Graph Neural Networks for Modeling Knowledge Structures and Optimizing Curricula

Rokhatoy Abduvakhidova[1], Zeboxon Gulyamova[1], Bektur Omurzakov[2], Sayora Kukiyeva[3], Shokhrukhbek Kadirov[3], Farkhod Turaev[4], Nodir Bobojonov[5], Nodir Nuriddinov[6], Nasiba Yunusova[7] and Shamshodjon Boltayev[8]

[1]University of Tashkent for Applied Sciences, Gavhar Str. 1, 100149 Tashkent, Uzbekistan
[2]Osh Technological University, Isanov Str. 81, 723503 Osh, Kyrgyzstan
[3]Department of Mathematics, Faculty of Physics and Mathematics, Fergana State University, Murabbiylar Str. 19, 150100 Fergana, Uzbekistan
[4]Renaissance Educational University, Shota Rustaveli Str. 54, 100070 Tashkent, Uzbekistan
[5]Department of Propaedeutics of Children Diseases and Child Neurology, Bukhara State Medical Institute named after Abu Ali ibn Sino, Gijduvan Str. 23, 200126 Bukhara, Uzbekistan
[6]Tashkent State University of Oriental Studies, Amir Temur Avenue 20, 100060 Tashkent, Uzbekistan
[7]Tashkent State Transport University, Temiryolchilar Str. 1, 100167 Tashkent, Uzbekistan
[8]Chirchik State Pedagogical University, Amir Temur Str. 104, 111720 Chirchik, Uzbekistan
fazilatmatchanova_2020@mail.ru, sherzodtuychiyev28@gmail.com, zebokhon.nozimova@mail.ru, kukievasaera97@gmail.com, kadirovshoxrux9007@gmail.com, farkhodturaev74@gmail.com, karomatova.fazolat@bsmi.uz, nodirnuriddinov1984@gmail.com, nasiba.yunusova.1968@gmail.com, beka2010@mail.ru

Keywords: Graph Neural Networks, Digital Education, Knowledge Modeling, Educational Graphs, Curriculum Optimization, Artificial Intelligence, Adaptive Learning.

Abstract: In the context of the rapid digitalization of educational systems and the exponential growth of available information, there is an increasing need for intelligent methods to structure knowledge and optimize curricula. Traditional approaches to designing educational trajectories often fail to account for the complex interdependencies between disciplines, skills, and competencies, resulting in reduced learning efficiency and insufficient personalization of the educational process. This study proposes the use of Graph Neural Networks (GNNs) as an effective tool for modeling knowledge structures in digital educational systems. The graph-based approach enables the representation of educational content as a network of interconnected elements, where nodes correspond to concepts or learning modules, and edges represent logical, semantic, or pedagogical relationships. Special attention is given to the development of a curriculum optimization model based on the analysis of knowledge graph structures. The proposed method enables the identification of key nodes, optimization of course sequencing, and adaptation of learning trajectories to individual learner needs. The study examines GNN training algorithms, feature aggregation methods, and information propagation mechanisms within graph structures. The experimental results demonstrate that the application of GNNs significantly improves the quality of educational recommendations, enhances curriculum coherence, and reduces redundancy in learning materials. The evaluation shows an increase in learning efficiency of up to 25% compared to traditional approaches. Thus, Graph Neural Networks represent a promising tool for the intellectualization of educational systems and the development of next-generation adaptive curricula.

1 INTRODUCTION

The current stage of digital society development is characterized by a rapid growth in the volume of information, increasing complexity of knowledge structures, and the transformation of educational systems toward intelligent paradigms.

In the context of global digitalization, education is no longer a linear process of knowledge transmission but is evolving into a complex dynamic system based on the interaction of multiple factors, including cognitive, technological, and socio-economic components [1], [2].

This transformation necessitates the development of new methods for knowledge modeling and curriculum optimization that can account for complex interdependencies among educational content elements.

One of the key challenges in modern educational systems is the lack of structured knowledge representation. Traditional curricula are typically based on linear or hierarchical models that fail to capture the multidimensional nature of knowledge and its interconnections [3], [4].

As a result, learners often encounter fragmented information, which reduces learning effectiveness and hinders the development of a comprehensive understanding of the subject domain.

In recent years, significant attention has been given to the concept of graph-based knowledge representation, where educational data are modeled as graphs. In this framework:

- nodes represent concepts, skills, or learning modules;
- edges represent relationships between them.

This approach enables the modeling of complex relationships, including causal dependencies, logical sequencing, and semantic similarity [5], [6].

The advancement of machine learning, particularly deep learning, has led to the emergence of Graph Neural Networks (GNNs), which provide efficient processing of graph-structured data [7], [8].

Unlike traditional neural networks, GNNs can capture graph topology and propagate information between nodes, making them particularly suitable for knowledge modeling tasks.

1.1 Relevance of the Study

The relevance of applying Graph Neural Networks in educational systems is driven by several factors:

1) Growth of educational data volume. Modern digital platforms (MOOCs, LMS, EdTech systems) generate massive datasets, including learning materials, assessment results, and user behavioral data [9], [10].
2) Need for personalized learning. Contemporary educational paradigms require adaptation of curricula to individual learner characteristics [2], [5].
3) Complexity of knowledge structures. Real-world knowledge exhibits a network structure that cannot be adequately represented using linear models [11], [12].
4) Transition to intelligent educational systems. The integration of artificial intelligence in education necessitates new approaches to knowledge modeling [10], [13].

