ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol. 14 (2024) No. 6, pp.93-112
DOI:10.33168/JSMS.2024.0607

Predicting Academic Success and Identifying At-Risk Students using Ensemble and Deep Learning Models

Rithesh Kannan[1], Chong How Wen [1], Yoong Kee Ng[1], Hu Ng [1, *], Timothy Tzen Vun Yap [2], Lai Kuan Wong [1], Fang Fang Chua [1], Vik Tor Goh [3], Yee Lien Lee [3], Hwee Ling Wong [3]

[1] Faculty of Computing and Informatics, Multimedia University, 63100 Cyberjaya, Malaysia

[2] School of Mathematical & Computer Sciences, Heriot-Watt University, Putrajaya, Malaysia

[3] Faculty of Engineering (FOE), Multimedia University

Abstract. Education has become an essential part of human lives. It can bring benefits to many aspects, including the development of personal, society and country. Success or failure in academics will bring a lot of impact to a person, positive or negative. Therefore, it is vital to have a predictive model to track students' performance and to help those weak students at an early stage. This paper aims to forecast the likelihood of a student's success in a course and identify those that are at-risk. The dataset used in this paper is obtained directly from a higher education institute and contains approximately 5488 student records and 94 features. The data cleaning process has been applied due to a lot of missing data. Several plots are used in this paper to explore and visualize the data. Class imbalance was handled using several resampling techniques including Random Oversampling (ROS), Synthetic Minority Oversampling Technique (SMOTE), etc. Feature selection was done using the Boruta algorithm. Several machine learning algorithms from ensemble learning (EL) and deep learning (DL) are applied with different test sizes and compared together in this paper. These include Random Forest (RF) and Logistic Regression (LR) models from EL, and Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models from DL. Accuracy, precision, recall, f1 and the receiver operating characteristic area under the curve (ROC AUC) score are the metrics used to evaluate the model performance. Result shows that the RF was the best EL model with a F1 score of 99.4% for binary classification, and 99.8% for multiclass classification. Meanwhile for DL, the LSTM model performed the best in binary classification, showing a F1 score of 69.8%. However, the results for multiclass classification were inconclusive due to underfitting issues.

Keywords: Academic At-Risk, Ensemble Learning, Classification, Deep Learning, LSTM, GRU, Logistic Regression, Random Forest.

1. Introduction

Nowadays, education has become an essential part of human lives. Accepting an education allows a person to gain knowledge and practical skills, and those knowledge and skills will be helpful throughout life. It helps to bring benefits to every aspect of a person and allows improvements for personal, social, and national development.

Furthermore, according to researchers, human needs are categorized into five main categories: physiological, safety, love and belonging, esteem, and self-actualization (Maslow, 1943; Maslow, Frager, Fadiman, McReynolds, & Cox, 1987). For example, safety needs refer to getting a job, maintaining a healthy body, avoiding injury and accidents, moving to a safer neighbourhood, etc. Education can fulfil most of the pyramid of needs. During school or university life, a person can meet a lot of new people that share similar interests and identity with them, and they can gain friendship through them (love and belonging). Educated people are also qualified for jobs (safety) and can achieve personal goals for themselves (self-actualization). When one becomes successful, he/she also can build confidence from it (esteem). But on the other hand, when one is faced with failure and needs to drop out, he/she will feel inferior. A good academic result will bring a lot of career choices and job security (Ong, Ting, Goh, Quek, & Cham, 2023).

Thus, it is important to identify students at-risk and predict student success at an early stage to prevent students from dropping out. In educational institutions, students' success is measured by their academic performance, for example examination results, coursework, co-curriculum achievement, graduating on time, final program status, and so on. Other researchers have identified students at-risk by predicting whether the student will drop out or not (Niyogisubizo, Liao, Nziyumva, Murwanashyaka, & Nshimyumukiza, 2022). There is not much research on predicting whether a student has graduated on time or not (GOT) and what will be the final program status of the student. Therefore, in this paper, student success and the students at-risk are identified by predicting two variables, GOT, and Program Status.

Moreover, academic datasets generally face two common issues, which are feature selection and class imbalance. Feature selection is an important issue in this field as determining which features contribute the most to the targeted output can greatly impact the performance of the model. This is because the model will be trained on the most important features rather than the entire set of features, which will improve model efficiency and reduce the time taken to build the model. At the same time most academic datasets are imbalanced, that is, the number of at-risk students and students not at risk are not the same. Thus, attempting to predict without handling this issue will lead to false assumptions on the results. There are many feature selection methods including filter-based methods like Info Gain, wrapper-based methods like forward selection and embedded methods like the Boruta algorithm (Alija, Beqiri, Gaafar, & Hamoud, 2023). Similarly, there are multiple class balancing methods, but researchers typically prefer oversampling techniques like Random Oversampling (ROS), Synthetic Oversampling Minority Technique (SMOTE), Adaptive Synthetic Sampling (ADASYN), Support Vector Machine SMOTE (SVM-SMOTE) as they generally perform better than other methods (Alija et al., 2023). Not many studies have included both feature selection and class balancing techniques together. Thus, this study will be using the Boruta algorithm to handle the feature selection issue and ROS, SMOTE, ADASYN and a hybrid resampling technique called SMOTE-Tomek Link to handle the class balancing issue.

In addition, it is possible to forecast students' success in a course and identify those that are at-risk using predictive modelling techniques. A predictive model can be used as an early warning system to identify at-risk students in a course and inform both the instructor and the students, to help the students to obtain a better result at the early stage. Generally, researchers have either used models from ensemble learning (EL) or deep learning (DL) to predict the students at risk. Ensemble learning refers to when two or more models are trained on the same data and their predictions are combined. Deep learning refers to enormous, interconnected models called neural networks that imitate the way the human brain works. Few researchers have utilized models from both ensemble learning and deep learning. Therefore, for this study, models from ensemble learning and deep learning are implemented and compared to find the optimal model to predict the GOT and the Program Status.

This research has contributed to the field in different ways. First, a real-world dataset from a Malaysian university was analysed. Second, the two issues in the dataset were handled appropriately. The first issue was class imbalance which was handled by implementing several different resampling techniques and choosing the best one. Next was the feature selection issue, whereby the features from the dataset were analysed and the best features that contribute to the goal the most were selected using the Boruta algorithm. Third, several predictive models from ensemble learning and deep learning fields, including RF, LR, LSTM, GRU, in addition to several preprocessing techniques were implemented to predict whether the students will graduate on time and their program status. The models with the best predictive capabilities from ensemble learning and deep learning were identified.

The objectives of this research are: (i) to acquire and prepare datasets of at-risk students, (ii) to identify prominent features for prediction of at-risk students, and (iii) to design the best prediction model of at-risk students by comparing algorithms from ensemble learning and deep learning.

2. Literature Review

In section 2, the previous work on obtaining relevant features and handling imbalanced class samples from academic at-risk datasets are discussed. The use of different types of machine learning techniques such as ensemble learning and deep learning to perform academic at-risk modelling by other researchers are reviewed, and the result of their work are discussed.

