EURASIAN JOURNAL OF MATHEMATICAL AND COMPUTER APPLICATIONS
ISSN 2306–6172
Volume 13, Issue 4 (2025) 159 – 167

MODELLING CURRICULA WITH GNN AND LSTM FOR LINK AND SEQUENCE PREDICTION

Kozhanov M. [1], **Varkonyi-Koczy A.**

Abstract A mathematical model of a university curriculum is proposed. A prerequisite structure is represented as a directed graph, while course order is represented as a sequence. Courses are represented by embeddings that are obtained from syllabus texts and metadata. These vectors are refined by a two layer graph convolutional network (GCN), which is used for link prediction of missing or potentially incorrect prerequisite relations. Course order is represented by a Long Short Term Memory (LSTM) network, which predicts the next course from a fixed window of previously completed courses. An experiment on two bachelor programmes, which are Information Systems and Electric Power Engineering, is reported for 92 courses and 90 explicit prerequisite links. Acceptable quality is obtained for link prediction and sequence reconstruction. Compact tables of recommended new prerequisites are produced for both programmes, which are suitable for curriculum revision decisions.

Keywords: curriculum; graph neural network; recurrent neural network; link prediction; educational trajectory; mathematical modelling; artificial intelligence; machine learning; data science.

AMS Mathematics Subject Classification: 68T07, 68T50, 68R10, 68Q32.

DOI: 10.32523/2306-6172-2025-13-4-159-167

1 Introduction

In university practice, a curriculum is used for education planning. Course sets, course order, and prerequisite relations are specified. Curriculum coherence is linked with competence formation and graduate outcomes, which are assessed by employers and accreditation bodies [1]. Curricula are revised due to technological change and labour market shifts [2]. Interdisciplinary programmes are expanded, e.g., information technology and electric power engineering. For this reason, a coherent progression is required from basic courses to advanced courses, while balance is maintained between fundamental training and applied competences. Traditionally, curriculum design is completed by expert committees. However, data volume is increased, which includes syllabi, learning outcomes, enrolment statistics, and performance indicators. As a result, manual analysis is limited. Therefore, educational data mining methods are used more often in curriculum analysis [3]. Also, digital platforms are used for university development tasks, in which competence databases are formed and expert evaluation is supported. Deep learning research in education is described in three strands. First, high accuracy is reported for student outcome prediction, which is obtained by models such as convolutional neural networks, recurrent networks, and deep feedforward networks [4]. This strand includes sequence tasks, in which learning behaviour patterns are inferred and next events are predicted. Also, link prediction in networks is reported as an analogous task, which informs relation inference problems in structured educational data [5, 6]. Second, early risk

[1]Corresponding Author.

detection is supported, which enables intervention and personalised learning, because individual learning needs are analysed. Third, constraints are reported, in which data volume, representativeness, interpretability, and scalability are described as limitations. Furthermore, hybrid modelling is proposed, in which deep learning is combined with traditional methods, while heterogeneous data integration is targeted.

A research gap is identified because many studies focus on student performance prediction. Explicit modelling of curricular structures is addressed less often, in which prerequisite links and learning sequences are inferred from structured representations. As a result, automated inference of missing prerequisite relations is limited for curriculum planning. For this reason, a joint approach is required, in which structural relations and temporal order are modelled together. Several related directions are reported for prerequisite inference. Prerequisite structures are inferred from learner trajectories and clickstream data [7]. In addition, prerequisite information is incorporated into knowledge tracing, which supports structured modelling of learning sequences [8]. These works indicate that implicit dependencies are inferred and are used to refine formal prerequisite graphs. In this paper, a combined mathematical model is introduced. A curriculum is represented as a directed graph, in which vertices represent courses and edges represent prerequisite relations. Course embeddings are obtained from syllabus texts, while sentence models are used for textual representations [9]. These representations are refined by a graph convolutional network [10], which enables link prediction of candidate prerequisites. Also, course order is represented as a sequence, which is modelled by a Long Short Term Memory network [11]. A next course prediction task is used, in which a fixed window of previous courses is processed.

2 Mathematical model of the curriculum

In this section, a mathematical model is defined. A graph structure and a set of course sequences are used. Evaluation metrics are defined for the prediction tasks.