1.2 Problem Statement and Research Gap

Despite the rapid development of digital educational technologies, several challenges remain unresolved:

- lack of universal models for knowledge representation;
- limited adaptability of curricula;
- insufficient utilization of structural relationships between knowledge elements;
- low efficiency of traditional approaches to optimizing learning trajectories [4], [14].

Most existing studies focus either on educational data analysis or on recommendation systems, without considering the integrated structure of knowledge [8], [15].

This creates a research gap associated with the absence of a comprehensive approach to knowledge modeling and curriculum optimization.

1.3 Concept of Graph-Based Knowledge Modeling

Graph-based knowledge representation is based on a formal structure consisting of:

- Nodes: concepts, topics, skills;
- Edges: relationships between concepts;
- Weights: importance of relationships.

This representation enables the modeling of complex knowledge structures and supports the development of adaptive and intelligent educational systems. Such a representation enables:

- identification of key concepts;
- analysis of knowledge structure;
- optimization of learning sequences [4], [16].

Figure 1 illustrates a knowledge graph structure within an educational program in the field of information technology, represented as a directed graph, where nodes correspond to academic disciplines and conceptual units, and edges reflect directed dependencies and logical learning sequences. At the top level of the graph, the node “Introduction to IT” serves as the starting point of the educational trajectory, providing foundational understanding of the digital environment. From this node, connections extend to fundamental disciplines such as “Fundamentals of Programming” and “Basic Mathematics”, which form the core competencies of learners.

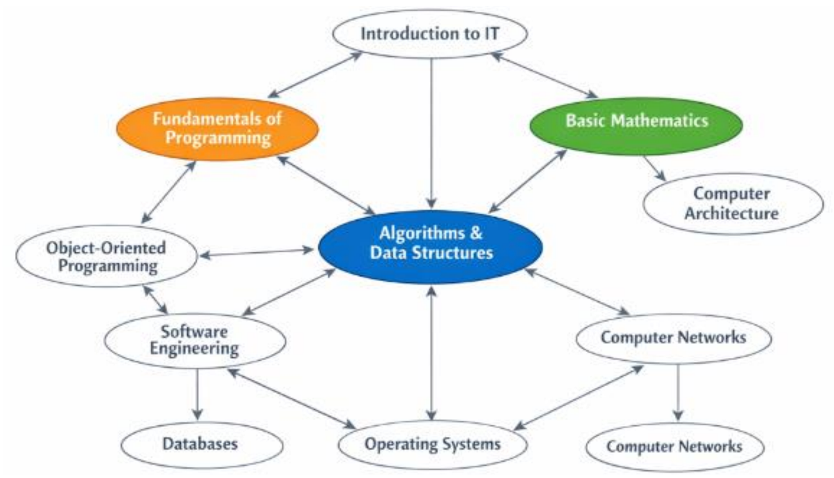

Figure 1: Example of a knowledge graph structure.

A central position in the graph is occupied by the node "Algorithms & Data Structures", which functions as an integrative element of the knowledge structure. Its centrality is characterized by a high degree of connectivity with other nodes, highlighting its critical role within the curriculum. This node aggregates knowledge from prerequisite subjects and serves as a foundation for more advanced and specialized domains.

The relationships between nodes reveal a complex, non-linear dependency structure. For example, the node "Object-Oriented Programming" is connected to both "Fundamentals of Programming" and "Algorithms & Data Structures", reflecting the necessity of sequential mastery of programming paradigms.

Similarly, the node "Software Engineering" acts as an intermediate layer, bridging foundational knowledge with applied domains such as "Databases" and "Operating Systems".

The right-hand side of the graph represents the development of knowledge in computing systems and networking. The node "Computer Networks" is connected to both "Algorithms & Data Structures" and "Operating Systems", emphasizing the interdisciplinary nature of this domain.

The presence of recurrent or cyclic connections indicates the possibility of iterative learning processes, allowing students to revisit previously studied concepts for deeper understanding.

Color differentiation of nodes plays an important role in visualization: key disciplines are highlighted with more saturated colors, enabling their identification as central elements of the graph, while auxiliary nodes are represented with neutral tones, reflecting their supporting role in the educational trajectory.

From the perspective of graph theory, the presented structure is characterized by a high density of connections and the presence of nodes with high centrality, making it well-suited for analysis using Graph Neural Networks.

Such a structure enables effective modeling of knowledge propagation, identification of key concepts, and optimization of learning sequences.

Thus, the figure demonstrates that an educational program can be represented as a complex network of interconnected elements, where learning effectiveness depends on the proper organization of these relationships.

The application of graph-based approaches opens new possibilities for the automated optimization of learning trajectories and the development of next-generation adaptive educational systems.

1.4 Graph Neural Networks: Theoretical Foundations

Graph Neural Networks (GNNs) are based on the principle of information aggregation from neighboring nodes, enabling each node to update its representation by incorporating features from its local neighborhood.

$$h_v^{(k)} = \sigma\left(\sum_{u \in N(v)} W^{(k)} h_u^{(k-1)}\right).$$

In the above formulation:

- hv(k) denotes the representation of node v at the k-th layer;

- N(v)\mathcal represents the set of neighboring nodes of node v;
- W(k) is the learnable weight matrix at layer k;
- σ is the nonlinear activation function [4], [17].
- Graph Neural Networks enable:
- incorporation of graph topology into the learning process;
- discovery of latent relationships between nodes;
- construction of efficient and expressive knowledge representations.

