International Journal of Educational Technology in Higher Education

RESEARCH ARTICLE Open Access

Extracting topological features to identify at-risk students using machine learning and graph convolutional network models

Balqis Albreiki[1,2*], Tetiana Habuza[1] and Nazar Zaki[1]

*Correspondence:
200907523@uaeu.ac.ae

[1] Department of Computer Science and Software Engineering, College of Information Technology, UAE University, Al Ain, UAE
[2] Office of Institutional Effectiveness, UAE University, Al Ain, UAE

Abstract

Technological advances have significantly affected education, leading to the creation of online learning platforms such as virtual learning environments and massive open online courses. While these platforms offer a variety of features, none of them incorporates a module that accurately predicts students' academic performance and commitment. Consequently, it is crucial to design machine learning (ML) methods that predict student performance and identify at-risk students as early as possible. Graph representations of student data provide new insights into this area. This paper describes a simple but highly accurate technique for converting tabulated data into graphs. We employ distance measures (Euclidean and cosine) to calculate the similarities between students' data and construct a graph. We extract graph topological features (*GF*) to enhance our data. This allows us to capture structural correlations among the data and gain deeper insights than isolated data analysis. The initial dataset (*DS*) and *GF* can be used alone or jointly to improve the predictive power of the ML method. The proposed method is tested on an educational dataset and returns superior results. The use of *DS* alone is compared with the use of $DS + GF$ in the classification of students into three classes: "failed","at risk", and "good". The area under the receiver operating characteristic curve (AUC) reaches 0.948 using *DS*, compared with 0.964 for $DS + GF$. The accuracy in the case of $DS + GF$ varies from 84.5 to 87.3%. Adding *GF* improves the performance by 2.019% in terms of AUC and 3.261% in terms of accuracy. Moreover, by incorporating graph topological features through a graph convolutional network (GCN), the prediction performance can be enhanced by 0.5% in terms of accuracy and 0.9% in terms of AUC under the cosine distance matrix. With the Euclidean distance matrix, adding the GCN improves the prediction accuracy by 3.7% and the AUC by 2.4%. By adding graph embedding features to ML models, at-risk students can be identified with 87.4% accuracy and 0.97 AUC. The proposed solution provides a tool for the early detection of at-risk students. This will benefit universities and enhance their prediction performance, improving both effectiveness and reputation.

Keywords: Student performance, Graph representation, Students at risk, Graph topological feature, Graph embedding, Graph convolutional network

Introduction

Recent technological developments have had significant impacts on education (Chen et al., 2020; Rodríguez-Hernández et al., 2021) resulting in the development of several online learning platforms such as Tutee, Intelligent Tutor, and Learning Partner (Hwang et al. 2020). These technology-assisted educational tools provide a new paradigm for the education field, offering the potential to monitor students' educational progress and predict their performance. The countries of the Organization for Economic Co-operation and Development have reported an alarming average dropout ratio of approximately 33% for undergraduate students (Ettorre et al. 2022). Such a considerable student dropout ratio leads to significant economic losses. Despite the recent development of many technology-assisted educational platforms, many higher education institutions are suffering from poor student performance (Barbosa Manhães et al., 2015). To address the issue of student dropouts, many automated systems have been developed to predict students' academic performance in higher education institutions (Ahadi et al., 2015; Rastrollo-Guerrero et al., 2020). Predicting student performance in the early stages of a course, particularly for at-risk students, can help the students, instructors, policymakers, and institutes in multiple ways. In particular, this enables the identification of students who will struggle to pass their courses and are at risk of dropping out (Albreiki et al., 2021b). Furthermore, predicting student performance with adequate accuracy can also help with the selection of students to receive grants and scholarships (Rodríguez-Hernández et al., 2021).

To carry out such experiments, several methodological approaches for predicting student academic performance have been developed, including correlation, regression, structural equation modeling, and analysis of variance (Albreiki et al., 2021a; Golino & Gomes, 2014). These traditional methods are based on certain assumptions, such as the independence of observations. However, such assumptions are not fulfilled by the actual data. The violation of these assumptions leads to type 1 and type 2 errors in predicting students' academic performance leading towards biased results (Nimon, 2012).

Recently, artificial intelligence-based techniques, including ML, have achieved promising results in predictive tasks across various fields. Several methods have been developed for predicting student academic performance based upon ML techniques such as artificial neural networks and support vector machines (Ahmad & Shahzadi, 2018; Lau et al., 2019). Existing approaches, however, have been observed to lack accuracy in predicting students' academic performance due to their dependency on multiple factors, both academic and non-academic (Shahiri & Husain, 2015).

Systems based on ML techniques, on one hand, have resulted in significantly improved predictions of students' academic performance. However, many critical challenges exist in developing workable ML methods for this task because of the complex nature of the enormous amounts of real-world data available from different technology-assisted learning platforms. The development of high-accuracy student academic performance prediction systems therefore remains a challenging task, requiring solutions to issues concerning data quality, quantity, and complexity.

To delve deeper, knowledge graph algorithms can help improve the classification and prediction performance by incorporating basic knowledge about the data.

Knowledge graph methods mainly focus on data correlation and can discover topological features that provide insights into interconnected data instances.

Novak (1991) introduced the idea of knowledge graphs as concept maps. Later, he used this framework to organize and interconnecting collected knowledge with existing knowledge. Many researchers have used maps to assess technology-based learning platforms (Trumpower et al. 2014; Valsamidis et al., 2012). This approach has been extended to enhance the underlying structure, resulting in a knowledge graph. While it is behind the scope of this paper to discuss the knowledge graph itself in detail, however a knowledge graph represents a network of entities and demonstrates the associations between them. The association-related information is visualized as a graph structure known as a knowledge graph. There are three main components of a knowledge graph: nodes, edges, and labels. A node represents a logical or physical entity. The association between nodes is represented by edges. Knowledge graph models can be applied in different domains, through generative graph models, knowledge graph construction/inference (Zaki et al., 2021), or network embedding.

From the studies mentioned above, it is clear that most of the current research is primarily concerned with employing basic features and ML approaches to predict students' performance with reasonable accuracy. ML and deep learning (DL) approaches with massive amounts of student data from various technology-assisted educational platforms provide acceptable predictions results of students' academic achievement and identification of at-risk students. However, these approaches currently struggle to extract additional useful features that can understand complex data structures and reflect connections among students' features. Knowledge graphs can uncover feature correlations and be used with promising ML and DL algorithms to predict and enhance student academic achievement. These techniques can be combined to detect at-risk students with better prediction results.

This study proposes a hybrid approach based on knowledge graphs and ML for predicting student academic performance, particularly for at-risk students. From the above discussion, it evident that the traditional method have many flaws and are not well suite for the state of the art students' performance evaluation. Furthermore, the ML techniques where provides a promising results, comes with its own challenges. There seems to be a debate on which techniques works well, and what is needed to be done. In light of these, this study proposed a novel method to extract topological features, by combing the ML and GCN model to predict and identify the poor performing or at risk students. The remainder of this paper is organized as follows. Section "Literature review" presents a literature review and identifies several gaps in existing research. Section "Methodology" states the research objectives and explains the methodology of the proposed approach, including descriptions of the dataset, knowledge representation, feature extraction, and research design. Section "Experimental results" presents the results given by the various experimental settings, before "Discussion and future work" section provides a comprehensive discussion of the results and a comparative analysis with state-of-the-art methods. Finally, "Conclusions" section concludes the paper and identifies areas for future work.

Literature review

The modern online learning platforms have resulted in massive amounts of educational data. The collected data can be analyzed to resolve critical issues in the educational field, such as the student dropout ratio (Mubarak et al., 2022), improved learning platforms Ferguson (2012), and the tracking of students' academic performance. Several efforts have been made in this direction, including the development of a course recommendation system (Liu et al., 2019), student behavior prediction system (Mubarak et al., 2021), user intention analysis system (Zhang et al., 2017), at-risk student prediction system (Mubarak et al., 2020), and knowledge tracing (Yang et al. 2020).

