 https://doi.org/10.24191/jeesr.v23i1.006

Mitigating Imbalanced Classification Problems in Academic Performance with Resampling Methods

A'zraa Afhzan Ab Rahim*, Norlida Buniyamin

***Abstract*—The imbalanced dataset is a common problem in the educational performance environment, where the number of students with poor performance is much less than those who perform well. This can create problems when predicting academic performance using machine learning algorithms, which assume that the datasets have a balanced distribution across all classes. We compared three resampling methods: SMOTE, Borderline SMOTE, and ADASYN, and used five different classifiers (Logistic Regression, Support Vector Machine, Naïve Bayes, K-Nearest Neighbor, and Decision Tree) on three imbalanced educational datasets. We used five-fold cross-validation to assess two performance indicators: accuracy and recall. Although accuracy indicates the overall performance, we focus more on recall values because it is more incumbent to identify poor-performing students so that necessary interventions can be executed promptly. Our results showed that when resampling improved recall values, ADASYN outperforms SMOTE and Borderline SMOTE consistently, better classifying the poor-performing students. Overall, our results suggest that resampling methods can be effective in addressing the problem of imbalanced classification in academic performance. However, the choice of resampling method should be carefully considered, as the performance of different methods can vary depending on the classifier used.**

***Index Terms* —Academic performance, imbalanced classification, machine learning, resampling algorithms.**

I. INTRODUCTION

A persistent issue in tertiary education is students' poor academic performance, which delays graduation or, even worse, leads to dropout. Ideally, all engineering undergraduate students who enrol in their chosen university should complete their studies on time, satisfy all minimum requirements, and obtain all learning outcomes within the stipulated time outlined by the university. The task of understanding, modeling, and analyzing student performance in higher education institutions (HEI) presents considerable hurdles in terms of developing accurate diagnostic models since a myriad of factors needs tobeconsidered[1]–[4].

According to the Malaysian Ministry of Higher Education [5], delayed graduation causes greater financial costs for a university. Therefore, it becomes important to develop a model that accurately predicts students' performance. In the Mid-term report by Malaysia's Economic Planning Unit (EPU), among the priority areas emphasized in the 11th Malaysian Plan is concerted efforts towards raising the quality of graduates so they have better employment opportunities and contribute to economic growth [6].

The potential to predict student performance paves the way for improving their educational outcomes by allowing educators to identify at-risk students, as indicated by lackluster GPAs, so that remedial targeted intervention could be implemented to avoid late graduation or, worse, immature withdrawals. To do this, HEIs can harvest data stored in their repositories. Nevertheless, evaluating a large volume of data to extract useful information is time-consuming if done manually; hence, educational Data Mining (EDM) can extract important and significant knowledge from the data [7]. A plethora of literature can be found on the applications of EDM using classifiers such as K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Naïve Bayes (NB), Logistic Regression (LR), Decision Tree (DT) on an educational dataset in the context of predicting attrition and retention [8]–[10] and predicting academic performance [11]–[14]. However, finding a suitable machine learning model for correctly estimating student achievement remains difficult and active research.

Despite much research that has successfully developed intelligent classifiers for predicting student academic performance using machine learning [13]–[15], the extant literature has sparsely given attention to cases of imbalanced educational data. Often overlooked, imbalanced datasets are nevertheless significant because the commonly used measurements of accuracy, applied as an indicator of a model's goodness of fit, can be misleading [7][16]–[18].

Imbalanced data refers to problems where the dataset contains an unequal distribution of instances in some classes over others [19]. Due to the imbalance in the distribution of training data, conventional classifiers are frequently overwhelmed by the majority class and perform badly in the minority class. The problem is that when there are fewer observations in the minority class, building predictive

This manuscript is submitted on 12th January 2023 and accepted on 17th April 2023. A.A.A.R is presently developing and comparing an academic performance prediction model using single versus ensemble models that consider cognitive, non-cognitive, and demographic factors on the imbalanced dataset (e-mail: azraa@uitm.edu.my).

N. Buniyaminis now a Professor of Electrical Engineering at UiTM, a Fellow of the Institution of Engineers, Malaysia, and an Hon Fellow of the ASEAN Federation of Engineering Organisation (AFEO).(e-mail: nbuniyamin@uitm.edu.my).

*Corresponding author
Email address: azraa@uitm.edu.my

boundaries to separate the minority class from other classes becomes nontrivial. This means that the model may not recognize patterns and features from the minority and majority groups. Therefore, samples from the minority class are often misclassified as if they belong to the majority class. However, when imbalanced datasets are involved, predicting the minority class is often more of interest [20].

For example, in fraud detection [21], [22], most transactions are non-fraudulent; however, fraudulent transactions occur in exceptional cases. Since non-fraudulent transactions dominate the samples and fraudulent cases are unique, it is necessary to address the imbalanced dataset to predict fraudulent transactions. Another example is when trying to predict rare medical diagnoses, for example, cancer detection [23], where the emphasis is on being able to predict the positive case, which lies in the minority class. In predicting rare cases, it is important to predict the rare cases correctly so that the correct treatment can be administered to the patient. Performing a classification algorithm without addressing the issue of imbalanced training data and using accuracy as the model's performance indicator misleadingly leads to high accuracy.