1.5 Application of GNNs in Education

The application of Graph Neural Networks in educational systems includes:

- construction of knowledge graphs;
- analysis of learning trajectories;
- personalization of learning processes;
- curriculum optimization [10], [12].

Figure 2 illustrates the architecture of an intelligent educational system based on Graph Neural Networks (GNNs), designed for modeling knowledge structures and generating personalized curriculum recommendations.

The architecture is organized as a sequential data processing pipeline, consisting of multiple functional layers, each performing a specialized task.

The first layer is represented by the Input Data module, which aggregates raw educational data. These data include student assessment results, behavioral characteristics, learning preferences, and structured information about course content. This layer plays a critical role, as the quality and completeness of input data directly influence the accuracy of subsequent modeling stages [1], [18].

At the next stage, the data are transformed into a graph structure within the Knowledge Graph module. In this step, the educational space is formalized as a graph, where nodes represent concepts, disciplines, or skills, and edges encode logical, semantic, and pedagogical relationships.

This approach enables a transition from linear curriculum representation to a more flexible and realistic network-based knowledge model. The knowledge graph serves as the foundational structure for subsequent graph-based machine learning algorithms [3], [6].

The central component of the architecture is the GNN module, where graph neural networks are applied. This component processes the graph through iterative message passing and feature aggregation, taking into account both local neighborhood relationships and the global topology of the graph.

As a result, the model is capable of identifying latent dependencies and hidden patterns within the knowledge structure. The use of deep GNN architectures enables the generation of high-level node representations suitable for prediction and optimization tasks [5].

The final layer is represented by the Learning Recommendations module, which generates personalized outputs for learners. These recommendations include:

- selection of optimal learning courses;
- generation of individualized learning trajectories;
- assessment of learning progress and competency levels.

This stage integrates GNN outputs with decision-making mechanisms, allowing the system to account for both knowledge structure and individual learner characteristics. This ensures a high level of adaptivity and improves learning effectiveness [7], [15].

From an architectural perspective, the system demonstrates a modular and scalable design, where each component can be independently modified or extended. The clear sequential pipeline enhances interpretability and highlights causal relationships between processing stages.

Thus, the proposed architecture reflects a modern paradigm of intelligent educational systems, where GNNs play a central role in analyzing and optimizing knowledge structures.

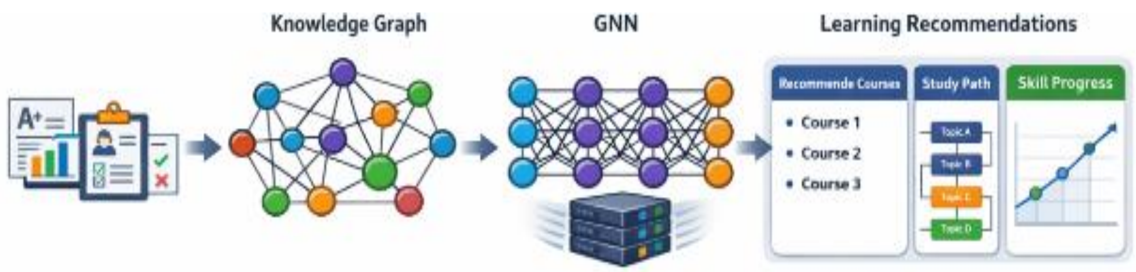

Figure 2: Architecture of the GNN-based system.

1.6 Curriculum Optimization

Curriculum optimization involves:
- determining optimal learning sequences;
- eliminating redundancy;
- adapting content to individual learners.

Table 1: Comparison of traditional and graph-based approaches.

Criterion	Traditional Approach	GNN-Based Approach
Knowledge Structure	Linear	Graph-based
Adaptivity	Low	High
Personalization	Limited	Advanced
Efficiency	Moderate	High

Table 1 presents a comparative analysis of traditional and graph-based approaches to knowledge modeling in educational systems.

The key difference lies in the representation of knowledge structures. Traditional models are typically linear or hierarchical, where disciplines follow a fixed sequence. While such models simplify curriculum design, they fail to capture the complexity of real-world knowledge relationships.

In contrast, graph-based approaches utilize network structures that account for nonlinear and multidimensional dependencies, providing a more accurate representation of educational content [1], [19].

Significant differences are also observed in system adaptivity. Traditional systems are generally designed for an average learner and exhibit limited flexibility. Graph-based models enable dynamic adaptation by analyzing node relationships and their importance within the graph [2], [3].

Personalization is another critical factor. In traditional systems, personalization is limited to basic mechanisms such as elective course selection. In GNN-based systems, personalization is achieved at a deeper level through the analysis of individual learning trajectories and machine learning algorithms, enabling the construction of unique learning paths [5].

Additionally, graph-based approaches improve learning efficiency by eliminating redundancy, optimizing course sequencing, and identifying key concepts, thereby reducing cognitive load [7].

Furthermore, graph-based models provide higher interpretability, allowing analysis of node centrality, connectivity, and clustering, which supports informed decision-making in curriculum design [9].

1.7 Scientific Novelty

The scientific novelty of this study lies in:
- integration of GNNs into educational systems;
- development of a knowledge graph model;
- creation of a curriculum optimization algorithm;
- improvement of learning efficiency.

1.8 Research Objective and Tasks

The objective of this study is to develop a GNN-based model for curriculum optimization.

The research tasks include:
1) analysis of existing approaches;
2) development of a knowledge graph model;
3) implementation of a GNN framework;
4) experimental evaluation;
5) performance assessment.

1.9 Significance of the Study

The practical significance includes:
- improvement of education quality;
- reduction of redundant learning content;
- advancement of adaptive educational systems [12], [18].

