Journal of Computer Science and Cybernetics, V.40, N.4 (2024), 299-314
DOI: 10.15625/1813-9663/21095

EARLY PREDICTION STUDENTS' GRADUATION RANK USING LAGT: ENHANCING ACCURACY WITH GCN AND TRANSFORMER

NGUYEN THI KIM SON[1,2,*], NGUYEN HUU QUYNH[3], BUI TUAN MINH[4]

[1] *Vietnam Academy of Science and Technology, Graduate University of Science and Technology, 18 Hoang Quoc Viet, Cau Giay, Ha Noi, Viet Nam*

[2] *Hanoi University of Industry, Hanoi, 298 Cau Dien, Nord Tu Liem, Ha Noi, Viet Nam*

[3] *CMC University, 84 Nguyen Thanh Binh, Ha Dong, Ha Noi, Viet Nam*

[4] *Thuyloi University, 175 Tay Son, Dong Da, Ha Noi, Viet Nam*

Abstract. Recent efforts to predict students' graduation ranks using machine learning and deep learning methods have faced challenges, particularly with small sample sizes which limit accuracy. This paper introduces the LAGT (Learning Analysis by Graph Convolutional Network and Transformer) method, a novel approach for early predicting of students' graduation ranks. LAGT integrates a Graph Convolutional Network (GCN) to enhance the training set with labeled samples and utilizes a Transformer to forecast graduation ranks. This method harnesses the semi-supervised learning capabilities of GCN to automatically label data, addressing the constraints of small sample sizes in training sets. Additionally, the Transformer leverages its proficiency in handling long sequences and capturing contextual information, thereby demonstrating superior effectiveness in models trained on larger datasets. We evaluated this method on three datasets from some universities (HNMU1, HNMU2, VNU) and achieved a maximum accuracy of 92.73%. Results indicate that the integrated LAGT method outperforms comparable approaches across multiple metrics including accuracy, prediction precision, and model sensitivity, achieving up to a 35.73% improvement. Notably, on the same HNMU1 dataset, the accuracy increased from 85% (reported by Son et al. [1]) to 90.91% with this model. Experimental comparisons underscore the superior performance of LAGT over alternative methodologies in similar scenarios.

Keywords. Early prediction of graduation classification, academic performance prediction, semi-supervised learning, Graph Convolutional Network, Transformer.

1. INTRODUCTION

Learning Analytics (LA) focuses on measuring and collecting data, as well as analyzing and reporting data to support educational decisions. One of the important applications of

Corresponding author.
E-mail addresses: sonntk@haui.edu.vn (N.T.K. Son), nhquynh@cmc-u.edu.vn (N.H. Quynh), 2051063681@e.tlu.edu.vn (B.T. Minh).

LA is to monitor and predict learners' learning outcomes and detect potential problems early so that timely intervention can be made [2]. Recently, higher education institutions have increased their interest in LA to meet demands for transparency and close oversight of their admissions and retention practices student. Universities are applying LA to improve service quality and achieve specific goals such as scores and student retention [3]. An important solution is to predict student learning outcomes early. This result helps students choose appropriate courses, allows managers and lecturers to identify students who need support to complete classes, and minimizes warnings or forced withdrawals due to poor learning results. This brings time and cost benefits to students, families, schools, and society. Therefore, predicting learning outcomes is an important research topic in the field of educational data analysis, attracting the attention of many researchers.

The emergence of advanced techniques in artificial intelligence (AI), deep learning models and their hybrid models [4], has created new opportunities for enhancing the accuracy of prediction systems. Graph Convolutional Network (GCN) effectively exploits the structure and relationships within graph data, enabling it to capture information from the connections and interactions among elements [5]. Meanwhile, Transformers excel in modeling long data sequences and capturing contextual information, demonstrating remarkable effectiveness across various applications.

Deep learning techniques are increasingly applied to analyze learners' outcomes as machine learning models evolve. Mubarak et al. [6] introduced a GCN-based model for classifying student engagement, achieving 84% accuracy compared to traditional methods. Sarwat et al. [7] developed a Conditional Generative Adversarial Network (CGAN) combined with a Support Vector Machine (SVM), demonstrating effectiveness in predicting learning outcomes. Hassan and Muhammad (2023) utilized K-nearest neighbors (KNN) and Decision Trees with attribute selection through genetic algorithms to predict student grades. Christou et al. [8] proposed a feature-building and selection method based on grammar evolution for Radial Basis Function (RBF) networks to forecast student outcomes. These studies highlighted the critical importance of deep learning models in processing educational data, particularly in predicting learners' outcomes.

A technical challenge in the problem of predicting learning outcomes is that educational data systems are not compatible with each other, so combining administrative data and survey data (before and after the learning process at school, personal, family, and social factors that influence learning outcomes) and process learning data remain a challenge. Due to the characteristics of each level, field of study, and regulations, the training program at each educational institution is different, making it difficult to synthesize and match data to build a large enough data set. On the other hand, the source of digitized education data, although much has been added in recent years, is still quite modest (compared to other data sources, such as ImageNet). Besides, LA also requires careful attention to student and faculty privacy as well as ethical obligations related to knowing and acting on student data. Not being able to reuse training data sets in published works is also one of the limitations of this approach. The automatic data generation mechanism to compensate for the above limitations is one of the priority mechanisms used in studies following the LA approach to small datasets.