Graph representation of the curriculum

A curriculum is represented by a directed graph [12]

$$G = (C, E),$$

where

$$C = \{c_1, c_2, \ldots, c_n\}$$

is the set of courses and

$$E \subseteq C \times C$$

is the set of edges. An edge $(c_i, c_j) \in E$ is interpreted as a prerequisite relation.

The graph is encoded by an adjacency matrix [12]

$$A \in \{0,1\}^{n \times n},$$

where

$$A_{ij} = \begin{cases} 1, & \text{if } (c_i, c_j) \in E, \\ 0, & \text{otherwise.} \end{cases}$$

Self loops are included as

$$\widetilde{A} = A + I_n,$$

where I_n is the identity matrix. A degree matrix is defined by

$$\widetilde{D}_{ii} = \sum_{j=1}^{n} \widetilde{A}_{ij}.$$

Vector representations of courses

Each course c_i is described by syllabus text and metadata. A feature vector is defined as

$$x_i \in \mathbb{R}^d.$$

Text representations are obtained by transformer language models [13]. Sentence level representations are obtained by a sentence embedding variant. Let

$$X \in \mathbb{R}^{n \times d}$$

denote the feature matrix, with row $X_{i:}$ equal to x_i. Normalisation is applied to improve training stability, which supports dot product based scoring [14].

Graph neural network

A graph convolutional network is used [10]. A GCN layer is defined by

$$H^{(l+1)} = \sigma\big(\widetilde{D}^{-1/2}\widetilde{A}\widetilde{D}^{-1/2}H^{(l)}W^{(l)}\big),$$

where $H^{(0)} = X$, $H^{(l)} \in \mathbb{R}^{n \times d_l}$, $W^{(l)} \in \mathbb{R}^{d_l \times d_{l+1}}$, and $\sigma(\cdot)$ is an activation function [15]. After L layers, an embedding matrix is obtained

$$Z = H^{(L)} \in \mathbb{R}^{n \times d_z},$$

in which row z_i represents course c_i.

Link prediction model

A link prediction task is defined for course pairs. A scoring function is defined as

$$s(i,j) = \varphi(z_i, z_j),$$

where φ is scalar. A logistic dot product score is used [14]

$$s(i,j) = \sigma\big(z_i^\top z_j\big),$$

in which σ is the logistic function.

Training is completed by positive and negative examples. Positive examples are edges $(i,j) \in E$. Negative examples are obtained by sampling from $C \times C \setminus E$. Negative sampling is used to control class balance [17]. A binary cross entropy loss is used [15]

$$L_{\text{link}} = -\sum_{(i,j)\in P} \log s(i,j) - \sum_{(i,j)\in N} \log\big(1 - s(i,j)\big),$$

where P is a set of positive pairs and N is a set of negative pairs.

Sequence model of courses

Course order is represented as a sequence

$$S = (c_{t_1}, c_{t_2}, \dots, c_{t_T}),$$

in which indices reflect semesters. A Long Short Term Memory model is used [11]. At step k, an embedding z_{t_k} is processed as

$$h_k = \mathrm{LSTM}(z_{t_k}, h_{k-1}).$$

A next course distribution is defined by

$$P\big(c_{t_{k+1}} \,\big|\, c_{t_1}, \dots, c_{t_k}\big) = \mathrm{softmax}(W_o h_k),$$

while the softmax formulation is used for multi class prediction [14]. A cross entropy loss is used [15]

$$L_{\mathrm{seq}} = -\sum_{k=1}^{T-1} \log P\big(c_{t_{k+1}} \,\big|\, c_{t_1}, \dots, c_{t_k}\big).$$

A combined loss is defined

$$L = L_{\mathrm{link}} + \lambda L_{\mathrm{seq}},$$

in which $\lambda > 0$ is a balancing coefficient.