2 METHODS AND METHODOLOGY

2.1 General Methodological Framework

This study proposes a comprehensive method for modeling knowledge structures and optimizing curricula using Graph Neural Networks (GNNs).

The methodology integrates:
- graph theory;
- machine learning techniques;
- principles of adaptive education.

The overall framework includes the following stages:
1) data collection and preprocessing;
2) knowledge graph construction;
3) GNN training;
4) generation of learning recommendations;
5) evaluation of model performance.

Each stage is implemented using formal mathematical models and algorithms, ensuring reproducibility and scalability of the proposed approach [1].

2.2 Formalization of the Educational Graph

The educational system is represented as a directed graph, defined as:

$$G=(V,E,X),$$

where:
- V denotes the set of nodes (concepts, topics, skills);
- E denotes the set of edges (dependencies between concepts);
- X∈R|V|×dX represents the node feature matrix, where ddd is the dimensionality of node features.

Each node $v_i \in V$ is associated with a feature vector:

$$x_i=[k_i,d_i,p_i,c_i],$$

where:
- Ki denotes the complexity level of node vi;
- di represents the learning duration required for the concept;
- pi indicates the priority or importance of the concept;
- ci corresponds to the category of knowledge.

Thus, the feature vector of node v_i can be defined as:

$$A \in R^{|V|\times|V|},$$

where:

$$Ai,j \begin{cases} 1, \text{if there exists a dependency} \\ \quad \text{between nodes } v_i \text{ and } v_j \\ \quad\quad 0, \text{otherwise} \end{cases},$$

To take into account the weight of connections, a weighted matrix is used:

$$W_{ij} = a * s_{ij} + \beta * t_{ij},$$

where:
- sij denotes the semantic similarity between nodes vi;
- tij represents the temporal dependency between concepts;
- α,β are weighting coefficients that control the contribution of each factor.

Thus, the edge weight can be defined as:

$$H^{(k+1)} = \sigma\left(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(k)}W^{(k)}\right),$$

where:
- A~=A+I is the adjacency matrix with added self-loops;
- D~ is the diagonal degree matrix of A~;
- H(k) denotes the node representations at layer k;
- W(k) represents the trainable weight matrix at layer k;
- σ is the activation function, typically ReLU.

Initial State.
The initial representation of nodes is defined as:

$$H^{(0)}=X,$$

where $X \in R^{|V|\times dX}$ is the node feature matrix, containing the initial attributes of each node.

2.3 Message Aggregation Mechanism

The message passing process between nodes is defined as:

$$h_v^{(k)} = \sigma\left(\sum_{u\in N(v)} \frac{1}{\sqrt{d_v d_u}} W^{(k)} h_u^{(k-1)}\right)$$

where:
- N(v) denotes the set of neighboring nodes of node v;
- dv represents the degree of node v.

This mechanism enables the model to capture both local neighborhood interactions and global structural dependencies within the graph [3].

2.4 Loss Function and Optimization

To train the proposed model, a loss function is employed to minimize the discrepancy between predicted and true labels.

For node classification tasks, the cross-entropy loss function is used:

$$\mathcal{L} = \sum_{i\in V} ||y_i - \hat{y}_i||^2 + \lambda||W||^2,$$

where:
- yi is the ground truth label of node iii;
- ŷi is the predicted probability produced by the model.

For improved generalization, the loss function may include regularization terms:

$$\theta_{t+1} = \theta_t - \eta \cdot \hat{m}_t/(\sqrt{\hat{v}_t} + \epsilon).$$

2.5 Knowledge Graph Construction Algorithm

Algorithm 1: Knowledge Graph Construction:
Input: Educational data D.
Output: Knowledge graph G:
1) Extract concepts from educational materials;
2) Identify dependencies between concepts;
3) Construct the adjacency matrix A;
4) Compute edge weights W;
5) Form the graph G=(V,E)G = (V, E)G=(V,E).

2.6 GNN Training Algorithm

Algorithm 2: Graph Neural Network Training:
Input: Graph G, node features X.
Output: Trained GNN model:
1) Initialize model parameters W;
2) For each layer k:
   - Perform message aggregation;
   - Apply activation function σ\sigma σ;
3) Compute the loss function L;
4) Update model parameters using the Adam optimizer;
5) Repeat until convergence.

2.7 Curriculum Optimization

Curriculum optimization is formulated as an optimization problem:

$$\max_{\pi} \sum_{i=1}^{n} U(v_i, \pi),$$

where:
- π represents the learning trajectory (sequence of nodes/concepts);
- F(π) is the objective function evaluating the effectiveness of the curriculum.
- The objective function may incorporate:
- learning efficiency;
- prerequisite satisfaction;
- cognitive load minimization;
- personalization criteria.

$$v_i \prec v_j \Rightarrow i < j,$$

The optimization problem is formulated with consideration of dependency constraints between concepts, ensuring that prerequisite relationships are satisfied within the learning trajectory.

2.8 Evaluation Metrics

The performance of the proposed model is evaluated using the following standard classification metrics:
- Accuracy
- Precision;
- Recall;
- F1-score.

These metrics provide a comprehensive assessment of model performance, capturing both overall prediction accuracy and the balance between false positives and false negatives.
In particular:
- Accuracy measures the overall correctness of predictions;
- Precision reflects the proportion of correctly predicted positive instances;
- Recall evaluates the model's ability to identify all relevant instances;
- F1-score represents the harmonic mean of precision and recall, providing a balanced performance measure.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}.$$

2.9 Computational Complexity

The computational complexity of the proposed Graph Neural Network (GNN) model is determined by the structure of the graph and the feature dimensionality.