In this paper, to overcome the above limitations, we propose a method that combines modern deep learning models, GCN and Transformer, into a LAGT method to deal with small datasets in the education field. This method takes advantage of GCN's semi-supervised learning advantages to automatically add labels to the data set, based on which Transformer can promote its prediction advantage with a larger training set (can double the number of original training samples).

This paper addresses the challenge of predicting a student's graduation classification using their survey results and academic performance during the first and second years of university. The main results of the paper are shown in the following aspects:

(1) Build 03 training data sets, processed from 03 raw data sets of 03 majors in 02 universities. These 03 data sets are good assets for use in research, data analysis, and providing recommendations and solutions in education - an area where quantitative research has not dominated the position so far.

(2) Propose a method to early predict students' graduation results through survey data and learning data of the first two years of the university according to the current deep learning approach. The proposed method takes advantage of the semi-supervised learning advantage of GCN to automatically generate labels, helping to increase the size of the training data set. Then use the flexibility and good clustering capabilities of the Transformer to predict the student's graduation type.

(3) Experimentally deploy the proposed model on 03 training datasets and compare our model with some machine learning models (Logistic Regression) and deep learning models (Transformer) methods to illustrate its effectiveness. The results show that the proposed integration method has superior performance compared to the matching methods on all three scales: accuracy, prediction accuracy, and model sensitivity. The highest difference is up to 35.73%. Moreover, for parallel results of the same dataset (HNMU1), the accuracy increases from 85% in [1] to 90.91% in this model. The highest accuracy is 92.73% for the VNU dataset confirming the effectiveness of our method.

The paper is organized as follows: An overview of the necessity and results of the research problem is presented in section 1. Some related research, proposed methods, and proposed models will be introduced in sections 2 and section 3. Experimental results with specific descriptions of the three datasets and results of different scenarios implemented above datasets are presented in section 4. Finally, there are conclusions and references.

2. RELATED STUDIES

Recent studies have demonstrated the effectiveness of machine learning and deep learning models in predicting student academic outcomes [9]. Iatrellis and colleagues [10] applied the K-means method to group students based on data and used Random Forest (RF) to cluster, providing more detailed predictions on course completion times and post-university enrollment rates of students (SEIPS). Okubo [11] utilized Recurrent Neural Networks (RNN) to predict student grades in a specific course. However, with data limited to only 108 students in a single course, the generalizability of the results cannot be fully evaluated. Fei and Yeung [12] experimented with State Space Models and Sequential Neural Networks on two MOOC datasets with the goal of detecting students likely to drop out.

Corrigan and Smeaton [13] used Random Forest, RNN, and simple LSTM to predict student success through a virtual learning environment. In the same virtual learning environment, Waheed et al. [14] deployed a Deep Neural Network (DNN) to analyze student interaction data, predicting students at risk of failing and proposing timely interventions. Fok et al. [15] discovered an optimal configuration of TensorFlow models to achieve higher prediction accuracy for this problem. Yousafzai et al. [16] improved results using Bidirectional Long Short-Term Memory (BiLSTM) with attention mechanisms. Li et al. [17] introduced the GNN, R2GCN model, which showed superior performance in predicting student performance in online learning groups. For predicting graduation outcomes, Son et al. [1] used Logistic Regression and feature selection to predict graduation outcomes based on admission data and first and second-year academic records at the university. The data was collected from HNMU's students with 993 samples and 23 related characteristics saved in the training management and the survey data. The accuracy with the Logistic Regression method was raised from 79% for the case of first-year data to 85% when the data was added to the second-year academic records.

A profound understanding of educational data through machine learning and deep learning not only optimizes student academic outcomes but also plays a crucial role in improving teaching methods and enriching the learning environment [3]. However, the application of deep learning in the field of educational data science is still in its early stages, with many recent studies emerging only in recent years with limited data resources.

3. METHOD

3.1. Overview of GCN

GCN is a generalization of Convolutional Neural Networks (CNNs) for structured graph data. The primary objective of GCN is to filter node attribute information and graph structure into a node representation vector, also known as embedding [5].

GCN operates by performing a series of linear transformations on the features of vertices in the graph. Each of these linear transformations is called a GCN layer. The feature of a vertex is a vector representing information about that vertex. This information may include the value of the vertex itself, its neighboring vertices, and the relationships between the vertex and its neighbors.

The linear transformation in a GCN layer uses a weight matrix to combine the features of neighboring vertices. This weight matrix is learned from training data. The basic structure of GCN is illustrated in Figure 1 with the model's input is a graph where vertices typically represent objects (such as people, products, and websites), and edges depict relationships between them.

Graph Representation: The graph is often represented as an adjacency matrix to describe the relationships between vertices. Each vertex can be represented by a feature vector, and a weight matrix can be used to represent edge weights.

Graph Convolutional Layer: GCN uses graph convolutional layers to compute new feature representations for each vertex based on neighborhood information. The number of layers denotes the furthest distance over which node features can propagate, essentially the maximum number of hops each node can move. These convolutional layers allow the model to "see" and "aggregate opinions" from neighboring vertices to enhance the representation of each vertex. The number of these convolutional layers can be customized.