Evaluation metrics

Ranking metrics are used for link prediction. For each query, a ranked list of candidate neighbours is formed. Mean reciprocal rank is defined as

$$\mathrm{MRR} = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{\mathrm{rank}_q}.$$

Hits at k is defined as

$$\mathrm{Hits@}k = \frac{1}{|Q|} \sum_{q \in Q} I\big(\mathrm{rank}_q \le k\big).$$

Next course accuracy is used for sequence prediction. Ranking evaluation is described in information retrieval literature, in which top k measures are standard [16]

$$\mathrm{Acc} = \frac{1}{T-1} \sum_{k=1}^{T-1} I\big(\hat{c}_{t_{k+1}} = c_{t_{k+1}}\big).$$

3 Methodology of the computational experiment

Data structure, feature construction, model configuration, and a recommendation procedure are described.

Data description and graph structure

Curricula of two bachelor programmes are used:

- Information Systems (programme IS);
- Electric Power Engineering (programme EE).

Each record in the course table includes identifier, code, title, programme, cycle, component, and order. The course matrix is reported as

$$\text{courses\_df} \in \mathbb{R}^{92\times 7}.$$

The prerequisite graph is reported by an edge table. Each edge defines a source course, a target course, and a relation type `prerequisite_chain`. The edge count is

$$|E| = 90,$$

which corresponds to two linear prerequisite chains, i.e., 45 edges per programme.

Course syllabi are stored in a table that includes identifier and syllabus text. The syllabus table is reported as

$$\text{syllabus\_df} \in \mathbb{R}^{92\times 6}.$$

After a join by course identifier, a merged table is obtained

$$\text{data} \in \mathbb{R}^{92\times 8}.$$

A binary adjacency matrix is constructed [12]

$$A \in \{0,1\}^{92\times 92},$$

and a normalised matrix is defined [10]

$$\widehat{A} = \widetilde{D}^{-1/2}\widetilde{A}\widetilde{D}^{-1/2},$$

in which $\widetilde{A} = A + I$.

Course embeddings

A feature vector is constructed for each course from syllabus text and metadata. Sentence embeddings are computed. Categorical features are encoded and are concatenated, when they are available. Therefore, a feature matrix is obtained

$$X \in \mathbb{R}^{92\times d},$$

with $d = 249$. Normalisation is applied before graph processing, which reduces scale effects in gradient based optimisation [15].

Configuration and training of the graph model

A two layer GCN is used [10]. The structure is defined as

$$H^{(1)} = \sigma\big(\widehat{A}XW^{(0)}\big), \qquad Z = \widehat{A}H^{(1)}W^{(1)}.$$

Dimensions are selected as

- input size $d_{\text{in}} = 249$;
- hidden size $d_{\text{hid}} = 128$;
- output size $d_z = 64$.

For link prediction, positive pairs are defined as observed edges. Negative pairs are sampled with equal count, which reduces label imbalance effects [17]. A logistic dot product score is used [5, 6]. Training is completed for 50 epochs on a CPU. Parameter optimisation is completed by Adam [18]. Regularisation is supported by dropout [19].

On the complete link set, the metrics are reported as

$$\text{MRR} \approx 0.2545, \qquad \text{Hits@10} \approx 0.7222.$$

Configuration and training of the sequence model

For each programme, a sequence of length $T = 46$ is defined by curriculum order. A window length $L = 5$ is selected. For each window, the next course is predicted. Therefore, $M = 82$ training examples are obtained.

Inputs are defined as sequences of five embeddings in $\mathbb{R}^{64}$, while targets are course indices. An LSTM layer is used with hidden size 128, followed by a fully connected layer with output size 92 [11]. Training is completed for 50 epochs. Cross entropy training is used for the classification objective [15]. Next course accuracy is reported as

$$\text{Acc} \approx 0.7561.$$

Construction of recommended prerequisite tables

A similarity matrix is computed from normalised embeddings

$$S = Z_{\text{norm}} Z_{\text{norm}}^{\top},$$

which corresponds to cosine similarity, while vector normalisation supports angular comparison [16]. For each course, top k neighbours are selected. Candidate pairs are filtered as follows:

1. existing prerequisite edges are removed;
2. only within programme pairs are kept;
3. target order is constrained to be larger than source order;
4. similarity is constrained by similarity_score $\geq \tau$;
5. service courses are removed, e.g., internships and final assessment.

At the initial stage, 460 candidate links are obtained. Then, strict filtering yields 116 recommendations, which include 60 links for IS and 53 links for EE. Summary tables are saved as `recommended_prereq_IS_summary.csv` and `recommended_prereq_EE_summary.csv`.

