FPT UNIVERSITY

Graduation Thesis Final Report

A University Student Dropout Detector Based on Academic Data – A case study at FPT University

Ngo Quang Hai; Nguyen Hoang Giang; Trinh Nhat Minh

Supervisor: Mr. Ngo Tung Son

April 25, 2023

Acknowledgment

We would like to express our sincere appreciation to our team members who have worked tirelessly throughout this project, generously sharing their knowledge and expertise. Their contribution has been invaluable in helping us achieve our objectives successfully.

We are also grateful to our supervisor, Mr. Ngo Tung Son, for his invaluable guidance, encouragement, and support throughout this project. His leadership, insight, and expertise were instrumental in ensuring the success of this project.

Furthermore, we would like to thank Mrs. Nguyen Thi Huong, the Head of Training Department, and Mrs. Nguyen Thi Thu Hang, the Head of Student Affairs Department, for their assistance in providing internal information and explaining the intricate details of the subject matter. Their inputs and suggestions have been of immense help to us.

Finally, we extend our gratitude to our teammates and supervisor for their contributions and support in completing this project. Their unwavering support and encouragement have been essential in making this project a success, and we are truly grateful for their assistance.

Abstract

Dropout at university has become a controversial problem in recent years since the crisis caused many severe consequences for students and universities. FPT University's (Hoa Lac campus) reputation and finances are also affected by student dropout. Therefore, we carried out our research on the early dropout prediction problem to provide school administrators with warning about students who have the risk of dropout so that the school can give proper solutions and support to those students. Our thesis is based on academic performance's influence on student dropout status. With FPT University, which includes information about students, subjects, and academic performance, we create a dataset that extracts features from the raw database to summarize critical information and partition features with similar characteristics into groups. In addition, we divide the problem into two phases based on FPT University program structure, which includes English preparation terms and Main terms. While FPT University's database consists of much valuable and massive information, the data dropout status is imbalanced, and many essential values are missing. With the generated datasets and the advance of deep learning neural networks, our research proposed three deep learning models: the convolution-based model (CNN model), the graph convolution network-based model (GCN model), and the tabular learning model (TabNet). Furthermore, compare the deep learning network with traditional machine learning algorithms: logistic regression (LR), support vector classifier (SVC), and light gradient boosting machine (LGBM) with feature selection supported. As a result, the proposed deep learning network performs better than tree-based algorithms, with 72% balance accuracy in the English preparation phase and 75% balance accuracy in primary terms. While TabNet trades off precision to achieve better recall, CNN and GCN models have more balanced results.

Keywords: Dropout prediction, Academic performance, Deep learning, Machine learning, Imbalanced, convolutional network, Graph Convolution network, Tabular learning, Logistic regression, Feature selection.

Table of Contents

List of Figures

List of Abbreviations and Acronyms

Abbreviations	Meaning
LR	Logistic regression
ANN	Artificial neural network
CS	Chi-Square
CSBA	Computer-supported behavior analytics
CSLA	Computer-supported learning analysis
CSPA	Computer-supported predictive analysis
CSVA	Computer-supported visualization analytics
CRF	Conditional Random Field
CNN	Convolution neural network
EDM	Education Data Mining
XGB	eXtreme Gradient Boosting
GR	Gain Ratio
GAT	Graph Attention Network
GCN	Graph convolution neural network
GNN	Graph neural network
IPF	Iterative Partitioning Filter
LGBM	Light Gradient Boosting machine
LSTM	Long short-term memory
MOOCs	Massive Open Online Courses
MSMOTE	Modified synthetic minority over-sampling technique
MTGNN	Multi-Topology Graph Neural Networks
NGSA II	non-dominated Sorting Genetic Algorithm
PSO	Particle swarm optimization
RBF	Radial basic function
RF	Random Forest
RNN	Recurrent neural network
SMOTEENN	SMOTE with Edited Nearest Neighbor
SNN	Spiking neural network
SVC	Support vector classifier
SVM	Support vector machine
SMOTE	Synthetic minority over-sampling technique
T-SNE	t-distributed stochastic neighbor embedding

1. Introduction

1.1 Topic background

The problem of student university dropout is a significant concern for both students and educational institutions. When students leave their university program before completing it without obtaining a degree, they miss out on valuable educational opportunities and face reduced career prospects. For universities, this issue can lead to revenue loss and reputational damage. Therefore, detecting the factors contributing to student dropout is crucial for improving student retention rates and ensuring the success of educational institutions.

In education, student dropout refers to students leaving school or university before completing their studies. Dropout can happen at any level of education, from primary school to university. Dropout can occur for various reasons, including financial constraints, academic difficulties, lack of motivation or interest in the subject matter, family or personal issues, and social or cultural factors. Detecting students who are at risk of leaving school prematurely at an early stage is crucial in mitigating the problem and directing appropriate interventions. Therefore, the timing aspect is essential. By identifying these students early, educators and policymakers can develop targeted interventions to address the underlying reasons for Dropout and provide support to help students stay in school and complete their studies. Dropout prediction can improve educational outcomes by allowing educators to intervene before students drop out, leading to better retention rates, improved academic achievement, and increased student opportunities.

FPT University is a private university in the Hoa Lac Hi-Tech Park in Hanoi, Vietnam. The university was established in 2006 as a member of the FPT Corporation, one of the largest IT companies in Vietnam. The Hoa Lac campus of FPT University is one of Vietnam's most modern and innovative campuses, equipped with high-class facilities and infrastructure. The campus covers an area of 50 hectares and offers a wide range of programs in various fields, such as Information Technology, Business Administration, Design and Digital Communications, and more. Over the year, FPT University's applications have increased significantly. However, there are still FPT students who choose to drop out. In 2020, 415 students decided to drop out, 579 in 2021, and 400 in 2022, including students who withdrew the application. Even though the number of dropout students is insignificant compared to those who remain, it still affects the university financially. Therefore, we must pay attention and give timely and appropriate support to potential dropout students.

1.2 The need for the research topic

Dropout from school or university can negatively affect individuals, communities, and societies. For instance, individuals who drop out of school or university may face limited job opportunities and lower earning potential than those who complete their studies. Drop out of school or university can limit an individual’s ability to move up the social ladder, may struggle to find stable and well-paying employment. In addition, Dropout has been linked to higher crime and incarceration rates, particularly among young males. Dropout can have high economic costs for society, including lost productivity and increased spending on social welfare programs.

In recent years, advances in machine learning and data analysis techniques have opened new possibilities for detecting student dropouts. By analyzing large amounts of data from various sources, such as student demographics, academic performance, and social interactions, researchers can identify the key factors contributing to Dropout and develop predictive models to help institutions intervene early and prevent students from dropping out. The motivation for this case study stems from the importance of understanding the factors contributing to student dropout and developing effective strategies for early intervention.

1.3. Research problem

Much research faces the following problem:

- **Class imbalance**: In many cases, the number of students who drop out of a program is much smaller than others, which can result in class imbalance, making it difficult to build accurate models that effectively predict student dropout.
- **Multifactorial**: There may be a wide range of factors that could contribute to a student's decision to leave school, including family circumstances, economic factors, academic performance, social environment, and many others. Therefore, it is often difficult to predict or determine with certainty whether a particular student will drop out, as the decision is likely to be influenced by multiple interacting factors.
- **Limited data availability**: Some schools or programs may not have comprehensive data on student demographics, academic performance, and other relevant factors impacting student dropout. This can limit the ability to develop accurate predictive models or identify effective interventions.
- **Lack of public dataset**: Refers to situations where there are not enough or publicly available datasets for a particular research or analysis task. This can be a significant problem, as researchers and analysts may be unable to perform their work accurately or effectively without sufficient data.

1.4. Research question

With the need for the research topic and the research problem in Sections 1.2 and 1.3, this study poses two research questions in the FPT University context:

- **Question 1**: How does academic performance influence student dropout?
- **Question 2**: Is Deep Learning better with the existing feature extraction technique?

1.5 Research objective

To develop a predictive model for student university dropout that addresses the challenges of class imbalance, to deal with multifactorial, limited data availability, and lack of public dataset by:

- We implement data cleaning and imputation techniques to ensure accurate, complete, and consistent structured datasets.
- We use a feature selection algorithm that balances the earliness in prediction and algorithm performance by identifying critical features influencing student dropout across multiple semesters.
- Using sampling techniques, modify the loss function to address the class imbalance and improve the accuracy of the predictive model.

1.6 Research scope

This project involves a case study using FPT University (Hoa Lac campus) dataset to analyze student dropout factors. The study will examine various data, including demographics and academic performance, to identify potential factors contributing to Dropout. The research limits the scope of FPT University (Hoa Lac campus) to Preparatory students and students from Information Technology majors to preserve the uniform subject structure.

1.7. Thesis outline

The following are the five sections of this thesis (excluding the abstract, reference, list of tables and figures, abbreviations, and acronyms list):

Section 1: Introduction summary of information about the research topic background, research problem, research questions, research objective, and research scope.

Section 2: The literature review presents an overview of education data mining and relevant research as the basis for our framework.

Section 3: Methodology discusses methods proposed for the thesis. Data preprocessing methods are clarified in this section and explained why those methods are used.

Section 4: Presents the proposed methods and experiment results and discusses the development for more insight into each approach.

Section 5: This section's conclusion responds to the research question by summarizing the highlights and analyzing thesis limitations for future work.

2. Literature review

2.1. Overview of educational data mining

Data Mining is discovering patterns and relationships in large datasets using various statistical, mathematical, and computational techniques. Data mining aims to extract valuable insights and knowledge from data that is difficult to obtain through manual analysis. Data mining involves two primary phases (Figure 1): the data preparation phase, the model, and the evaluation phase. In the data preparation phase, data is collected, prepared, cleaned, and transformed into a format that can be analyzed. In the modeling phase, algorithms are used to identify patterns and relationships in the data and to create models that can be used to make predictions or classify data.

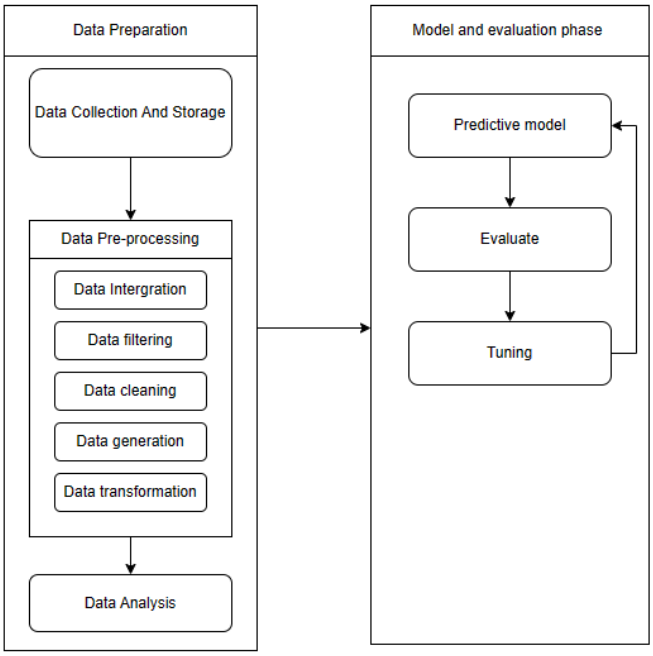

Figure 1: Pipeline of data mining process in EDM problem

Data collection is gathering and measuring information on variables of interest using established methods and tools. In Educational Data Mining (EDM), there are three main methods for collecting data: Surveys, observations, and secondary data analysis. Surveys use questionnaires or interviews to gather data from individuals or groups, which can be conducted in person, over the phone, through the mail, or online. For instance, Simón and Puerta [1] (2022) surveyed 894 first-year undergraduate students at the University of Granada, Spain. The survey included 54 questions related to 4 psychological, sociological, economic, and organizational subjects. Hegde and Prageeth [2] (2018) built a dropout system using survey questions involving academic information, demographical factors, psychological factors, social integration, and general social media information. Observations involve systematically recording behaviors, events, or activities as they occur. In EDM, this collecting method is primarily used in Massive Open Online Courses (MOOCs) since MOOCs own a massive log and recording system. Vasić et al. [3] (2015) and Haiyang et al. [4] (2018) both use the system logs to build the dataset for their research. Ding et al. [5] (2019), Dalipi et al. [6] (2018), KDD Cup 2015, … collected data from clickstreams and videos of MOOCs courses. Secondary data analysis is the use of existing data. For example, Berens et al. [7] (2019) used the Higher Education Statistics Act, student database of Bangor University was used by Gray and Perkins [8] (2019).

Data preprocessing includes five stages: data integration, data filtering, data cleaning, feature generation, and data transformation. The first stage in data preprocessing is combining data from different sources and presenting it in a unified view to provide a comprehensive data picture. In the filtering stage, a subset is selected from the large dataset based on specific criteria to reduce the dataset size and focus on a particular subgroup relevant to the specific task. The next stage is identifying and correcting or removing errors and inaccuracies in a dataset. New features are generated from the cleaned dataset to improve the machine learning model's performance. Several standard methods exist, such as creating ratio features or one-hot encoders. Alternatively, using their health insurance, Berens et al. [7] (2019b) determine students' migration by zip code, address, and financial status. To reflect abnormalities in academic performance, Pérez et al. [9] (2018b) generate enrollment age and standard deviation of Semester cumulative GPA as new features or Sandoval-Palis et al. [10] (2020)'s vulnerable features, which are the scoring base on socioeconomic. In the data transformation stage, data is converted from one format or structure to another to enhance the algorithm. The idea of converting interaction feature into a time-series format by accumulating the number of behaviors in a unit of time is favored by researchers since it can explore the sequence and time-related information that origin features do not process: Chen et al. [11] (2019), Berens et al. [7] (2019), Ding et al. [5] (2019), Qiu et al. [12] (2019), ... Duong et al. [13] (2022) utilized the aggregation and means of multiple features. Aside from standard transformation methods like aggregation and normalization, handling imbalance and feature selection are implemented occasionally to confront the dataset limitations.

In education, data mining involves extracting knowledge and insights from education data such as student records, learning management system data, assessment results, and student engagement data to identify areas where students struggle and develop interventions to improve student performance. By its application, EDM is grouped into four categories: computer-supported learning analytics (CSLA), computer-supported predictive analytics (CSPA), computer-supported behavioral analytics (CSBA), and computer-supported visualization analytics (CSVA) (Figure 2).

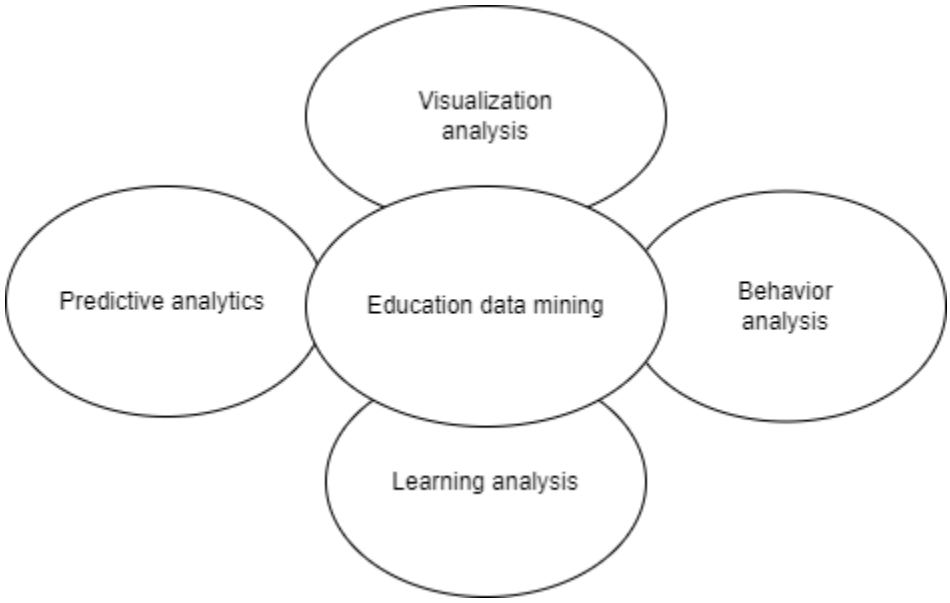

Figure 2: Data mining scheme in higher education

Visualization analytics is a technical inquiry used to visualize and analyze data by creating graphs, tables, charts, maps, etc., to highlight complex data patterns and relationships in an interactive way. This technique mainly identifies student behavior and performance patterns in EDM, providing the overview to enhance decision-making, course-building, and recommendation. Paiva et al. [14] (2018) proposed three visualizations: a segmented bar graph to measure the amount of interaction, an order weight of linear regression to highlight the most impactful feature, and combines interaction using association rule to show the best combination of features, to get an insight view of an online math course. MoocViz, created by Qu and Chen [15] (2015), is a framework that provides cross-course, cross-platform visual analytics for online course data. MoocViz can provide basic visualization like heatmaps and line charts, complex charts like state transition diagrams to demonstrate the learning analytics results, or visual analytic systems to visualize conversation dates in blogs.