2.1. Feature Selection and Class Balancing

As has been stated, many studies that have applied machine learning techniques with the overarching goal of combating student dropout face the issues of feature selection and class imbalance. Based on these studies by other researchers, the commonly used features selection and class balancing methods are stated below.

Researchers have experimented with using information gain measure to rank the features extracted from a Virtual Learning Environment (VLE) (Pongpaichet, Jankapor, Janchai, & Tongsanit, 2020). The most relevant features they found was the total number of submissions of quizzes and assignments from students. The researchers also found the best method to handle their imbalanced dataset was oversampling the minority class.

Revathy, Kamalakkannan, and Kavitha (2022) proposed combining a feature selection method called Principal Component Analysis (PCA) with SMOTE to create an algorithm called PCA-SMOTE that would help accurately identify the issues that cause students to drop out.

The length of course can also be one of the important parameters in academic at-risk modelling. The difficulty of studying might also depend on the year of the study in the program (Adnan et al., 2021). There are more parameters like financial data, health issues, lecturer's opinion, student behaviour, drop-out time, distance of education, high school performance and parents' information. All these parameters can have great importance in predicting academic at-risk.

The Boruta algorithm has been studied and seems to perform very well compared to other methods (Naseem, Chaudhary, Sharma, & Lal, 2019). Another feature selection method called Gain ratio has been applied on one of the previous studies as well with promising results (Ahmed & Khan, 2019). Several algorithms are also seen performing well using other feature selection techniques including forward selection, backward elimination, and evolutionary method (Nagy & Molontay, 2018). Lastly there is a features selection algorithm called 'cfsSubsetEval' that performed well (Sahlaoui, Alaoui, Nayyar, Agoujil, & Jaber, 2021).

2.2. Ensemble Learning Techniques

Based on the papers studied, previous researchers have utilized various ensemble learning models to do different prediction on academic at-risk modelling. Researchers have done early detection of at-risk students using Naïve Bayes, Decision Tree, Random Forest, Logistic Regression, Stochastic Gradient Descent and Support Vector Machine (Pongpaichet et al., 2020).

Multiple researchers have also focused on the prediction of dropout students from the education university using SMOTE and ensemble learning (Mulyani, Hidayah, & Fauziati, 2019; Revathy et al., 2022). The machine learning techniques they used includes Random Forest, Logistic Regression and K-Nearest Neighbour.

Soobramoney and Singh (2019) have identified students at-risk with an ensemble of machine learning algorithms including Naïve Bayes, Decision Tree, Random Forest, Logistic Regression, Support Vector Machine, Adaboost, Artificial Neural Network and K-Star.

Adnan et al., (2021) have done a study using ensemble learning to predict at-risk students at different percentages of course length for early intervention using machine learning models. This prediction is done by using an ensemble learning model called Extra tree classifier. The Extra Tree Classifier included Random Forest, Stochastic Gradient Descent, Support Vector Machine, Adaptive Boosting (Adaboost) and K-Nearest Neighbour models.

2.3. Deep Learning Techniques

Dropouts have a negative impact on many different levels. Students' financial, emotional, and personal growth are negatively impacted, as are other elements in their environment, such as society as a whole (Maldonado, Miranda, Olaya, Vásquez, & Verbeke, 2021). Due to this reason, Maldonado et al. (2021) proposed a predictive model to predict the student dropout.

In order to forecast five crucial students' dropout risks—dropout in future, dropout in the upcoming semester, dropout type, dropout duration, and dropout reason—Ul Alam (2022) developed a Multi-modal Neural Fusion network for Multi-Task Cascade Learning (MSNF-MTCL).

At the Budapest University of Technology and Economics (BME), Baranyi, Nagy, and Molontay (2020) proposed using gradient boosted trees and deep neural networks to predict students' final grades.

By utilizing data from a VLE, Waheed et al. (2020) carried out a study using deep artificial neural networks to predict at-risk students and provide strategies for early intervention in such circumstances.

Sun et al. (2019) conducted a study to predict the dropout rate in massive open online courses (MOOCs). Recurrent neural networks (RNN) are used as the foundation of a dropout rate prediction model, and an Uniform Resource Layer (URL) embedding layer is suggested as a solution.

Using the information supplied at enrolment (secondary school performance, personal details), Nagy and Molontay (2018) carried out research to identify students at-risk and forecast students' dropout rate of university programs. Different input settings have been used to train a wide variety of classifiers, including Deep Learning, Linear Models, k-NN, Naive Bayes, and Decision Tree-based algorithms.

Tsai, Chen, Shiao, Ciou, and Wu (2020) used a statistical learning method and a machine learning method based on deep neural networks to forecast Taiwan students' probability of dropping out. According to the findings, in the case of prioritizing the high sensitivity in predicting dropouts, student academic performance, student loan applications, the frequency of absences from class, and the number of alerted subjects successfully predicted whether students would drop out of university with an accuracy rate of 68% when the statistical learning method was used, and 77% for the deep learning method.

There are some predictions that are done using Rule Learners, Neural Network, Recurrent Neural Network and Linear Regression (Agnihotri & Ott, 2014; Hegde & Prageeth, 2018; Nagy & Molontay, 2018; Omar Alkhamisi & Mehmood, 2020; Sahlaoui et al., 2021).

3. Methodology

This section will describe the strategy used to get the result and will also describe the steps taken. Each process will be explained in different sections which include data acquisition, exploratory data analysis, data pre-processing, feature selection, class balancing, model construction and model evaluation.

3.1. Data Acquisition

In this paper, the data has been provided by a Malaysian university directly. As it is confidential data, the details of the dataset cannot be provided directly, instead it is aggregated and anonymized. More details can be seen in the next subsection (Figures 1 to 4). Some of the attributes of the dataset are also shown in Tables 1a and 1b. The dataset consists of 5488 undergraduate student records and 94 features. All students belong to the same program. The target classes are GOT2 and PROG_STATUS.

Table 1a. Dataset description

Name	Description
GOT2	Graduated on time or not
NewID	Unique ID for students
ACAD_CAREER	Type of degree the student was taking
PROG_STATUS	Short form of the program status of student
PROG_ACTION	Long description of the program status of student
ADMIT_TERM	The term when the student was admitted
BEGIN_DT	Beginning date of student
END_DT	Ending date of student
EXP_GRAD_TERM	Expected term the student will graduate according to program
CAMPUS	Campus the student belongs to
SAD_LOAD_DESCR	Mode of study the student is in
ACAD_PLAN	Code for the specific study program the student is taking
ACAD_PROG_DESCR	Short form of the study program
TRNSCR_DESCR	Long, descriptive form of the study program
ACAD_ORG	Faculty the student belongs to
DISABILITY	Type of disability the student has
NATIONALITY	Nationality of the student
RACE	Race of the student
SEX	Sex of the student
MUET	MUET (Malaysian University English Test) score for the student
IELTS	IELTS score for the student
LOAN	Loan belonging to the student have
SPONSOR	Sponsorship belonging to the student
SCHOLAR	Scholarship belonging to the student
TOT_CUMULATIVE	Total cumulative credits
N_FINAL_RSLT_DESCR	Description of final program results
N_HONOUR_DESCR	Honours belonging to the student

Table 1b. Dataset description

CREDITREQUIRED	Total credits required to graduate
INFO1	SPM and STPM grades of student
Prog_Length(Term)	Number of terms student has done
T1 to T17: CUR_GPA	Current GPA for the term
T1 to T17: CUM_GPA	Cumulative GPA (CPGA) for the term

3.2. Exploratory Data Analysis

To understand the dataset, four data visualizations are created using a tool called Tableau. The variables used to generate the visualizations are graduate on time (GOT2), program status (PROG_STATUS), faculty name and admit term.