Recently, ML and data mining techniques have been successfully applied to the prediction of students' performance in higher education (Aleem & Gore, 2020). These techniques play a vital role in identifying many trends in educational data related to student performance and the teaching–learning process. Aleem and Gore (2020) concluded that there is no single technique that can meet all requirements of the educational system, such as predicting students' academic performance. Instead, limited technologies exist that can be integrated with current e-learning platforms to help students, instructors, and institutions to assess students' performance, particularly at-risk students. Yadav et al. (2012) analyzed ML-based predictive models for student retention assessment. They concluded that decision trees provide better performance and more interpretable output in understanding student retention in educational institutes. Their experimental results demonstrate the effectiveness of ML-based predictive models for predicting student retention with adequate accuracy and identifying at-risk students. Predictive models thus improve the student dropout ratio in educational institutes. However, the dataset was relatively small and geographically restricted to a region in India. The dataset comprises 432 participants and it has been taken from the institute records from 1997 to 2012 with a gap between 2000 to 2009. Such data may not indicate some viable features that may correctly predict student performance. The paper does not explain the full technique in adequate detail.

Kolo and Adepoju (2015) used decision trees and ordinal regression approaches using IBM Statistical Package for Social Studies. Mainly the prediction of either a participant will "pass" or "fail is highlighted in this work. To do so, features that were considered are financial status, pass fails status, motivation to learn, and gender. The focus of the work is also to identify factors that affect the performance of the students. The data set is quite small i.e., three years data of second-year data structure course participants. As in the case of Kolo and Adepoju (2015), the dataset is geographically restricted to Nigerian colleges of education.

Similarly, Dhanalakshmi et al. (2016) has explored opinion mining using supervised learning algorithms. ML and Natural language processors are the focus of the research work. The author has mainly provided a comparative analysis of SVM, Naive Bayes, K nearest neighbor and Neural network classifier for binomial classifications and to find the polarity of the students from their provided feedback. The data set here used is from Middle east college, Oman, and is not diverse in nature. Furthermore, it is not clear from the results if the students' progress can be predicted.

The study of Mesarić and Sebalj (2016) involved dividing students into two groups based on their academic performance in high school and during the first year of their current studies. Decision trees were created using various algorithms, and the most effective one was selected based on its classification rate and statistical significance. While the REPTree algorithm had the highest classification rate, it was not as successful at accurately categorizing students from both groups. The most influential factors in the study were the total points earned on a state exam, points earned in high school, and points earned on a Croatian language exam.

The current ML approaches suggest that there is a positive connection between student engagement and performance. The recent work of Moubayed et al. (2020), personalizes the learning experience and strives to engage the students and keep them motivated in order to retain them. The k-means algorithm is used to cluster students based on twelve engagement metrics. Furthermore, two main categories are established which are based on the interaction of the students and their respective efforts. Qualitative analysis is performed in order to identify those students who may need help, or otherwise, they may be considered at risk of dropout.

With the availability of massive data from online learning platforms, graph-based techniques are a promising method of analyzing data structures and identifying correlations in terms of nodes and edges. Here, the nodes define unlabeled and labeled data sets, whereas edges represent the similarity between various nodes (Zha et al., 2009).

In order to cope with the state-of-the-art learning systems, where a massive number of students are enrolled from different parts of the world and the learning experience is different than the conventional method. There is a need for a more advanced and robust system to be deployed. Several researchers have used DL methods to predict student performance and dropout ratios, Karimi et al. (2020) has highlighted the low completion rate of online courses and also relates it to the unconventional teacher-to-student relation, especially in evaluation. They proposed a Deep Online Performance Evaluation (DOPE) and make use of knowledge graphs and advanced graph neural networks to predict the students' performance for each course. The dataset has total participants of 32,953 and each course ranges from 35–40 weeks long. The system uses the features such as distinction, pass, fail, and withdrawn and translate it into their respective interpretation of the system. In the same direction Fei and Yeung (2015) used the long short-term memory model to extract relevant features from a student questionnaire, video lectures, and problems. They used a very reliable dataset from Coursera[1] and Edx.[2] The research work mainly deals with dropout prediction or identifying students at risk of being dropped out. Dropout prediction has been approached using simple ML methods like support vector machines and logistic regression, which use features related to student activities such as watching lecture videos and participating in forum discussions on a massive online open course (MOOC) platform. However, temporal models using recurrent neural networks with long short-term memory cells have been found to be more effective in predicting dropout, based on experiments conducted on MOOCs offered on Coursera and edX. These results outperformed both baseline methods and other proposed methods by a significant margin. Whitehill et al. (2017) also proposed a neural network system. They strongly suggest that in order to get a fair result the

[1] coursera.com.

[2] edx.org.

system should be trained on one dataset and tested on a different one. Training and testing the system on the same dataset may increase the accuracy by several percentages. Their dataset is based on 528,349 participants from HarvardX. Feng et al. (2019) proposes Context-aware Feature Interaction Network (CFIN) to model and to predict users' dropout behavior. Experiments on two large datasets show that the proposed method achieves better performance than several state-of-the-art methods. The proposed method model has been deployed on a real system to help improve user retention. The dataset is closer to 700,000 enrollments.

DL representations of data samples have been widely used with knowledge graphs in recent years. A knowledge graph mainly focuses on entities and their associations, as represented in the form of a graph. There has been significant progress in the knowledge graph area specifically, which predicts the strong research interests in the subject area, as highlighted in Luo and Fang (2018) and Lin et al. (2015). Knowledge graphs learn embedded information that can be used in different applications such as association extraction, similarity computation, and link prediction.

Many researchers have integrated DL techniques with knowledge graphs to improve model predictions and classification effectiveness. Knowledge graphs can be integrated with DL methods in two ways. The first is to integrate the semantic information extracted from the graph into DL and ML. In this way, a discrete knowledge graph, represented as a continuous vector of expert knowledge, is applied to the DL method.

Previous reports on the application of knowledge graphs and DL to predict student academic performance, including at-risk students. Another such example includes the work of Gaur et al. (2021), this article discusses how knowledge, represented in a knowledge graph, can be integrated into DL methods using a strategy called knowledge-infused learning. The use of this approach is demonstrated through a case study in the field of education. By incorporating domain knowledge and a student's historical knowledge state into the model, it becomes possible to trace academic weaknesses back to their root causes, including any underlying pre-requisite concepts that may be impacting a student's performance.

Knowledge graphs have been successfully employed in MOOC platforms (Zheng et al., 2017). They have been used in different education-related domains, teaching and classroom resources, education management, and educational technologies. In terms of teaching and classroom resources, the K12EduKG system has been developed based on knowledge graphs using the K-12 educational subjects (Chen et al., 2018). This system is based on specific educational knowledge from the Chinese mathematics curriculum. The developers of K12EduKG identified knowledge concepts and associations based on probabilistic association rules and a conditional random field model. Su and Zhang (2020) suggested a knowledge graph-based method for accommodating educational big data. Their knowledge graph incorporated an online encyclopedia and subject teaching resources. Similarly, Zhao et al. (2019) used knowledge graphs to build a system called mathGraph. This was initially developed through crowdsourcing to consider dissimilar mathematical objects and their operations. It can be concluded that the combination of ML and DL methods, along with access to large amounts of student data from educational technology platforms, enables accurate predictions of academic performance and identification of at-risk students. However, current approaches have difficulty

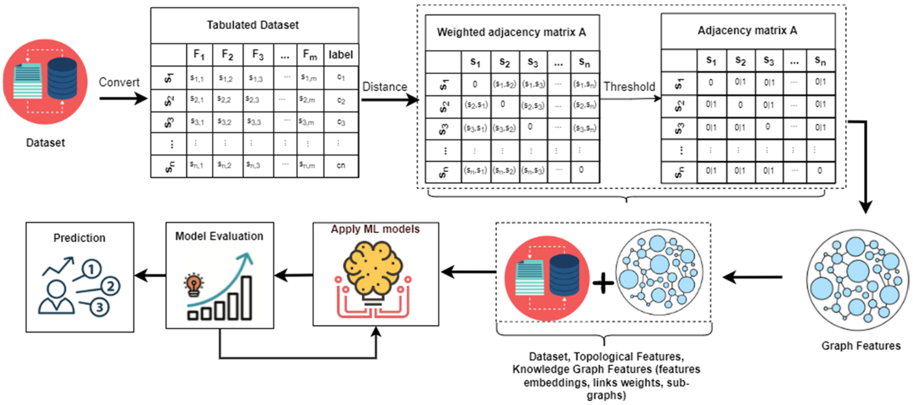


Fig. 1 Overview of the suggested knowledge graph methodology

extracting relevant features that can understand complex data structures and represent correlations between them. Knowledge graphs, on the other hand, are able to effectively extract these correlations and can be integrated with ML and DL methods to improve performance prediction. Although there has been limited research on using advanced techniques such as ML, DL, and knowledge graphs to identify and predict the academic performance of at-risk students, the potential benefits of integrating these approaches make it a promising avenue for further study.