Behavioral analytics in EDM involves the analysis of large and complex datasets such as engagement, participation, and interaction with learning material to identify patterns and trends related to student behavior, performance, and the learning process.

In EDM, learning analytics extract insight into how learners engage with education resources, improve academic performance, and recommend a learning path. Learning analytics enhances students' learning experience by providing instructors with actionable information. For instance, identify the challenge in student's study, provide suggestions and support, and recommend a course based on learner learning and behavior.

Predictive analysis education datamining refers to using data mining techniques to extract knowledge and insights from educational data, intending to predict student performance, behavior, and outcomes. This prediction can be based on various factors, including the student's past academic performance, demographic information, socioeconomic status, motivation, study habits, and environment. By analyzing these variables, machine learning models can identify patterns and correlations in student data, which can then be used to inform teaching practices and improve educational outcomes or identify at-risk students so

targeted interventions can be implemented, such as mentoring, tutoring, and academic counseling, to help them stay on track and succeed academically.

2.2. Dropout prediction

Based on the problem's dataset, the dropout prediction problem in EDM can be categorized into two classes: MOOCs and offline education. MOOCs (Massive Open Online Courses) are online courses that are open to a large number of students from around the world. These courses are typically free or low-cost and are offered by universities and other educational institutions. Since MOOCs usually systematically record student activities, clickstream features significantly influence the prediction problem. In contrast, offline education refers to traditional classroom-based instruction, where students attend in-person classes and interact with instructors and peers face-to-face. This form of education has been the dominant model for centuries and is still the most common in many parts of the world.

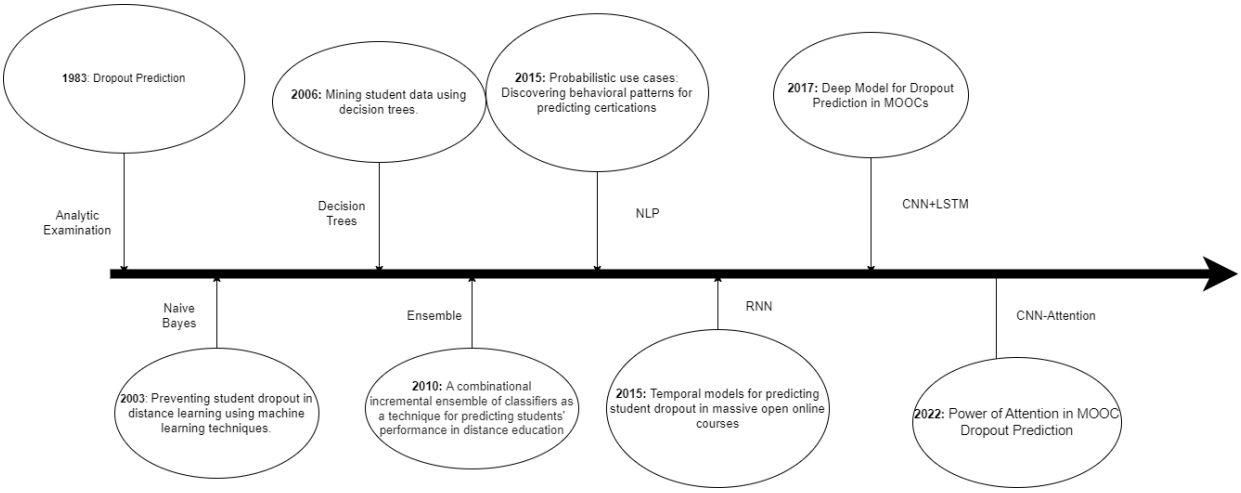

Figure 3: Timeline of the dropout prediction problem

Figure 3 illustrates the timeline of the dropout prediction problem. Before the 21st century, dropout prediction in education data mining does not attract much attention, and the typical approach for this classification problem is analytic examination. The first research on dropout prediction is [16] by Curtis and Jonathan, which analyzed the students' demographic and academic performance at a secondary school in Austin, Texas, to identify the student likely to drop out. In the early 21st century, researchers began applying machine learning algorithms to the prediction model. In 2003, Kotsiantis et al. [17] were the first to use machine learning algorithms for the education dropout problem instead of the analytic approach. In 2006, Al-Radaideh [18] proposed using the tree-based algorithm. In 2010, Kotsiantis et al. [19] applied an ensemble model. In the twenty-tens decade, deep learning networks proved to perform better than traditional algorithms and gained influence over the year. Therefore, while machine learning approaches still be favored by many researchers, other studies have begun to apply deep neural networks to the dropout prediction problem. In 2015, Coleman et al. [20] constructed the time series features and proposed using recurrent neural networks (RNN) to solve the time-series classification problem. In the same year, Fei and Yeung [21] (2015) applied Natural language processing (NLP) to the dropout prediction problem. Later researches mainly focus on comparing different machine learning algorithms, modeling students for a specific model like time series and graphs, and proposing a hybrid model to enhance the performance. In 2017, W. Wang et al. [22] offered a pipeline that includes transforming the clickstream feature into two dimensions array, and prediction

architecture used a convolutional layer to extract the most critical features followed by an Long short term memory (LSTM) layer to grasp the sequential feature of the input, which later be the baseline for other researches, especially in MOOCs problems.

2.2.1. Traditional supervise algorithm

Traditional supervise algorithm is a general approach for education data mining problems. Since those algorithms have short computation time, the education problems already have structured data, and usually, researchers can only access a limited amount of information.

Overview

Logistic regression, Support Vector Machines (SVMs), and Light gradient boosting machine (LightGBM) represent linear-based, kernel-based, and tree-based supervised machine learning algorithms that can be used for classification tasks.

Logistic regression

Logistic regression [23] is a popular statistical model used to predict the probability of a binary outcome, such as whether a student will drop out. Logistic regression can be employed to construct a predictive model for predicting student dropout that considers factors that may influence a student's decision to drop out, such as academic performance, attendance, socioeconomic background, and family support.

The basic idea behind logistic regression is to model the relationship between a set of predictor variables (also called independent variables or features) and a binary outcome variable (also called dependent variable or response) using a logistic function. The logistic function maps any real-valued input to a value between 0 and 1, which can be interpreted as the probability of the binary outcome being positive (i.e., Dropout in this case).

The logistic regression model uses a logistic function (or sigmoid function) to transform a linear combination of the input variables into a probability value between 0 and 1. The output of the logistic regression model can be interpreted as the probability that the event will occur given the input variables, in this case, is student dropout or not.

The formula for logistic regression can be expressed as follows:

$$\hat{y} = sigmoid(\theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n)$$

, where θ_0 is the bias term; $\theta_1, \theta_2, \dots, \theta_n$ are the coefficients or weights associated with each input variable $x_1, x_2, \dots, x_n$.

In logistic regression, the cost function is used to measure the error or loss of the model. The goal of the model is to minimize this cost function by adjusting the model parameters (coefficients) during the optimization process. The formula of the cost function (where m is the number of examples):

$$J(\theta) = -\frac{1}{m}\sum_{i=0}^{m}[y^{(i)} \log\left(\widehat{y^{(i)}}\right) + \left(1 - y^{(i)}\right) \log\left(1 - \widehat{y^{(i)}}\right)]$$

To find $\theta_0, \theta_1, \dots, \theta_n$ We need to repeat the gradient descent equation until convergence:

$$\theta_j \coloneqq \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \quad where \quad j = 0,1, \dots, n$$

, α is the learning rate, controls the step size at each iteration of the gradient descent algorithm. A learning rate that is too large can cause the algorithm to diverge, while a learning rate that is too small can cause the algorithm to converge slowly.

Support vector machine

Support Vector Machines (SVMs) [24] are a type of supervised machine learning algorithm that can be used for classification tasks, such as predicting whether a student will drop out. SVMs are based on finding the hyperplane that best separates the classes in the input space. The hyperplane is the decision boundary that maximizes the margin between the classes.

SVMs may be preferred over logistic regression for several reasons. First, SVMs are generally more effective than logistic regression in handling complex, non-linear relationships between the input features and the target variable. This is because SVMs can use non-linear kernels, such as the radial basis function (RBF) kernel, to transform the input features into a higher-dimensional space where the classes may be more separable.

Second, SVMs are less prone to overfitting than logistic regression, especially when the number of input features is large relative to the number of training samples. SVMs achieve this by maximizing the margin between the classes, which encourages the algorithm to find the decision boundary that is most robust to new, unseen data.

SVMs are particularly useful when the number of dimensions in the dataset is very high, making it difficult to visualize and analyze the data. In addition, SVMs can handle non-linearly separable datasets by mapping the input data to a higher-dimensional space using a “kernel” function without explicitly computing the coordinates of the data in that space. The most used kernels are the linear, polynomial, and radial basis functions (RBF). The linear kernel is used for linearly separable data, while the polynomial kernel is used when the decision boundary is curved. The RBF kernel is the most used as it can model more complex decision boundaries.

LightGBM

LightGBM [25] is a gradient-boosting framework designed to be highly efficient and scalable, making it well-suited for handling large datasets with high-dimensional input features. LightGBM may be preferred when predicting dropout students’ tasks over logistic regression or SVMs.

First, LightGBM can handle non-linear relationships between the input features and the target variable, just like SVMs. However, unlike SVMs, LightGBM does not rely on selecting a kernel function, which can be tedious and time-consuming. Instead, LightGBM constructs decision trees to model the relationships between the input features and the target variable and then combines the predictions of these trees in a gradient-boosting framework to make the final prediction.

Second, LightGBM is highly optimized for speed and memory efficiency, which can be crucial when dealing with large datasets with many input features. LightGBM achieves this by using histogram-based algorithms for binning the input features, reducing the memory needed to store the data, and speeding up the training process. Additionally, LightGBM uses a leaf-wise strategy to grow decision trees, which can result in a more compact tree structure and faster training times.

Finally, LightGBM includes several built-in mechanisms for handling imbalanced datasets, which can be helpful in the problem of predicting dropout students where the number of students who drop out may be significantly smaller than the number of students who do not. For example, LightGBM allows for using

class weights to give more importance to the minority class or oversampling or undersampling techniques to balance the class distribution.

At its core, LightGBM uses decision trees as weak learners, where each tree is trained on a subset of the data using a gradient-based optimization process. The resulting model is an ensemble of decision trees, where the final prediction is obtained by combining the outputs of all trees in the ensemble.

In contrast to other gradient boosting frameworks, LightGBM utilizes a novel approach called "Gradient-based One-Side Sampling" (GOSS) to select which samples to use during each iteration of the training process. This approach selects samples with large gradients while discarding samples with small gradients, thereby reducing the computational cost of training the model without sacrificing accuracy.

Feature selection

Feature selection identifies a subset of relevant features or variables from a dataset's more extensive set of features. Feature selection aims to improve the performance of machine learning models, reduce computational complexity, and enhance the interpretability of the results by selecting the most informative features relevant to dropout risk while ignoring redundant, irrelevant, or noisy features that may decrease the accuracy of the model.

Chi-square [26] feature selection can be used to identify the most relevant features associated with dropout risks, such as academic performance and demography. The technique involves calculating the chi-square statistic between each feature and the dropout variable and ranking the features based on their statistical significance. Features with higher chi-square values are more likely to be associated with dropout risk and, therefore, more important for prediction. The chi-squared test is a statistical analysis tool that helps to establish whether there is a significant relationship between two categorical variables. This test examines the contrast between the expected frequencies and those observed in a contingency table.

In a contingency table, the rows represent one categorical variable, and the columns represent another. Each cell in the table represents the count of observations that fall into a particular combination of categories.

The expected frequency is computed as follows:

$$Expected\ frequency = \frac{row\ total * column\ total}{grand\ total}$$

, where the row total is the sum of the counts in a particular row, the column total is the sum of the counts for a specific column, and the total is the total count in the table.

The chi-square test calculates the sum of the squared differences between the observed and expected frequencies, divided by the expected frequencies:

$$X^2 = \frac{\sum(Observed\ frequency - Expected\ frequency)^2}{Expected\ frequency}$$

The higher the chi-squared value, the more significant the association between the feature and the target variable, and the more likely the feature should be retained.

The resulting test statistic follows a chi-squared distribution, which can be used to determine the p-value and evaluate the null hypothesis that there is no association between the two variables.

The F-test is a statistical test that measures the difference between two variances. In the context of feature selection, the F-test compares the variance explained by a single feature to the variance explained by the

other features in the dataset. The F-test is calculated for each feature, and features with a high F-test score are considered more relevant for the regression model. The F-test score for a feature is calculated as follows:

Pearson correlation feature selection is a technique used to explore the linear relationship between two continuous variables, and it is commonly used in machine learning for feature selection. The Pearson correlation coefficient measures the strength and direction of the linear relationship between two variables, ranging from -1 (perfect negative correlation) to 1 (perfect positive correlation). In the context of feature selection, the Pearson correlation coefficient can be used to identify features highly correlated with the target variable. Highly correlated features may contain redundant information and can negatively impact the performance of machine learning models. By removing these features, we can improve the accuracy and efficiency of the model. We first compute the pairwise Pearson correlation coefficient between each feature and the target variable to perform Pearson correlation feature selection. We then select the essential elements with the highest absolute correlation coefficients.

Pearson's correlation coefficient is the covariance of the two variables divided by the product of their standard deviations:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$

, where

- $cov(X,Y)$ is the covariance
- σ_X is the standard deviation of X
- σ_Y is the standard deviation of Y

Past researches

Many researchers prefer the traditional supervised machine learning algorithm to solve the dropout prediction problem in MOOCs and offline education. Nuanmeesri et al. [28] (2022) and Panagiotakopoulos et al. [29] (2021) both compared the efficiency of different machine learning algorithms, which include LightGBM, extremely randomized trees (Extra) algorithm, ridge classification method (Ridge), gradient boosting classifier, Random Forest (RF), logistic regression (LR), classification and regression tree (CART) algorithm, AdaBoost boosting algorithm, linear SVM with stochastic gradient descent (SVM-SGD) algorithm, decision tree, SVM, MLP, etc. Both studies and similar research results usually show that tree-based models or LightGBM perform better than different algorithms. Jin [30] (2021) improved the traditional ML model by intelligent optimization to initialize model weight. Chen et al. [11] (2019) partitioned students' historical behavior records by week, which optimizes information by week. The study implemented an embedding feature selection approach, which involves a decision tree as the embedding model for feature selection and uses the DT structure to create the ELM model.

In offline education, the most common type of information in EDM is students' historical academic performance and demographic since those variables are required for all higher education systems. Berens et al. [7] (2019) composed multiple classifiers: logit regression model, MLP, and decision tree algorithm utilizing the AdaBoost algorithm, and investigated the proposed model on two different university datasets, which is collected according to the Higher Education Statistics Act. Pérez et al. [9] (2018b) integrate admission information, transcript records, and graduation dates and grouped courses based on their department to create a dataset that includes 31 features from 762 students enrolled in the Systems Engineering Program at a private university in Bogota, Colombia. The study then analyzed the efficiency of the decision tree, random forest, logit regression, and Native Bayes on their dataset. Baranyi et al. [31]

(2020) researched the influence of students' high school performance on Dropout. Permutation importance and SHAPE value were used to determine the feature importance of two predictive models XGBoosted and fully connected neural network. Segura et al. [32] (2022) ranked and selected features from socioeconomic variables, enrollment variables, and academic performance of 1st Semester using correlation-based feature selection and experienced different prediction model, which is ANN, SVM, KNN, DT, and LR. Yaacob et al. [33] (2020) implemented k-NN, Naive Bayes, Decision Tree, Neural Network, Logistic Regression, and Random Forest models to predict and analyze the relationship between each subject and Dropout. Tenpipat and Akkarajitsakul [34] (2020) explored the relationship among students' personal information, academic records, and academic status from their previous semesters and compared decision tree, random forest, and gradient boosting algorithm performance. Lottering et al. [35] (2020) transformed the dropout prediction problem from a classification problem into a regression by defining the risk ratio features of students, which is the number of credits accumulated over the years. Then the study compared the performance of various machine learning models using students' demographic and past academic performance. Da Silva et al. [36] (2022), Yukselturk et al. [37] (2014), and many other researchers also implemented and compared multiple machine learning algorithms to analyze their influence on the study's dataset. From those comparison experiences, most results show that tree-based algorithms usually perform better than machine-learning algorithms.