GOT2 is a binary categorical variable having Y or N, as the values. PROG_STATUS is a multiple categorical variable having six values, which are AC (Active in Program), CM (Completed), CN (Cancelled), DC (Discontinued), DM (Dismissal), and LA (Leave of Absence). The faculty name and admit terms were anonymized to preserve data privacy. There are seven faculties in total, ranging from FA1 to FA7, while there are six admit terms, from D1 to D6.

In Figures 1 and 2, the target variable is GOT, and bar plots showing the distribution of students among different faculties and admit terms are shown. From Figure 1, the faculties with the highest distribution of students who did not GOT are FA2 (60.33%), followed by FA4 (55.44%) and then FA3 (54.83%).

In Figure 2, the distribution of students who GOT or did not GOT across all admit terms is shown. From this, it is observed that the highest distribution of students who did not GOT are students who entered on D6 (53.98%), followed by D4 (53.55%) and D1 (52.22%).

From Figure 1 and 2, the problematic faculties and admit terms where the students are not able to GOT can be identified, and further investigation is done by focusing on these faculties and admit terms to understand the reason the students are not able to GOT.

For Figures 3 and 4, the target variable is program status, and multiple bar plots showing the distribution of students among different faculties and admit terms are shown. From Figure 3, FA1 (74.65%), FA5 (74.37%), and FA6 (72.66%) are the faculties with the highest distribution of students who completed their program (CM as program status).

From Figure 4, the distribution of students with their program status across all admit terms is shown. It can be seen clearly that D2 has the highest distribution (78.22%) of students who completed the program, followed by D3 (76.11%) and D1 (75.32%).

From Figures 3 and 4, the faculties and admit terms where majority of students were able to complete their program was identified, and further investigation can be done into these faculties and admit terms to understand the reason for the success of students.

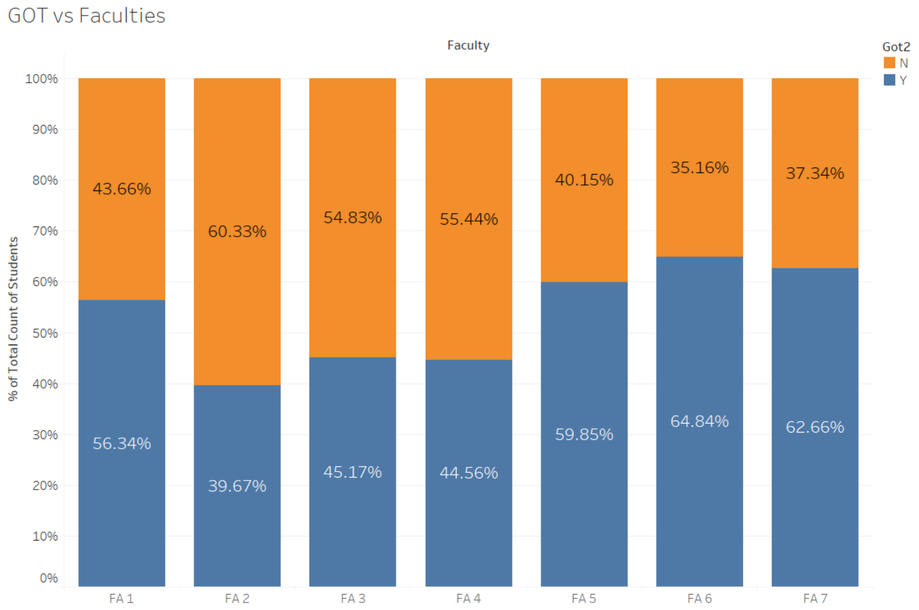

Fig.1: Bar chart representing distribution of students among the different faculties.

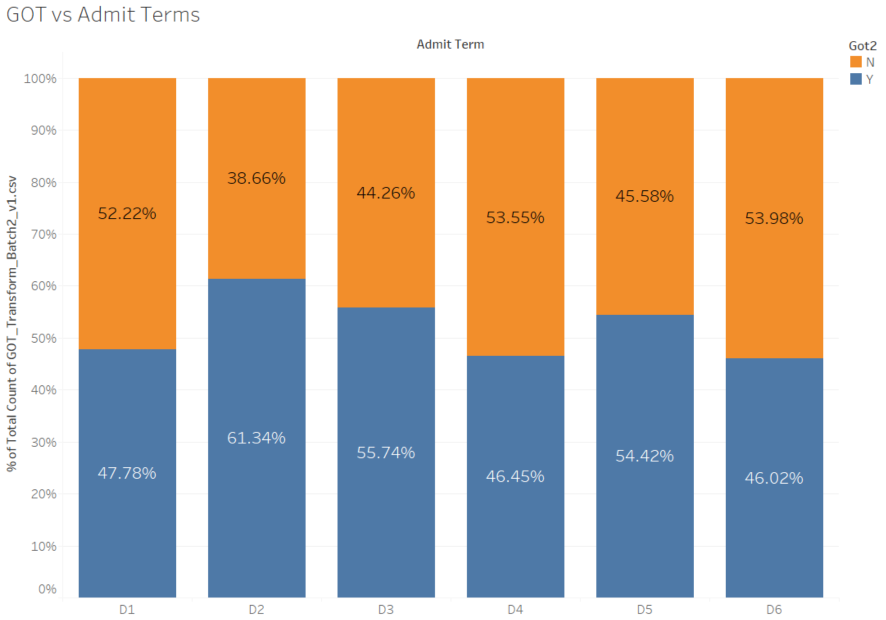

Fig.2: Bar chart representing distribution of students among the different admit terms.