Activation Function and Pooling Layer: After each convolutional layer, an activation function may be applied to introduce non-linearity. Pooling layers can also be used

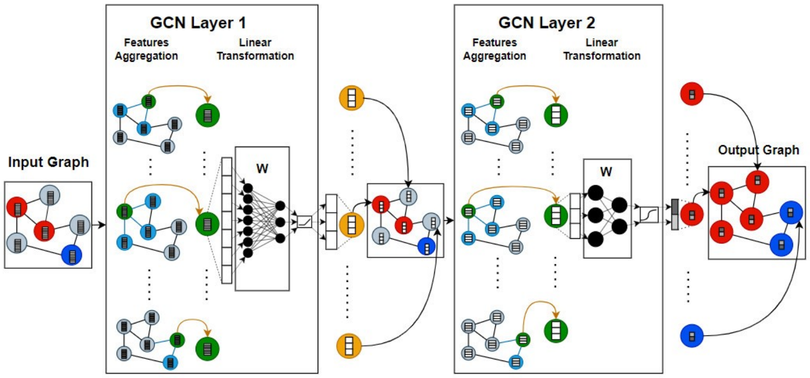

Figure 1: Graph Convolutional Network Architecture with Two Layers

to reduce the dimensionality of representations, enhancing overall efficiency and reducing model complexity.

Fully Connected (FC) Layer: After information has passed through several convolutional layers, a fully connected layer can be added to aggregate information and produce the final representation.

Output Layer (Classification): The output layer typically uses a Softmax function to convert the final representation into probabilities for each class. The final result of the model is a probability distribution, with the class having the highest probability often chosen as the classification result.

GCN is a relatively new technology still under development. Researchers continue to explore ways to improve its performance and expand its range of applications.

3.2. Overview of Transformer

Computers cannot learn directly from raw data such as images, text files, audio files, or video clips. They require a process of encoding information into numerical form and decoding from numerical form to output results. This process involves two main stages: Encoder and Decoder [18].

Encoder: This phase transforms the input into machine-understandable features. In neural networks, the Encoder consists of hidden layers. For CNN models, the Encoder comprises a sequence of hidden layers of convolution and max-pooling. In RNN models, the Encoder process includes embedding layers and recurrent neural networks.

Decoder: The output from the Encoder serves as the input to the Decoder. The Decoder's objective is to determine the probability distribution from the features obtained in the Encoder phase, thereby identifying the output label. The result can be a single label for classification models or a sequence of labels over time for seq2seq models.

In this paper, we use the Transformer model with Encoder using Attention transformation, combined with convolution; Combine some additional layers of full connections, Dropout, and Decoder with Pooling-creating connections with the number of dimensions in the Unit. The application of node reduction techniques to predict students' graduation outcomes is consistent with previous models.

3.3. Our method

In this section, we present a proposed model for predicting graduation ranks.

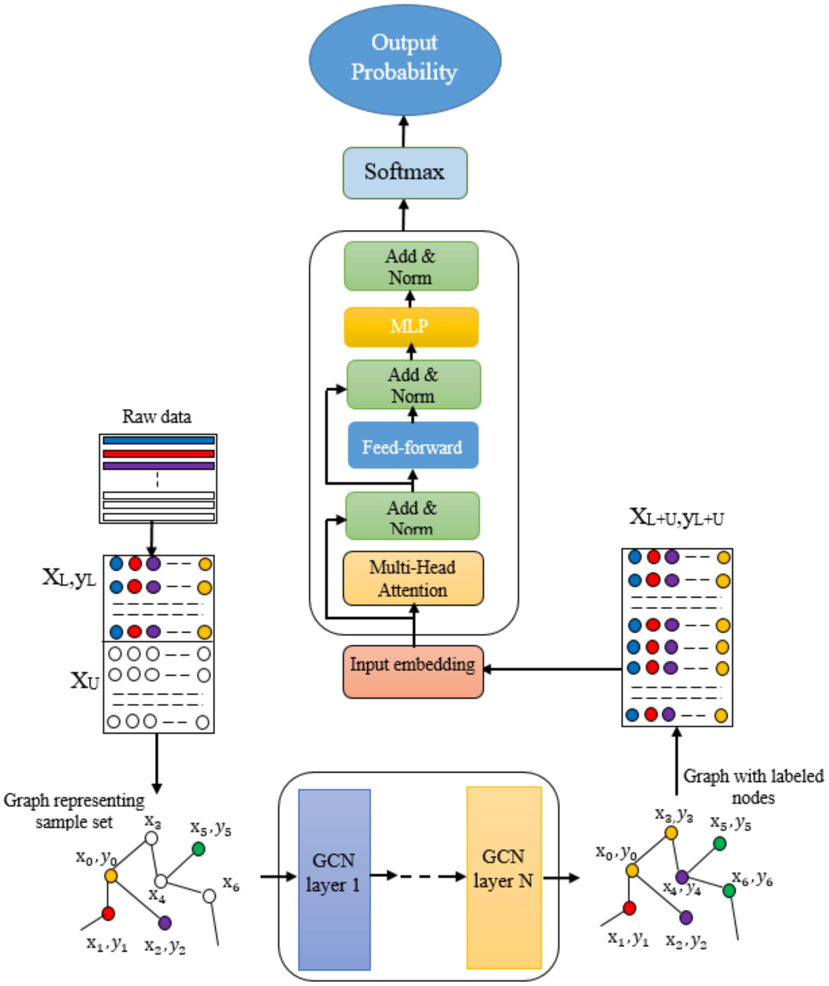

Figure 2: Proposed graduation rating prediction model.