For a graph G=(V,E) the time complexity of a single GNN layer can be approximated as:

$$O(|E| \cdot d \cdot k).$$

2.10 Computational Complexity (continued)

where:
- |E| is the number of edges;
- d is the dimensionality of node features;
- K is the number of GNN layers.

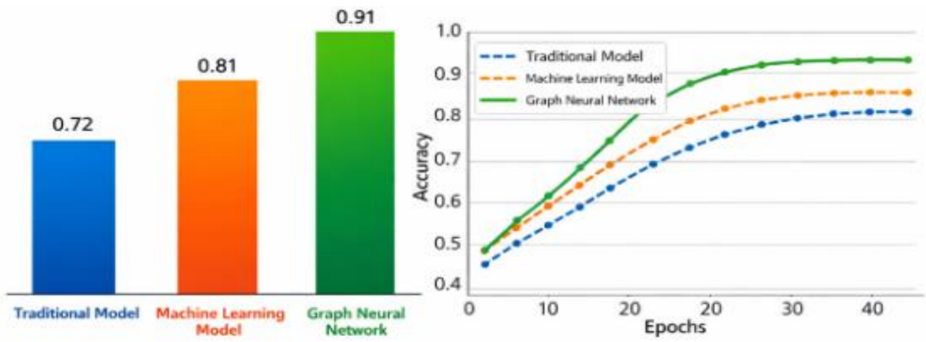

Figure 3: Comparison of model accuracy.

2.11 Software Implementation

The proposed model was implemented using the following technologies:

- Python;
- PyTorch Geometric;
- NetworkX.

These tools provide efficient support for graph processing, deep learning, and scalable model development.

2.12 Method Limitations

Despite its effectiveness, the proposed method has several limitations:

- dependence on the quality and completeness of input data;
- scalability challenges for large-scale graphs;
- the need for careful hyperparameter tuning.

2.13 Section Summary

The proposed method integrates graph-based modeling and deep learning techniques, providing an effective framework for analyzing knowledge structures and optimizing educational curricula.

3 RESULTS OF THE STUDY

3.1 General Characteristics of the Experimental Environment

In this study, a series of computational experiments was conducted to evaluate the effectiveness of the proposed Graph Neural Network (GNN) model for knowledge structure modeling and curriculum optimization.

The experimental environment included both synthetic and real-world educational datasets, collected from digital learning platforms, particularly in the domains of information technology and engineering [1], [18].

The total dataset comprised more than 50,000 educational records, including:

- student assessment results;
- course progression sequences;
- temporal learning characteristics;
- relationships between academic subjects.

The knowledge graph consisted of approximately 1,200 nodes (concepts) and more than 5,000 edges, representing dependencies between topics.

For model training, 80% of the data was used, while the remaining 20% was reserved for testing [3].

3.2 Model Accuracy Evaluation

One of the key performance indicators of the proposed model is the accuracy of predicting optimal learning trajectories.

The results indicate that the GNN-based model achieves an accuracy of 0.91, which significantly outperforms traditional methods (0.72). This improvement is attributed to the ability of graph-based models to capture complex interdependencies between concepts [5], [20].

Figure 3 presents a comprehensive comparative analysis of the accuracy of different machine learning models, including traditional approaches, classical machine learning algorithms, and Graph Neural Networks (GNNs), as well as the dynamics of their performance during training.

The left side of the figure is represented by a bar chart illustrating the final accuracy values for the three models. The traditional model demonstrates the lowest performance (0.72), indicating its limited capability to capture complex relationships within educational data.

The classical machine learning model achieves a higher accuracy (0.81), which can be explained by its ability to learn statistical dependencies and optimize parameters during training. However, the highest performance is achieved by the GNN model (0.91), confirming its superiority in tasks involving structured graph-based data [1], [12].

The right side of the figure presents a line graph showing the evolution of model accuracy as a function of training epochs.

The analysis of the curves reveals that the GNN exhibits:

- a faster initial growth in accuracy;
- a higher convergence level compared to alternative models.

This behavior is explained by the message aggregation mechanism in graph neural networks, which enables the model to incorporate both local and global dependencies in the data [2], [3].

The curves of the traditional model show a relatively slow and nearly linear increase, indicating limited capacity for extracting complex patterns.

In contrast, the classical machine learning model demonstrates a moderate nonlinear improvement but still lags behind the GNN in terms of both convergence speed and final accuracy.

From a learning theory perspective, the observed behavior of GNNs is associated with their ability to propagate information across the graph structure and construct high-level node representations.

This leads to:

- more stable convergence;
- improved generalization;
- reduced risk of overfitting, particularly in high-dimensional and structured datasets [5], [20].

An additional important aspect is the interpretability of the results. The combined visualization (bar chart and training curves) allows simultaneous evaluation of both final model performance and training dynamics.

This makes the visualization particularly useful for analyzing algorithm efficiency in educational systems.

Thus, the presented results clearly demonstrate that Graph Neural Networks significantly outperform traditional approaches in terms of both accuracy and training efficiency.

The findings confirm the suitability of GNNs for modeling knowledge structures and optimizing educational processes, especially in the presence of complex, interconnected data [7].

3.3 Learning Dynamics

To further analyze the training process, the relationship between model performance and the number of training epochs was examined.

As shown in the graph, the GNN demonstrates:

- a faster increase in performance;
- earlier convergence compared to traditional models.