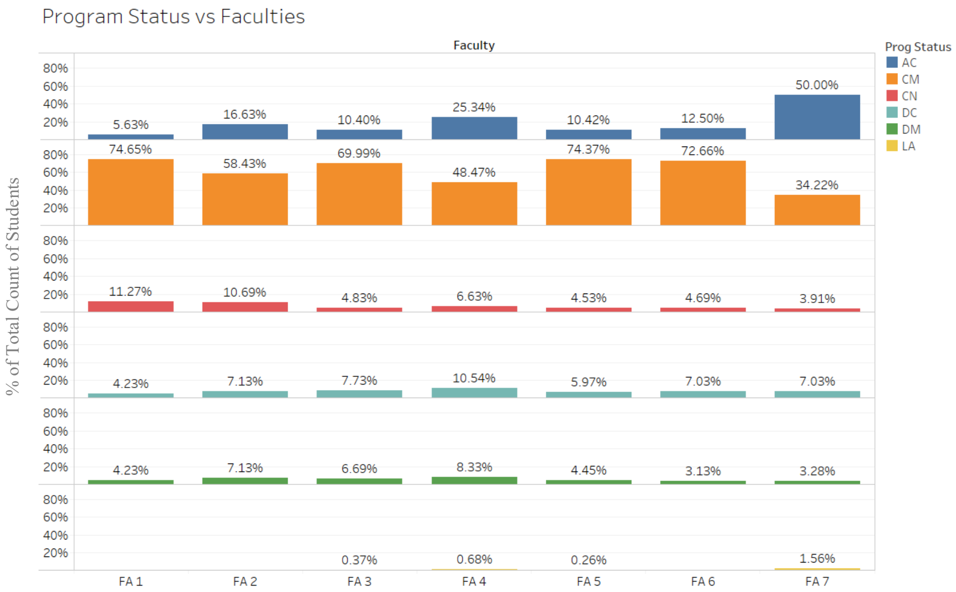

Fig.3: Bar chart representing distribution of students among the different faculties, with the status of their program representing the different colours.

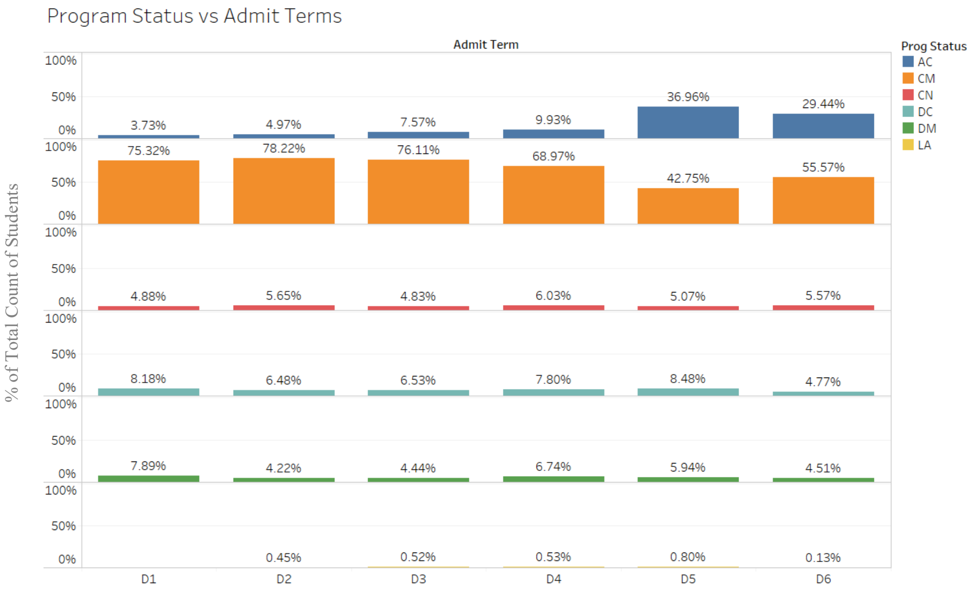

Fig.4: Bar chart representing distribution of students among the different admit terms, with the status of their program representing the different colours.

3.3. Data Preprocessing

In the dataset, there are a lot of columns with many missing values. Most of these will be removed. Some of these columns with just a little missing value, will be filled in with the mode value of the columns. The distinctness of the dataset is also checked, and any duplicate records are removed. Finally, label encoding is applied to convert the categorical features into numerical representations. This is because generally, the categorical data cannot be interpreted directly by the machine learning models.

3.4. Data Augmentation

Uniform noise addition is a technique used to introduce random noise to data by adding values sampled from a uniform distribution. In the context of this paper's data, it involves modifying the existing data points by adding random values within a specified range to each data point.

$$\boldsymbol{Z = a + (b - a) * rand\,(0, 1)} \tag{1}$$

- Z is the modified value of the column
- a is the minimum value of the column (min_value)
- b is the maximum value of the column (max_value)
- *rand* (0, 1) generates random values from a uniform distribution between 0 and 1

Uniform noise addition can be beneficial for several reasons:

- Increasing data variability: By adding random noise, the data points become more varied, which can help capture the natural variations present in the real-world scenarios.
- Enhancing model robustness: The added noise helps make the model more robust to small fluctuations or outliers in the data. It makes the model more resilient to noise in unseen data.
- Avoiding overfitting: By introducing randomness, uniform noise addition can prevent the model from overfitting by memorizing the specific training data and instead learn the underlying patterns and generalizations.
- Data augmentation: Adding random noise can effectively increase the size of the dataset without collecting additional data. This is particularly useful when working with limited data, as it provides more samples for the model to learn from.

However, it is essential to note that the effectiveness of uniform noise addition depends on the specific problem and the characteristics of the dataset. It may not always lead to improvements in model performance, and careful experimentation and evaluation are necessary to determine its impact.

3.5. Feature Selection

Before applying predictive analytics, feature selection is an essential stage. Feature selection aims to eliminate unnecessary variables from the data and speed up the computation of predictive models. Boruta is the algorithm of choice in this paper for feature selection. The reason Boruta is chosen is because it follows an all-relevant variable section method compared to the common minimal optimal method. This allows it to consider a wider selection of features that are relevant to the target class, causing it to have better performance than other methods. The algorithm uses Random Forest classifier as the estimator and ranks the features. For this paper's dataset, the top four features were academic career and the GPA results of students in terms six, ten and nine, respectively, in that order.

3.6. Resampling

The datasets often exhibit class imbalance, where one class significantly outweighs the other. This imbalance can lead to biased models, poor generalization, and reduced accuracy in predicting the minority class or rare events. For the dataset, the main class imbalance issue was with the Program Status target class. For GOT, the number of students who graduated on time was 2934 and students who did not graduate on time was 2554. For Program Status, there were 938 students who had Accepted (AC), 3554 students who had Completed (CM), 292 students who had Cancelled (CN), 390 students who had Discontinued (DC), 299 students who had Dismissed (DM), and 25 students who had Leave of Absence (LA). This makes sense as a majority of students would have completed the program on time. The remaining students might have had some issues causing them to not have Completed as their program status. By employing resampling techniques such as oversampling and undersampling, the class distribution can be rebalanced, allowing machine learning models to better capture the complexities of the classes.

In this paper four oversampling techniques (Random oversampling, SMOTE, ADASYN, SVM-SMOTE) and one hybrid sampling technique (SMOTE-Tomek links) were compared. The four oversampling techniques were considered as they generally add samples to the minority class to make it equal to the majority class, which makes the dataset into a normal dataset that most predictive model can be trained on. The disadvantage is sometimes the models trained can overfit, but methods can be implemented to counteract that issue. Undersampling techniques were not considered as they generally remove samples from the majority class to make it equal with the minority class. This leads to loss of data and will generally make the predictive model underfit. However, when undersampling techniques are combined with oversampling techniques to form hybrid sampling techniques, the disadvantages are reduced. In hybrid sampling, oversampling generally occurs first, increasing the samples of the minority class and then undersampling occurs, whereby similar samples from the minority and majority classes are removed. Therefore, this paper included one of the best hybrid sampling techniques called SMOTE-Tomek Link.