Methodology

Research objectives

This study aims to integrate graph theory with a prediction system to improve the accuracy of students' performance predictions and help identify hidden structures and similarities between different student behaviors. It is anticipated that the proposed solution will be of benefit to universities, as it will enable them to accurately predict performance and subsequently implement remedial plans to address the factors associated with low performance and dropout, thereby maintaining their reputation for delivering academic excellence. To address the main goal, we formulate the following tasks:

- Incorporate structural correlations between students and extract additional features by converting the tabulated data to graph data. This will allow graph features to be extracted and combined with the original features to improve the ML classification results.
- Combine the extracted graph features with a GCN to identify at-risk students.
- Implement graph embedding methods to represent entities, and develop ML models to identify at-risk students with better accuracy.

Research design

The proposed method is designed to identify at-risk students using knowledge graphs and conventional ML methods, as illustrated in Fig. 1.

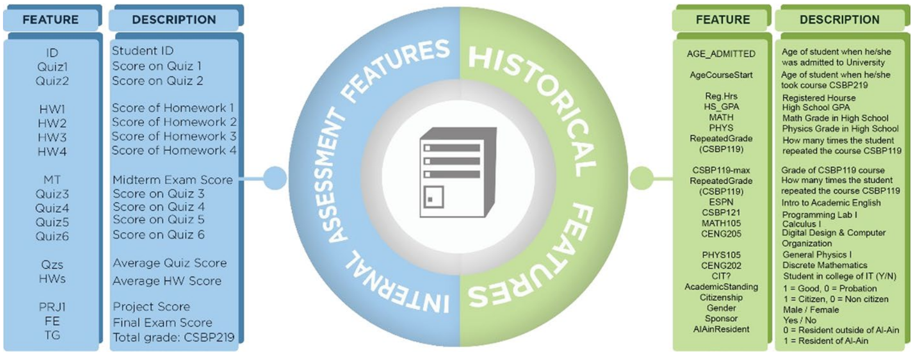
For the first objective of our framework, we explore possibilities for converting tabulated data into a graph as it is suggested by Zaki et al. (2021). We start by building an adjacency matrix for a graph considering students as nodes. First, we explore the distance norms that can be used to represent the link between data points (students). Different distance metrics or norms, such as Euclidean, Manhattan, cosine, correlation, and Chebyshev, are used to calculate the similarity between the data points and generate the graph's edge weights. We extract the new graph topological features (GF) from the preprocessed dataset (see Table 4) and add them to the original dataset (DS). From DS, we use all the features collected prior to the midterm exam (MT). Next, we employ advanced ML techniques to the combined DS & GF features to identify at-risk students as early as possible. Students are classified as Good, At-Risk, and Failed based on their total grades in the course. We employ multiclassification prediction models using state-of-the-art ML methods such as the XGB classifier, LightGBM, SVM linear, ExtraTrees, Random Forest, and multilayer perceptron. Five-fold cross-validation is used to generalize the true error rate at the population level. Feature selection methods are applied to find the most informative features based on the model output. Finally, the performance of the proposed models is compared using different sets of input features (historical data only, mixed data including historical data and performance in course) with and without the newly calculated topological features.

For the second objective, we combine the extracted graph features with the original features and use a GCN to identify at-risk students. The GCN is a type of convolutional neural network that can work directly on graphs and take advantage of their structural information. The core of the GCN model is the graph convolution layer. This layer is similar to a conventional dense layer, augmented by the graph adjacency matrix A to use node information, such as the topological information created in the first step. In conjunction with the GCN, graph topological features are expected to provide more meaningful information, leading to highly accurate node classification.

For the third objective, we construct a knowledge graph from the dataset and extract the graph embedded features to further improve the performance of the classification models. First, we propose (subject, predicate, object) tuples to create a semantic network of the students. From the knowledge graph, we extract the complex relations among features and efficiently store the entities and their relationships. We utilize the Neo4j database to store and process the created knowledge graph. To improve the classification accuracy, we then generate embedded vectors from the graph nodes using the node2vec algorithm. Finally, we combine the extracted embedded vectors with the original and topological features extracted earlier, and feed them to the ML classification models.

Data collection and data preprocessing

For this study, educational data were collected from a variety of sources, including the Banner system, which contains information about students, instructors who taught programming courses, and documents manually extracted from the Ministry of Education's portal. The main data used in this study are related to programming courses offered at the College of Information Technology (CIT) at United Arab Emirates University (UAEU). These courses are a requirement for graduation at the university for CIT students. The courses may be taken as electives by students from other colleges. The data presented here represent student performance in programming courses (fall and spring) from academic year 2016/2017 until 2020/2021. We added demographics, course



FEATURE	DESCRIPTION
ID	Student ID
Quiz1	Score on Quiz 1
Quiz2	Score on Quiz 2
HW1	Score of Homework 1
HW2	Score of Homework 2
HW3	Score of Homework 3
HW4	Score of Homework 4
MT	Midterm Exam Score
Quiz3	Score on Quiz 3
Quiz4	Score on Quiz 4
Quiz5	Score on Quiz 5
Quiz6	Score on Quiz 6
Qzs	Average Quiz Score
HWs	Average HW Score
PRJ1	Project Score
FE	Final Exam Score
TG	Total grade: CSBP219

FEATURE	DESCRIPTION
AGE_ADMITTED	Age of student when he/she was admitted to University
AgeCourseStart	Age of student when he/she took course CSBP219
Reg.Hrs	Registered Hourse
HS_GPA	High School GPA
MATH	Math Grade in High School
PHYS	Physics Grade in High School
RepeatedGrade (CSBP119)	How many times the student repeated the course CSBP119
CSBP119-max	Grade of CSBP119 course
RepeatedGrade (CSBP119)	How many times the student repeated the course CSBP119
ESPN	Intro to Academic English
CSBP121	Programming Lab I
MATH105	Calculus I
CENG205	Digital Design & Computer Organization
PHYS105	General Physics I
CENG202	Discrete Mathematics
CIT?	Student in college of IT (Y/N)
AcademicStanding	1 = Good, 0 = Probation
Citizenship	1 = Citizen, 0 = Non citizen
Gender	Male / Female
Sponsor	Yes / No
AlAinResident	0 = Resident outside of Al-Ain 1 = Resident of Al-Ain

Fig. 2 Features description

Table 1 Summary of the datasets used

	D1	D2	D3
Dataset (size)	218	230	201
Attributes	Checkpoints	Checkpoints & Historical Features	Checkpoints
Target performance in	TG	TG	TG
Classification	Multiclassification: Good/At-Risk/Failed	Multiclassification: Good/At-Risk/Failed	Multiclassification: Good/At-Risk/Failed

TG: total grade

registration, and campus information to the data. Before data analysis and classification, there were 730 records and 44 features in the original dataset. After removing inconsistent rows and features using univariate feature processing, the final dataset consisted of 649 samples and 38 features. The courses were not directed or specially designed for the experiments described in this paper. We constructed different non-overlapping datasets based on the features of the data. One dataset used in this study includes records of 230 students enrolled in "Object-oriented programming." We collected data on students' prior performance and demographics, as well as their enrollment. Figure 2 shows the features of the dataset.

The data preprocessing included six phases. First, the course assessment files, student data (from Banner), and manually extracted documents were synthesized. Second, the compiled data were cleaned to remove any unnecessary entries. Third, the data were homogenized (structure unification) to remove inconsistencies in file structures due to different instructors teaching courses. In the next step, missing data values were treated using an imputation technique in which the average values of coursework components were assigned to missing entries. Following data aggregation, it was necessary to apply standardization to convert the categorical data to numerical values, integrate the data into the same CSV file, and normalize the data by applying min-max normalization (rescaling to [0, 1]). As a final step, we added a column based on rules and significant milestones in student performance. Using their total grade (TG) performance, we divided the students into three main categories: Good ($TG \geq 70\%$), At-Risk ($60\% < TG < 70\%$), and Failed ($TG \leq 60\%$). To ensure the generality of our model, we used other datasets to validate our findings. Dataset details can be found in Table 1.