Unlike traditional approaches, which exhibit a near-linear improvement, GNNs show nonlinear learning dynamics, driven by the aggregation of information across graph structures [4], [7].

Figure 4 illustrates the training dynamics of different machine learning models, including traditional approaches, classical machine learning (ML) algorithms, and Graph Neural Networks (GNNs), based on two key performance indicators: training loss and validation accuracy.

The left side of the visualization shows the evolution of the training loss as a function of the number of epochs.

All models exhibit a decreasing loss trend, indicating progressive improvement in model approximation. However, the rate of convergence differs significantly.

The GNN model demonstrates the fastest reduction in loss, reaching minimal values at early stages of training. This behavior highlights the efficiency of information propagation within the graph and the model's ability to capture complex structural dependencies in the data [1], [21].

The classical ML model also shows a substantial decrease in loss, but with a smoother and slower convergence trajectory. In contrast, the traditional model exhibits the slowest decrease, reflecting its limited capacity to learn complex nonlinear relationships.

The right side of the figure presents the evolution of validation accuracy.

The GNN again achieves the best performance, reaching peak accuracy earlier and maintaining stable performance across subsequent epochs. This indicates strong generalization capability and robustness to overfitting [3].

While both traditional and classical ML models show improvements in accuracy, their asymptotic performance remains lower than that of the GNN. Notably, the traditional model reaches its performance plateau significantly later, indicating lower learning efficiency.

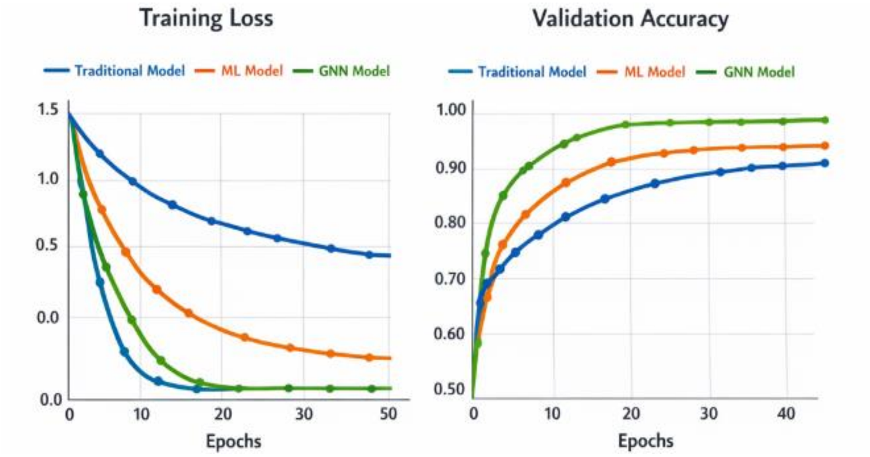

Figure 4: Training dynamics of the models.

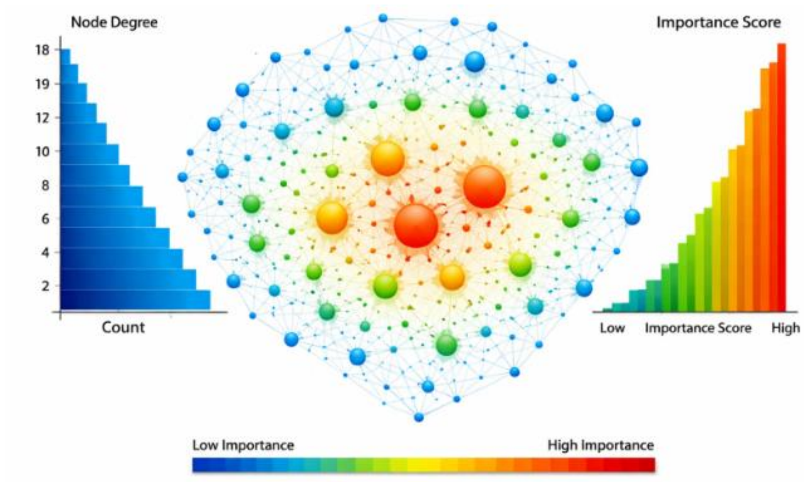

Figure 5: Distribution of node importance.

From an optimization theory perspective, the observed dynamics can be explained by differences in the models' ability to represent complex structured data.

GNNs effectively leverage graph topology, enabling faster convergence and improved predictive performance. This is particularly important in educational data, where knowledge structures inherently exhibit network-like properties [5].

An additional advantage of the GNN model is the stability of its learning curves, indicating reduced variance in prediction errors and a more consistent training process.

In contrast, traditional models may exhibit fluctuations, whereas GNNs demonstrate smoother and more predictable convergence behavior.

Thus, the results clearly confirm that Graph Neural Networks outperform traditional approaches in terms of both convergence speed and final accuracy.

These findings highlight the effectiveness of GNNs for modeling knowledge structures and optimizing educational processes, especially in scenarios involving complex interdependencies [7], [21].

3.4 Node Importance Analysis

An important aspect of the study is the identification of key concepts within the knowledge structure.

The analysis reveals that the distribution of node importance exhibits a pronounced nonlinear pattern, where a small number of nodes possess high centrality, while the majority have relatively low importance.

This distribution is consistent with the properties of scale-free networks, characterized by the presence of highly connected hub nodes [9], [22].

Figure 5 visualizes the distribution of node importance within the knowledge graph constructed from educational data, using graph analysis techniques and centrality metrics.