By employing these four oversampling methods, the paper aims to tackle the challenges posed by imbalanced datasets. Each method offers a unique approach to address class imbalance, and their effectiveness will be evaluated and compared based on relevant evaluation metrics such as accuracy, precision, recall, F1 score, and ROC AUC score. This comprehensive evaluation will provide insights into which oversampling method is most suitable for handling the imbalanced nature of the dataset and improving the performance of machine learning models.

3.6.1. Random Oversampling

Random oversampling is the most basic resampling technique in which the minority class samples are increased by randomly sampling from the majority class. In the end, both class samples will be equal.

3.6.2. SMOTE

Synthetic Minority Oversampling Technique (SMOTE) is a popular oversampling technique used to address class imbalance. It works by generating synthetic samples for the minority class to increase its representation in the dataset. The synthetic samples are created by interpolating between neighbouring minority class samples.

3.6.3. Tomek Links

Tomek Links is an under-sampling technique that aims to remove the samples from the majority class that are close to the decision boundary between the minority and majority classes. It identifies pairs of samples (one from the majority class and one from the minority class) that are nearest neighbours to each other. It removes most class samples from these pairs, making the decision boundary between the classes more distinct.

3.6.4. ADASYN

Adaptive Synthetic (ADAYSN) is another oversampling technique that generates synthetic samples for the minority class, like SMOTE. The difference is that ADASYN creates synthetic data according to data density which usually causes it to outperform SMOTE.

3.6.5. SMOTE-Tomek Links

First, SMOTE is applied to oversample the minority class, creating synthetic samples to balance the class distribution. Then, Tomek Links is applied to remove majority class samples that are close to the minority class samples, further improving the separation between the classes. The result is a modified dataset with a balanced class distribution and a clearer decision boundary.

By combining these techniques, the hybrid sampling approach helps address class imbalance while preserving the overall distribution of the data. It can enhance the performance of classification models by providing more representative samples and reducing the potential bias towards the majority class.

3.7. Model Construction

In this paper, multiple classification models were compared to predict the GOT and program status within the dataset used. The classification models that have been used in this paper are Decision Tree (DT), K-Nearest Neighbour (KNN), Logistic Regression (LR), Random Forest (RF), Stacking ensemble, Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LGB), Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU).

DT and LR were implemented as they are simple models that are easy to build and can act as a baseline for other models. KNN was implemented as it is an unsupervised model that can provide another perspective into the issue of predicting GOT. RF, XGBoost, LGB were all implemented as they are considered the best models from ensemble learning from literature review. Stacking ensemble was used to combine DT, KNN and RF together into one model to see if its performance is better than its individual components. LSTM and GRU were selected as they are the best models from deep learning to help understand and predict patterns in the dataset with respect to time.

The dataset was split into train sets and test sets. All the train-test splits ratios were performed on the dataset to evaluate and compare the performance of the models, from 90-10, 80-20, 70-30, 60-40 to 50-50. Evaluation metrics such as accuracy, precision, recall, F1 score, and ROC-AUC score will be used to assess and compare the effectiveness of these models in capturing the complex relationships and patterns in the data. After constructing the models, hyperparameter tuning has been done by applying grid search method to find the best combination of the hyperparameters that improve the models' performance.

In this paper, the target variables for binary classification is GOT and multiclass classification is program status. For each type of classification, ensemble learning, and deep learning methods are applied, and the results compared. Different types of class balancing methods have also been tested to find the best method.

3.8. Model Evaluation

After the model construction is done, model evaluation is an important step to perform. The aim of the model construction is to evaluate and find the best model with the best performance and how will the best model perform in the future. In this paper a few different metrics has been used to compare the performance of the models. They are accuracy, precision, recall and F1 Score. In addition, confusion matrix was used to compare the performance of the models as well. Results of the models will be discussed in the next section.

$$Accuracy = \frac{TN + TP}{TP + TN + FP + FN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$F1\,Score = 2\left(\frac{Precision \times Recall}{Precision + Recall}\right) \tag{5}$$

4. Results and Analysis

This section will analyse the results of the different types of modelling. The section is divided into binary classification and multiple classification sub sections.

For each sub section, the different types of machine learning techniques are compared against each other based on their F1 score and ROC AUC score. This is because for imbalanced datasets, accuracy is not a good measure as it will show misleading results. For example, the accuracy results of a model may be 99% but the model itself may be simply always predicting the majority class. This is similar in behaviour to just randomly predicting the class, which defeats the purpose of building a machine learning model.

F1 score and ROC AUC are much better measures for imbalanced datasets. F1 score can show the true model performance as it is calculated using harmonic mean of both precision and recall of the class which will show poor results if the model is simple predicting the majority class. Similarly, the ROC AUC curve is sensitive to class imbalance and the prediction of the minority class will have a strong impact on the ROC AUC value.

Each machine learning technique have also been tested with multiple class balancing techniques. However, the best method is the only one reported here to save space.

4.1. Binary classification

4.1.1. Logistic Regression and Random Forest

From Tables 2 and 3, it is observed that for binary classification, both RF and LR performs well, with above 90% F1-performance scores. The best scores overall however, belonged to the RF model, with 97.7% F1 score. For both tables, the best train test split ratio was 80-20. In general, applying resampling techniques to balance the data increased the F1 score performance of the model by around 4%.

4.1.2. Ensemble Learning

The Stacking ensemble model is built using RF Classifier, K-NN Classifier and DT Classifier. Tables 4 and 5 shows that the best model seems to be the RF classifier with F1 scores of 98.6% and 99.4% respectively. For both tables, the best train test split ratio was 90-10. The Stacking ensemble model performs well but not as good as the single RF classifier despite the ensemble model containing a RF classifier as one of its models. Applying random oversampling has increased the F1 score performance of the RF classifier by 8%.