Besides academic performance and demographic features, which school administrators automatically record, some studies focus on collecting and researching the influence of other factors that cause student dropout. For example, Gray and Perkins [9] (2019) investigated students' attention influence over Dropout. The research also proposed a student's attention modeling and ANN model for the prediction task. Sandoval-Palis [10] (2020) investigated the economic and academic variables' impact on Dropout in the first year. The study generated a vulnerability index, a sum weight value of the synthesis of monetary values, filtered features using correlation and independence analysis, and implemented and conducted experiences on Logit regression and ANN prediction model. Mujica et al. [38] (2019) analyzed the impact of student status, motivation, resolve, and discipline, ... on Dropout. Simón and Puerta [1] (2022) surveyed four main factors influencing Dropout: psychological, sociological, economic, organizational, and interactions to collect data and analyze the information using chi-square and student t-test for feature selection and logit regression for dropout prediction. Korenkova et al. [39] (2020) research explored the influence of psychological factors on student dropout rates.

In addition, many studies investigate the efficiency of different machine learning models in other problems of education datamining. For instance, Amornsinlaphachai [40] (2016) compared the efficiency of seven algorithm algorithms that are Artificial Neural Network, K-Nearest Neighbor, Naive Bayes, Bayesian Belief Network, JRIP, ID3, and C4.5 in predicting the performance of a learning group, then developed a web-model for cooperative learning using C4.5, the best-performing model. Pallathadka et al. [41] (2021) and Yağcı [42] (2022) both implemented multiple data mining algorithms for their university. Duong et al. [13] (2022) built a warning system to warn students about their performance at the beginning and the end of a semester. The system uses the lightGBM model to predict students' performance and warn them if their grade is lower than a standard threshold. In addition, the study groups the past academic performance features of the student into Semester and type of exam to gain more detailed information. While many studies try to handle the performance prediction task in university, Siddique et al. [43] (2021) and Roslan [44] (2022) focus on predicting the performance at the secondary level, which tends to be the foundation of student's learning progress in higher education. Roslan [44] (2022) implemented different DM techniques to predict secondary school students' performance in English and Mathematics subjects for the Malaysian Certificate of Examination (MCE) in 2021. The study results show that DT and Naïve Bayes (NB) techniques have the best predictive performance for English and Mathematics subjects, respectively. In

addition, the result also indicates that historical academic performance is a critical feature in the prediction. Siddique et al. [43] (2021) compare combinations of single and ensemble-based classifiers, where the best performer is the fusion of MultiBoost with MLP. To improve the performance of the prediction model, some researchers applied feature selection using filter methods such as Duong et al. [13] (2022) used Pearson correlation, Rachburee and Punlumjeak [45] (2015) compared different methods: greedy algorithm, information gain ratio, chi-square, Minimum Redundancy Maximum Relevance (mRMR), or XGBoost feature importance in Dalipi et al. [48] (2018). On the other hand, Farissi et al. [46] (2020) and Alraddadi et al. [47] (2021) enhanced prediction performance in exchange for computing expenses by developing a hybrid model using wrapper methods in feature selection, where both researchers applied optimization algorithm as the feature selection model which is the genetic algorithm and Binary Teaching-Learning Based Optimization. Besides feature selection, different hybrid methods are proposed to increase the model's efficiency. For instance, Niyogisubizo et al. [49] (2022) improved the model's performance by stacking ensemble algorithms: random forest, XGBoost, GradientBoost, embedding output, then passing the stacked embed through the FNN layer. Both Hassan et al. [50] (2022) and Almasri et al. [51] (2020) 's approach uses the clustering method. Hassan et al. [50] (2022) proposed a hybrid model of k-means and LassoCV to improve the original linear regression model. In addition, the Community of Inquiry framework (CoI) is also utilized as the student feature modeling since the framework focuses on social features. Almasri et al. [51] (2020) partitioned students into groups based on their performance, then used the prediction model for each cluster.

In conclusion, since education data mining problems usually have structured datasets and limited amounts of data, the traditional supervised algorithm is efficient and popular in those problems. Because of that, in this thesis, we implemented and compared the performance of some traditional machine learning algorithms.

2.2.2 Deep learning approach

While the traditional algorithm is suitable for dropout structure data of student probation problems, standard algorithms cannot explore the inexplicit information of features that may influence student dropout status. A solution for this problem is to represent students' data in latent space to learn the data's explicit features. In addition, deep learning can automatically discover and extract relevant features from the input data, whereas other methods often require feature selection or engineering. Figure 4 illustrates a sample of neural networks.

Overview

Neural network

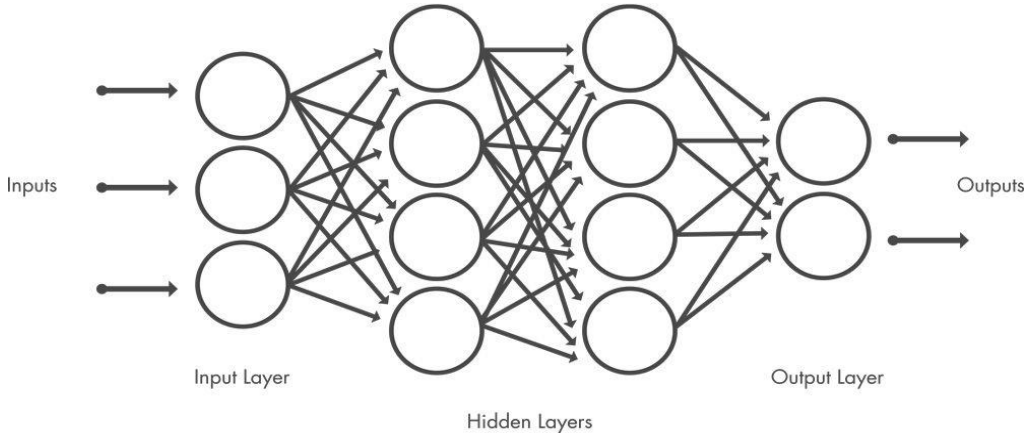

Figure 4 A sample neural network

A neural network is a machine learning algorithm designed to learn patterns and relationships in data and use this knowledge to make predictions or decisions. It is inspired by the way the human brain works, with neurons that communicate with each other to process information. A neural network contains:

Unit: A neuron or unit is the basic building block of a neural network, denoted as a round shape in Figure X. It receives one or more inputs, performs a computation on those inputs, and produces an output. The output of one neuron can be fed as input to another neuron.

Layer: A layer in a neural network is a group of neurons that process the same input type. A neural network has several layers, including input, output, and hidden layer(s).

Activation function: An activation function is a non-linear function that is applied to the output of each neuron. It allows the network to model complex, non-linear relationships between inputs and outputs. Common activation functions include the sigmoid, tanh, and ReLU functions. Figure 5 illustrates sigmoid, Tanh, and ReLU activation functions.

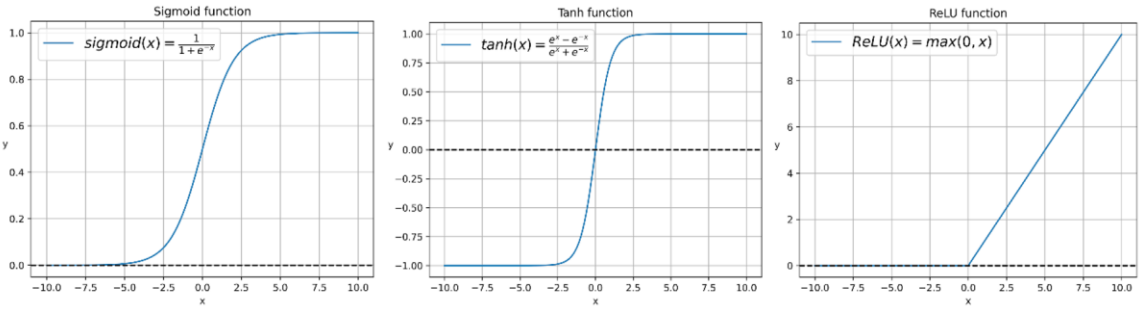

Figure 5 Sigmoid, Tanh, ReLU activation function

In a neural network, data is fed into the input layer and processed through a series of hidden layers that transform the data and extract important features. Each layer contains a set of neurons that perform a simple computation, such as a weighted sum of the inputs, followed by an activation function. The output of each layer is then passed to the next layer until the final result is produced.

During training, the neural network learns to adjust the weights and biases of the neurons in each layer to minimize the difference between the predicted and actual outputs. This is done using backpropagation, which involves computing the gradient of the loss function for the weights and biases and using this information to update the parameters.

Convolutional neural network

In a convolutional neural network (CNN) [52] , the input data passes through one or more convolutional layers before reaching the fully connected (FC) layer(s). CNNs are inspired by the structure and function of the human visual cortex and are designed to learn and extract features from raw image data automatically. The critical component of a CNN is the convolutional layer, which applies a set of filters to the input image to detect different features, such as edges, shapes, and textures. The output of the convolutional layer is then passed through a series of pooling and activation layers to reduce the dimensionality of the data and increase its nonlinearity. Figure 6 illustrates the operation of the convolution layer.

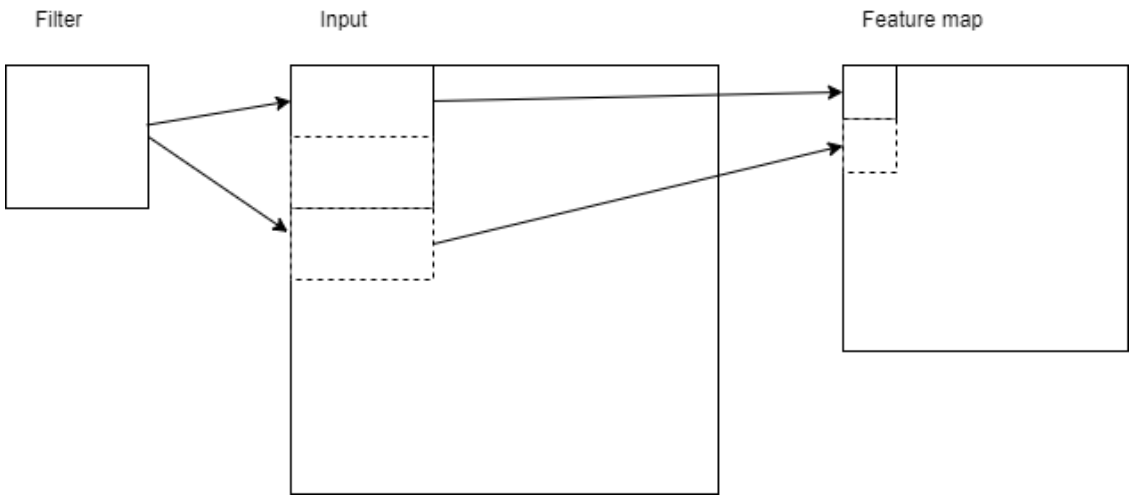

Figure 6 Example of a Filter Applied to a Two-Dimensional Input to Create a Feature Map

Past research

While there, many research studies prefer classical machine learning models like regression, kernel-base, tree-base, and ensemble models in EDM because of their time computing and data structure. Some research has been conducted on the architecture of neural networks in the context of the pedagogical domain. For instance, Giannakas et al. [53] (2021) experimented with different activation and optimizer combinations of a two-layer neural network for their dataset in the early prediction problem. Salam et al. [54] (2022) and Thaher and Jayousi [55] (2020) implemented simple neural networks, which are CNN, and Multilayer Perceptrons (MLPs), respectively. On the other hand, many studies propose a different variant of the neural networks model, which may suit their study dataset and scope. For instance, with the idea of concentrating the embedding features of two 2D CNN models, Poudyal et al. [56] (2022) propose a hybrid 2D CNN model that takes converted 2D data, outperforming other classical models of accuracy. Instead of the traditional Artificial Neural Network Model (ANN), Liu et al. [57] (2022) established an evolutionary spiking neural network (SNN) and compared it to six different data mining algorithms. Unlike traditional ANNs, which typically use continuous-valued activations, SNNs use discrete, binary spikes to communicate information between neurons. In an SNN, neurons receive input from other neurons or the environment, integrating this input over time. When the neuron's internal state reaches a certain threshold, it generates a spike, which is transmitted to other neurons downstream. The downstream neurons then integrate these spikes, which may generate their spikes. They have several advantages over traditional ANNs, including improved energy efficiency, more biological realism, and the ability to encode information during spikes. Considering student privacy protection, B. Xu et al. [58] (2022) propose a framework based on federated transfer learning for students' grades classification. In addition, the framework also introduces the domain adaptation method, which supports maintaining the feature distribution after feature extraction. To deal with the imprecision of academic performance, Hussain et al. [59] (2020) propose using a fuzzy-based neural network (FNN). In addition, since gradient-based learning methods limit FNN's performance, a metaheuristic learning approach Henry Gas Solubility Optimization (HGSO) algorithm, replaces the tunning parameter task. Instead of past academic performance, Gao et al. [60] (2022) study the influence of students' cognitive states on performance prediction. The student's skill proficiency is first modeled using the survey answers in this research. Additive attention is used to obtain the similarity between the student's skill proficiency and the corresponding problem and add the skill interaction model. Finally, deterministic input, noisy-and (DINA), a discrete Cognitive diagnosis model, used students' problem proficiency to predict their scores.

In exchange for performance improvement, deep learning models require a large amount of data and computing time. Since MOOCs system has student's behavior log data and a massive amount of internet student, many studies implement neutral network models on MOOCs prediction problem. On the other hand, offline academic has stricter enrollment requirements and limited features since the system usually updates after a semester. Therefore, the offline education dropout prediction problem takes little attention in the deep learning network.

2.2.3 Sequence deep learning network approach

Because features are captured continuously for each student over time, dropout prediction is essentially a time series prediction problem (Fei & Yeung, 2015b). Especially in MOOCs problems because MOOCs' clickstream log system records student behavior in short time units (day, hour, minutes). To utilize the time information, many researchers implement a time series-based algorithm.

Overview

Recurrent Neural Network

Recurrent Neural Network (RNN) is a neural network that processes sequential data, such as speech, text, and time series data. Unlike feedforward neural networks, which process input data in a single pass, RNNs can maintain an internal state or memory of previous inputs, which allows them to capture temporal dependencies in the data. The critical feature of RNNs is their ability to maintain a hidden state that captures information about the previous inputs in a sequence. This hidden state is updated at each time step using a recurrent function, combining the current input with the previous hidden state to produce a new one. The output at each time step is computed based on the current hidden state. Figure 7 shows the primary mechanism of RNN.

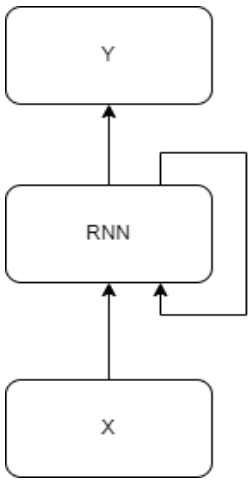

Figure 7 RNN pipeline

Long Short-Term Memory

Long Short-Term Memory is (LSTM) a variant of RNN. The critical innovation of LSTM networks is their ability to selectively store and forget information over long periods, which is particularly useful for processing sequences with long-term dependencies. LSTMs achieve this through a set of gates that regulate

the flow of information through the network. These gates include the input gate, which controls the flow of new information into the network, and the forget gate, which contains the old data out of the network. And the output gate controls the information flow to the next layer in the network. LSTMs have been used successfully in various applications, including speech recognition, language translation, and sentiment analysis. They are beneficial in applications where the context of a sequence is essential for making accurate predictions, such as in natural language processing or financial forecasting.