4 Results of the computational experiment

Experimental setup and prediction results are summarised for programmes IS and EE.

Experimental setup

Three data sets are used:

- course table with identifiers, titles, programme, cycle, component, and order;
- edge table with source and target pairs and relation type `prerequisite_chain`;
- syllabus table with syllabus texts.

The merged data set includes 92 courses and 90 prerequisite links. Embeddings of dimension $d = 249$ are computed for all syllabi [9]. All computations are completed on a CPU.

Link prediction quality

A two layer GCN is used, which maps $249 \to 128 \to 64$ [10]. Binary cross entropy training is used with 90 positive and 90 negative pairs [15]. The metrics are reported as

$$\mathrm{MRR} \approx 0.25, \qquad \mathrm{Hits@10} \approx 0.72.$$

Therefore, true neighbours are ranked near the top for many queries.

Sequence prediction quality

An LSTM model is trained for next course prediction [11]. The accuracy is reported as

$$\mathrm{Acc} \approx 0.76.$$

Hence, curriculum order is reconstructed with reasonable precision.

Recommended prerequisite links

After filtering, 116 recommended links are obtained. For IS, recommendations connect mathematics and discrete mathematics with foundational and advanced information technology courses. For EE, recommendations connect mathematics and physics with electrical engineering and power engineering courses. These tables are used as data based proposals for prerequisite revision.

Discussion

Three observations are reported. First, prerequisite links are recovered with acceptable ranking quality, because graph topology and content representations are combined [6, 10]. Second, curriculum order is reproduced with reasonable accuracy, because sequence dependence is captured by LSTM [11]. Third, recommendation tables remain interpretable for curriculum experts, because pairs are filtered by programme, order, and similarity threshold.

Further model variants are relevant. Graph attention can be used to support adaptive neighbourhood aggregation [20]. Inductive aggregation can be used for curriculum updates, in which new courses are added after training [21]. Relational graph modelling can be used when multiple edge types are defined, e.g., prerequisite and corequisite relations [22]. Baseline representations can be obtained by random walk embeddings [23, 24]. These options can be used for comparative analysis, which supports robustness checks [25].

4 Conclusions

In this study a curriculum model is proposed, in which prerequisite relations are represented as a directed graph, while course order is represented as a sequence. Course embeddings are obtained from syllabus texts. A graph convolutional network is used for prerequisite link prediction. A Long Short Term Memory model is used for next course prediction. An experiment is reported for two bachelor programmes with 92 courses and 90 explicit prerequisite links. Acceptable quality is reported for link prediction and for sequence reconstruction. Tables of recommended prerequisites are produced for both programmes. In this study two programmes were analysed, while broader coverage is required for generalisation. Embeddings are obtained from a general sentence model without domain specific adaptation. Additional variables are not incorporated, e.g., grades and workload constraints. Future work is directed to larger and multilingual syllabus sets, stronger graph architectures, and richer educational signals, which include student performance traces and competence descriptors.

Acknowledgements

The use of large language models in this article is limited to minor language editing. After the preparation of the first draft and after the preparation of revised versions, Chatgpt is used to improve readability and language and to remove typographical errors. The content is reviewed and edited by the authors, who take full responsibility for the final manuscript. Furthermore, the authors declare they have prior joint publications with one member of the editorial board.

References

[1] Jantassova D., Kozhanov M., Shebalina O., *Digital Platform as a Tool for Internationalization: Model for Formation of International Competences' Database Applying of Hierarchy Analysis Method*, Journal of Theoretical and Applied Information Technology, 99(21) (2021), 4942–4957.

[2] Lin Y., Chen Y., Xia Y., Lin W., Wang K., Liu J., *A Comprehensive Survey on Deep Learning Techniques in Educational Data Mining*, Data Science and Engineering, 10 (2025), 564–590. DOI: 10.1007/s41019-025-00303-z.

[3] Song X., Li J., Cai T., Yang S., Yang T., Liu C., *A Survey on Deep Learning Based Knowledge Tracing*, Knowledge-Based Systems, 258 (2022), 110036. DOI: 10.1016/j.knosys.2022.110036.