Table 2. Results comparison between LR and RF using imbalanced dataset for Binary Classification

GOT - Binary Classification						
Imbalanced						
Model	**Train-Test Split**	**Accuracy**	**Precision**	**Recall**	**F1 score**	**ROC AUC**
Logistic Regression	50-50	0.929	0.929	0.929	0.929	0.986
	60-40	0.926	0.926	0.926	0.926	0.987
	70-30	0.914	0.914	0.914	0.914	0.978
	80-20	0.910	0.910	0.910	0.910	0.976
	90-10	0.913	0.913	0.913	0.913	0.978
Random Forest	50-50	0.966	0.966	0.966	0.966	0.995
	60-40	0.970	0.970	0.970	0.970	0.997
	70-30	0.975	0.975	0.975	0.975	0.997
	80-20	0.977	0.977	0.977	0.977	0.998
	90-10	0.976	0.976	0.976	0.976	0.998

Table 3. Results comparison between LR and RF using Random oversampling for Binary Classification

GOT - Binary Classification						
Random Oversampling						
Model	**Train-Test Split**	**Accuracy**	**Precision**	**Recall**	**F1 score**	**ROC AUC**
Logistic Regression	50-50	0.922	0.922	0.922	0.922	0.984
	60-40	0.919	0.919	0.919	0.919	0.986
	70-30	0.917	0.917	0.917	0.917	0.985
	80-20	0.924	0.924	0.924	0.924	0.982
	90-10	0.918	0.918	0.918	0.918	0.984
Random Forest	50-50	0.968	0.968	0.968	0.968	0.996
	60-40	0.972	0.972	0.972	0.972	0.997
	70-30	0.976	0.976	0.976	0.976	0.997
	80-20	0.981	0.981	0.981	0.981	0.998
	90-10	0.976	0.976	0.976	0.976	0.998

Table 4. Results comparison between RF, K-NN, DT and Stacking ensemble using imbalanced dataset for Binary Classification

GOT - Binary Classification						
Imbalanced						
Model	**Train-Test Split**	**Accuracy**	**Precision**	**Recall**	**F1 score**	**ROC AUC**
Random Forest	50-50	0.987	0.991	0.981	0.986	0.999
	60-40	0.987	0.990	0.981	0.986	0.999
	70-30	0.985	0.987	0.982	0.984	0.999
	80-20	0.985	0.988	0.980	0.984	0.999
	90-10	0.987	0.988	0.984	0.986	0.999
K-NN	50-50	0.959	0.970	0.941	0.955	0.987
	60-40	0.957	0.959	0.948	0.954	0.987
	70-30	0.959	0.967	0.945	0.956	0.986
	80-20	0.960	0.970	0.943	0.956	0.984
	90-10	0.954	0.960	0.941	0.950	0.980
Decision Tree	50-50	0.959	0.975	0.937	0.955	0.977
	60-40	0.950	0.940	0.953	0.947	0.977
	70-30	0.944	0.946	0.932	0.939	0.977
	80-20	0.934	0.942	0.916	0.929	0.971
	90-10	0.969	0.980	0.953	0.966	0.977
Stacking Ensemble Learning (RF + K-NN + DT)	50-50	0.986	0.984	0.986	0.985	0.998
	60-40	0.985	0.984	0.984	0.984	0.999
	70-30	0.985	0.984	0.983	0.984	0.998
	80-20	0.982	0.984	0.977	0.980	0.998
	90-10	0.982	0.980	0.980	0.980	0.997

Table 5. Results comparison between RF, K-NN, DT and Stacking ensemble using Random oversampling for Binary Classification

GOT - Binary Classification						
Random Oversampling						
Model	**Train-Test Split**	**Accuracy**	**Precision**	**Recall**	**F1 score**	**ROC AUC**
Random Forest	50-50	0.988	0.987	0.988	0.988	0.999
	60-40	0.990	0.990	0.989	0.990	0.999
	70-30	0.990	0.992	0.987	0.990	0.999
	80-20	0.990	0.992	0.988	0.990	0.999
	90-10	0.994	0.996	0.992	0.994	1.000
K-NN	50-50	0.960	0.973	0.947	0.960	0.990
	60-40	0.958	0.971	0.945	0.958	0.991
	70-30	0.965	0.976	0.954	0.965	0.989
	80-20	0.970	0.976	0.963	0.969	0.988
	90-10	0.963	0.972	0.953	0.963	0.987
Decision Tree	50-50	0.964	0.975	0.951	0.963	0.968
	60-40	0.966	0.959	0.974	0.966	0.985
	70-30	0.934	0.942	0.926	0.934	0.961
	80-20	0.919	0.910	0.930	0.920	0.961
	90-10	0.926	0.940	0.910	0.925	0.965
Stacking Ensemble Learning (RF + K-NN + DT)	50-50	0.984	0.981	0.986	0.984	0.998
	60-40	0.986	0.983	0.988	0.986	0.999
	70-30	0.986	0.988	0.983	0.986	0.997
	80-20	0.983	0.984	0.982	0.983	0.997
	90-10	0.980	0.980	0.980	0.980	0.994

4.1.3. Deep Learning

Overall, from observing the results in Tables 6 and 7, the performance of both deep learning models for binary classification are similar. For all train-test ratios the F1 score is around 70% for both LSTM and GRU model. The best performance was GRU in Table 6 with 70.8% F1 score and LSTM in Table 7 with 69.8% F1 score. From the results in Table 7, it can be observed that applying data augmentation has generally not affected the F1 score performance of the models. The performance slightly decreases by 1%.

Table 6. Results comparison between LSTM and GRU using imbalanced dataset for Binary Classification

GOT - Binary Classification						
Imbalanced						
Model	**Train-Test Split**	**Accuracy**	**Precision**	**Recall**	**F1 score**	**ROC AUC**
LSTM	50-50	0.921	0.890	0.975	0.704	0.967
	60-40	0.899	0.853	0.983	0.702	0.935
	70-30	0.917	0.884	0.977	0.703	0.965

	80-20	0.928	0.904	0.972	0.708	0.960
	90-10	0.893	0.835	0.986	0.680	0.930
GRU	50-50	0.901	0.871	0.960	0.704	0.972
	60-40	0.904	0.877	0.956	0.702	0.970
	70-30	0.899	0.858	0.975	0.703	0.971
	80-20	0.898	0.867	0.962	0.708	0.969
	90-10	0.853	0.784	0.986	0.680	0.965

Table 7. Results comparison between LSTM and GRU using data augmentation for Binary Classification

GOT - Binary Classification						
With Data Augmentation (Uniform Noise Addition)						
Model	**Train-Test Split**	**Accuracy**	**Precision**	**Recall**	**F1 score**	**ROC AUC**
LSTM	50-50	0.946	0.927	0.977	0.698	0.986
	60-40	0.937	0.902	0.989	0.697	0.988
	70-30	0.947	0.985	0.915	0.695	0.990
	80-20	0.964	0.976	0.955	0.689	0.992
	90-10	0.955	0.944	0.972	0.685	0.990
GRU	50-50	0.915	0.902	0.944	0.698	0.977
	60-40	0.912	0.876	0.973	0.697	0.978
	70-30	0.925	0.909	0.954	0.695	0.978
	80-20	0.920	0.901	0.953	0.689	0.977
	90-10	0.9135	0.9613	0.8689	0.685	0.9764

4.2. Multiple Classification

4.2.1. Logistic Regression and Random Forest

From Tables 8 and 9, it is observed that the RF model scores better than LR in all cases for multiclass classification. The best performance in both tables is the 90-10 train test split and was 99.5% F1 score for Table 8 and 99.8% F1 score for Table 9. Among all the methods, RF with ADASYN achieves the highest scores, which is an increase of 3% compared to the imbalanced model. All the F1 score outcomes of RF are more than 90%, while most of the outcomes of LR are less than 90%, with some of them only scoring around 60%.