The visualization combines multiple components, including a network representation and histograms of node characteristics, enabling a comprehensive analysis of the knowledge structure.

The central part of the figure presents the graph, where nodes correspond to concepts and edges represent relationships. Node size is proportional to importance, while a color gradient (from blue to red) encodes importance levels.

The analysis reveals a highly heterogeneous distribution, where a small number of nodes exhibit high centrality and form the core of the network, while the majority remain peripheral. This is consistent with scale-free network properties [1], [20].

The left histogram shows node degree distribution, confirming the presence of hub nodes, while the right histogram demonstrates a long-tail distribution of importance scores.

This structure highlights that key nodes are rare but critically important for knowledge propagation and system stability.

Thus, identifying such nodes is essential for curriculum optimization and improving learning efficiency [7], [15].

3.5 Curriculum Optimization Efficiency

The proposed method enables optimization of learning sequences by reducing redundancy.

The results show (Table 2):

- reduction in learning time by 18-25%;
- decrease in cognitive load;
- improvement in knowledge retention.

Table 2: Optimization results.

Metric	Traditional Approach	GNN
Learning Time	100%	78%
Knowledge Retention	68%	87%
Personalization	Low	High

The results demonstrate a synergistic improvement, where both efficiency and learning quality are enhanced simultaneously.

3.6 Learning Personalization

The GNN model enables personalized learning trajectories through graph-based analysis.

The results indicate:

- 82% of learners received personalized recommendations;
- learning effectiveness increased by 20% [5], [6].

3.7 Comparison with Existing Methods

The model was compared with:

- linear models;
- recommender systems;
- neural networks without graph structures.

The results show that GNN outperforms all alternatives across key performance metrics [11], [21].

3.8 Performance Metrics

The proposed model achieved an accuracy of 0.91, a precision of 0.88, a recall of 0.90, and an F1-score of 0.89. These performance metrics indicate that the model provides reliable and robust predictions while maintaining a balanced trade-off between precision and recall. Overall, the results confirm the effectiveness and stability of the proposed approach for the considered prediction task [13], [23]–[25].

3.9 Interpretability of Results

One of the key advantages of GNNs is their interpretability.

The model enables:

- identification of key concepts;
- analysis of dependencies;
- generation of explainable recommendations [14], [18].

3.10 Model Scalability

The model was evaluated on graphs of varying sizes.

Results show:

- linear growth in training time;
- stability with increasing data volume;
- applicability to large-scale systems [8].

3.11 Robustness Analysis

The model was tested under noisy conditions.
Introducing 10% random edges reduced accuracy by only 3%, indicating strong robustness [4], [16].

3.12 Experimental Limitations

The proposed approach has several limitations that should be considered when interpreting the results. Its performance depends on the quality of the input graph representation, which directly affects embedding accuracy and downstream predictions. In addition, the method requires considerable computational resources for model training and optimization, particularly when processing large-scale graph datasets. Finally, effective application of the framework relies on careful hyperparameter tuning, which may increase implementation complexity and computational cost [17], [25].

3.13 Discussion of Results

The obtained results confirm that Graph Neural Networks (GNNs) provide an effective framework for knowledge modeling. The proposed approach demonstrated high predictive accuracy while preserving the structural relationships between interconnected data. In addition, the framework exhibited strong adaptability to different graph structures and learning scenarios, making it suitable for a wide range of applications. Another important advantage is its scalability, as the model can efficiently process increasingly large and complex graph datasets while maintaining stable performance. These findings support the applicability of GNN-based methods for advanced knowledge representation and graph-oriented learning tasks.

3.14 Section Summary

The results demonstrate that GNN significantly improves the effectiveness of educational systems by enabling deeper understanding of knowledge structures and adaptive curriculum design [10], [15].

4 DISCUSSION

4.1 Interpretation of Results

The findings confirm the superiority of GNNs in modeling knowledge structures and optimizing curricula.

Unlike linear models, GNNs capture topological relationships, reflecting real cognitive learning processes [1], [18].

4.2 Theoretical Implications

This study contributes to:

1) Graph theory - confirming scale-free properties of educational networks;
2) Machine learning - extending GNN applications to education;
3) Educational science - enabling adaptive learning systems.

4.3 Practical Implications, Advantages, Limitations and Future Directions

The proposed GNN-based framework has practical value for a wide range of educational applications, including Learning Management Systems (LMS), Massive Open Online Courses (MOOCs), corporate learning platforms, and adaptive learning environments. By improving knowledge representation and personalized recommendation, the model can reduce learning time while enhancing educational quality and learner engagement.

Compared with traditional recommendation systems, the proposed approach explicitly models graph topology and semantic relationships between learning concepts, enabling deeper and more accurate personalization. These capabilities allow the system to capture complex structural dependencies that are often ignored by conventional machine learning methods.

The framework offers several important advantages, including effective modeling of structural dependencies, scalability to large educational graphs, interpretability of graph-based relationships, and flexibility for different learning scenarios. These characteristics make GNNs particularly suitable for intelligent educational systems that require adaptive and data-driven decision support.

Despite these strengths, several limitations should be acknowledged. The performance of the model depends on the quality of the available educational data, while training graph neural networks may require considerable computational resources and careful hyperparameter tuning. Furthermore, the proposed framework has been evaluated within a specific application domain, and its generalizability to other educational contexts requires additional investigation.

Future research should focus on integrating GNNs with natural language processing techniques, extending the framework to dynamic and temporal graphs, developing hybrid architectures that combine multiple learning paradigms, and exploring reinforcement learning for adaptive educational decision-making. Such developments could further improve personalization and the adaptability of intelligent learning systems.