Table 2 Structure of the educational datasets $D_i, i = \overline{1,3}$

	F_1	F_2	F_3	$\cdots$	F_m	**Category**
$Student_1$	$s_{1,1}$	$s_{1,2}$	$s_{1,3}$	$\cdots$	$s_{1,m}$	g_1
$Student_2$	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$	$\cdots$	$s_{2,m}$	g_2
$Student_3$	$s_{3,1}$	$s_{3,2}$	$s_{3,3}$	$\cdots$	$s_{3,m}$	g_3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$Student_n$	$s_{n,1}$	$s_{n,2}$	$s_{n,3}$	$\cdots$	$s_{n,m}$	g_n

Table 3 Distance norms employed to convert dataset to graph representation

Norm	Formula	Description
Chebyshev	$d_{i,j} = \max_p \lvert s_{i,p} - s_{j,p} \rvert$	Metric induced by the supremum norm
Euclidean	$d_{i,j} = \sqrt{\sum_{p=1}^{m} (s_{i,p} - s_{j,p})^2}$	Length of a line segment between two vectors
Manhattan	$d_{i,j} = \sum_{p=1}^{m} \lvert s_{i,p} - s_{j,p} \rvert$	Sum of the lengths of the projections onto axes
Correlation	$d_{i,j} = 1 - \frac{(s_i - \overline{s_i}) \cdot (s_j - \overline{s_j})}{\sqrt{\sum_{p=1}^{m} s_{i,p}^2 \cdot \sum_{p=1}^{m} s_{i,p}^2}}$	Correlation distance between two vectors
Cosine	$d_{i,j} = 1 - \frac{(s_i, s_j)}{\sqrt{\sum_{p=1}^{m} s_{i,p}^2 \cdot \sum_{p=1}^{m} s_{i,p}^2}}$	Cosine distance between two vectors

$(s_i, s_j) = \sum_{p=1}^{m} s_{i,p} \cdot s_{j,p}$

$\overline{s_i}$ is the mean of all elements in s_i

Knowledge and graph representation

Knowledge graphs can be developed automatically using ML and graph mining techniques, eliciting new insights into a particular area. Knowledge graphs reveal information in structures and eliminate data abstraction, making it easier to understand a given field. This research presents a simple, but extremely accurate, way of converting tabulated data into graphs and graph features, allowing significant improvements in ML classification. The graph features can help identify hidden structures between different student behaviors that are hidden to standard classification algorithms.

Let us consider a dataset D that consists of n students $s_i, i = \overline{1,n}$. Every student st_i is represented by m attributes F_i, $A_i \in$ {checkpoints, historical features}, $i = \overline{1,m}$ (see Fig. 2 for more details). All attributes are defined and can be represented as continuous or categorical values, as listed in Table 2. Students are graded according to the course or program requirements. We use the total grade in the course to categorize students as Good, At-Risk, or Failed. Therefore, $g_i \in C$, C={Good, At-Risk, Failed}, $i = \overline{1,n}$.

The data representation (features) input to ML algorithms have a significant impact on the classification performance. Additional features may improve the model's outcome. We propose to capture the topological relations between students by using a graph representation of the data. We consider our data as a set of points in m-dimensional space. A measure of proximity for any two data points s_i and s_j is the distance between them, calculated by one of the formulae listed in Table 3. Using these formulae, we can convert the dataset into a graph by creating an adjacency matrix. In this instance, the adjacency matrix $A = \{a_{i,j} : a_{i,j} = d_{i,j} \in \mathbb{R}, i,j = \overline{1,n}\}$ is representative of a weighted graph G having n nodes. To create an unweighted graph, we clip the adjacency matrix cells with the threshold $threshold_i = \frac{1}{n} \sum_{j=1}^{n} \dot{a}_{i,j}$.

Table 4 Definitions of topological features in graph theory

	Topological feature	Definition
1	Center	Node with eccentricity equal to the radius
2	Density	Defined as $d = \frac{2N_E}{N_v(N_v-1)}$ for an undirected graph, where N_E and N_v are the number of edges and nodes in a graph *G*
3	Radius	Minimum eccentricity of a graph
4	Diameter	Maximum eccentricity of a graph
5	Periphery	Periphery is a subgraph with eccentricity equal to the diameter of *G*
6	Triangle	Number of triangles having a node *v* as one vertex
7	Transitivity	Fraction of all possible triangles present in *G*
8	Degrees	Nnumber of edges connected to the node
9	In degree	Number of head ends adjacent to a node
10	Out degree	Number of tail ends adjacent to a node
11	Weighted degree (Newman, 2001)	Summation of edges connected to the node
12	Eccentricity (Harary & Norman, 1953)	Maximum distance from node *v* to all other nodes in *G*
13	Hub (Kleinberg et al., 2011)	Number of highly authoritative nodes a node *v* is pointing to
14	Authority (Kleinberg et al., 2011)	Amount of valuable information that a node *v* carries
15	PageRank (Page et al., (1999)	Importance of a node *v* in the graph *G*
16	Closeness centrality (Sabidussi, 1966)	Time it takes to move from node *v* to other nodes in the graph *G*
17	Betweenness centrality (Brandes, 2001)	Sum of the fraction of all-pairs shortest paths that pass through a node *v*
18	Information centrality (Brandes & Fleischer, 2005)	Current-flow closeness centrality based on effective resistance between nodes in a network
19	Harmonic centrality (Marchiori & Latora, 2000)	Sum of the reciprocal of the shortest path distances from node *v* to all other nodes in *G*
20	Eigenvector centrality (Bonacich, 1987)	Connectivity or transitive influence of a node *v*

We propose an approach that uses topological features extracted from a graph representation of the dataset. These topological features can be integrated with original features to enhance the useful graph information and find correlations between instances. Table 4 presents definitions of the topological features used in this research. These features are employed with the ML classification models to improve the accuracy of the classifier.

Employing graph convolutional network

Recently, graph-based DL models, particularly GCNs, have achieved excellent performance compared with several ML-based methodologies in solving complex problems (Zhang et al., 2019). Moreover, GCNs have attained a novel ability to understand graph-based representations and provide robust performance in many complex and unsolved problems (Zhang et al., 2019). This study aims to utilize the power of GCNs with the combination of graph-based topological features to conduct node classification. According to the training idea of Kipf and Welling (2016), the objective of a GCN is to learn the features of a graph $G = (V, E)$ from a description of the graph structure as an adjacency matrix A and a feature descriptor f_v for each node v, as summarized in a feature matrix S.

Using the above set of inputs, the GCN produces a unique output Y, which is a $l \times o$ feature matrix in which o is the number of output features per node, $Y = \{y_{a,b} : z_{a,b} \in \mathbb{R}, a = \overline{1,l}, b = \overline{1,o}\})$. The graph-level outputs can be enhanced by adding pooling layers (Duvenaud et al., 2015). Thus, each of the neural network layers can be described as a nonlinear function of the form

$$P^{(q+1)} = f(P^{(q)}, A), q = \overline{0,Q}, \tag{1}$$

where $P^{(0)} = S$, $P^{(Q)} = Y$, and Q is the number of layers. Thus, models differ only in terms of the chosen $f(\cdot,\cdot)$ and its parameters. Hence, the GCN is exploited to model the binary classifiers. The (processed) graph and the relevant features are used to assemble the adjacency matrix A, feature matrix S, and degree matrix Φ Kipf and Welling (2016). Moreover, an identity matrix I is included in A to establish the self-connections $\tilde{A} = A + I$. Then, the output matrix is normalized by exploiting the degree matrix Φ as follows: $\widehat{A} = \sqrt{\Phi}\tilde{A}\sqrt{\Phi}$. The degree matrix can be described as $\Phi = \{\phi_{i,i} = \sum_{j=1}^{n} \tilde{a}_{i,j}\}$. Thus, the input to the GCN model consists of the feature matrix S and the normalized adjacency matrix $\widehat{A}$. In this research, we use a four-layer GCN model with weight matrices W_1, W_2, W_3, W_4. At the start of the training, the weight matrices are initialized with random values between 0 and 1. During the training, these weight matrices are optimized using a backpropagation-based error correction algorithm (the Adam optimization function). Hence, the output of our proposed GCN model can be described as follows:

$$GCN_{(W_1,W_2,W_3,W_4)}(\widehat{A}, S) = \sigma(\widehat{A}\varphi(\widehat{A}\varphi(\widehat{A}\varphi(\widehat{A}SW_1)W_2)W_3)W_4), \tag{2}$$

where $\varphi(\cdot)$ is the ReLU activation function and $\sigma(\cdot)$ is the softmax activation function. The final layer provides the prediction for each node. At the end of every GCN layer, the dropout operation is applied with a rate of 0.3.