Past researches

In this approach, it is necessary to transform the behavior interaction feature into a time-series matrix to apply a time-series algorithm that takes maximum advantage of time information. Haiyang et al. [4] (2018) propose a time series forest algorithm, while other researchers like Ding et al. [5] (2019), Tang et al. [62] (2018), Fei & Yeung [61] (2015b), Xiong et al. [63] (2019), and Mubarak et al. [54] (2021b) employed time-series neuron networks model like LSTM or RNN. Jin [64] (2020) experienced support vector regression (SVR) on a time-series dataset and optimized the SVR parameter using improved quantum particle swarm optimization (PSO) algorithm. Qiu et al. [12] (2019) developed a CNN with three layers and one fully connected layer for the time-series dataset. Based on W. Wang et al. [22] 's pipeline, Feng et al. [65] (2019) fusion of context information and learning behavior by attention layer. In addition, Feng et al. [65] deployed their model on the university system. Another variant of W. Wang et al. [22] 's framework is in which Wu et al. (2019) add an SVM layer to identify the class border. Yin et al. [67] (2020) and Cai and Zhang [68] (2021) both improve the CNN-LSTM framework by adding an encoder and decoder, which use the attention layer and word2vec layer, respectively. Li et al. [69] (2022b) developed an end-to-end deep learning model that automatically extracts features from multiple students' heterogeneous behaviors to predict academic performance. The deep learning pipeline includes 3 phases: first, a one-dimension CNN to extract embedding features, then LSTM to learn time-series features, and lastly, a two-dimension CNN is applied to learn the correlation between different behavior. Uliyan et al. [70] (2021) indicate that the prediction of student retention is possible with a high level of accuracy using LSTM and Conditional Random Field (CRF) deep learning techniques in offline education.

While the sequence-based algorithms can utilize the time-series property of the education predictive problem, because the time unit of offline education data is large, it is hard to optimize the balance between the prediction time and the model’s accuracy.

2.3 Graph neutral network

In recent years, graph neural networks (GNN) have received attention because of their potential to deal with graph-structure data. GNNs are a type of neural network that operates directly on graph-structured data, allowing for end-to-end learning of graph representations that can be used for tasks such as graph classification. In contrast to traditional neural networks, designed to operate on data structured as grids or sequences, GNNs can learn and reason about the relationships between entities in a graph. Based on the purpose of the model, GNN can be categorized into three types: graph-level, node-level, and edge-level.

Graph Convolutional Networks (GCNs) are a basic variant of GNN. The idea behind GCNs is to use a convolutional operation that can be applied to graphs, similar to how convolutional layers are used in traditional image-processing tasks. However, since graphs are non-Euclidean data, traditional convolutional layers cannot be directly applied. Instead, GCNs use a variation of the convolutional operation, called graph convolution, defined in the spectral domain of the graph Laplacian.

In a GCN, the input is a graph represented as an adjacency matrix A and a node feature matrix X. The adjacency matrix represents the edges between nodes. In contrast, the feature matrix contains features for

each node. These matrices are multiplied to obtain a graph signal that is then convolved with a filter parameter θ. This filter is learned during training and controls how the convolution operation is applied to the graph signal. The output of the convolution operation is a new set of node features, which can be fed into additional layers of the GCN.

The convolution operation in GCNs can be expressed mathematically as:

$$H = \sigma(D^{-\frac{1}{2}}AD^{-\frac{1}{2}}X\theta)$$

Where H is the new set of node features, σ is a non-linear activation function, D is the diagonal degree matrix of the graph, and θ is the trainable filter parameter. The multiplication by $D^{-\frac{1}{2}}$ On both sides of the adjacency matrix, A is used to normalize the rows and columns of the adjacency matrix, ensuring that the convolution operation is scale-invariant.

After applying the convolution operation, the output node features can be aggregated into a single graph-level representation, which can be used for downstream graph classification tasks. This aggregation can be done in various ways, such as taking the max, mean, or sum of the output features across all nodes.

In EDM, some researchers employed GNN in students' performance prediction problems. For example, M. Li et al. [71] (2022) represent each student as a node and build multiple relationship graphs using the different distant measures: cosine similarity, Euclidean Distance, and Manhattan Distance, then propose Multi-Topology Graph Neural Networks (MTGNN), which use an attention layer to fuse multiple graph embeddings, for the student performance prediction task. The pipeline considers the secret relationship between students with similar characteristics, further enhanced using multiple similarity measurements. Hu and Rangwala [72] (2019) applied Graph Attention Network (GAT) to predict the performance of a future course using the information of prerequisite subjects. Nakagawa et al. [73] (2019) transformed the knowledge structure into a graph to convert the knowledge tracking task into a node-level classification problem in GNN. They proposed a GNN-based knowledge tracing method to handle the situation.

2.4 TabNet

TabNet [74] is a deep learning model designed for tabular data written by Arik and Pfister from Google Cloud AI, which is data organized in a table with rows and columns; it is a combination of techniques such as feature selection, feature masking, and attention mechanisms to learn from tabular data. It is a supervised learning model that can be used for both classification and regression tasks. TabNet can provide interpretable results. The model can identify essential features or columns in the input data and provide insights into why it made specific predictions.

One of the critical benefits of TabNet is its ability to handle numerical and categorical features without requiring one-hot encoding. This thing can be beneficial when dealing with large datasets that contain a mix of data types, as it can help reduce the feature space's dimensionality and improve model performance.

In addition, TabNet incorporates a sparsity-inducing regularization term during training, which helps to prevent overfitting and improve generalization performance. This can be especially important when dealing with imbalanced datasets, such as those familiar with predicting student dropout.

The main ideas from TabNet are: it has Attentive Transformer Blocks that can instance-wise feature selection for better performances, built-in explainability derived from masks, and efficient learning capacity; the Feature Transformer Blocks contain basic MLP blocks with Gated Linear Unit Activation and

shared layers across different steps; it is act like sequential steps that can mimic ensembling and large model capacity. Figure 8 is the visualize of TabNet network.

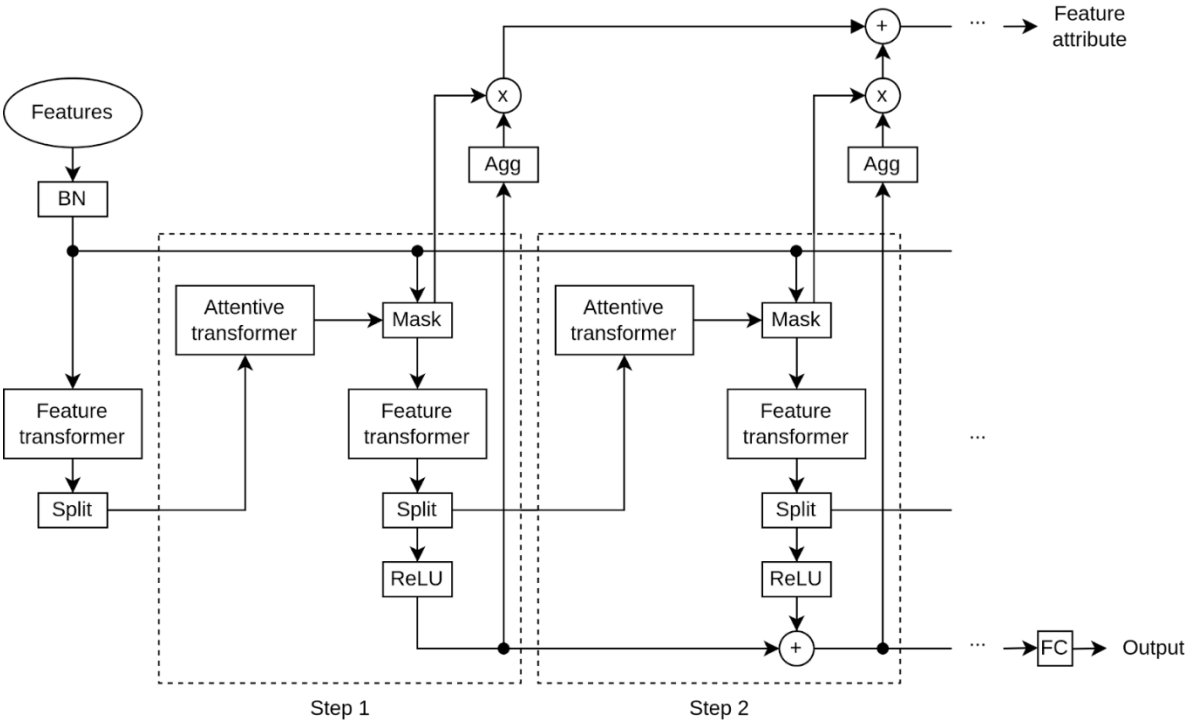

Figure 8 TabNet encoder architecture with two steps.

The architecture starts with a Batch Norm layer and the raw input features of size $f \in \mathbb{R}^{B*D}$ (where B is the batch size, D is feature dim) go through this layer. The TabNet model is based on sequential multi-step processing, with N_{steps} decision steps and all the steps are the same: Attentive transformer creates a mask (Mask block) that will mask our feature for the next Feature Transformer block then each Feature Transformer block creates both predictions and the following feature for the Attentive Transformer block in the next step. The model output is summed in every step prediction and passed into the Linear layer for classification or regression tasks. The Mask block also gives information about the model used to make the predictions.

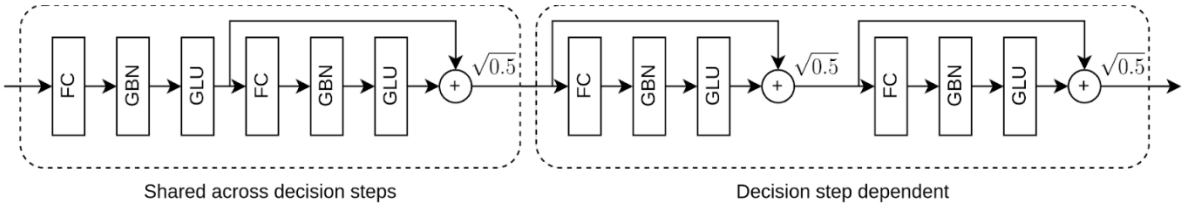

Figure 9 Feature transformer block. An example of two layers is shared across all decision steps, and two layers are decision step-dependent

As shown in Figure 9, each Feature Transformer block consists of a certain amount of steps and different kinds of blocks in each step. In each block, there is a fully connected layer (FC), a ghost batch normalization (GBN) layer, and a Gated Linear Unit (GLU) activation function. A GLU function is $GLU(x) = \sigma(x) \odot x$, it is the sigmoid times the input features. Each of these blocks may be shared or independent. A Feature

Transformer should comprise layers shared across all steps for parameter-efficient and robust learning with high capacity. There is a skip connection between two consecutive FC - GBN - GLU blocks. Normalization $\sqrt{0.5}$ the summation of the skip connection and the presentation after the FC - GBN - GLU block helps to stabilize learning by ensuring that the variance throughout the network does not change dramatically. The input size is $n_{features}$ to initial, output size = $n_d + n_a$, nd is the decision dimension and n_a is the attentive dimension (done by Split Block that will split the representation for the next step). The decision goes through the ReLU activation function, and the attentive part goes to the Attentive Transformer in the next step.

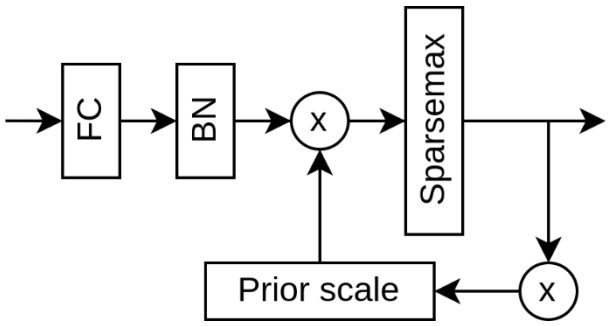

Figure 10 Attentive transformer block.

The Attentive Transformer block has an FC layer and BN layer, a Prior scale layer, and a Sparsemax layer.

The prior gives the understanding and how much use the features in the previous step. In the beginning, the first prior is a matrix full of ones: $P[0] = 1^{B*D}$, for the next step, the prior is going to be a multiplication of all the previous steps: $P[i] = \prod_{j=1}^{i}(\gamma - M[j])$ where M is a mask of size $\mathbb{R}^{B*D}$, γ is the parameter of this architecture which is always greater than 1. If we set γ close to 1, the model does not use the same features at every different step (which is, we have different features for every step). If we set larger γ, we can reuse the same feature at every step.

SparseMax is a sparser version of the SoftMax function that has the following properties:

- $\sum_{i=1}^{n} sparsemax(x)_i = 1, \forall x \in \mathbb{R}^n$
- Many dimensions will be zeros: instance-wise feature selection.

After the Sparsemax layer, we have output many zeros values; the rest will sum to 1. That is how the mask is created, directly applied to the input feature.

The input size of the Attentive transformer block is $B * n_a$, the output size is $B * n_{features}$ is the learnable mask M so we can apply the masking to the Feature transformer in the next step: $M * f$

TabNet incorporates a sparsity-inducing regularization term during training, which helps to prevent overfitting and improve generalization performance. The sparsity regularization has the following formula:

$$L_{sparse} = -\sum_{i=1}^{N_{steps}} \sum_{b=1}^{B} \sum_{j=1}^{D} \frac{M_{b,j}[i] \log\left(M_{b,j}[i] + \varepsilon\right)}{N_{steps} * B}$$

, where ε is a small number for numerical stability, $M_{b,j}[i]$ is the value of the learnable mask at step i

The overall loss function:

$$L = L_{CE} - \lambda_{sparse} * L\_sparse$$

, where L_{CE} is Cross-Entropy loss, λ_{sparse} is the regularization term for L_{sparse}.

3. Methodology

3.1. Model modeling

The overview model aims to predict whether a student will drop out in the following semesters, using features from all previous semesters. Key variables used in the model include:

- k is the chosen semester for prediction
- S is the number of students
- D is the number of department
- C_s is the set of courses that student s^{th} have learned before semester k
- h_s is the demographic feature of students s^{th}
- t_c is the features of course c^{th}
- $f_{s,d}$ is the feature of student s^{th} in department d^{th}
- $X_s = h_s f_{s,1} \dots f_{s,D}$ represent the feature of student s^{th}
- $G_s = \{V_s, E_s\}$, where $V_s = \{t_c, c \in C_s\}$ and $E_s = M^s$ is the represent the feature of student s^{th} in the graph.

The model will use the feature set (X_s or G_s), which was extracted from $(k-1)$ previous semesters to predict whether students drop out after $(k-1)$ semester.

3.2. Dataset preprocessing

3.2.1. Data cleaning

Data cleaning is a crucial step in preparing data for analysis and decision-making, as it ensures the accuracy and reliability of the data. In this case, we need to remove specific categories of students from the dataset to improve its quality. These categories include:

- Students who have never taken any courses: These students have no grades, and their presence in the dataset can lead to misleading conclusions.
- Remove students with conflict information: For instance, those who have negative or small age.
- Remove students with specific conditions: Students who exit the system but not the dropping out the case.
- Remove students with missing key features.
- Fill the missing feature with the mean value of columns.

3.2.2. Data transforming

This thesis builds a new prediction dataset from the raw Dataset provided by FPT University. Because of the transformation of curriculum over the years and features variations of students' learning results, it is necessary to represent similar features to new ones.

Based on FPT University's education system, each subject's grade consists of four component grades: participation (student's behavior and attendance), progress (assignment and progress test), practice (practice exam), and final (final exam). Each component has a weight, which sum of four weights is one.