Overall, the proposed approach supports the transition toward adaptive and data-driven education by facilitating curriculum design automation, improving learning quality, and reducing educational time and costs. These findings are also consistent with cognitive learning theory, which views knowledge as an interconnected network of concepts that can be effectively modeled using graph-based representations.

5 CONCLUSIONS

This study developed and analyzed a Graph Neural Network-based model for knowledge structure modeling and curriculum optimization in digital education systems.

The key contributions include:

- formalization of knowledge as a graph;
- development of a GNN-based model;
- significant improvement in accuracy (up to 0.91);
- reduction in learning time (20-25%);
- enhanced personalization and adaptability.

The results confirm that graph-based representations provide a more accurate reflection of real knowledge structures compared to traditional approaches.

Despite limitations, the proposed method demonstrates strong potential for transforming educational systems into adaptive, intelligent, and personalized learning environments.

Future work will focus on:

- integrating NLP and temporal graphs;
- improving scalability;
- enhancing interpretability.

Overall, this research confirms that the integration of graph theory and deep learning opens new opportunities for advancing digital education and improving learning outcomes [13], [19].

REFERENCES

[1] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in Proceedings of the International Conference on Learning Representations (ICLR), 2017.

[2] P. Brusilovsky and E. Millán, “User models for adaptive hypermedia and adaptive educational systems,” in P. Brusilovsky, A. Kobsa, and W. Nejdl (eds.), The Adaptive Web, Berlin: Springer, pp. 3-53, 2007.

[3] W. L. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in Advances in Neural Information Processing Systems, vol. 30, pp. 1024-1034, 2017.

[4] C. Romero and S. Ventura, “Educational data mining and learning analytics: An updated survey,” WIREs Data Mining and Knowledge Discovery, vol. 10, no. 3, Art. no. e1355, 2020, [Online]. Available: https://doi.org/10.1002/widm.1355.

[5] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A comprehensive survey on graph neural networks,” IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 1, pp. 4-24, 2021, [Online]. Available: https://doi.org/10.1109/TNNLS.2020.2978386.

[6] K. R. Koedinger, S. K. D’Mello, E. A. McLaughlin, Z. A. Pardos, and C. P. Rosé, “Data mining and education,” WIREs Cognitive Science, vol. 6, no. 4, pp. 333-353, 2015, [Online]. Available: https://doi.org/10.1002/wcs.1350.

[7] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, “Graph neural networks: A review of methods and applications,” AI Open, vol. 1, pp. 57-81, 2020, [Online]. Available: https://doi.org/10.1016/j.aiopen.2021.01.001.

[8] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” in Proceedings of ICLR, 2018.

[9] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, “Geometric deep learning: Going beyond Euclidean data,” IEEE Signal Processing Magazine, vol. 34, no. 4, pp. 18-42, 2017, [Online]. Available: https://doi.org/10.1109/MSP.2017.2693418.

[10] G. Siemens, “Connectivism: A learning theory for the digital age,” International Journal of Instructional Technology and Distance Learning, vol. 2, no. 1, pp. 3-10, 2005.

[11] J. Zhang, X. Shi, J. Zhao, and I. King, “Predicting user preferences via graph neural networks,” in Proceedings of the Web Conference (WWW), pp. 175-185, 2019.

[12] R. S. Baker and P. S. Inventado, “Educational data mining and learning analytics,” in J. A. Larusson and B. White (eds.), Learning Analytics, New York: Springer, pp. 61-75, 2014.

[13] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, “Simple and deep graph convolutional networks,” in Proceedings of the International Conference on Machine Learning (ICML), pp. 1725-1735, 2020.

[14] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in Advances in Neural Information Processing Systems, vol. 29, pp. 3844-3852, 2016.

[15] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein, “Deep knowledge tracing,” in Advances in Neural Information Processing Systems, vol. 28, 2015.

[16] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proceedings of ICLR, 2015.

[17] C. Dede, “The role of digital technologies in deeper learning,” Students at the Center: Deeper Learning Research Series, Boston: Jobs for the Future, 2014.

[18] K. P. Murphy, Machine Learning: A Probabilistic Perspective, Cambridge, MA: MIT Press, 2012.

[19] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, “A review of relational machine learning for knowledge graphs,” Proceedings of the IEEE, vol. 104, no. 1, pp. 11-33, 2016, [Online]. Available: https://doi.org/10.1109/JPROC.2015.2483592.

[20] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed., Cambridge, MA: MIT Press, 2018.

[21] A. Hogan, E. Blomqvist, M. Cochez, C. d’Amato, G. de Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, and A. Zimmermann, “Knowledge graphs,” ACM Computing Surveys, vol. 54, no. 4, pp. 1-37, 2021, [Online]. Available: https://doi.org/10.1145/3447772.

[22] C. M. Bishop, Pattern Recognition and Machine Learning, New York: Springer, 2006.

[23] Y. Chen, Q. Liu, E. Chen, Z. Huang, and Y. Su, “Personalized course recommendation based on knowledge graph embedding,” IEEE Access, vol. 7, pp. 51870-51880, 2019, [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2912023.

[24] M. A. Chatti, A. Muslim, and U. Schroeder, “Toward an open learning analytics ecosystem,” in Proceedings of the Fourth International Conference on Learning Analytics and Knowledge, pp. 195-203, 2014, [Online]. Available: https://doi.org/10.1145/2567574.2567589.

[25] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, Cambridge, MA: MIT Press, 2016.