Evaluation measures

To assess the quality of the outcomes given by the classification methods, we calculated the sensitivity, specificity, area under the receiver operating characteristic (ROC) curve (AUC), accuracy, and balanced accuracy metrics. A confusion or error matrix was constructed for each predictive model to show how well it distinguished between classes. The ROC curve and its AUC were used to evaluate the performance of the classifiers and summarize the trade-off between the true positive rate (TPR) and false positive rate (FPR) using different probability thresholds. We define

$$TPR\,(sensitivity) = \frac{TP}{TP + FN}, \tag{3}$$

$$TNR\,(specificity) = \frac{TN}{TN + FP}. \tag{4}$$

The overall accuracy of the model is defined as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{5}$$

where *TP*, *TN*, *FP*, *FN* are the numbers of true positives, true negatives, false positives, and false negatives representing the confusion matrix of the classification model, respectively. All models were trained using *k*-fold cross-validation. The metrics were calculated for each fold separately, and then the averaged values were used as the final measure.

Experimental results

This section describes experimental work undertaken to improve the classification accuracy by converting tabulated data structures into graph data structures. The use of graphs is expected to capture more significant correlations between instances, which are typically ignored in classification. We evaluate the benefits of adding graph-related features to the original features using conventional ML methods, a GCN model, and knowledge graph embeddings. We conducted experiments with the initial dataset features (*DS*) only, and then with both *DS* and the graph topological features *GF*. These combined features were ensembled with the GCN to achieve superior results. Finally, the graph embeddings were exploited to achieve state-of-the-art performance. The results are described in detail below. In summary, the proposed method of adding graph-related features to the original features produces better results than the current state-of-the-art ML classification using only the original features in the dataset.

ML classification model that integrates graph features with original features to improve performance

We used five different norms to calculate adjacency matrices for the graphs and extract graph topological features (see Table 3). The extracted features were combined with features from the original dataset (*DS*) and fed to the classification models. The best performance was obtained using the Euclidean and cosine norms. These two metrics were therefore used for further analysis. Table 5 presents the results given by the multi-classification ML methods fed with the original dataset features only and with the graph topological features (GF) combined with DS. We employed six different classifiers to the features from datasets *D*1, *D*2, *D*3. The mean accuracy of dataset *D*1 increased from 73.0% to 76.7% under the Euclidean metric and from 73.0% to 74.8% under the cosine metric. Similarly, the mean AUC for *D*1 increased from 0.894 to 0.913 and to 0.898 for the Euclidean and cosine metrics, respectively. For dataset *D*2, the mean accuracy increased from 75.8% to 79.5% and to 82.5% under the Euclidean and cosine metrics, respectively, while the mean AUC increased from 0.907 to 0.925 and to 0.942, respectively. The accuracy of dataset *D*3 remained the same at 88.8% under the Euclidean norm, but increased to 91.0% when using the cosine distance metric. Finally, the mean AUC for *D*3 decreased from 0.962 to 0.960 with the Euclidean norm and increased from 0.962 to 0.966 with the cosine norm. Hence, the prediction performance typically improves with the addition of GF to the dataset; the exception is for dataset *D*3 with the Euclidean distance metric.

For the next experiments, we focused on dataset *D*2, as this contained historical and checkpoint features. When sufficient information about the students is included, the

Table 5 Classification performance when employing *DS* only and combining *DS* and *GF* for datasets *D*1, *D*2, and *D*3

	ML Model	**Accuracy**					**AUC**				
		DS	**Euclidean**		**Cosine**		**DS**	**Euclidean**		**Cosine**	
			+GF	Δ_{ACC}	**+GF**	Δ_{ACC}		**+GF**	Δ_{AUC}	**+GF**	Δ_{AUC}
D1	XGB	0.780	0.840	7.692	0.800	2.564	0.915	0.932	1.930	0.933	1.959
	Light GBM	0.780	0.800	2.564	0.810	3.846	0.914	0.928	1.527	0.934	2.160
	SVM linear	0.690	0.660	− 4.348	0.650	− 5.797	0.840	0.861	2.469	0.813	− 3.273
	Extra Trees	0.790	0.820	3.797	0.770	− 2.532	0.929	0.950	2.252	0.962	3.523
	Bagging	0.630	0.670	6.349	0.670	6.349	0.838	0.853	1.769	0.830	− 1.039
	Random Forest	0.710	0.810	14.085	0.790	11.268	0.926	0.955	3.138	0.915	− 1.183
	Mean	0.730	0.767	5.023	0.748	2.616	0.894	0.913	2.181	0.898	0.358
D2	XGB	0.780	0.860	10.256	0.860	10.256	0.895	0.950	6.232	0.966	7.976
	Light GBM	0.790	0.860	8.861	0.840	6.329	0.915	0.935	2.176	0.959	4.740
	SVM linear	0.660	0.640	− 3.030	0.760	15.152	0.866	0.849	− 1.972	0.888	2.537
	Extra Trees	0.810	0.870	7.407	0.890	9.877	0.938	0.970	3.360	0.971	3.541
	Bagging	0.730	0.700	− 4.110	0.730	0.000	0.899	0.882	− 1.934	0.897	− 0.270
	Random Forest	0.780	0.840	7.692	0.870	11.538	0.927	0.963	3.828	0.972	4.822
	Mean	0.758	0.795	4.513	0.825	8.859	0.907	0.925	1.948	0.942	3.891
D3	XGB	0.900	0.920	2.222	0.950	5.556	0.951	0.979	2.967	0.978	2.800
	LightGBM	0.920	0.910	− 1.087	0.970	5.435	0.978	0.975	− 0.339	0.985	0.665
	SVM linear	0.780	0.800	2.564	0.790	1.282	0.917	0.907	− 1.052	0.917	0.036
	Extra Trees	0.960	0.920	− 4.167	0.950	− 1.042	0.995	0.980	− 1.548	0.990	− 0.530
	Bagging	0.850	0.810	− 4.706	0.860	1.176	0.940	0.945	0.531	0.944	0.477
	Random Forest	0.920	0.920	0.000	0.940	2.174	0.989	0.973	− 1.680	0.983	− 0.648
	Mean	0.888	0.880	− 0.862	0.910	2.430	0.962	0.960	− 0.187	0.966	0.467

DS: initial dataset; GF: graph topological features

Δ shows the boost in performance achieved by adding GF to DS

prediction results are expected to improve. As a result, Table 6 describes the classification results when historical features are added to *D*2. That is, *D*2 now consists of

- checkpoints,
- historical features,
- graph topological features (see Table 4).

From Table 6, the mean accuracy improves overall, even in the case where only *DS* is used, i.e., 84.5% accuracy. Moreover, the accuracy of $DS + GF$ increases from 84.5% to 87.3% for the cosine metric and from 84.5% to 84.7% for the Euclidean metric. When *DS* is combined with *GF*, the mean AUC increases from 0.948 to 0.957 for the cosine metric and from 0.948 to 0.967 for the Euclidean metric. Thus, the overall result of adding *GF* to *DS* with historical checkpoints is a significant improvement in classification accuracy.