The total grade of a subject is computed as the total of each element multiplied by corresponding weights. Finally, the student's GPA (Grade Point Average) is calculated by the following formula:

$$GPA = \frac{\sum_1^n avg_i * credit_i}{\sum_1^n credit_i}$$

Where:

- n: the number of subjects
- avg_i : the average grade of the subject i^{th}
- $credit_i$: the credit of subject i^{th}

In this research, students academic grade is partitioned by the department. The new features are grouped and calculated by the following formulas:

Group GPA: The list of features contains the GPA of each department and the total average grade of all subjects in the selected department.

$$s(d) = \frac{\sum_1^{N(d)} avg_{d,i} * credit_{d,i}}{\sum_1^{n(d)} credit_{d,i}}$$

$$s0(d) = \frac{\sum_1^{N(d)} avg_{d,i}}{N(d)}$$

Group Average Component Grades (ACG): Group ACG composes the average of each type of component grade in the responding department.

$$avg(j) = \frac{\sum_1^{N(d)} score(j)_{d,i} * coef(j)_{d,i} * credit_{d,i}}{\sum_1^{N(d)} credit_{d,i}}$$

Group Coefficient Component Grades (CCG): Group ACG composes the average of each type of component weight in the responding department Data transform.

$$coef(j) = \frac{\sum_1^{N(d)} coef(j)_{d,i} * credit_{d,i}}{\sum_1^{N(d)} credit_{d,i}}$$

Group Ratio: Student learning grade rank by the following rule: 9.0–10.0 (A+), 8.0–9.0 (A), 7.0–8.0 (B+), 6.0–7.0 (B), 5.0–6.0 (C+), 4.0–5.0 (C), 3.0–4.0 (D), <3 (F). The group ratio is the list of each rank ratio.

$$ratio(r) = \frac{rank(r)_d}{\sum_1^r rank(r)_d}$$

Where:

- d: is the representative of the department.
- N(d): is the number of subjects in d^{th} department
- avg_i : the average grade of the subject i^{th}
- $credit_i$: the credit of subject i^{th}

- $score(j)_{d,i}$: the j[th] component grade for subject i[th] in department d^{th}
- $coef(j)_{d,i}$: the j[th] component weight for subject i[th] in department d^{th}
- $rank(r)$: the number of grades that have rank r in the deparstment d^{th}

In addition, from the built Dataset, we generate new features, which are composed based on our origin feature to extract hidden information.

avg_final sum of component grades in the same department subject (sum of group ACG).

coef_final sum of component weights in the same department (sum of group CCG).

avg10 is the converted component average grade calculated using avg_final divided by coef_final.

3.2.3. Data filtering

Due to the differences in the educational systems among the big majors, it is necessary to conduct independent evaluations of the different majors. The scope of the evaluation will be narrowed down to students in the field of Information Technology. Even though these students are in the same area, the curriculum of the previous generations, specifically before Generation 13, exhibits a significant degree of inconsistency. Furthermore, there is also an uncontrollable amount of missing data for Generation 13. Therefore, **data only used academic information from the 13[th] generation of information technology major and all English preparation info with all students.**

3.2.4. Data sampling

Based on the highly imbalanced dropout dataset, with a significantly smaller number of students who dropped out than those who did not. Data sampling techniques were applied to address this imbalance and prevent the model from being biased toward the majority class. Specifically, the Synthetic Minority Over-sampling Technique (SMOTE) and the combination of SMOTE and Edited Nearest Neighbor (SMOTEENN) were used to increase the number of dropout instances in the dataset. SMOTE creates synthetic samples of the minority class by interpolating between existing samples. At the same time, SMOTEENN first applies SMOTE and then removes samples identified as noise or classified as belonging to the majority class by the nearest neighbor classifier. This approach helps to improve the model's performance by providing a more balanced dataset for training and testing.

3.3. Predictive Model

Our study implements and compares the performance of the four approaches in the FPT University dropout prediction problem.

The first approach is centered around utilizing LR, SVC, and LGBM models with input features derived from X_s. Furthermore, the research incorporates a filter method for feature selection to enhance model performance. It is worth noting that the filter method is exclusively leveraged in this particular approach.

The second approach involves employing a custom CNN model illustrated in Figure 12.This model comprises two blocks of 1d convolution layer with the batch norm, followed by a fully connected layer. Given the structured nature of the data and its relatively small scale, it is imperative to avoid overfitting by limiting the depth of the model.

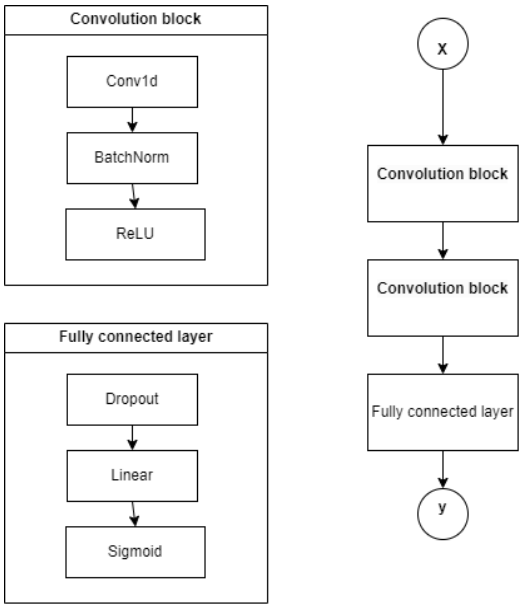

Figure 12 CNN architecture

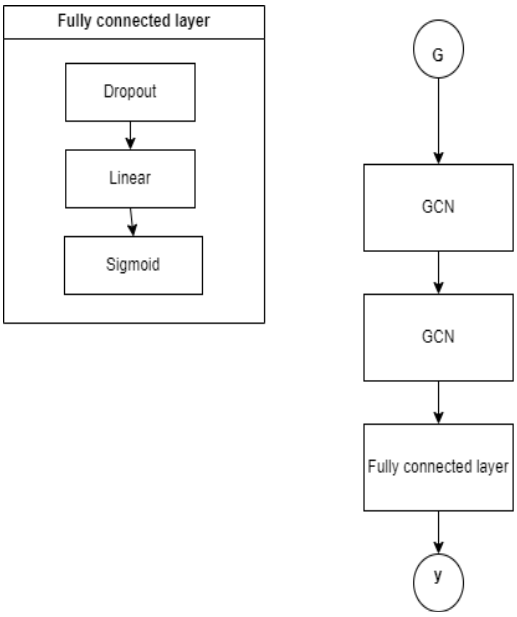

Figure 11 GCN architecture

In the third approach, the research employs TabNet for prediction, where X_s serves as the input variable.

Finally, our fourth approach involves transforming X_s into a graph structure G_s, which serves as the input for the GCN model depicted in Figure 11. Analogous to the CNN model, this GCN model only utilizes two layers of GCN and a fully connected layer.

In the context of predicting dropout students, the problem can be considered imbalanced if many students do not drop out compared to those who do. In such cases, a standard loss function, such as cross-entropy loss, may not be able to effectively distinguish between the positive and negative samples, leading to a bias towards the majority class. Focal loss is a loss function often used in machine learning models to improve the accuracy of predictions for imbalanced datasets.

3.4. Data Collection and Storage

The data used in this study was collected from the University Academic portal of FPT University (FAP). Due to the sensitivity of the data, certain features were not collected or encrypted to protect individuals' privacy. The data collection process was overseen by a supervisor who directly accessed the data and controlled the data collection process to ensure that all ethical guidelines were followed and appropriate permissions were obtained. Further details regarding the Dataset will be presented in the following section.

The collected data was then stored in an MSSQL database for further analysis. The data storage process was designed to ensure the security and confidentiality of the data. Access to the database was strictly controlled and limited to authorized personnel only. Appropriate measures were taken to protect against data loss and ensure data integrity.

The raw dataset provides comprehensive information about academic students, including details of each exam for each subject in each Semester and demographic information such as age and gender. The dataset

is essential for conducting the research presented in this thesis, and its thorough description is necessary to provide context and a complete understanding of the data used. The following sections will discuss the data and its preprocessing and transformation procedures. To better understand the database schema, a diagram is included below.

Dataset Schema

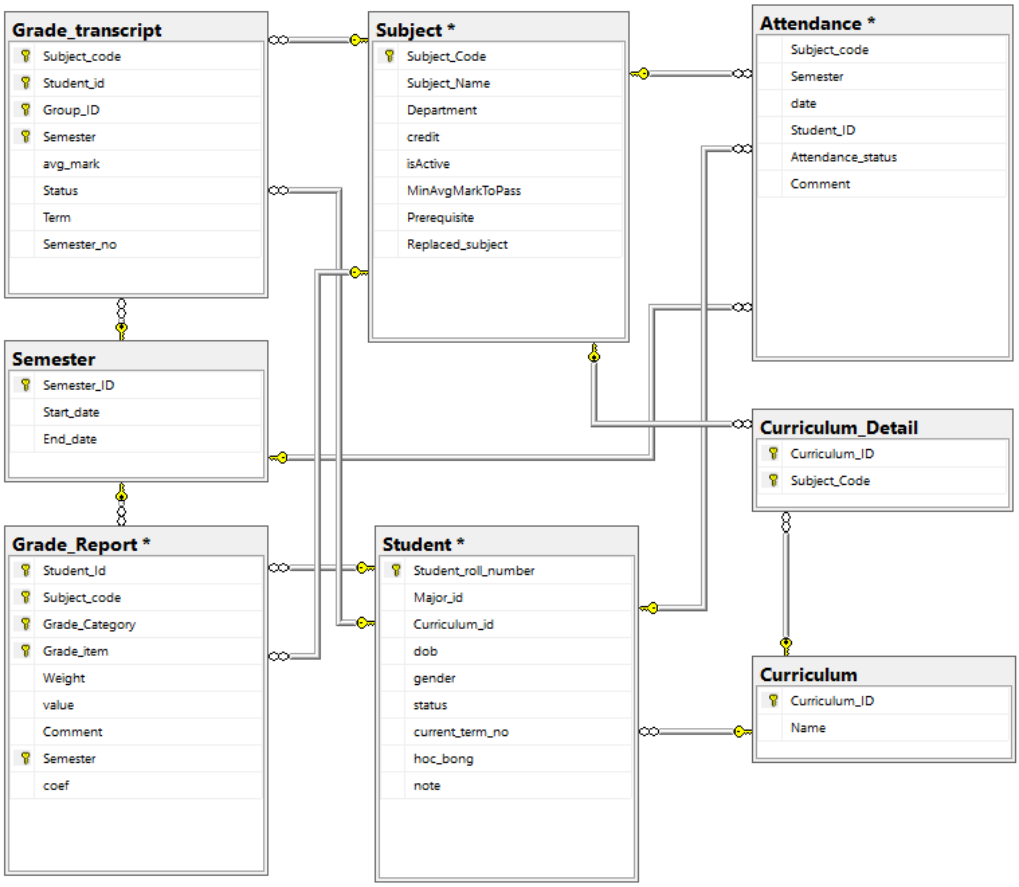

Figure 13: Dataset Schema

Entity 1: Student

The "Student" entity contains information about student ID, demographic data, and status information of all students.

Column Name	Data type	Description	Example
Student_roll_number	String	Unique identifier of student	HE153632
major_id	String	Specific major identifier of each student. Each major contains many different narrow majors.	BIT
curriculum_id	String	Specific identifier curriculum.	BIT_AI_15B_DS
dob	Date	Date of birth	01/01/2001
gender	Int	female is "1," and male is "0."	
status	String	The current status of the student.	TH - TH, Is progress
current_term_no	int	The current term no of student	
hoc_bong	Int or Null	The scholarship amount of a student. Null if the student does not have a scholarship	100

note	String	Specific note of administrator for students based on the corresponding status.	

Table 1: Student Entity

Entity 2: Curriculum

The "Curriculum" entity contains information about the detailly name of the curriculum ID.

Column name	Data type	Description	Example
Curriculum_ID	String	Specific identifier curriculum.	BIT_AI_15B_DS
Name	String	Detailed name of the corresponding major that contains it	Bachelor Program of Information Technology, Artificial Intelligence Major

Table 2: Curriculum Entity

Entity 3: Curriculum_Detail

The "Curriculum_Detail" entity contains all detailed subjects in each curriculum.

Column name	Data type	Description	Example
Curriculum_ID	String	Specific identifier curriculum.	BIT_AI_15B_DS
Subject_Code	String	The Subject code that is in the corresponding curriculum_id	MAI391

Table 3: Curriculum_Detail

Entity 4: Subject

The "Subject" Entity contains information detailly each subject.

Column name	Data type	Description	Example
Subject_Code	String		MAI391
Subject_name	String	The exact name of the related subject.	Mathematics for Machine Learning
Department	String	The department contains the subject.	Mathematics
credit	Int	Number credit of the corresponding subject.	3
isActive	Boolean	"True" if the subject code is active; else, "False"	
MinAvgMarkToPass	Float	Min final average mark corresponding subject_code to pass	5.0
Prerequisite	String	The List contains all Prerequisite Subject_Code of corresponding Subject_Code.	MAE101
Replaced_subject	String	The Subject code that has replaced similar subject codes in older curriculum	

Table 4: Subject Entity

Entity 5: Grade_transcript

The "Grade_transcript" Entity contains the Final grade of each subject in

Column name	Data type	Description	Example
Subject_code	String	Subject code	MAI391
Student_id	String	Unique identifier of student	
Semester	String		Summer2021
avg_mark	Float	Final average mark of corresponding subject code	8.3
status	String		Passed
Term	Int		4
Semester no	Int	The semester count from begin of the academic term	4

Table 5: Grade transcript Entity

Entity 6: Grade_Report

The "Grade_Report" Entity contains detailly grade items for each subject of all students.

Column name	Data type	Description	Example
Student_id	String	Unique identifier of student	
Subject_code	String		MAI391
Grade_catergory	String	The wrapped categories of list grade with the same.	Assignment
Grade_item	String	Detail of each item grade in each grade category.	Assignment1, Assignment 2, Assignment 3
Value	Int	Value of the grade.	8
Comment	String	Specific note	Cải thiện điểm
Semester	String	The Semester contains the corresponding subject code	Summer2021
Coef	float	Weighted percentage of corresponding Grade_item	10

Table 6: Grade report Entity

Entity 7: Attendance

The "Attendance" Entity contains information detail in each slot of each subject of all students.

Column name	Data type	Description	Example
Suject_code	String		MAI391
Semester	String	The Semester contains the corresponding subject code	Summer2021
Date	Date	Date of Slot	Monday 10/05/2021
Student_ID	String	Unique identifier of student	HE153632
Attendance_status	String	Absent, Present or Future	Absent
Comment	String	Specific note by lecture	

Table 7: Attendance Entity

Entity 8: Semester

The "Semester" Entity contains information about the start and finish of all Semesters.

Column name	Data type	Description	Example
Semester	String		Summer2021
Start_date	Date	Start date of Semester	2021-05-10

Table 8: Semester Entity

Important Terminologies and School dropout labeling facility

- **Semester**: At FPT University, a year consists of 3 semesters per year: Spring (from January to April), Summer (from May to August), and Fall (from September to December). Only the semesters with at least one assigned subject are counted when counting a student's semesters. The Semester count reflects the **student's actual study time**.

- **Term**: At FPT University, there are a maximum of 15 terms, where terms from -5 to 0 represent the preparation stages for English levels 1 to 6, and terms 1 to 9 represent nine specialized semesters. The term count of a student will not level up if the student does not meet the conditions to assign any courses in the next Term. The term count reflects the **progress of completing the curriculum.**
    - There are two special values in Term No:
        - 10: indicate for Student has Graduated – "G" or has not Graduated – "KTN."

- -6: refers to students who immediately withdraw their profile after entering the school.

- **Dropout**: Dropout refers to students who have discontinued their studies. The "status" column in the "Student" entity determines a student dropout at FPT University. Suppose the value of the column is "TH" (abbreviation for "Thôi Học," which means "Stop Studying") or "KTN" (abbreviation for "Không tốt nghiệp" which means "Not Graduated"). In that case, the student is considered a dropout. "KTN" is a particular case of "TH " in which the student has gone through all coursework but failed to pass some subjects and did not clear it. It is crucial to remove a student who dropped out in Semester (or Term) k^{th} was not in all previous Semesters (or Terms). Therefore, the label for a dropout student is dynamic and depends on the model's predictions. A dropout student is labeled 1, while others are labeled 0.

4. Experimental and result

4.1 Experimental design

4.1.1. Experimental data

The data used in this study was collected from the University Academic portal of FPT University (FAP). Due to the sensitivity of the data, certain features were not collected or encrypted to protect individuals' privacy. The data collection process was overseen by a supervisor who directly accessed the data and controlled the data collection process to ensure that all ethical guidelines were followed and appropriate permissions were obtained. Further details regarding the Dataset will be presented in the following section.

Based on the FPT university program structure, this thesis divides the problem into the English preparation (EP) phase and the main term phase of Information technology (IT) students. In addition, because there are too many missing values in students before K13, we filter the data for students that enroll from 2017 to 2021. According to Figure 14, we can see that the trends of dropping over the semesters in the English preparation phase and the main phase are similar. Major of the student chose to drop out in the first and second semesters of both phases, and over the stander number of semesters (9 in the main phase and 5 in the preparation phase), the number of dropouts in each semester was steady. Finally, only insignificant students will choose to drop out after spending more than the required number of semesters in the school program. Since the number of dropout students is small, the prediction will take place in the first semester of each phase to minimize the number of students who have already dropouts.