The operation of the model in Figure 2 is as follows. Firstly, the samples in “Raw data” (labeled and unlabeled) collected from reality will be preprocessed to obtain a data array of samples labeled (X_L, y_L) and unlabeled samples X_U. From this data table, we proceed to build a “Graph” with each student data record as a vertex of the graph. The construction of a “Graph” is done according to the principle that if the Euclidean distance of two data information records is less than a certain threshold γ, there will be an edge connecting those two vertices in the graph. The constructed “graph” will be fed into an N-layer graph convolutional network (block in the bottom middle) to obtain a “Graph with labeled nodes” (graph in the lower right corner). The reason for using a graph convolutional network here is because the problem we are solving has very few training samples (labeled samples even less) and takes advantage of the power of this approach semi-supervised based on the GCN graph to increase the number of training samples from unlabeled samples. After passing through GCN, the unlabeled samples of the dataset will be labeled and added. Thus we have a training sample table with an increased number of labeled samples $(X_{L+U},\ y_{L+U})$. Secondly, this data table is passed through the Transformer model (upper middle block) to make predictions. It should be noted here that the Transformer model is modern and very effective, but with a small number of training samples, the model does not maximize its effectiveness. This is also the reason why we combine the GCN and Transformer networks in our model.

GCN Model

We use a 2-layer model with the HNMU1, HNMU2, and VNU datasets. For the HNMU1, HNMU2, and VNU datasets, we use a two-layer model. For the HNMU1 dataset, the first layer has a hidden size of 6 and uses the ReLU activation function. The second layer is the output layer with 5 dimensions (corresponding to the number of classes in the HNMU1 dataset) and uses the Softmax activation function. For the HNMU2 dataset, the first layer has a hidden size of 8 and uses the ReLU activation function. The second layer is the output layer with 4 dimensions (corresponding to the number of classes in the HNMU2 dataset) and uses the Softmax activation function. For the VNU dataset, the first layer has a hidden size of 8 and uses the ReLU activation function. The second layer is the output layer with 3 dimensions (corresponding to the number of classes in the VNU dataset) and uses the Softmax activation function. The detail parameters of GCN are given in Table 1.

Table 1: GCN model parameter table on the HNMU1, HNMU2, and VNU datasets

	First layer	Activation function	Second layer	Activation function
HNMU1	6	Relu	5	Softmax
HNMU2	8	Relu	4	Softmax
VNU	8	Relu	3	Softmax

Transformer Model

The Transformer model for HNMU1 will select the multi-head value as 1. The feed-forward layer in each encoder layer has a size of 64. The number of Transformer encoder layers is 1. The dropout rate is 0.5. After that, it is passed through a fully connected layer with an output size of 5. The output of this network will be 5 (corresponding to the number of classes in the HNMU1 dataset).

The Transformer model for HNMU2 will select the multi-head value as 2. The feed-forward layer in each encoder layer has a size of 64. The number of Transformer encoder layers is 1. The dropout rate is 0.5. After that, it is passed through a fully connected layer with an output size of 4. The output of this network will be 4 (corresponding to the number of classes in the HNMU2 dataset).

The Transformer model for VNU will select the multi-head value as 4. The feed-forward layer in each encoder layer has a size of 64. The number of Transformer encoder layers is 1. The dropout rate is 0.5. After that, it is passed through a fully connected layer with an output size of 3. The output of this network will be 3 (corresponding to the number of classes in the VNU dataset). The detail parameters of Transformer are given in Table 2.

Table 2: Transformer model parameter table on the HNMU1, HNMU2, and VNU datasets

	Multi-head	Feed-forward layer	Number of Encode	Fully connected layer	Activation function
HNMU1	1	64	1	5	Softmax
HNMU2	2	64	1	4	Softmax
VNU	4	64	1	3	Softmax

4. EXPERIMENTAL

4.1. Description of three experimental datasets

4.1.1. First dataset (HNMU1)

The dataset includes 2,763 students majoring in elementary education, who has studied at HNMU from 2014 to 2021. The dataset was collected and processed for 18 months, from March 2020 to September 2021. We extracted 73 student-related characteristics to get a comprehensive view of students' training history. These features fall into main categories: academic performance (e.g., GPA, credits completed), financial information, information before admission, etc. Each observation sample is represented by 1 row. We use data from HNMU for training management and survey data to incorporate various characteristics of each student. The data analysis process involves aligning data and prioritizing data updates, as well as reducing overall data sparsity. This results in a new, valuable training dataset that improves the reliability of our prediction outcomes.

The data is cleaned, removing unnecessary data variables and variables not assessed in this study (some physical or aptitude subjects, and variables related to student finances). At the same time, attributes with too little data or lots of empty data are also removed. Electives largely fall into this blank data area. We also focus on specific test score data and eliminate the letter grade portion of the test score data. We selected student-specific variables to test their correlation with the variable of interest. Therefore, from 2,763 samples with 73 attribute variables (20 survey characteristics and 53 GPA of university), the dataset was cleaned to include data from only 933 observed samples and the variables were limited to 23 variables for training (3 survey characteristics and 20 GPA of the first and second year of university).

4.1.2. Second dataset (HNMU2)

The dataset was collected from students majoring in Mathematics and Physics Education at the Faculty of Education, Hanoi Metropolitan University, encompassing the years 2014 to 2023. Raw data was provided by the Faculty of Education's Department of Training Management and Student Affairs, and survey data, including student management details (tuition, personal information, etc.), admission scores, foreign language scores, computer science scores, module scores across 8 semesters spanning 4 years of study, and results of related factor survey. Scores were recorded on both a 10-point and 4-point scale, accompanied by letter grades and module credits. This dataset underwent collection and processing over 2 years from 2022 to 2023, yielding over 1,000 samples and encompassing 89 attributes, of which 35 survey characteristics. Following pre-processing, the dataset includes 744 observation samples from students majoring in Mathematics Education, encompassing 55 attribute variables (35 survey characteristics and 20 GPA of the first and second year of university), with 551 observation samples having actual labels. The HNMU2 dataset exhibits significant class imbalance, with the following distribution of samples: the medium class contains 19 samples, the good class has 338 samples, the very good class includes 190 samples, and the excellent class has only 4 samples.