Improving the classification model performance using a GCN

The next set of experiments used the *D*2 dataset with historical features included and *GF* added. We employed a GCN to evaluate the classification performance. The overall prediction results improve significantly. Figure 3 shows that, with the GCN model, the accuracy increases from 87.3% to 88.2% for the cosine metric and from 84.7% to 85%

Table 6 Classification performance when employing initial D2 dataset and combining it with topological features using cosine and Euclidean metrics

	ML Model	Accuracy			AUC		
		DS	+GF	Δ_{ACC}, %	DS	+GF	Δ_{AUC}, %
Cosine	XGB	0.920	0.924	0.435	0.961	0.979	1.873
	Light GBM	0.890	0.924	3.820	0.967	0.983	1.655
	SVM linear	0.760	0.773	1.711	0.909	0.922	1.430
	Extra Trees	0.870	0.950	9.195	0.980	0.995	1.531
	Random Forest	0.870	0.874	0.460	0.965	0.981	1.658
	MLP	0.760	0.790	3.947	0.907	0.943	3.969
	Mean	0.845	0.873	3.261	0.948	0.967	2.019
Euclidean	XGBClassifier	0.920	0.874	− 5.000	0.961	0.963	0.208
	Light GBM	0.890	0.820	− 0.899	0.967	0.972	0.517
	SVM linear	0.760	0.798	5.000	0.909	0.928	2.090
	Extra Trees	0.870	0.882	1.379	0.980	0.975	-0.510
	Random Forest	0.870	0.891	2.414	0.965	0.983	1.865
	MLP	0.760	0.756	− 0.526	0.907	0.920	1.433
	Mean	0.845	0.847	0.395	0.948	0.957	0.934

DS: initial dataset; GF: graph topological features

Δ shows the boost in performance achieved by adding GF to DS

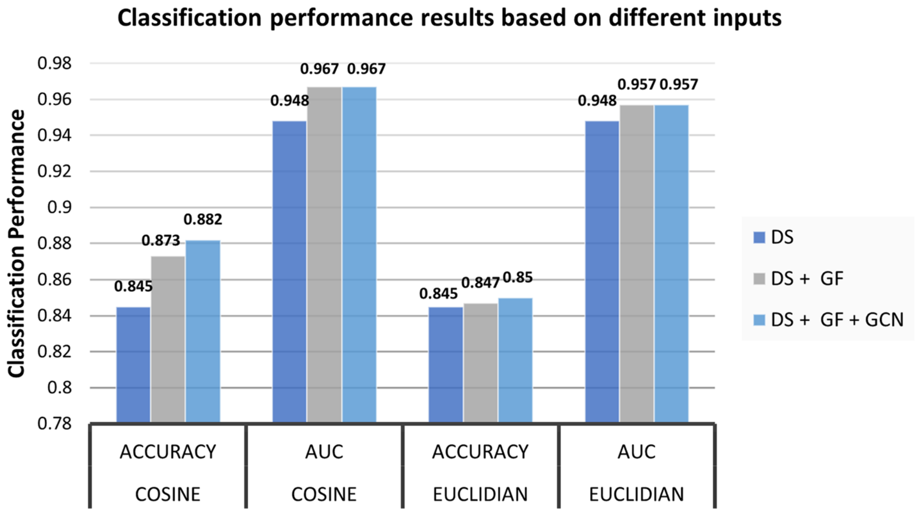


Fig. 3 Comparison of prediction results based on different inputs

for the Euclidean norm. Similarly, the mean AUC improves from 0.967 to 0.972 for the cosine norm; however, it remains the same at 0.957 for the Euclidean distance. Finally, Table 7 provides an overall comparison of the models. The GCN model trained on a combination of *DS* and *GF* features provides a strong classification model for predicting at-risk students at the early stages of a course.

Entity representations and graph embedded vectors

By extracting knowledge from the acquired data, a student-related knowledge graph can be constructed. The main task is to construct an abstract ontology. For this, we use student dataset *D*2, as it contains both historical and checkpoint features. In dataset *D*2, we

Table 7 Comparative analysis of classification models applied to data enhanced with graph concepts

Metrics	Data used	ACC	Δ_{ACC}, %	AUC	Δ_{AUC}, %
Original dataset	DS	0.845	Ref.	0.948	Ref.
	DS+GE	0.86	+1.775	0.976	+2.954
Cosine metric	DS+GF	0.873	+3.314	0.967	+2.004
	DS+GF+GE	0.874	+3.432	0.970	+2.321
	GCN (DS+GF)	0.882	+4.379	0.972	+2.532
Euclidean metric	DS + GF	0.847	+0.237	0.957	+0.949
	DS + GF + GE	0.863	+2.130	0.975	+2.848
	GCN (DS + GF)	0.850	+0.592	0.957	+0.949

Δ shows the boost in performance compared with original dataset (Ref.)

DS: initial dataset; GF: graph features; GE: graph embeddings

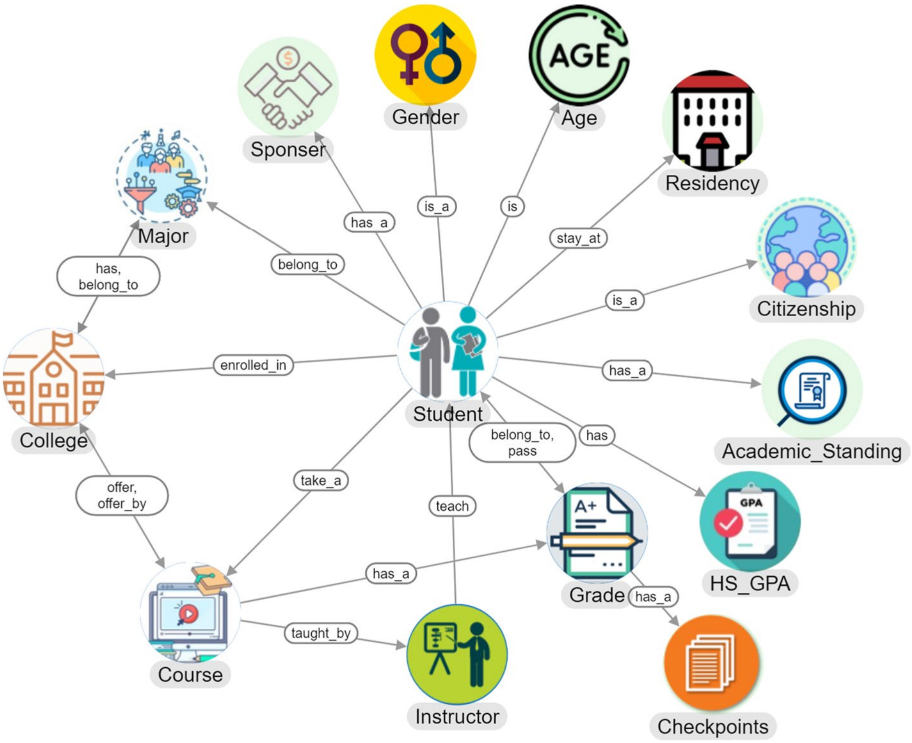


Fig. 4 Knowledge graph with entities and relationships

define 14 entities and 17 relationships. Table 8 defines each of these 14 types. The relationships between the entities are shown in Fig. 4.

The knowledge graph utilizes a graph structured data model or topology to integrate data. The general structure of the knowledge graph consists of a network of entities, their semantic types, properties, and relationships. In this research, the relationships among these entities are defined (i.e., a student is enrolled on a course, that student is taught by a particular instructor, has a particular high school GPA, and the total grade is linked to these checkpoints). The knowledge graph extracts the complex relations among features. To efficiently store the entities and their relationships, we utilize the Neo4j database.

Table 8 Entity descriptions

#	Entity type	Connection	Description
1	S	Student	Students enrolled in the course Object-Oriented Programming "CSPB219"
2	Ge	Gender	There is either a male or female student in the class
3	I	Instructor	The university instructor who taught the course
4	C	College	College of the student: Information Technology (IT), Business (BE), Science (SC), Engineering (EN), etc.
5	M	Major	There is a large range of majors, including Biochemistry, Information Security, Statistics, etc.
6	Cu	Course	During their degree program, students took different courses to fulfill the requirements
7	G	Grade	Each student receives a final grade for each course. Grade range is [0–1]
8	Cp	Checkpoints	There can be several types of checkpoints in a course, such as homework, quizzes, projects, midterms, finals, etc.
9	H	HS_GPA	High school GPA
10	A	Age	Age of student when CSPB219 course was taken
11	Sp	Sponsor	A sponsor is an entity that provides financial support to students during their studies
12	R	Residency	Residency indicates whether a student stays in a hostel or not. As the campus is located in Al Ain, students from Al Ain do not have access to hostels. Hostels are available to students who live outside of Al Ain
13	Z	Citizenship	Students from the UAE are considered citizens, while those from other countries are considered non-citizens
14	As	Academic_Standing	A student's academic standing refers to their current status. They either have a good academic standing or they are on probation

Neo4j is a high-performance NoSQL graph database, and is an embedded disk-based JAVA persistence engine that supports massive data storage and rapid graph enquiries. The Neo4j database uses a corresponding key-value pair structure to store entities and relations, which can then be visualized using different queries. To improve the classification accuracy, we generate the embedded vectors from the graph nodes using the node2vec algorithm. These graph embedding features are then integrated with the graph features and the results are analyzed. By combining the graph embedding (*GE*) features, we can observe how the performance of the employed models improves compared with the results of the basic *DS* and *GF* features (see Table 7). With the cosine metric, there is a 3.43% boost in accuracy and a 2.3% boost in AUC. Similarly, the Euclidean norm produces a 2.13% improvement in accuracy and a significant 2.83% boost in AUC. Therefore, our proposed method of combining the initial dataset features with graph topological features and graph embedding features substantially boosts the overall performance, indicating that these features are essential for accurate predictions of student performance and should not be neglected. The performance may be further enhanced by employing dimensionality reduction or feature selection techniques.