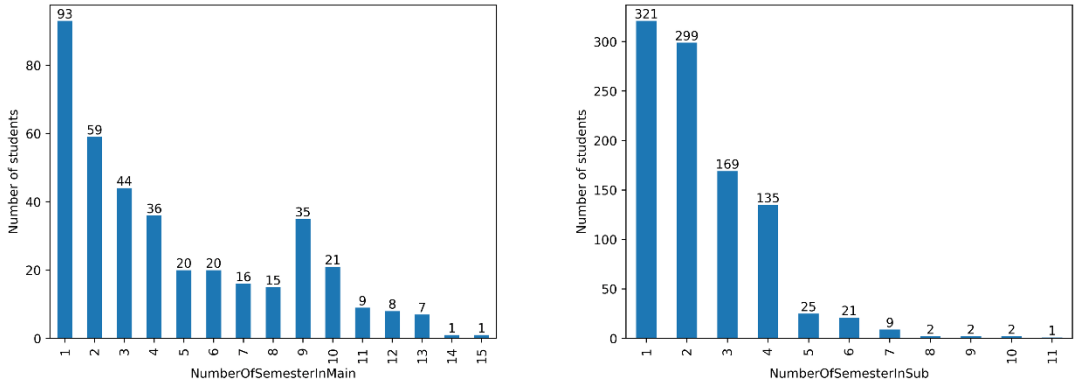

Figure 14 Number of dropout students over semesters. A- in the main term and B- in the preparation term

Data splitting

In this case study, the dataset is split into three parts: a training set, a test set, and a validation set. The training set comprises 80% of the data, while the remaining 20% is equally divided between the test and validation sets.

4.1.1.1. English preparation dataset

This dataset is created using the first-semester information of students in the English preparation phase. In addition, we only label the student dropout in the English preparation phase as a dropout. The EP dataset includes 21429 students, of which 20443 non-dropout students and 986 dropout students. Since the dropout student is 5% compared to non-dropout, the dataset is imbalanced. The dataset consists of 29 features of student performance in English preparation subjects and five demographic features.

For this dataset, we implement LR, SVC, LGBM, CNN model, and TabNet and compare the results. In this dataset, we do not deploy the GCN model because there is only one subject in the first semester, so there is no relationship between courses.

4.1.1.2. Information technology dataset

This dataset is created using IT students' first main semester information and their grade in the English preparation phase. We must drop students who already drop out before the first main semester. The IT dataset includes 7836 students, of which 7458 non-dropout students and 378 dropout students. Since the dropout student is 5% compared to non-dropout, the dataset is imbalanced. The dataset consists of 5 demographic features and 29 performance feature each in departments: English Preparation Course, Traditional Instrument, Computing Fundamental, Soft Skill, Information Technology Specialization, Mathematics, Japanese, Software Engineering, and Physical Training, which make a total of 266 features.

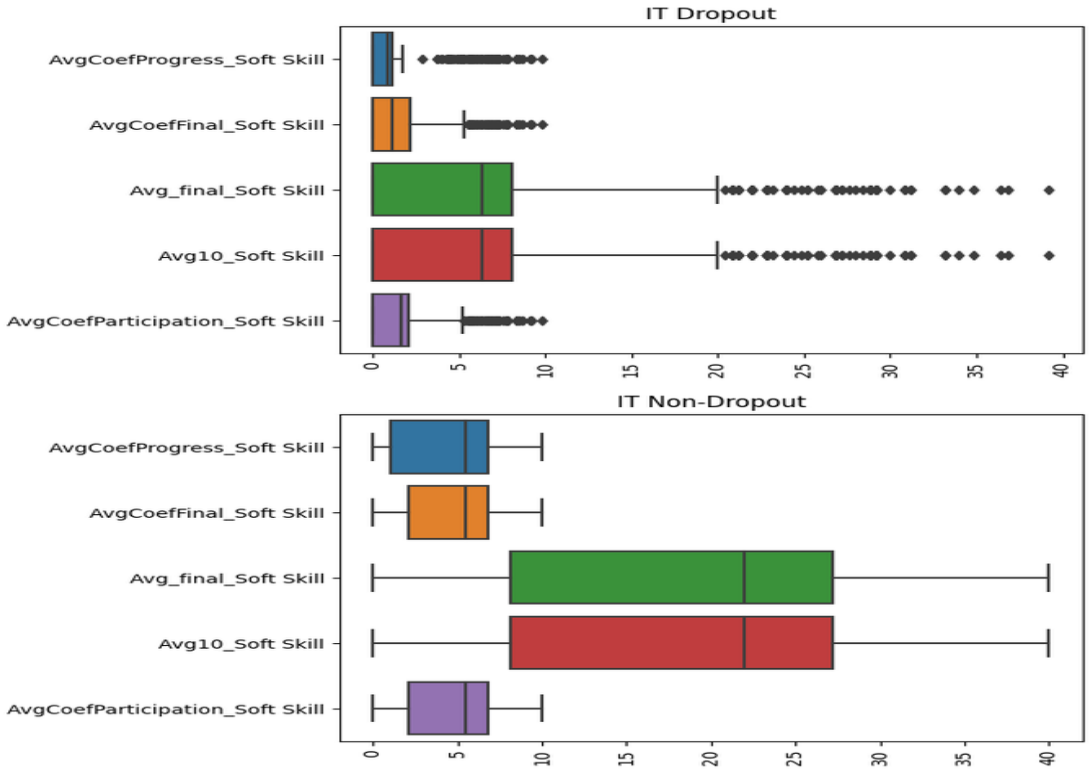

Figure 15 The best five features, based on Pearson correlation, description partitioned by dropout and non-dropout of IT dataset

According to Figure 15, based on the distribution, we can see that the best five features base on Pearson correlation ranking can create a linear bounder between two classes, which may be able to improve the model performance. However, the outliners of dropouts are mixed with the mean of non-dropout students, affecting the precision of the models.

For this dataset, we implement LR, SVC, LGBM, CNN model, GCN model, and TabNet, then compare the results.

4.1.2. Evaluation metric

Because of the dataset imbalance makes those metrics biased toward non-dropout class results. Therefore, we use the macro average for evaluation since the metric is the means of each class evaluation individually. The average macro calculation is as follows:

$$PrecisionMacroAvg = \frac{(Prec_1 + Prec_2 + \cdots + Prec_n)}{n}$$

$$RecallMacroAvg = \frac{(Recall_1 + Recall_2 + \cdots + Recall_n)}{n}$$

4.1.3. Experimental environment

Table 9 summarizes the performance evaluation environment. We trained all models to run on a single GPU RTX 2070 Super with Intel i9-9900KF 5.0GHz, and it was implemented using Python.

Feature	Contents
Exprimental system	Ubuntu 18.04.5 LTS
CPU	Intel i9-9900KF 5.0 GHz
GPU	NVIDIA Gefore RTX 2070 SUPER
Memory	32 GB x 2
Disk	4 TB
Program language	Python

Table 9 Performance evaluation environment

4.2 Result and Discussion

4.2.1 English preparation experience

	Accuracy	Precision-macro	Recall-macro	F1-macro
LR	0.74	0.54	**0.68**	0.52
SVC	0.76	0.54	0.67	0.52
LGBM	0.94	0.68	0.64	**0.66**
LGBM + Pearson	**0.95**	**0.74**	0.60	0.64
LGBM + Chi2	0.90	0.59	0.65	0.61

Table 10: ML performance on EP dataset

Table 10 shows the results of logistic regression (LR), support vector classification (SVC), light gradient boosting machine (LGBM), and light gradient boosting machine with Pearson correlation as feature selection. According to Table 10, it can be seen that while LR and SVC achieve better recall, LGBM-based algorithms have better precision. In addition, the Chi-square feature selection method improves LGBM recall and reduces the precision-macro and f1-macro.

Further insight into the results can be seen in Figure 16. LR and SVC predict more true dropout students than LGBM model. Nevertheless, compared to LR and SVC algorithms, LGBM can propose more precision predictions. This is because the two classes' features are mixed, and no clear linear boundary separates the two classes, which makes the linear-based and kernel-based model inaccuracy.

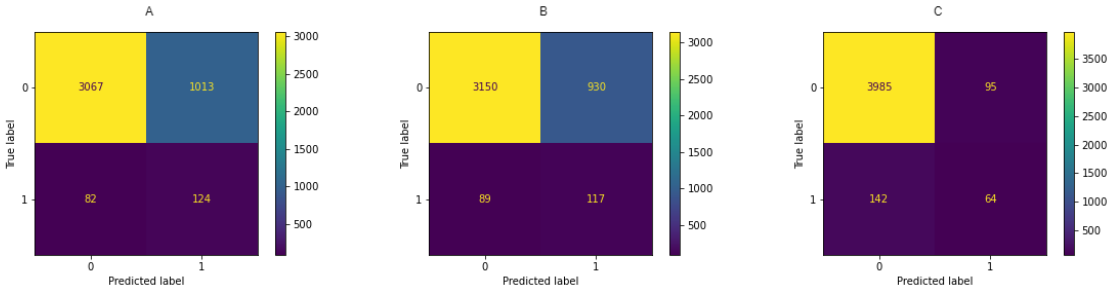

Figure 16 Confusion matrixes of ML algorithms on EP dataset. A-LR, B-SVC, C-LGBM

In this thesis experience, we use Pearson correlation as a statistical measure to examine the relationship between the dropout state (binary value) and other features (continuous variables). According to Figure 18**,** it can be seen that a course failed and pass status greatly influence dropout prediction. In addition, the ratio of F and A grades (lowest and highest rank) also affect the students' motivation. We experience Pearson correlation as feature selection and achieve results with significantly better precision in exchange for recall, as shown in Figure 17. Furthermore, LGBM predicted the highest accuracy meant LGBM model could achieve high accuracy on non-dropout students.

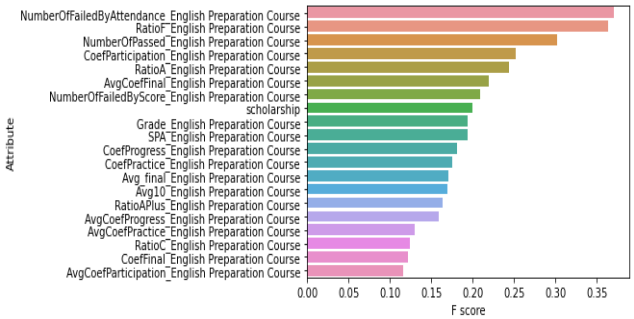

Figure 18 Features ranking based on Pearson correlation measurement

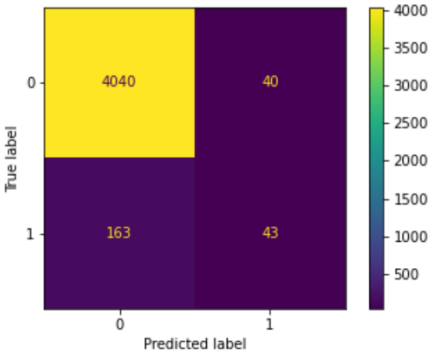

Figure 17 Confusion matrix of LGBM with feature selection on EP dataset

	Mean	Std	Max
True = 0 & pred = 0	6.325354	2.581064	9.2
True = 0 & pred = 1	4.446269	3.262376	8.6
True = 1 & pred = 0	5.551538	2.910479	8.6
True = 1 & pred = 1	3.22500	2.64485	8.3

Table 11: English preparation course GPA statistics in LGBM predict

We propose a more insightful view of LGBM results by showing the statistic of the English preparation course GPA of each case of confusion matrix in
Table 11. It can be seen that those students whom the LGBM model predicts Dropout have lower grades compared to those who do not, which stated that from the GPA of the first English preparation semester, it is hard to propose an accuracy model.

Since the thesis problem is to predict dropout students to support and keep them remained in school, therefore our studies focus more on recall values, which represent the proportion of actual positive cases that are correctly identified by the model, as well as keep the precision as high as possible to avoid biased. Table 12 shows results from deep learning approaches, a two layers CNN model, and TabNet. Those deep learning models can improve the recall and precision-macro values compared to LR and SVC. While those deep learning algorithms trade off precision to recall compared to the LGBM-base algorithm, because of the properties of the problem, the trade-off is acceptable.

Figure 19 illustrates the confusion matrix of CNN and TabNet models. According to the Figure, the deep learning approach can predict more true dropouts than machine learning algorithms and reduce the number of wrong-predicted dropout students.

	Accuracy	Precision-macro	Recall-macro	F1-macro
CNN	0.82	0.56	0.70	0.56
TabNet	0.73	0.56	**0.72**	0.52
CNN + Focal	**0.956**	**0.81**	0.57	0.61
Tabnet + Focal	0.953	0.75	0.60	**0.64**

Table 12: Deep learning models results on EP dataset

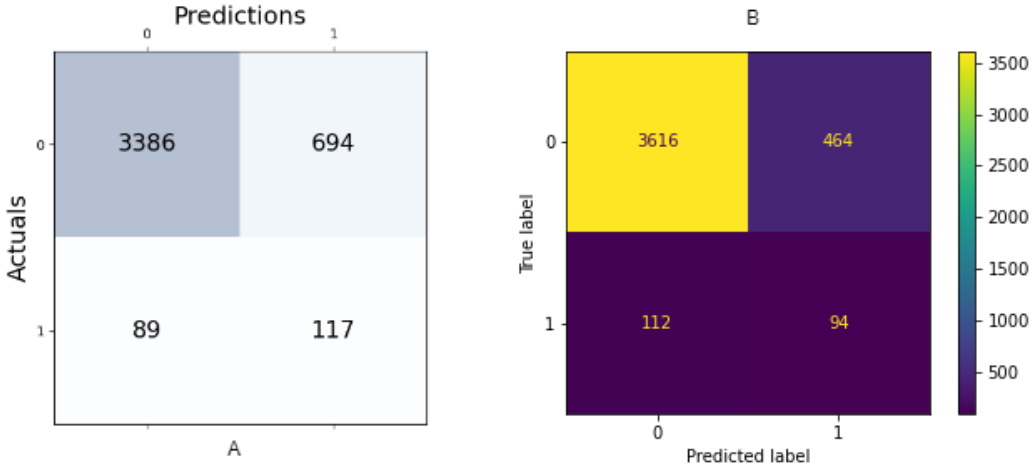

Figure 19:Confusion matrix of CNN and TabNet models with entropy loss. A- CNN models and B-TabNet models

On the other hand, we applied Focal loss to handle the imbalance problem. From the results of Table 12 and confusion matrixes in Figure 20, we can see that Focal loss improves the model precision and overall F1-score in exchange for recall value. However, from what we observe, Focal loss, like LGBM algorithm, is biased to predict non-dropout students, which may lead to the inability to identify dropout students.

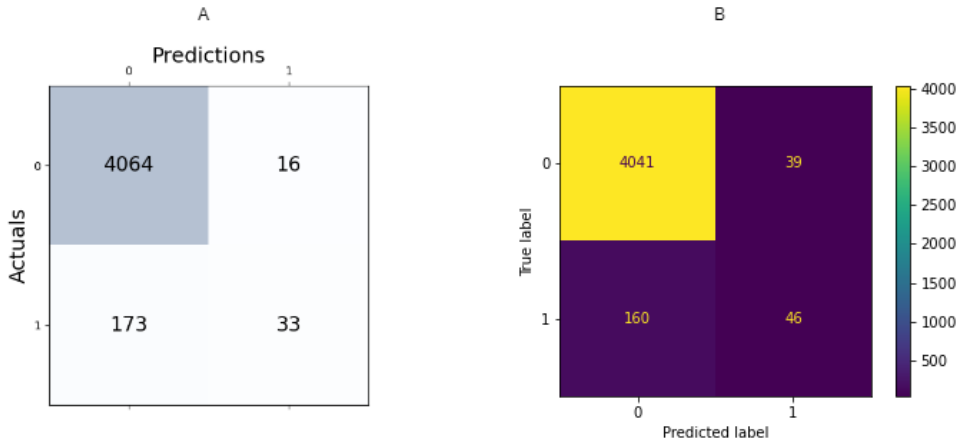

Figure 20:Confusion matrix of CNN and TabNet models with Focal loss. A- CNN models and B-TabNet models

4.2.2 Information technology experience

	Accuracy	Precision-macro	Recall-macro	F1-macro
LR	0.78	0.55	0.69	0.54
SVC	0.81	0.55	0.69	0.56
LGBM	**0.91**	0.62	0.71	0.65
LGBM+Pearson	0.90	**0.62**	**0.73**	**0.65**
LR + Pearson	0.70	0.52	0.74	0.56
LGBM + Chi2	0.90	0.61	0.70	0.63

Table 13 ML performance on IT dataset

Unlike the English preparation dataset, LGBM fusion with Pearson correlation is the best model compared to other ML algorithms like LR and SVC with the IT dataset (Table 13). Since the IT dataset contains the English preparation phase and the program's first Semester, it has more information features with values mixed, making the linear boundary between classes vaguer. In addition, the results also show that the Pearson correlation is a better supporter than the chi-square feature selection in ranking features. This is because chi-square is more suitable for categorical or discrete features. Figure 21 shows the results of four algorithms, and it can be seen that the values that differ between the four algorithms in accurate dropout predictions are not too large. However, the number of wrong label dropouts makes the distinction between models.