4.1.3. Third dataset (VNU)

The dataset collects information from 2,791 students at the University of Education, VNU from 2014 to 2023 of different pedagogical majors (Literature Education, Mathematics Education, Physic Education, etc.). These datasets are collected and basic-processed from 2021 to 2023. There is a fact that data with different majors do not match each other. Thus, we only format data for a specific major. For example, we use data collected from 668 students majoring in Literature Education at the University of Education, VNU from 2014 to 2023 with 92 attributes (mixed from survey data and academic data). Data after pre-processing were cleaned, removing unnecessary data variables and variables not evaluated. We selected student-specific variables to test their correlation with the variable of interest. After performing the pre-processing steps, we have only obtained 271 observation samples consistent with 69 actual labeled attribute variables (49 survey characteristics and 20 GPA of the first and second year of university). The dataset consists of three main categories of variables: (A) Personalization factors of participating students, such as gender and parents' educational backgrounds; (B) Factors affecting learning outcomes, which include study hours, time spent on social media, scholarships, health status, and employment status; and (C) Academic performance metrics, covering both pre-university and university achievements.

In this paper, for experimental results, only survey admissions data and academic data from students' first and second years were included, as previous studies have shown that early in a student's college career is the most important stage for retention and graduation outcomes [19, 20].

4.2. Evaluation measures

The metrics used include: Accuracy, Prediction Accuracy (Precision), Sensitivity (Recall), and $F1$-Score ($F1$). They are calculated by using the following formula ([21]).

Accuracy $= \dfrac{AP + AN}{\text{Total number of samples}}$, Precision $= \dfrac{TP}{TP + FP}$, Recall $= \dfrac{TP}{TP + FN}$, and $F1 = \dfrac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$.

In which the indicators TP, FP, TN, and FN have the following meanings. TP (True Positive) is the total number of positive pattern-matching prediction cases. TN (True Negative) is the total number of negative pattern-matching prediction cases. FP (False Positive) is the total number of cases that predict observations belonging to the negative label to be positive and FN (False Negative) is the total number of cases that predict observations belonging to the positive label to be negative.

4.3. Experimental results

The proposed model given in subsection 3.3 improves the accuracy of the problem of predicting graduation grades with a small training data set size. To demonstrate the effectiveness of our proposed model, this subsection centers on conducting experiments using three separate datasets: the HNMU1 dataset, the HNMU2 dataset, and the VNU dataset. Through the evaluation of these real-world datasets, we intend to substantiate the effectiveness of our model.

We performed experiments employing three distinct methods:

(a) Transformer on the original dataset.

(b) Transformer on the original dataset and data augmentation using SMOTE.

(c) Proposed Method - LAGT: The approach involves using a Transformer trained on an expanded dataset (comprising original and additional data), followed by applying GCN on the original dataset. Specifically, the dataset is divided as follows: 60% original data, 20% additional unlabeled data, 20% test data.

On the other hand, to estimate the effectiveness of our model, DNN and GAT are used to compare with similar scenarios (input data consists of 60% of the original labeled data and 20% of the test data).

By comparing the performance of these established methods with our proposed model, we aim to evaluate how leveraging synthetic data generated by LAGT improves predictive accuracy. These analyses will help determine whether our approach surpasses traditional techniques and highlights its potential for enhancing predictive performance, particularly in data-limited scenarios. For the GCN model, we use the Adam optimizer with a learning rate of 0.01 and weight decay of 0.0005 and each layer will use a dropout of 0.5. For the Transformer model, we use the Adam optimizer with a learning rate of 0.005 and weight decay of 0.0005 and each layer will use a dropout of 0.5.

4.3.1. Results on the first data set (HMNU1)

We train the model on this first dataset with 1,000 epochs. The principle of choosing the best model is that we take the average of the training loss and validation loss values. Whichever epoch gives the smallest value will be selected at that epoch. On that principle, with the model in Figure 3, the model selected at the 969^{th} epoch has a train loss of 0.0740 and a validation loss of 0.7925. In Figure 4, the model is selected at the 674^{th} epoch. has a train loss of 0.4651 and a validation loss of 0.4985. With the model in Figure 5, the model is selected at the 375^{th} epoch. has a train loss of 0.3419 and a validation loss of 0.4266. With the model in Figure 6, the model selected at the 418^{th} epoch has a train loss of 0.2739 and a validation loss of 0.4665.