Discussion and future work

Traditional feature engineering techniques in ML can be replaced by powerful representation learning methods. In recent years, representation learning on graphs has steadily improved on tasks like node classification and connection prediction for graph-structured data (Hamilton et al., 2017). Current studies suggest that topology is a promising approach for improving classification performance (Bhatti et al., 2018; Dey et al., (2017; Hofer et al., (2017). Our work supports this statement, as topological features and graph embeddings have significantly improved the performance of classification models (see Table 7).

The conversion of tabulated data to graph data as a means of extracting valuable features and improving the performance of a classification model was also conducted by Zaki et al. (2021). The authors used the scalar product to extract the relations between nodes. Following their approach, we tested various distance norms. We hypothesized that educational datasets contain information about student performance, and so a distance metric may adequately catch hidden links between two instances (students). The proposed method was tested on three educational datasets and produced superior results. Furthermore, a comparison was made between various settings of input features, such as *DS* alone, $DS + GF$, and $DS + GF + GE$. The resultant model was able to classify the students into classes of Failed, At-Risk, and Good. With inputs of *DS* and $DS + GF$, the model reached AUC scores of 0.948 and 0.964, respectively. Similarly, the accuracy increased from 84.5% to 87.3% with the addition of *GF* to *DS*. Adding *GF* improved the performance by 2.019% in terms of AUC and 3.261% in terms of accuracy. Moreover, the *GCN* model incorporating graph topological features enhanced the prediction performance by 0.5% in terms of accuracy and 0.9% in terms of AUC for the cosine metric, and by 3.7% in terms of accuracy and 2.4% in terms of AUC for the Euclidean norm.

The proposed model identified at-risk students with 87.4% accuracy and 0.970 AUC under the cosine matric when graph embedding features were added. Similarly, adding graph embedding features resulted in 86.3% accuracy and 0.975 AUC under the Euclidean norm. The proposed solution may serve as a tool for the early detection of at-risk students. This will benefit universities and allow them to make better predictions of performance, thus improving their effectiveness and reputations. Overall, the proposed model can be applied to track students' performance. This may provide decision-makers and instructors with feedback about at-risk students failing a course, allowing stakeholders to decide the responses that may augment the final outcomes of the course.

We conducted a comparative analysis of different baseline methods reported in the literature with the one proposed in this paper. Our method achieves superior results to these state-of-the-art approaches. Starting from the random forest model developed by Mubarak et al. (2022), which has an accuracy of 79.00%, the SVM offers a slight improvement of 79.10%. Mubarak et al. (2022) also developed a graph neural network that achieved 84.00% accuracy. However, our proposed method boosts the accuracy to 84.50% using only the initial dataset features. When the initial graph features are combined with graph topological features, the accuracy increases to 87.30%, which is a significant boost. Finally, with our proposed ensemble method of *GCN* with *DS* and *GF*, the accuracy is further improved to 88.20%. We have also achieved a significant boost in terms of AUC; however, as the baseline methods only provided results in terms of accuracy, we simply claim that our proposed method achieves state-of-the-art accuracy that outperforms current methods.

Conclusions

Predict students' performance and their retention in institutions are vital issues in the learning analysis field, especially in virtual learning environments and MOOCs. This paper has presented a novel method for estimating students' performance based on the original dataset features and the features extracted from a graph representation of the data, combined with a GCN. We employed the Euclidean and cosine distance measures to evaluate the similarities between students' data and construct a graph. We then extracted topological features from the graph to enhance the data, providing the ability to capture structural correlations between data and gain deeper insights into historical feature points in terms of data analysis. The extracted original dataset features were then combined with graph features to improve the predictive power of the ML methods applied. Moreover, we have incorporated graph topological features using a GCN, which significantly increased the prediction performance. We also achieved superior results by employing an ensemble technique of adding graph embedding features with various ML models.

Our future work will focus on a knowledge graph approach and attempt to identify similarities between student behavior data from social networks to make better predictions about their performance, while considering both assessment and non-assessment factors. Furthermore, we will use hybrid methods based on ML to identify similar knowledge concepts across different courses and predict similar courses. This is expected to have a significant impact on improving learning systems.

Furthermore, we will construct entity representations using knowledge graph embedding methods to retain their semantic information. Our objective is to find better ways to rank students based on their previous performance by integrating academic and non-academic data from previous years.

Abbreviations

Acc	Accuracy
AI	Artificial Intelligence
AUC	Area Under the Curve
CSV	Comma-Separated Value
DL	Deep Learning
DS	Dataset Features
FPR	False Positive Rate
GCN	Graph Convolutional Network
GE	Graph Embeddings
GF	Graph topological Features
ITS	Intelligent Tutoring Systems
HW	Homework assignment
ML	Machine Learning
MLP	Multi-Layer Perceptron
MOOC	Massive Open Online Courses
MT	Mid-Term exam
RF	Random Forest
ROC	Receiver Operating Characteristic Curve
SVM	Support Vector Machine
TG	Total Grade
TPR	True Positive Rate
Qz	Quiz
XGB	EXtreme Gradient Boosting

Acknowledgements

The authors would like to acknowledge the continuous support from the College of Information Technology and Office of Institutional Effectiveness, UAEU.

Authors' contributions

Conceptualization, methodology, software, statistical analysis, writing—original draft preparation: B.A., T.H., and N.Z.; data curation: B.A. and T.H.; writing—review and editing: all authors; visualization: B.A. and T.H.; supervision: N.Z.; data analysis, literature review, discussion: all authors. All authors have read and agreed to the published version of the manuscript.

Availability of data and materials

The data of this study "EduRisk" is available from the corresponding author upon reasonable request through bi-dac.com/download.

Declarations

Ethics approval and consent to participate

The study was conducted according to the UAEU Human Research Ethics Committee. Approval was acquired under Ref. Nos. ERS_2020_6085 and ERS 2019_5869. No potentially identifiable personal information is presented in this study.

Competing interests

The authors declare that they have no competing interests.

Received: 1 September 2022 Accepted: 9 March 2023
Published: 10 April 2023

References

Ahadi, A., Lister, R., Haapala, H., & Vihavainen, A. (2015). Exploring machine learning methods to automatically identify students in need of assistance. In Proceedings of the eleventh annual international conference on international computing education research (pp. 121–130).

Ahmad, Z., & Shahzadi, E. (2018). Prediction of students' academic performance using artificial neural network. *Bulletin of Education and Research, 40*(3), 157–164.

Albreiki, B., Habuza, T., Shuqfa, Z., Serhani, M. A., Zaki, N., & Harous, S. (2021). Customized rule-based model to identify at-risk students and propose rational remedial actions. *Big Data and Cognitive Computing, 5*(4), 71–77.

Albreiki, B., Zaki, N., & Alashwal, H. (2021). A systematic literature review of student' performance prediction using machine learning techniques. *Education Sciences, 11*(9), 552–555.

Aleem, A., & Gore, M. M. (2020). Educational data mining methods: A survey. In 2020 ieee 9th international conference on communication systems and network technologies (csnt) (pp. 182–188).

Barbosa Manhães, L. M., da Cruz, S. M. S., & Zimbrão, G. (2015). Towards automatic prediction of student performance in stem undergraduate degree programs. In Proceedings of the 30th annual acm symposium on applied computing (pp. 247–253).

Bhatti, N., Hanbury, A., & Stottinger, J. (2018). Contextual local primitives for binary patent image retrieval. *Multimedia Tools and Applications, 77*(7), 9111–9151.

Bonacich, P. (1987). Power and centrality: A family of measures. *American Journal of Sociology, 92*(5), 1170–1182.

Brandes, U. (2001). A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology, 25*(2), 163–177.

Brandes, U., & Fleischer, D. (2005). Centrality measures based on current flow. In Annual symposium on theoretical aspects of computer science (pp. 533–544).

