class analysis of non-formal learners' self-directed learning patterns in open educational resource repositories. *Br. J. Educ. Technol.* **2019**, *50*, 3420–3436. . [CrossRef]
32. Watson, S.L.; Watson, W.R.; Yu, J.H.; Alamri, H.; Mueller, C. Learner profiles of attitudinal learning in a MOOC: An explanatory sequential mixed methods study. *Comput. Educ.* **2017**, *114*, 274–285. [CrossRef]
33. Tempelaar, D.T.; Rienties, B.; Giesbers, B. In search for the most informative data for feedback generation: Learning analytics in a data-rich context. *Comput. Hum. Behav.* **2015**, *47*, 157–167. [CrossRef]
34. Zamecnik, A.; Kovanović, V.; Joksimović, S.; Liu, L. Exploring non-traditional learner motivations and characteristics in online learning: A learner profile study. *Comput. Educ. Artif. Intell.* **2022**, *3*, 100051. [CrossRef]
35. Koopman, M.; Beijaard, D. Nuances and details in student approaches to learning in innovative secondary education using profiles. *Int. J. Educ. Res. Open* **2024**, *6*, 100316. [CrossRef]
36. Meghji, A.F.; Shaikh, F.B.; Wadho, S.A.; Bhatti, S.; Ayyasamy, R.K. Using educational data mining to predict student academic performance. *Vfast Trans. Softw. Eng.* **2023**, *11*, 43–49. [CrossRef]
37. Liao, C.H.; Wu, J.Y. Learning analytics on video-viewing engagement in a flipped statistics course: Relating external video-viewing patterns to internal motivational dynamics and performance. *Comput. Educ.* **2023**, *197*, 104754. [CrossRef]
38. Yu, J.; Luo, G.; Xiao, T.; Zhong, Q.; Wang, Y.; Feng, W.; Luo, J.; Wang, C.; Hou, L.; Li, J.; et al. MOOCCube: A Large-scale Data Repository for NLP Applications in MOOCs. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 10 July 2020; pp. 3135–3142. [CrossRef]
39. Yu, J.; Wang, Y.; Zhong, Q.; Luo, G.; Mao, Y.; Sun, K.; Feng, W.; Xu, W.; Cao, S.; Zeng, K.; et al. MOOCCubeX: A Large Knowledge-centered Repository for Adaptive Learning in MOOCs. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM '21), Virtual Event, 1–5 November 2021; ACM: New York, NY, USA, 2021; pp. 4643–4652. [CrossRef]
40. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), Long Beach, CA, USA, 4–9 December 2017; pp.6000–6010.
41. Hu, Z.; Dong, Y.; Wang, K.; Sun, Y. Heterogeneous graph transformer. In Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 2704–2710.
42. Rizvi, S.; Rienties, B.; Khoja, S.A. The role of demographics in online learning; A decision tree based approach. *Comput. Educ.* **2019**, *137*, 32–47. [CrossRef]
43. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of tricks for efficient text classification. *arXiv* **2016**, arXiv:1607.01759. [CrossRef]
44. Zhang, Y.; Yu, X.; Cui, Z.; Wu, S.; Wen, Z.; Wang, L. Every Document Owns Its Structure: Inductive Text Classification via Graph Neural Networks. *arXiv* **2020**, arXiv:2004.13826. [CrossRef]

45. Chowdhary, K.; Chowdhary, K. Natural language processing. In *Fundamentals of Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 603–649.
46. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 8026–8037.
47. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
48. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
49. Llugsi, R.; El Yacoubi, S.; Fontaine, A.; Lupera, P. Comparison between Adam, AdaMax and Adam W optimizers to implement a Weather Forecast based on Neural Networks for the Andean city of Quito. In Proceedings of the 2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM), Cuenca, Ecuador, 12–15 October 2021; IEEE: New York, NY, USA, 2021; pp. 1–6.
50. Smith, L.N.; Topin, N. Super-convergence: Very fast training of neural networks using large learning rates. In Proceedings of the Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, Baltimore, MD, USA, 15–17 April 2019; SPIE: Washington, DC, USA, 2019; Volume 11006, pp. 369–386.