The model obtained after training is applied to the test set and results are obtained as shown in Table 3. Table 3 shows the accuracy, i.e. the percentage of correct predictions over the total number of test samples, of the proposed method is 6.95% higher than the accuracy of the Transformer method. Moreover, the accuracy of LAGT is 5.91% higher than the accuracy of the Logistic Regression method, that was used by Son et al. [1]. Besides, prediction accuracy (the ratio of correct predictions to the total number of positive predictions of the model) and sensitivity (the ratio of positive samples correctly identified by the model to the total number of actual positive samples) of the proposed method are also higher than

Table 3: Prediction results on the first dataset (HNMU1)

Methods	Accuracy	Precision	Recall	$F1$-Score
Logistic Regression Son et al. (2022)	85	-	-	-
DNN	81.28	36.92	43.91	38.67
GAT	74.33	42.01	64.72	47.19
Transformer	83.96	42.91	37.74	39.66
Transformer + Smote	85.03	68.39	41.25	46.96
LAGT	**90.91**	**71.76**	**67.55**	**69.39**

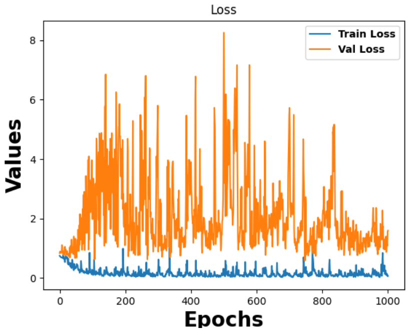

Figure 3: Transformer on the original dataset

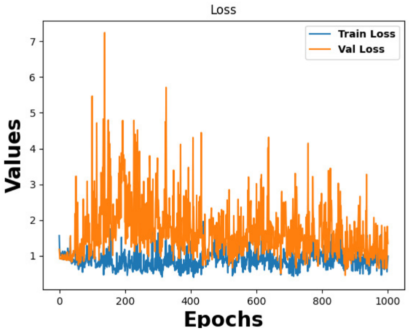

Figure 4: Transformer on the original dataset and SMOTE-generated data

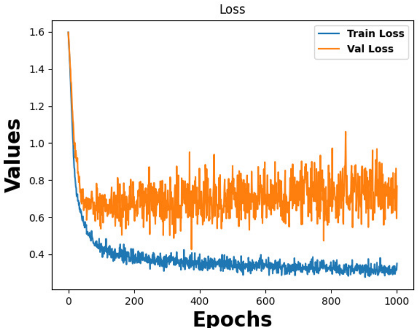

Figure 5: GCN on the original dataset

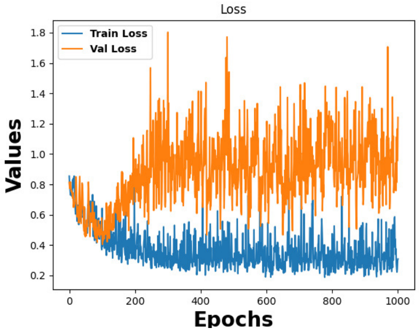

Figure 6: Transformer (after adding training samples)

the Transformer method by 28.85% and 29.81%, respectively. This shows that the proposed method's ability to accurately classify positive samples is much higher than that of the Transformer method. The $F1-$Score, i.e. the harmonic average of prediction accuracy and sensitivity, of the proposed method is 29.73% higher than that of the Transformer method. This shows a better balance between prediction accuracy and sensitivity of the proposed method compared to the Transformer method.

4.3.2. Results on the second data set (HMNU2)

Do the same principle we train the model on second dataset with 1,000 epochs. With the model in Figure 7, the model selected at the 96th epoch has a train loss of 0.0326 and a validation loss of 0.7590. With the model in Figure 8, the model is selected at the 410th epoch has a train loss of 0.2704 and a validation loss of 0.4526.With the model in Figure 9, the model selected at the 159th epoch has a training loss of 0.4682 and a validation loss of 0.7812. With the model in Figure 10, the model selected at the 11th epoch has a train loss of 0.3919 and a validation loss of 0.6049. The model obtained after training is applied to the test set and results are obtained as shown in Table 4.

Table 4: Prediction results on the second data set (HNMU2)

Methods	Accuracy	Precision	Recall	F1-Score
DNN	81.82	58.30	57.06	57.42
GAT	86.36	57.60	61.36	59.42
Transformer	87.27	59.46	60.96	59.99
Transformer + Smote	89.09	78.96	79.45	79.06
LAGT	**91.82**	**82.40**	**78.04**	**79.74**

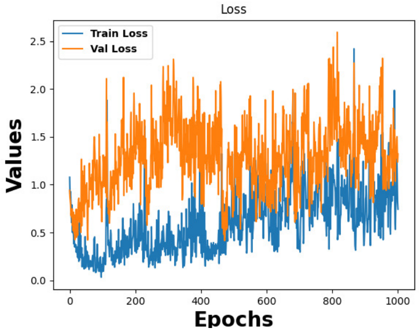

Figure 7: Transformer on the original dataset

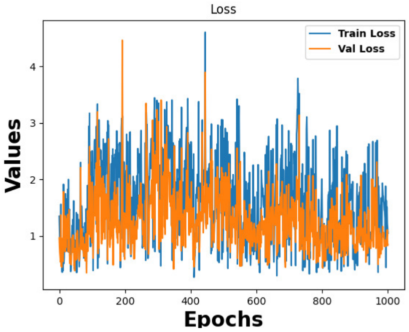

Figure 8: Transformer on the original dataset and SMOTE-generated data

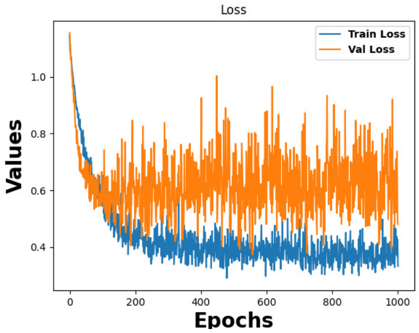

Figure 9: GCN on the original dataset

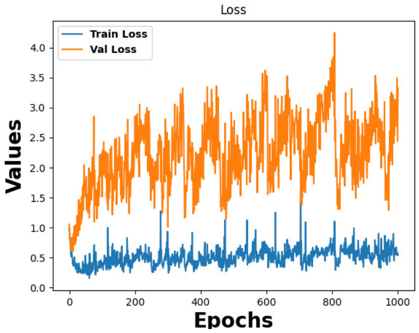

Figure 10: Transformer (after adding training samples)