Table 8. Results comparison between LR and RF using imbalanced dataset for Multiple Classification

PROG_STATUS - Multiple Classification						
Imbalanced						
Model	**Train-Test Split**	**Accuracy**	**Precision**	**Recall**	**F1 score**	**ROC AUC**
Logistic Regression	50-50	0.831	0.831	0.831	0.831	0.907
	60-40	0.805	0.805	0.805	0.805	0.909
	70-30	0.812	0.812	0.812	0.812	0.925
	80-20	0.808	0.808	0.808	0.808	0.927
	90-10	0.794	0.794	0.794	0.794	0.925
Random Forest	50-50	0.991	0.991	0.991	0.991	0.999

	60-40	0.992	0.992	0.992	0.992	0.999
	70-30	0.993	0.993	0.993	0.993	1.000
	80-20	0.989	0.989	0.989	0.989	0.999
	90-10	0.995	0.995	0.995	0.995	0.999

Table 9. Results comparison between LR and RF using ADASYN for Multiple Classification

PROG_STATUS - Multiple Classification						
ADASYN						
Model	**Train-Test Split**	**Accuracy**	**Precision**	**Recall**	**F1 score**	**ROC AUC**
Logistic Regression	50-50	0.699	0.699	0.699	0.699	0.860
	60-40	0.696	0.696	0.696	0.696	0.853
	70-30	0.687	0.687	0.687	0.687	0.881
	80-20	0.687	0.687	0.687	0.687	0.874
	90-10	0.619	0.619	0.619	0.619	0.848
Random Forest	50-50	0.993	0.993	0.993	0.993	0.999
	60-40	0.996	0.996	0.996	0.996	0.999
	70-30	0.997	0.997	0.997	0.997	1.000
	80-20	0.996	0.996	0.996	0.996	0.999
	90-10	0.998	0.998	0.998	0.998	0.999

4.2.2. Ensemble Learning

From Tables 10 and 11, the performance of both XGBoost and LGB models are good, with above 90% score results for all train-test split ratios. After applying the hybrid resampling technique (SMOTE-Tomek Link), the overall scores for both models increased. The best model for imbalanced data seems to be LGB with 92.9% F1 score and for the balanced dataset, it seems to be XGBoost with 93.1%. For both cases, the best train-test split is 90-10.

Table 10. Results comparison between XGBoost and LGB using imbalanced dataset for Multiple Classification

PROG_STATUS - Multiple Classification						
Imbalanced						
Model	**Train-Test Split**	**Accuracy**	**Precision**	**Recall**	**F1 score**	**ROC AUC**
XGBoost	50-50	0.919	0.916	0.919	0.916	0.980
	60-40	0.916	0.914	0.916	0.914	0.979
	70-30	0.924	0.921	0.924	0.921	0.982
	80-20	0.917	0.916	0.917	0.915	0.984
	90-10	0.918	0.917	0.918	0.916	0.987
LGB	50-50	0.920	0.916	0.920	0.917	0.978
	60-40	0.920	0.917	0.920	0.918	0.979
	70-30	0.923	0.921	0.923	0.920	0.980
	80-20	0.918	0.917	0.918	0.917	0.983
	90-10	0.931	0.930	0.931	0.929	0.987

Table 11. Results comparison between XGBoost and LGB using hybrid resampling for Multiple Classification

PROG_STATUS - Multiple Classification						
SMOTE-Tomek Links						
Model	**Train-Test Split**	**Accuracy**	**Precision**	**Recall**	**F1 score**	**ROC AUC**
XGBoost	50-50	0.922	0.921	0.922	0.921	0.982
	60-40	0.915	0.915	0.915	0.914	0.981
	70-30	0.916	0.915	0.916	0.915	0.979
	80-20	0.917	0.916	0.917	0.916	0.984
	90-10	0.931	0.932	0.931	0.931	0.987
LGB	50-50	0.921	0.920	0.921	0.919	0.980
	60-40	0.921	0.919	0.921	0.919	0.979
	70-30	0.917	0.916	0.917	0.915	0.981
	80-20	0.916	0.916	0.916	0.916	0.985
	90-10	0.922	0.923	0.922	0.922	0.986

4.2.3. Deep Learning

In this multiclass classification, a challenge is faced as both the deep learning models cannot produce proper results. As can be seen in the comparison tables, the performance scores for all the train-test ratios are not in the expected range. There also does not seem to be much effect from applying data augmentation Thus, it can be concluded that both the LSTM and GRU are not suitable for multiclass classification in this paper, or that the model should be trained using a lot more data to see a satisfying outcome. There is also a possibility that using Uniform Noise Addition as the data augmentation method may be incorrect for multiple classification, and other methods need to be applied.

Table 12. Results comparison between LSTM and GRU using imbalanced dataset for Multiple Classification

PROG_STATUS - Multiple Classification						
Original						
Model	**Train-Test Split**	**Accuracy**	**Precision**	**Recall**	**F1 score**	**ROC AUC**
LSTM	50-50	0.174	0.000	0.000	0.383	0.576
	60-40	0.177	0.984	0.136	0.380	0.587
	70-30	0.176	1.000	0.142	0.382	0.863
	80-20	0.174	1.000	0.137	0.383	0.887
	90-10	0.186	1.000	0.141	0.389	0.770
GRU	50-50	0.174	0.000	0.000	0.383	0.357
	60-40	0.179	0.000	0.000	0.380	0.453
	70-30	0.176	0.000	0.000	0.382	0.501
	80-20	0.174	0.000	0.000	0.383	0.561
	90-10	0.186	0.000	0.000	0.389	0.710

Table 13. Results comparison between LSTM and GRU using data augmentation for Multiple Classification

PROG_STATUS - Multiple Classification						
With Data Augmentation (Uniform Noise Addition)						
Model	**Train-Test Split**	**Accuracy**	**Precision**	**Recall**	**F1 score**	**ROC AUC**
LSTM	50-50	0.170	1.000	0.132	0.388	0.889
	60-40	0.170	1.000	0.136	0.389	0.809
	70-30	0.174	1.000	0.163	0.389	0.852
	80-20	0.173	1.000	0.113	0.391	0.870
	90-10	0.173	1.000	0.161	0.386	0.889
GRU	50-50	0.170	0.000	0.000	0.388	0.528
	60-40	0.170	0.000	0.000	0.389	0.719
	70-30	0.174	0.000	0.000	0.389	0.649
	80-20	0.173	0.000	0.000	0.391	0.533
	90-10	0.173	0.000	0.000	0.386	0.628

5. Conclusions

In conclusion, this study presented ensemble and deep learning models for identifying academically at-risk students. The Random Forest model achieved the best performance with F1-score of 99.4% for binary classification (predicting GOT) and 99.8% for multiple classification (predicting Program Status), demonstrating the efficacy of ensemble techniques. Applying class balancing has generally always increased the performance of the models. The deep learning methods showed promise but underperformed on multi-class prediction. Applying data augmentation using Uniform Noise Addition has not greatly affected the performance of the model. This was somewhat expected, and more testing and experimentation needs to be done to find a way to improve the model performance using this method. Other methods may also be explored in the future. Overall, this study makes useful contributions in applying advanced ML techniques to guide student interventions. However, the models can be improved by using larger datasets and better hyperparameter tuning. As future work, incorporating additional student features like social, behavioural and financial data could provide more holistic insights. Testing different network architectures and ensemble combinations could also help advance the state-of-the-art in student risk prediction. In summary, this study takes an important step toward enabling proactive analytics to enhance student success and retention rates.