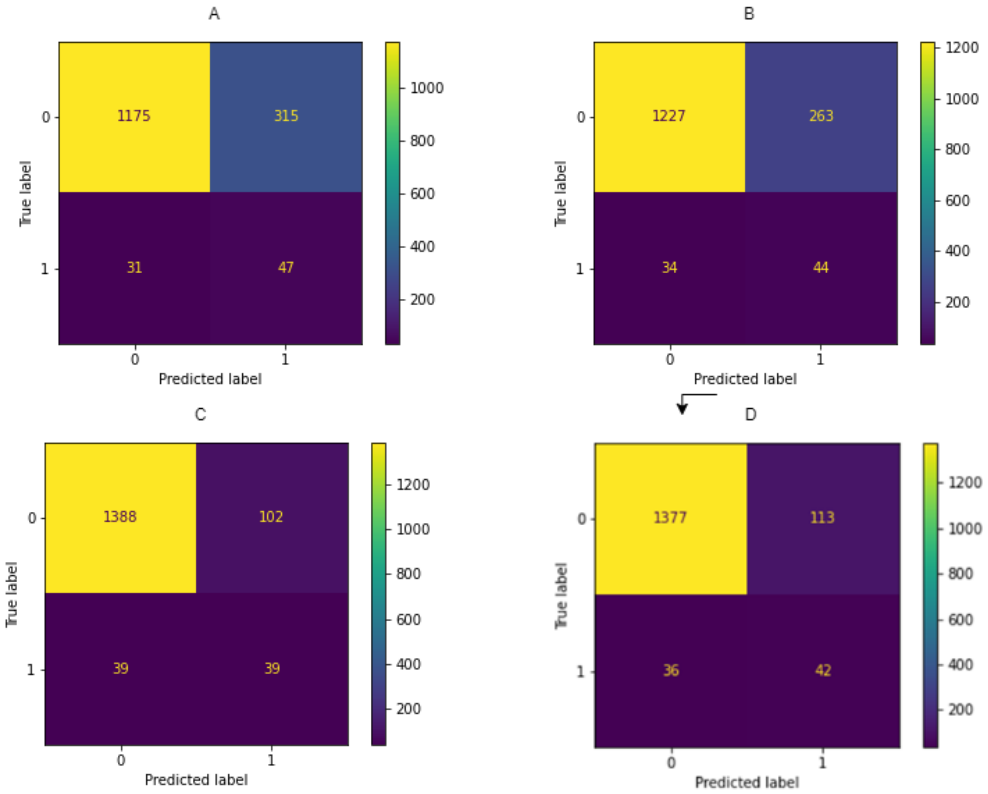

Figure 21:Confusion matrix of ML algorithms. A-LR, B-SVC, C-LGBM, and D-LGBM with feature selection

The features ranking based on the Pearson correlation measure is shown in Figure 22**.** According to Figure 22**,** soft skill subjects significantly influence the dropout state after the first Semester, followed by Computing Fundamental and Traditional Instruments. With the result in Figure 23 and Figure 23, we

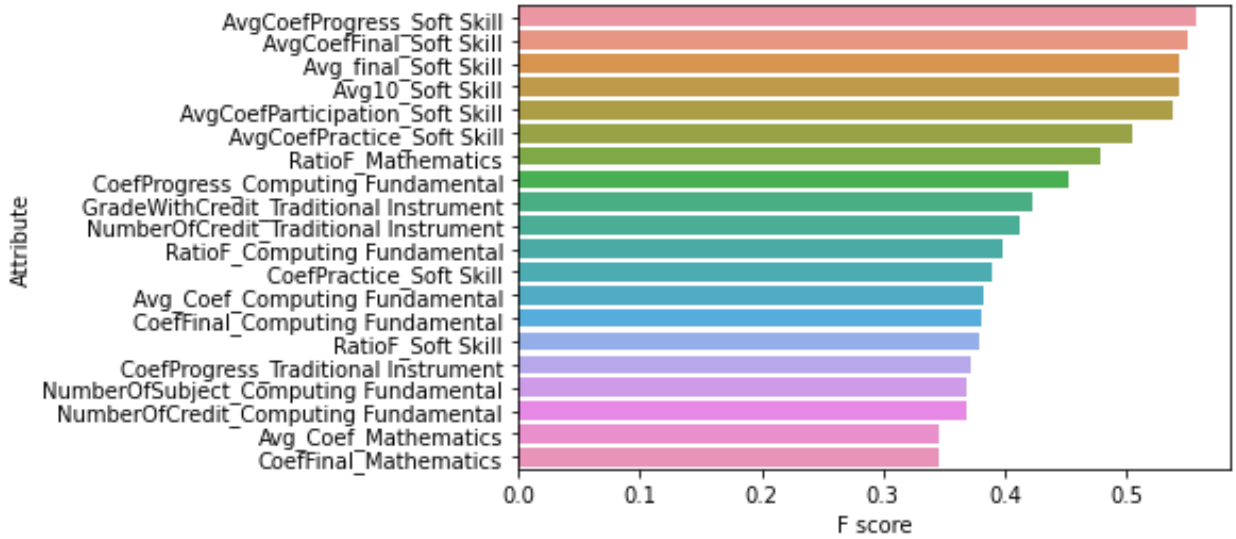

Figure 22 Features ranking based on Pearson correlation measurement

explore the linear boundary between two classes by applying LR with Pearson correlation-based feature selection. Figure 23 shows the effect of LR combined with feature selection. The result shows that it achieves a better balance accuracy score in exchange for precision. In addition, LR with Pearson has a high recall which almost gets all the dropouts but also labels one-third of non-dropout students wrong.

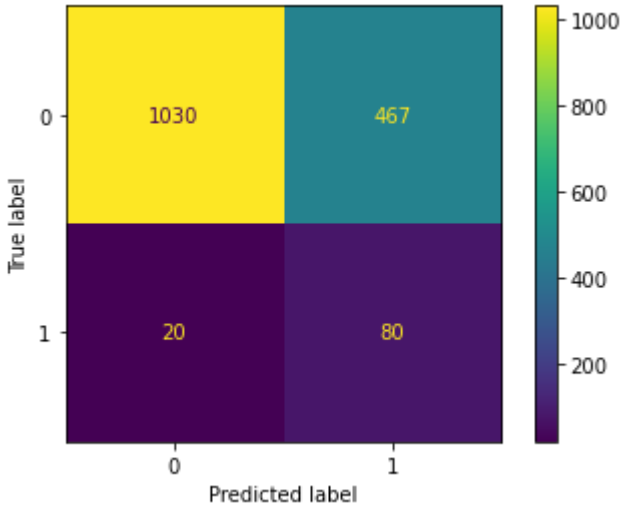

Figure 23 Confusion matrix of LR with feature selection

Like the English preparation phase, deep learning models can perform better than traditional machine learning algorithms, as shown in Table 14. In comparison, TabNet and GCN achieve the recall macro of 75%. However, GCN can preserve precision, which means GCN is better at handling bias than TabNet.

	Accuracy	Precision-macro	Recall-macro	F1-macro
CNN	0.858	0.58	0.71	0.60
TabNet	0.73	0.55	**0.75**	0.53
GCN	0.864	**0.60**	**0.75**	0.62
CNN + Focal	0.95	0.47	0.5	0.48
TabNet + Focal	0.94	0.72	0.58	0.61
GCN + Focal	0.95	0.47	0.5	0.48

Table 14 Deep learning models results on IT dataset.

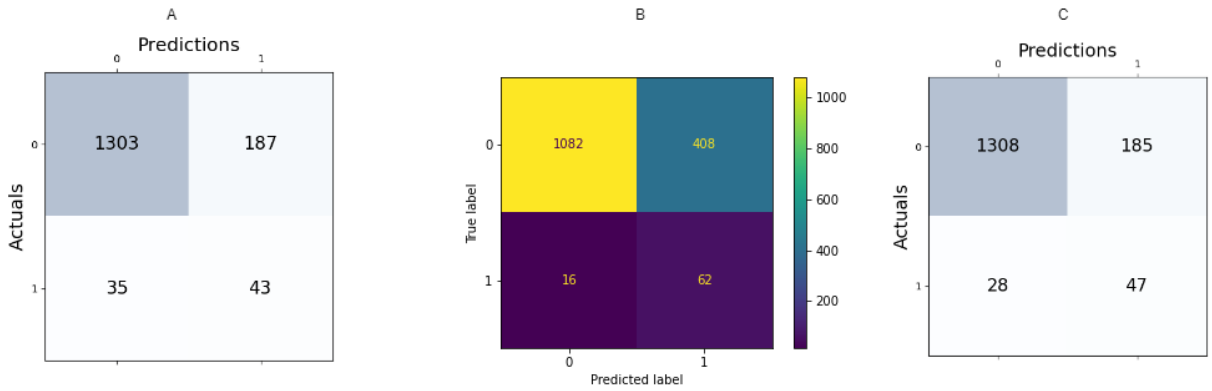

Figure 24 Confusion matrix result of deep learning models. A-CNN model, B-TabNet, and C-GCN model

In Figure 24**,** we can see the confusion matrix of three deep learning models. While GCN and CNN models both achieve similar values, the scale of dropout students in the IT dataset is too small, which makes the evaluation different. TabNet exchanges the result precision with the correct dropout prediction compared to the other two models. However, those deep learning models still lose to LGBM in terms of accuracy, which causes bias.

In conclusion, our experiences show that the tree-based algorithm or LGBM can see outperformance in precision and accuracy, which is biased because of dataset imbalance dataset, compared to other algorithms, including the deep learning model. However, the dropout prediction priority reduces the missing Dropout predict students. Therefore, the deep learning model performs better than LGBM and outperforms other algorithms with significant features and datasets. More specifically, TabNet can indicate the most dropout student in exchange for precision, while CNN and GCN show better balance in precision and recall. In addition, from the result of both datasets, it can be observed that the later the phase to prediction, there is more information, in exchange for the scale of the dataset, since those dropouts can not be used for prediction. This problem also causes the dataset to become more imbalanced.

5. Conclusion

In conclusion, this thesis analyzes and transforms the raw FPT University database into features for the machine learning algorithm. This study also partitioned the dropout problem into two phases: English preparation and main course phases. In addition to traditional machine learning algorithms, we propose deep learning models based on CNN and GCN and implement TabNet, a deep learning model for tabular learning. After conducting experience and comparison between algorithms, our study achieves 72% and 75% balance accuracy in the English preparation and IT first-semester datasets, respectively. On the other hand, LGBM performs with the precision of both datasets. However, since the main priority of the dropout

problem is to detect most dropout students, our proposed model proved to be more efficient than traditional machine learning algorithms in FPT university dropout prediction problems.

On the other hand, our research is obstructed by some limitations. The biggest problem is the dataset imbalance since the dropout students only take up 5% of the total dataset. This issue cause the deployed models to be biased and insufficient data for the deep learning model. Another problem is the missing data because of the change in curriculum and FPT's privacy system. Due to the missing dataset, our research has to generate a dataset for a few missing features, while those with too much data are unusable. In our case is the attendance status, even though attendance significantly influences students' behavior based on our experience with feature ranking. After filtering and cleaning the database based on the prediction phase, our training dataset scale can be too small for deep learning. Especially dropout students' data, because the number of dropout students has to be dropped after each Semester. Therefore, choosing when to suggest the prediction is also a challenging problem for us to solve.

In addition, from our results, it can be seen that students' performance does not significantly influence dropout status since, by visualizing the dataset in t-sne space, it can be seen that features of students' performance in each class are mixed. Furthermore, according to our experience with feature ranking using the Pearson correlation measure, attendance failed status has a significant relationship with dropout status. However, we cannot use attendance-related features for our research due to the lack of a dataset in attendance features. The binary state of dropout status also obstructs our studies because there are too few dropouts, and the probability of those remaining students dropouts in the following semesters. A solution for this problem is constructing a dropout rate representing the student's dropout probability. While dropout prediction problems, sequence, and time-series algorithms like LSTM are prevalent in MOOCs, our studies can not apply those approaches since our dataset unit of time is too large, which may cause data loss when building a time-series dataset. In brief, the thesis contribution is:

- Analysis of FPT University (Hoa Lac campus) students' performance and determine factors influencing Dropout.
- Investigate the influence of grade categorical and subject department in dropout prediction problems.
- Construct FPT university (Hoa Lac campus) students' performance dataset based on subject department and grade detail.
- Analyze the efficiency of machine learning algorithms, convolution, graph neural networks, and tabular learning in academic dropout prediction problems.

Overall, our research proposes a solution for the dropout prediction for FPT University. While our results may not achieve the expected performance, we can analyze the influence of students' academic grades and deep learning models influence in FPT University's dropout problem.

Reference

[1] Simón, E. J. L., & Puerta, J. M. (2022). Prediction of early Dropout in higher education using the SCPQ. *Cogent Psychology*, *9*(1). https://doi.org/10.1080/23311908.2022.2123588

[2] Hegde, V., & Prageeth, P. G. (2018). Higher education student dropout prediction and analysis through educational data mining. *2018 2nd International Conference on Inventive Systems and Control (ICISC)*. https://doi.org/10.1109/icisc.2018.8398887

[3] Vasić, D., Kundid, M., Pinjuh, A., & Šerić, L. (2015). Predicting student's learning outcome from Learning management system logs. *International Conference on Software, Telecommunications and Computer Networks*. https://doi.org/10.1109/softcom.2015.7314114

[4] Haiyang, L., Wang, Z., Benachour, P., & Tubman, P. (2018). A Time Series Classification Method for Behaviour-Based Dropout Prediction. *International Conference on Advanced Learning Technologies*. https://doi.org/10.1109/icalt.2018.00052

[5] Ding, M., Yang, K., Yeung, D., & Pong, T. (2019). Effective Feature Learning with Unsupervised Learning for Improving the Predictive Models in Massive Open Online Courses. *ArXiv (Cornell University)*. https://doi.org/10.1145/3303772.3303795

[6] Dalipi, F., Imran, M., & Kastrati, Z. (2018). MOOC dropout prediction using machine learning techniques: Review and research challenges. *Global Engineering Education Conference*. https://doi.org/10.1109/educon.2018.8363340

[7] Berens, J., Schneider, K., Görtz, S., Oster, S., & Burghoff, J. (2019). Early Detection of Students at Risk - Predicting Student Dropouts Using Administrative Student Data from German Universities and Machine Learning Methods. *Zenodo (CERN European Organization for Nuclear Research)*. https://doi.org/10.5281/zenodo.3594771

[8] Gray, C. C., & Perkins, D. (2019). Utilizing early engagement and machine learning to predict student outcomes. *Computers & Education*, *131*, 22–32. https://doi.org/10.1016/j.compedu.2018.12.006

[9] Pérez, B. T., Castellanos, C., & Correal, D. (2018b). Predicting Student Dropout Rates Using Data Mining Techniques: A Case Study. *Communications in Computer and Information Science*, 111–125. https://doi.org/10.1007/978-3-030-03023-0_10

[10] Sandoval-Palis, I., Naranjo, D., Vidal, J., & Gilar-Corbi, R. (2020). Early Dropout Prediction Model: A Case Study of University Leveling Course Students. *Sustainability*, *12*(22), 9314. https://doi.org/10.3390/su12229314

[11] Chen, J. M., Feng, J., Sun, X., Wu, N., Yang, Z., & Chen, S. (2019). MOOC Dropout Prediction Using a Hybrid Algorithm Based on Decision Tree and Extreme Learning Machine. *Mathematical Problems in Engineering*, *2019*, 1–11. https://doi.org/10.1155/2019/8404653

[12] Qiu, L., Liu, Y., Hu, Q., & Liu, Y. (2019). Student dropout prediction in massive open online courses by convolutional neural networks. *Soft Computing*, *23*(20), 10287–10301. https://doi.org/10.1007/s00500-018-3581-3

[13] Duong, H.TH., Tran, L.TM., To, H.Q. *et al.* Academic performance warning system based on data driven for higher education. *Neural Comput & Applic* **35**, 5819–5837 (2023). https://doi.org/10.1007/s00521-022-07997-6

[14] Paiva, R., Bittencourt, I. I., Lemos, W., Vinicius, A., & Dermeval, D. (2018). Visualizing Learning Analytics and Educational Data Mining Outputs. *Springer EBooks*, 251–256. https://doi.org/10.1007/978-3-319-93846-2_46

[15] Qu, H., & Chen, Q. (2015). Visual Analytics for MOOC Data. *IEEE Computer Graphics and Applications*, *35*(6), 69–75. https://doi.org/10.1109/mcg.2015.137

[16] Curtis, Jonathan, et al. (1983). Dropout Prediction. Austin Independent School District, TX. Office of Research and Evaluation.