Chen, P., Lu, Y., Zheng, V. W., Chen, X., & Yang, B. (2018). Knowedu: A system to construct knowledge graph for education. *IEEE Access, 6*, 31553–31563.

Chen, X., Zou, D., & Xie, H. (2020). Fifty years of British journal of educational technology: A topic modeling based bibliometric perspective. *British Journal of Educational Technology, 51*(3), 692–708.

Dey, T., Mandal, S., & Varcho, W. (2017). Improved image classification using topological persistence. In Proceedings of the conference on vision, modeling and visualization (pp. 161–168).

Dhanalakshmi, V., Bino, D., & Saravanan, A. M. (2016). Opinion mining from student feedback data using supervised learning algorithms. In 2016 3rd mec international conference on big data and smart city (icbdsc) (pp. 1–5).

Duvenaud, D. K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., & Adams, R. P. (2015). Convolutional networks on graphs for learning molecular fingerprints. Advances in neural information processing systems, 28.

Ettorre, A., Michel, F., & Faron, C. (2022). Prediction of students' performance in e-learning environments based on link prediction in a knowledge graph. In The 23rd international conference on artificial intelligence in education (aied 2022).

Fei, M., & Yeung, D.-Y. (2015). Temporal models for predicting student dropout in massive open online courses. In 2015 ieee international conference on data mining workshop (icdmw) (pp. 256–263).

Feng, W., Tang, J., & Liu, T. X. (2019). Understanding dropouts in moocs. In Proceedings of the aaai conference on artificial intelligence (Vol. 33, pp. 517–524).

Ferguson, R. (2012). Learning analytics: drivers, developments and challenges. *International Journal of Technology Enhanced Learning, 4*(5/6), 304–317.

Gaur, M., Faldu, K., & Sheth, A. (2021). Semantics of the black-box: Can knowledge graphs help make deep learning systems more interpretable and explainable? *IEEE Internet Computing, 25*(1), 51–59.

Golino, H., & Gomes, C. M. A. (2014). Four machine learning methods to predict academic achievement of college students: a comparison study. *Revista E-Psi, 1*, 68–101.

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Representation learning on graphs: Methods and applications. arXiv preprint arXiv:1709.05584.

Harary, F., & Norman, R. Z. (1953). Graph theory as a mathematical model in social science (No. 2). University of Michigan, Institute for Social Research Ann Arbor.

Hofer, C., Kwitt, R., Niethammer, M., & Uhl, A. (2017). Deep learning with topological signatures. Advances in neural information processing systems, 30.

Hwang, G.-J., Xie, H., Wah, B. W., & Gašević, D. (2020). Vision, challenges, roles and research issues of artificial intelligence in education (Vol. 1). Elsevier.

Karimi, H., Derr, T., Huang, J., & Tang, J. (2020). Online academic course performance prediction using relational graph convolutional neural network. International Educational Data Mining Society.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

Kleinberg, J. M., Newman, M., Barabási, A.-L., & Watts, D. J. (2011). *Authoritative sources in a hyperlinked environment*. Princeton University Press.

Kolo, D. K., & Adepoju, S. A. (2015). A decision tree approach for predicting students academic performance. International Journal of Education and Management Engineering.

Lau, E., Sun, L., & Yang, Q. (2019). Modelling, prediction and classification of student academic performance using artificial neural networks. *SN Applied Sciences, 1*(9), 1.

Lin, Y., Liu, Z., Sun, M., Liu, Y., & Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion. In Twenty-ninth aaai conference on artificial intelligence.

Liu, M., Zha, S., & He, W. (2019). Digital transformation challenges: A case study regarding the mooc development and operations at higher education institutions in china. *TechTrends, 63*(5), 621–630.

Luo, S., & Fang, W. (2018). Potential probability of negative triples in knowledge graph embedding. In International conference on neural information processing (pp. 48–58).

Marchiori, M., & Latora, V. (2000). Harmony in the small-world. *Physica A: Statistical Mechanics and its Applications, 285*(3–4), 539–546.

Mesarić, J., & Sebalj, D. (2016). Decision trees for predicting the academic success of students. *Croatian Operational Research Review, 7*(2), 367–388.

Moubayed, A., Injadat, M., Shami, A., & Lutfiyya, H. (2020). Student engagement level in an e-learning environment: Clustering using k-means. *American Journal of Distance Education, 34*(2), 137–156.

Mubarak, A. A., Cao, H., Hezam, I. M., & Hao, F. (2022). Modeling students performance using graph convolutional networks. *Complex & Intelligent Systems, 8*(3), 2183–2201.

Mubarak, A. A., Cao, H., & Zhang, W. (2020). Prediction of students early dropout based on their interaction logs in online learning environment. *Interactive Learning Environments, 30*(8), 1414–1433.

Mubarak, A. A., Cao, H., Zhang, W., & Zhang, W. (2021). Visual analytics of video-clickstream data and prediction of learners performance using deep learning models in moocs courses. *Computer Applications in Engineering Education, 29*(4), 710–732.

Newman, M. E. (2001). Scientific collaboration networks. ii. Shortest paths, weighted networks, and centrality. *Physical review E*, *64*(1), 016132.

Nimon, K. F. (2012). Statistical assumptions of substantive analyses across the general linear model: a mini-review. *Frontiers in psychology, 3*, 322.

Novak, J. (1991). Clarify with concept maps. *The science teacher, 58*(7), 44.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. (Tech. Rep.). Stanford InfoLab.

Rastrollo-Guerrero, J. L., Gómez-Pulido, J. A., & Durán-Domínguez, A. (2020). Analyzing and predicting students performance by means of machine learning: A review. *Applied sciences, 10*(3), 1042.

Rodríguez-Hernández, C. F., Musso, M., Kyndt, E., & Cascallar, E. (2021). Artificial neural networks in academic performance prediction: Systematic implementation and predictor evaluation. *Computers and Education: Artificial Intelligence, 2*, 100018.

Sabidussi, G. (1966). The centrality index of a graph. *Psychometrika, 31*(4), 581–603.

Shahiri, A. M., Husain, W., et al. (2015). A review on predicting student's performance using data mining techniques. *Procedia Computer Science, 72*, 414–422.

Su, Y., & Zhang, Y. (2020). Automatic construction of subject knowledge graph based on educational big data. In Proceedings of the 2020 the 3rd international conference on big data and education (pp. 30–36).

Trumpower, D. L., Filiz, M., & Sarwar, G. S. (2014). Assessment for learning using digital knowledge maps. In Digital knowledge maps in education (pp. 221–237). Springer.

Valsamidis, S., Kontogiannis, S., Kazanidis, I., Theodosiou, T., & Karakos, A. (2012). A clustering methodology of web log data for learning management systems. *Journal of Educational Technology & Society, 15*(2), 154–167.

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

Yadav, S. K., Bharadwaj, B., & Pal, S. (2012). Mining education data to predict student's retention: a comparative study. arXiv preprint arXiv:1203.2987.

Yang, S., Zhu, M., Hou, J., & Lu, X. (2020). Deep knowledge tracing with convolutions. arXiv preprint arXiv:2008.01169.

Zaki, N., Mohamed, E. A., & Habuza, T. (2021). From tabulated data to knowledge graph: A novel way of improving the performance of the classification models in the healthcare data. medRxiv.

Zha, Z.-J., Mei, T., Wang, J., Wang, Z., & Hua, X.-S. (2009). Graph-based semi-supervised learning with multiple labels. *Journal of Visual Communication and Image Representation, 20*(2), 97–103.

Zhang, H., Sun, M., Wang, X., Song, Z., Tang, J., & Sun, J. (2017). Smart jump: Automated navigation suggestion for videos in moocs. In Proceedings of the 26th international conference on world wide web companion (pp. 331–339).

Zhang, S., Tong, H., Xu, J., & Maciejewski, R. (2019). Graph convolutional networks: a comprehensive review. *Computational Social Networks, 6*(1), 1–23.

Zhao, T., Chai, C., Luo, Y., Feng, J., Huang, Y., Yang, S., Li, H., Li, K., Zhu, F., & Pan, K. (2019). Towards automatic mathematical exercise solving. *Data Science and Engineering, 4*(3), 179–192.

Zheng, Y., Liu, R., & Hou, J. (2017). The construction of high educational knowledge graph based on mooc. In 2017 ieee 2nd information technology, networking, electronic and automation control conference (itnec) (pp. 260–263).

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.