Acknowledgements

This research is supported by Fundamental Research Grant Scheme (FRGS), FRGS/1/2022/ICT06/MMU/03/3, and TM Research & Development Grant (TM R&D), MMUE/220028.

References

Adnan, M., Habib, A., Ashraf, J., Mussadiq, S., Raza, A. A., Abid, M., Bashir, M., & Khan, S. U. (2021). Predicting at-Risk Students at Different Percentages of Course Length for Early Intervention Using Machine Learning Models. *IEEE Access*, *9*, 7519–7539. https://doi.org/10.1109/ACCESS.2021.3049446

Agnihotri, L., & Ott, A. (2014). Building a Student At-Risk Model: An End-to-End Perspective From User to Data Scientist. *Educational Data Mining*. https://api.semanticscholar.org/CorpusID:16485228

Ahmed, S. A., & Khan, S. I. (2019). A machine learning approach to Predict the Engineering Students at risk of dropout and factors behind: Bangladesh Perspective. *2019 10th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2019*. https://doi.org/10.1109/ICCCNT45670.2019.8944511

Alija, S., Beqiri, E., Gaafar, A. S., & Hamoud, A. K. (2023). Predicting Students Performance Using Supervised Machine Learning Based on Imbalanced Dataset and Wrapper Feature Selection. *Informatica*, *47*(1), 11–20. https://doi.org/10.31449/INF.V47I1.4519

Baranyi, M., Nagy, M., & Molontay, R. (2020). Interpretable Deep Learning for University Dropout Prediction. *SIGITE 2020 - Proceedings of the 21st Annual Conference on Information Technology Education*, 13–19. https://doi.org/10.1145/3368308.3415382

Hegde, V., & Prageeth, P. P. (2018). Higher education student dropout prediction and analysis through educational data mining. *Proceedings of the 2nd International Conference on Inventive Systems and Control, ICISC 2018*, 694–699. https://doi.org/10.1109/ICISC.2018.8398887

Maldonado, S., Miranda, J., Olaya, D., Vásquez, J., & Verbeke, W. (2021). Redefining profit metrics for boosting student retention in higher education. *Decision Support Systems*, *143*, 113493. https://doi.org/10.1016/J.DSS.2021.113493

Maslow, A. H. (1943). A theory of human motivation. *Psychological Review*, *50*(4), 370–396. https://doi.org/10.1037/H0054346

Maslow, A. H., Frager, R., Fadiman, J., McReynolds, C., & Cox, R. (1987). Motivation and personality (3rd). *New York*.

Mulyani, E., Hidayah, I., & Fauziati, S. (2019). Dropout Prediction Optimization through SMOTE and Ensemble Learning. *2019 2nd International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2019*, 516–521. https://doi.org/10.1109/ISRITI48646.2019.9034673

Nagy, M., & Molontay, R. (2018). Predicting Dropout in Higher Education Based on Secondary School Performance. *INES 2018 - IEEE 22nd International Conference on Intelligent Engineering Systems, Proceedings*, 000389–000394. https://doi.org/10.1109/INES.2018.8523888

Naseem, M., Chaudhary, K., Sharma, B., & Lal, A. G. (2019). Using Ensemble Decision Tree Model to Predict Student Dropout in Computing Science. *2019 IEEE Asia-Pacific Conference on Computer Science and Data Engineering, CSDE 2019*. https://doi.org/10.1109/CSDE48274.2019.9162389

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 and Education: Artificial Intelligence*, *3*, 100066. https://doi.org/10.1016/J.CAEAI.2022.100066

Omar Alkhamisi, A., & Mehmood, R. (2020). An Ensemble Machine and Deep Learning Model for Risk Prediction in Aviation Systems. *Proceedings - 2020 6th Conference on Data Science and Machine Learning Applications, CDMA 2020*, 54–59. https://doi.org/10.1109/CDMA47397.2020.00015

Ong, S. Y., Ting, C. Y., Goh, H. N., Quek, A., & Cham, C. L. (2023). Workplace Preference Analytics Among Graduates. *Journal of Informatics and Web Engineering*, *2*(2), 233–248. https://doi.org/10.33093/JIWE.2023.2.2.17

Pongpaichet, S., Jankapor, S., Janchai, S., & Tongsanit, T. (2020). Early Detection At-Risk Students using Machine Learning. *International Conference on ICT Convergence*, *2020-October*, 283–287. https://doi.org/10.1109/ICTC49870.2020.9289185

Revathy, M., Kamalakkannan, S., & Kavitha, P. (2022). *Machine Learning based Prediction of Dropout Students from the Education University using SMOTE*. 1750–1758. https://doi.org/10.1109/ICSSIT53264.2022.9716450

Sahlaoui, H., Alaoui, E. A. A., Nayyar, A., Agoujil, S., & Jaber, M. M. (2021). Predicting and Interpreting Student Performance Using Ensemble Models and Shapley Additive Explanations. *IEEE Access*, *9*, 152688–152703. https://doi.org/10.1109/ACCESS.2021.3124270

Soobramoney, R., & Singh, A. (2019). Identifying students at-risk with an ensemble of machine learning algorithms. *2019 Conference on Information Communications Technology and Society, ICTAS 2019*. https://doi.org/10.1109/ICTAS.2019.8703616

Sun, D., Mao, Y., Du, J., Xu, P., Zheng, Q., & Sun, H. (2019). Deep learning for dropout prediction in MOOCs. *Proceedings - 2019 8th International Conference of Educational Innovation through Technology, EITT 2019*, 87–90. https://doi.org/10.1109/EITT.2019.00025

Tsai, S. C., Chen, C. H., Shiao, Y. T., Ciou, J. S., & Wu, T. N. (2020). Precision education with statistical learning and deep learning: a case study in Taiwan. *International Journal of Educational Technology in Higher Education*, *17*(1), 1–13. https://doi.org/10.1186/S41239-020-00186-2/FIGURES/3

Ul Alam, M. A. (2022). College Student Retention Risk Analysis from Educational Database Using Multi-Task Multi-Modal Neural Fusion. *Proceedings of the AAAI Conference on Artificial Intelligence*, *36*(11), 12689–12697. https://doi.org/10.1609/AAAI.V36I11.21545

Waheed, H., Hassan, S. U., Aljohani, N. R., Hardman, J., Alelyani, S., & Nawaz, R. (2020). Predicting academic performance of students from VLE big data using deep learning models. *Computers in Human Behavior*, *104*, 106189. https://doi.org/10.1016/J.CHB.2019.106189