Table 4 shows the accuracy of the proposed method is significantly higher (4.55% higher) than the method using only the Transformer, indicating that the proposed model is capable of predicting more accurately than the Transformer. The prediction accuracy of the proposed method is much higher (22.94%) than that of the Transformer, this means that in the positive predictions of the model, the correct prediction rate is higher, minimizing the cases of false positives. The sensitivity of the proposed method is also significantly higher (about 17.08%), showing that the proposed model is capable of detecting more real positive samples and minimizing false negative cases. The $F1-$Score of the GCN and Transformer combination is much higher (about 19.75%). $F1-$Score is a composite index that combines both prediction accuracy and sensitivity, and this improvement shows that the proposed method achieves a better balance between accurately predicting positive samples and many actual positive samples were detected.

4.3.3. Results on the third data set (VNU)

We also train the model on third dataset with 1,000 epochs. In Figure 11, the model selected at the 9^{th} epoch has a training loss of 0.2096 and a validation loss of 0.3005. With the model in Figure 12, the model is selected at the 666^{th} epoch has a train loss of 0.4841 and a validation loss of 0.2416. With the model in Figure 13, the model is selected at the 986^{th} epoch has a train loss of 0.1833 and a validation loss of 0.1687. With the model in Figure 14, the model selected at the 17^{th} epoch has a train loss of 0.1609 and a validation loss of 0.2728. The model obtained after training is applied to the test set and results are obtained as shown in Table 5.

Table 5: Prediction results on the third data set (VNU)

Methods	Accuracy	Precision	Recall	$F1$-Score
DNN	81.82	61.67	92.91	67.70
GAT	85.45	63.27	58.98	58.70
Transformer	87.27	60.46	60.14	59.92
Transformer + Smote	89.09	96.23	55.56	63.08
LAGT	**92.73**	**73.80**	**95.87**	**78.15**

From Table 5 we can see that the accuracy of the proposed method is significantly higher (5.46%) than the method using only a Transformer, showing that the proposed model is capable of predicting more correctly in total test samples. The prediction accuracy of the proposed method is much higher (13.34%), which means that in the positive predictions of the proposed model, the correct prediction rate is higher, minimizing the positive cases fakeness. The sensitivity of the proposed method is also significantly higher (35.73%), showing that this model is capable of detecting more true positive samples, minimizing false negative cases. The $F1-$Score of the proposed method is much higher (18.23%). $F1-$Score is a composite index that combines both prediction accuracy and sensitivity, and this improvement shows that the proposed method achieves a better balance between accurately predicting positive samples and many actual positive samples were detected.

Through experimental results on three data sets (Tables 3-5), we can confirm that the proposed method is better than the method using only Transformer in all evaluation indicators. These results indicate that combining GCN and Transformer significantly improved the model's learning and prediction capabilities, making the proposed method superior to using Transformer alone.

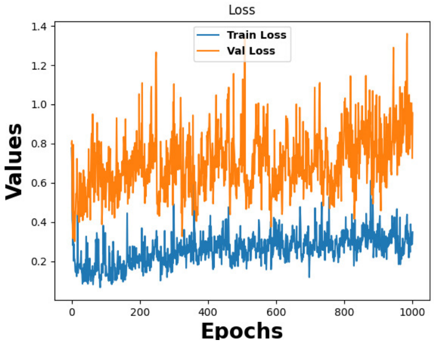

Figure 11: Transformer on the original dataset

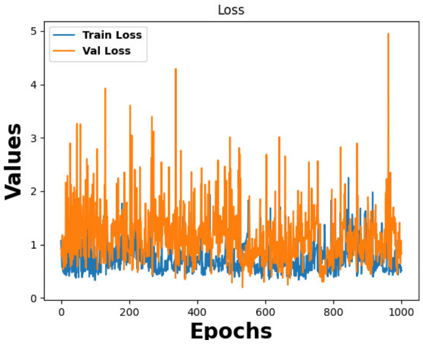

Figure 12: Transformer on the original dataset and SMOTE-generated data

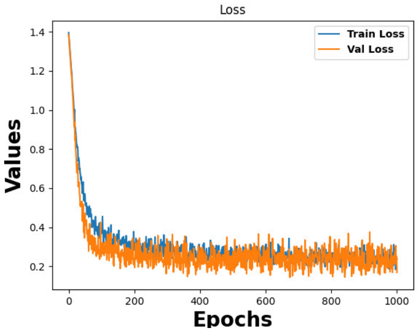

Figure 13: GCN on the original dataset

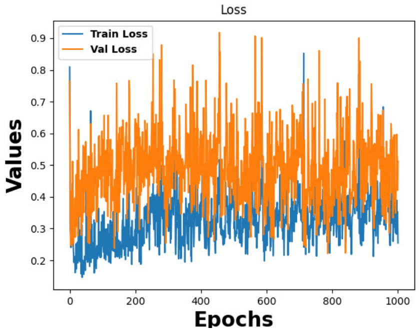

Figure 14: Transformer (after adding training samples)

5. CONCLUSION

This paper presents the LAGT method, a new approach designed to enhance the accuracy of early student graduation classification predictions. By integrating Graph Convolutional Networks (GCNs) to enrich the training set with labeled samples and utilizing Transformers for prediction, the LAGT method effectively addresses the challenges of traditional techniques in managing small sample sizes. Experimental results across three datasets from various universities demonstrate that the LAGT method achieves a remarkable accuracy of up to 92.73%, significantly surpassing several competing models, including DNN, GAT, and Transformer, in scenarios utilizing single machine learning models. Furthermore, LAGT shows enhanced performance when using models paired with a Transformer model, such as when combined with data augmentation techniques like SMOTE. This indicates that LAGT not only boosts prediction accuracy but also excels relative to other methods under similar conditions. The synergy of GCN and Transformer maximizes the extraction of information from the data, yielding reliable and timely predictions. We aim to enhance the model's capability to automatically generate data and optimize its performance in future research.