[4] Piech C., Bassen J., Huang J., Ganguli S., Sahami M., Guibas L., Sohl-Dickstein J., *Deep Knowledge Tracing*, Advances in Neural Information Processing Systems, 28 (2015).

[5] Bordes A., Usunier N., Garcia-Duran A., Weston J., Yakhnenko O., *Translating Embeddings for Modeling Multi Relational Data*, Advances in Neural Information Processing Systems, 26 (2013).

[6] Kipf T. N., Welling M., *Variational Graph Auto Encoders*, arXiv: 1611.07308, 2016.

[7] Chen W., Lan A. S., Cao D., Brinton C. G., Chiang M., *Behavioral Analysis at Scale: Learning Course Prerequisite Structures from Learner Clickstreams*, Proceedings of the 11th International Conference on Educational Data Mining, 2018.

[8] Chen Y., Lu Y., Zheng Q., Pian Y., *Prerequisite Driven Deep Knowledge Tracing*, Proceedings of the 11th International Conference on Educational Data Mining, 2018.

[9] Reimers N., Gurevych I., *Sentence BERT: Sentence Embeddings using Siamese BERT Networks*, Proceedings of EMNLP-IJCNLP (2019), 3982–3992. DOI: 10.18653/v1/D19-1410.

[10] Kipf T. N., Welling M., *Semi Supervised Classification with Graph Convolutional Networks*, International Conference on Learning Representations, 2017. arXiv: 1609.02907.

[11] Hochreiter S., Schmidhuber J., *Long Short Term Memory*, Neural Computation, 9(8) (1997), 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[12] Newman M. E. J., *Networks: An Introduction*, Oxford University Press, 2010.

[13] Devlin J., Chang M.-W., Lee K., Toutanova K., *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, Proceedings of NAACL-HLT (2019), 4171–4186.

[14] Bishop C. M., *Pattern Recognition and Machine Learning*, Springer, 2006.

[15] Goodfellow I., Bengio Y., Courville A., *Deep Learning*, MIT Press, 2016.

[16] Manning C. D., Raghavan P., Schutze H., *Introduction to Information Retrieval*, Cambridge University Press, 2008.

[17] Mikolov T., Sutskever I., Chen K., Corrado G. S., Dean J., *Distributed Representations of Words and Phrases and Their Compositionality*, Advances in Neural Information Processing Systems, 26 (2013), 3111–3119.

[18] Kingma D. P., Ba J., *Adam: A Method for Stochastic Optimization*, International Conference on Learning Representations, 2015. arXiv: 1412.6980.

[19] Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R., *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*, Journal of Machine Learning Research, 15 (2014), 1929–1958.

[20] Velickovic P., Cucurull G., Casanova A., Romero A., Lio P., Bengio Y., *Graph Attention Networks*, International Conference on Learning Representations, 2018. arXiv: 1710.10903.

[21] Hamilton W., Ying Z., Leskovec J., *Inductive Representation Learning on Large Graphs*, Advances in Neural Information Processing Systems, 30 (2017).

[22] Schlichtkrull M., Kipf T. N., Bloem P., van den Berg R., Titov I., Welling M., *Modeling Relational Data with Graph Convolutional Networks*, European Semantic Web Conference (2018), 593–607.

[23] Perozzi B., Al-Rfou R., Skiena S., *DeepWalk: Online Learning of Social Representations*, Proceedings of KDD (2014), 701–710.

[24] Grover A., Leskovec J., *node2vec: Scalable Feature Learning for Networks*, Proceedings of KDD (2016), 855–864.

[25] Kozhanov M., et al., *Syllabus design and planning with artificial intelligence and the potential of generative AI*, In 2024 IEEE 24th International Symposium on Computational Intelligence and Informatics (CINTI), pp. 257–264, IEEE (2024).

Murat Kozhanov,
Doctoral School of Applied Informatics and Applied Mathematics, Obuda University, Budapest, Hungary;
Abylkas Saginov Karaganda Technical University, Karaganda, Kazakhstan;
e-mail: `m.kozhanov@ktu.edu.kz`.

Annamaria R. Varkonyi-Koczy,
John von Neumann University, Kecskemet, Hungary;
J. Selye University, Komarno, Slovakia;
.

Received 09.09.2025 , Accepted 17.11.2025, Available online 31.12.2025