[17] Kotsiantis, S., Pierrakeas, C., & Pintelas, P. E. (2003). Preventing Student Dropout in Distance Learning Using Machine Learning Techniques. *Springer eBooks*, 267–274. https://doi.org/10.1007/978-3-540-45226-3_37

[18] Al-Radaideh, Qasem & Al-Shawakfa, Emad & Al-Najjar, Mustafa. (2006). Mining Student Data Using Decision Trees. The International Arab Journal of Information Technology - IAJIT.

[19] Kotsiantis, S., Patriarcheas, K., & Xenos, M. (2010). A combinational incremental ensemble of classifiers as a technique for predicting students' performance in distance education. *Knowledge Based Systems*. https://doi.org/10.1016/j.knosys.2010.03.010

[20] Coleman, Cody & Seaton, Daniel & Chuang, Isaak. (2015). Probabilistic Use Cases: Discovering Behavioral Patterns for Predicting Certification. 141-148. 10.1145/2724660.2724662.

[21] Fei, M., & Yeung, D. (2015). Temporal Models for Predicting Student Dropout in Massive Open Online Courses. *International Conference on Data Mining*. https://doi.org/10.1109/icdmw.2015.174

[22] Wang, W., Yu, H., & Miao, C. (2017). Deep Model for Dropout Prediction in MOOCs. *Proceedings of the 2nd International Conference on Crowd Science and Engineering - ICCSE'17*. https://doi.org/10.1145/3126973.3126990

[23] Kleinbaum, David G., et al. Logistic regression. New York: Springer-Verlag, 2002.

[24] Hearst, Marti A., et al. "Support vector machines." IEEE Intelligent Systems and their applications 13.4 (1998): 18-28.

[25] Ke, Guolin, et al. "Lightgbm: A highly efficient gradient boosting decision tree." Advances in neural information processing systems 30 (2017).

[26] Plackett, Robin L. "Karl Pearson and the chi-squared test." International statistical review/revue internationale de statistique (1983): 59-72.

[27] Mukherjee, Sabyasachi, et al. "Ensemble Method of Feature Selection Using Filter and Wrapper Techniques with Evolutionary Learning." Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2022, Volume 2. Singapore: Springer Nature Singapore, 2022. 745-755.

[28] Nuanmeesri, S., Poomhiran, L., Chopvitayakun, S., & Kadmateekarun, P. (2022). Improving Dropout Forecasting during the COVID-19 Pandemic through Feature Selection and Multilayer Perceptron Neural Network. *International Journal of Information and Education Technology*, *12*(9), 851–857. https://doi.org/10.18178/ijiet.2022.12.9.1693

[29] Panagiotakopoulos, T., Kotsiantis, S., Kostopoulos, G., Iatrellis, O., & Kameas, A. (2021). Early Dropout Prediction in MOOCs through Supervised Learning and Hyperparameter Optimization. *Electronics*, *10*(14), 1701. https://doi.org/10.3390/electronics10141701

[30] Jin, C. (2021). Dropout prediction model in MOOC based on clickstream data and student sample weight. *Soft Computing*, *25*(14), 8971–8988. https://doi.org/10.1007/s00500-021-05795-1

[31] Baranyi, M., Nagy, M., & Molontay, R. (2020). Interpretable Deep Learning for University Dropout Prediction. *Conference on Information Technology Education*. https://doi.org/10.1145/3368308.3415382

[32] Segura, M., Mello, J., & Hernández, A. (2022). Machine Learning Prediction of University Student Dropout: Does Preference Play a Key Role? *Mathematics*, *10*(18), 3359. https://doi.org/10.3390/math10183359

[33] Yaacob, W. F. W., Sobri, N. M., Nasir, S. a. M., Norshahidi, N. D., & Husin, W. Z. W. (2020). Predicting Student Dropout in Higher Institution Using Data Mining Techniques. *Journal of Physics: Conference Series*, *1496*, 012005. https://doi.org/10.1088/1742-6596/1496/1/012005

[34] Tenpipat, W., & Akkarajitsakul, K. (2020). Student Dropout Prediction: A KMUTT Case Study. *International Conference on Big Data*. https://doi.org/10.1109/ibdap50342.2020.9245457

[35] Lottering, R., Hans, R. T., & Lall, M. (2020). A model for the identification of students at risk of Dropout at a university of technology. *International Conference on Artificial Intelligence*. https://doi.org/10.1109/icabcd49160.2020.9183874

[36] Da Silva, D. P. C., Pires, E. J. S., Reis, A., Oliveira, P. M., & Barroso, J. (2022). Forecasting Students Dropout: A UTAD University Study. *Future Internet*, *14*(3), 76. https://doi.org/10.3390/fi14030076

[37] Yukselturk, E., Özekes, S., & Türel, Y. K. (2014). Predicting Dropout Student: An Application of Data Mining Methods in an Online Education Program. *The European Journal of Open, Distance and E-Learning*, *17*(1), 118–133. https://doi.org/10.2478/eurodl-2014-0008

[38] Mujica, A. D., Villalobos, M. P. S., Gutiérrez, A., Fernández-Castañón, A. C., & García, J. a. S. (2019). Affective and cognitive variables involved in structural prediction of university dropout. *Psicothema*, *31*(4), 429–436. https://doi.org/10.7334/psicothema2019.124

[39] Korenkova, M. M., Shadrina, E., & Oshmarina, O. E. (2020). Educational Data Mining for Prediction of Academically Risky Students Depending on Their Temperament. *Communications in Computer and Information Science*, 277–290. https://doi.org/10.1007/978-3-030-71214-3_23

[40] Amornsinlaphachai, P. (2016). Efficiency of data mining models to predict academic performance and a cooperative learning model. *International Conference on Knowledge and Smart Technology*. https://doi.org/10.1109/kst.2016.7440483

[41] Pallathadka, H., Wenda, A., Ramirez-Asís, E., Asís-López, M., Flores-Albornoz, J., & Phasinam, K. (2021). Classification and prediction of student performance data using various machine learning algorithms. *Materials Today: Proceedings*. https://doi.org/10.1016/j.matpr.2021.07.382

[42] Yağcı, M. (2022). Educational data mining: prediction of students' academic performance using machine learning algorithms. *Smart Learning Environments*, *9*(1). https://doi.org/10.1186/s40561-022-00192-z

[43] Siddique, A., Jan, A., Majeed, F., Qahmash, A., Quadri, N. N., & Wahab, M. A. (2021). Predicting Academic Performance Using an Efficient Model Based on Fusion of Classifiers. *Applied Sciences*, *11*(24), 11845. https://doi.org/10.3390/app112411845

[44] Roslan, Haziq & Chen, Chwen Jen. (2022). Predicting students' performance in English and Mathematics using data mining techniques. Education and Information Technologies. 28. 10.1007/s10639-022-11259-2.

[45] Rachburee, N., & Punlumjeak, W. (2015). A comparison of feature selection approach between greedy, IG-ratio, Chi-square, and mRMR in educational mining. *International Conference on Information Technology and Electrical Engineering*. https://doi.org/10.1109/iciteed.2015.7408983

[46] Farissi, A., Dahlan, H. M., & Samsuryadi. (2020). Genetic Algorithm Based Feature Selection With Ensemble Methods For Student Academic Performance Prediction. *Journal of Physics: Conference Series*, *1500*(1), 012110. https://doi.org/10.1088/1742-6596/1500/1/012110

[47] Alraddadi, S. A., Alseady, S., & Almotiri, S. (2021). Prediction of Students Academic Performance Utilizing Hybrid Teaching-Learning based Feature Selection and Machine Learning Models. *2021 International Conference of Women in Data Science at Taif University (WiDSTaif)*. https://doi.org/10.1109/widstaif52235.2021.9430248

[48] Dalipi, F., Imran, M., & Kastrati, Z. (2018). MOOC dropout prediction using machine learning techniques: Review and research challenges. *Global Engineering Education Conference*. https://doi.org/10.1109/educon.2018.8363340

[49] Niyogisubizo, J., Liao, L., Nziyumva, E., Murwanashyaka, E., & Nshimyumukiza, P. C. (2022). Predicting student's Dropout in university classes using two-layer ensemble machine learning approach: A novel stacked generalization. Computers & Education: Artificial Intelligence, 3, 100066. https://doi.org/10.1016/j.caeai.2022.100066

[50] Hassan, Y., Elkorany, A. S., & Wassif, K. T. (2022). Utilizing Social Clustering-Based Regression Model for Predicting Student's GPA. *IEEE Access*, *10*, 48948–48963. https://doi.org/10.1109/access.2022.3172438

[51] Almasri, A., Alkhawaldeh, R. S., & Celebi, E. (2020). Clustering-Based EMT Model for Predicting Student Performance. *Arabian Journal for Science and Engineering*, *45*(12), 10067–10078. https://doi.org/10.1007/s13369-020-04578-4

[52] Albawi, Saad, Tareq Abed Mohammed, and Saad Al-Zawi. "Understanding of a convolutional neural network." 2017 international conference on engineering and technology (ICET). Ieee, 2017.

[53] Giannakas, F., Troussas, C., Voyiatzis, I., & Sgouropoulou, C. (2021). A deep learning classification framework for early prediction of team-based academic performance. *Applied Soft Computing*, *106*, 107355. https://doi.org/10.1016/j.asoc.2021.107355

[54] Salam, A., Zeniarja, J., & Anthareza, D. M. (2022). Student Graduation Prediction Model using Deep Learning Convolutional Neural Network (CNN). *2022 International Seminar on Application for Technology of Information and Communication (iSemantic)*. https://doi.org/10.1109/isemantic55962.2022.9920449

[55] Thaher, T., & Jayousi, R. (2020). Prediction of Student's Academic Performance using Feedforward Neural Network Augmented with Stochastic Trainers. *Advanced Industrial Conference on Telecommunications*. https://doi.org/10.1109/aict50176.2020.9368820

[56] Poudyal, S., Mohammadi-Aragh, M. J., & Ball, J. E. (2022). Prediction of Student Academic Performance Using a Hybrid 2D CNN Model. *Electronics*, *11*(7), 1005. https://doi.org/10.3390/electronics11071005

[57] Liu, C., Wang, G., Du, Y., & Yuan, Z. (2022). A Predictive Model for Student Achievement Using Spiking Neural Networks Based on Educational Data. *Applied Sciences*, *12*(8), 3841. https://doi.org/10.3390/app12083841

[58] Xu, B., Yan, S., Li, S., & Du, Y. (2022). A Federated Transfer Learning Framework Based on Heterogeneous Domain Adaptation for Students' Grades Classification. *Applied Sciences*, *12*(21), 10711. https://doi.org/10.3390/app122110711

[59] Hussain, K., Talpur, N., Qin, Z., & Zakria, Z. M. (2020). A Novel Metaheuristic Approach to Optimization of Neuro-Fuzzy System for Students' Performance Prediction. *JOURNAL OF SOFT COMPUTING AND DATA MINING*, *01*(01). https://doi.org/10.30880/jscdm.2020.01.01.001

[60] Gao, L., Zhao, Z., Li, C., Zhao, J., & Zeng, Q. (2022). Deep cognitive diagnosis model for predicting students' performance. *Future Generation Computer Systems*, *126*, 252–262. https://doi.org/10.1016/j.future.2021.08.019

[61] Fei, M., & Yeung, D. (2015b). Temporal Models for Predicting Student Dropout in Massive Open Online Courses. *International Conference on Data Mining*. https://doi.org/10.1109/icdmw.2015.174

[62] Tang, C., Ouyang, Y., Rong, W., Zhang, J., & Xiong, Z. (2018). Time Series Model for Predicting Dropout in Massive Open Online Courses. *Lecture Notes in Computer Science*, 353–357. https://doi.org/10.1007/978-3-319-93846-2_66

[63] Xiong, F., Zou, K., Liu, Z., & Wang, H. (2019). Predicting learning status in MOOCs using LSTM. *Proceedings of the ACM Turing Celebration Conference - China*. https://doi.org/10.1145/3321408.3322855

[64] Jin, C. (2020). MOOC student dropout prediction model based on learning behavior features and parameter optimization. *Interactive Learning Environments*, 1–19. https://doi.org/10.1080/10494820.2020.1802300

[65] Feng, W., Tang, J., & Liu, T. X. (2019). Understanding Dropouts in MOOCs. *Proceedings of the . . . AAAI Conference on Artificial Intelligence*, *33*(01), 517–524. https://doi.org/10.1609/aaai.v33i01.3301517

[66] Wu, N., Zhang, L., Gao, Y., Zhang, M., Sun, X., & Feng, J. (2019). CLMS-Net. *Proceedings of the ACM Turing Celebration Conference - China*. https://doi.org/10.1145/3321408.3322848

[67] Yin, S., Lei, L., Wang, H., & Chen, W. (2020). Power of Attention in MOOC Dropout Prediction. *IEEE Access*, *8*, 202993–203002. https://doi.org/10.1109/access.2020.3035687

[68] Cai, L., & Zhang, G. (2021). Prediction of MOOCs Dropout based on WCLSRT Model. *IEEE Advanced Information Technology, Electronic and Automation Control Conference*. https://doi.org/10.1109/iaeac50856.2021.9390886

[69] Li, X., Zhang, Y., Cheng, H., Li, M., & Yin, B. (2022). Student achievement prediction using deep neural network from multi-source campus data. *Complex & Intelligent Systems*, *8*(6), 5143–5156. https://doi.org/10.1007/s40747-022-00731-8

[70] Uliyan, D. M., Aljaloud, A., Alkhalil, A., Amer, H. S. A., Mohamed, M. I., & Alogali, A. (2021). Deep Learning Model to Predict Students Retention Using BLSTM and CRF. *IEEE Access*, *9*, 135550–135558. https://doi.org/10.1109/access.2021.3117117

[71] Li, M., Wang, X., Wang, Y., Chen, Y., & Chen, Y. (2022). Study-GNN: A Novel Pipeline for Student Performance Prediction Based on Multi-Topology Graph Neural Networks. *Sustainability*, *14*(13), 7965. https://doi.org/10.3390/su14137965

[72] Hu, Q., & Rangwala, H. (2019). Academic Performance Estimation with Attention-based Graph Convolutional Networks. *arXiv (Cornell University)*. https://www.arxiv.org/pdf/2001.00632

[73] Nakagawa, H., Iwasawa, Y., & Matsuo, Y. (2019). Graph-based Knowledge Tracing: Modeling Student Proficiency Using Graph Neural Network. *Web Intelligence*. https://doi.org/10.1145/3350546.3352513

[74] Arik, S. O., & Pfister, T. (2019). TabNet: Attentive Interpretable Tabular Learning. *Proceedings of the . . . AAAI Conference on Artificial Intelligence*, *35*(8), 6679–6687. https://doi.org/10.1609/aaai.v35i8.16826