REFERENCES

[1] N. T. K. Son, N. V. Bien, N. H. Quynh, and C. C. Tho, "Machine learning based admission data processing for early forecasting students' learning outcomes," *International Journal of Data Warehousing and Mining*, vol. 18, no. 1, pp. 1-14, 2022.

[2] M. Bienkowski, M. Feng, and B. Means, *Enhancing teaching and learning through educational data mining and learning analytics*, U.S. Department of Education, Office of Educational Technology, Washington, D.C., 2012.

[3] M. Khalil and M. Ebner, "Learning analytics: Principles and constraints," *EdMedia: World Conference on Educational Media and Technology*, Association for the Advancement of Computing in Education (AACE), pp. 1789-1799, 2015.

[4] L. M. Tuan, L. T. Huong and H. M. Tan, "A hybrid model using the pre-trained bert and deep neural networks with rich feature for extractive text summarization," *Journal of Computer Science and Cybernetics*, vol. 37, no. 2, pp. 123-143, 2021.

[5] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017), Conference Track Proceedings*, Toulon, France, 2017.

[6] A. A. Mubarak, H. Cao, and I. M. Hezam, "Modeling student's performance using graph convolutional networks," *Complex Intelligent Systems*, vol. 8, pp. 2183–2201, 2022.

[7] S. Sarwat, N. Ullah, S. Sadiq, R. Saleem, M. Umer, A. A. Eshmawi, A. Mohamed, and I. Ashraf, "Predicting students' academic performance with conditional generative adversarial network and deep SVM," *Sensors*, vol. 22, no. 13, pp. 4834, 2022.

[8] V. Christou, I. Tsoulos, V. Loupas, A. T. Tzallas, C. Gogos, P. Karvelis, N. Antoniadis, E. Glavas, and N. Giannakeas, "Performance and early drop prediction for higher education students using machine learning," *Expert Systems with Applications*, vol. 225, pp. 120079, 2023.

[9] T. Doleck, D. J. Lemay, R. B. Basnet, and P. Bazelais, "Predictive analytics in education: A comparison of deep learning frameworks," *Education and Information Technologies*, vol. 25, no. 3, pp. 1951–1963, 2020.

[10] O. Iatrellis, I. K. Savvas, P. Fitsilis, and V. C. Gerogiannis, "A two-phase machine learning approach for predicting student outcomes," *Education and Information Technologies*, vol. 26, pp. 69–88, 2021.

[11] F. Okubo, T. Yamashita, A. Shimada, and H. Ogata, "A neural network approach for students' performance prediction," *LAK '17: Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, 2017, pp. 598–599, 2017.

[12] M. Fei and D. Y. Yeung, "Temporal models for predicting student dropout in massive open online courses," *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pp. 256–263, 2015.

[13] O. Corrigan and A. F. Smeaton, "A course agnostic approach to predicting student success from VLE log data using recurrent neural networks," *European Conference on Technology Enhanced Learning*, Springer, pp. 545–548, 2017.

[14] H. Waheed, S. Hassan, N. R. Aljohani, J. Hardman, S. Alelyani, and R. Nawaz, "Predicting academic performance of students from VLE big data using deep learning models," *Computers in Human Behavior*, vol. 104, pp. 106189, 2020.

[15] W. W. T. Fok, Y. S. He, H. H. A. Yeung, K. Y. Law, K. Cheung, Y. Ai, and P. Ho, "Prediction model for students' future development by deep learning and TensorFlow artificial intelligence engine," *2018 4th International Conference on Information Management (ICIM)*, pp. 103–108, 2018.

[16] B. K. Yousafzai, S. A. Khan, T. Rahman, I. Khan, I. Ullah, A. U. Rehman, M. Baz, H. H. Hamam, and O. Cheikhrouhou, "Student-Performulator: Student academic performance using hybrid deep neural network," *Sustainability*, vol. 13, no. 17, pp. 9775, 2021.

[17] H. Li, H. Wei, Y. Wang, Y. Song, and H. Qu, "Peer-inspired student performance prediction in interactive online question pools with graph neural network," *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*, pp. 2589–2596, 2020. [Online].

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS 2017)*, pp. 5998–6008, 2017.

[19] K. E. Arnold and M. D. Pistilli, "Course signals at Purdue: Using learning analytics to increase student success," *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge ACM*, pp. 267-270, 2012.

[20] V. Tinto, "Research and practice of student retention: What next?," *Journal of College Student Retention: Research, Theory & Practice*, vol. 8, no. 1, pp. 1–19, 2006.

[21] J. Lever, M. Krzywinski, and N. Altman, "Classification evaluation," *Nature Methods*, vol. 13, pp. 603–604, 2016.

Received on June 04, 2024
Accepted on November 03, 2024