The same scenario happens in the educational domain, as highlighted by [24], where they investigated three educational datasets with three different modeling tasks: forum post classification, drop-out prediction, and predicting whether a student pass or fails. The imbalance of the three datasets is contributed by the male/female student ratio. Imbalance can also be in terms of the number of students who graduate on time being far greater than those who graduate late. In our work, the number of poor-performing students is significantly less than those who perform average and excellent, contributing to class imbalance, similar to[25].

A. Problems associated withimbalanced dataset

The factors that influence the ability of a classifier to correctly identify the class labels are the size of the dataset, the separability of the classes, and the presence or absence of within-class clusters [26]–[29]. Sample size plays a crucial role in determining the goodness of a model. If a sample size is limited, finding patterns inherent to the small class is challenging, but the prediction error may decrease as the dataset increases. In [30], the authors investigate the effect of sample size and class distribution in assessing credit risk, emphasizing real-life imbalanced data sets. According to them, classification algorithms demonstrated varying levels of sample size sensitivity. When the sample size decreased, logistic regression and neural networks found a clear trend of deteriorating accuracy. However, the effect was much smaller than anticipated, indicating that reducing the sample size does not result in a major loss in performance which constitutes an important discovery for credit scoring.

Class separability refers to how different the observations of the different classes are [29], [31]. And intuitively, the more different they are, the easier they are to classify. If patterns among the classes overlap, it becomes more complicated for an algorithm to find the boundaries that separate one class from another. On the other hand, linearly separable domains are insensitive to any amount of imbalance [28]. The authors in [28] also discovered that the more imbalanced a class causes an increase in concept complexity, and the lower number of sample sizes in the training set, the greater the influence of class imbalances on classifiers affected by the problem. This discovery was attributed to high degrees of complexity and imbalance, in addition to small training setsizes, producing small subclusters that cannot be correctly categorized. However, not all the classifiers were affected equally by the class imbalance problem: the decision tree was the most affected by class imbalance, Neural Networks (NN) demonstrated sensitivity variations, and Support Vector Machines (SVM) were the least susceptible, exhibiting immunity toward class imbalance.

In classification problems, a single class can be composed of sub-clusters indicating that the class is not homogeneous. As explained by [32], a frequent issue encountered by classification algorithms is when samples of the same class are separated in the input space. The idea that a class can be divided into several sub-clusters dispersed across the input space is common. The absence of homogeneity is especially troublesome in algorithms centered on dividing and conquering (e.g., decision trees) and set covering (e.g., rule induction), whereby the sub-clusters lead to the formation of small disjuncts. Sub-clusters within a class increase the complexity of the minority class, making it harder to detect the boundaries that separate the classes.

In a nutshell, data imbalance per se does not automatically suggest that it is more difficult to build a model that correctly predicts the minority classes. Instead, factors like having sufficient samples for the algorithms to learn from, the classes being well separated, and the presence or absence of sub-clusters also play a role [26], [29].

B. Strategies to address the imbalance.

Throughout the literature, there is a consistent trend pointing to two types of solutions for the imbalance classification problem. The first is the data-level or also referred to as the external approach, and the second type is the algorithm-level or internal approach [33]–[39].

Data-level approaches refer to altering the class distribution of the training set. The algorithm-level requires the development of new classification algorithms or changing the existing classification algorithms to overcome the bias imposed by the class imbalance [39]. According to Estabrooks et al. [35], the downside of algorithm-level, although they may be highly efficient in certain circumstances, they are algorithm-dependent. This is a concern since datasets with different features may be better classified by different algorithms, and it would be infeasible to apply the proposed change for the class imbalance problem from one classifier to another. Alternatively, data-level methods are model-independent and, therefore, more flexible. We concur with [35] hence this article is limited to investigating only data-level methods to overcome the class imbalance.

Data-level methods can be grouped into oversampling and undersampling. Oversampling approaches increase the number of examples in the minority class. Undersampling methods eliminate instances from the majority class, especially in problems where a big dataset is available, and the observations provide redundant information [34]. Random under-sampling is simple to implement and does not require a learning algorithm. It also has the advantage of reducing the size of the

training set, which may improve the learning speed. However, under-sampling may remove potentially useful instances from the majority class, which may reduce the accuracy of the classifier.

The Synthetic Minority Oversampling Technique (SMOTE) creates new observations from the minority class by interpolation instead of merely duplicating the samples in the minority class like random undersampling [18]. The new samples from the minority class are not identical to the original ones, and this way, it overcomes the main limitation associated with random oversampling. However, if outliers from the minority class are present in the majority class, SMOTE generates synthetic samples between the minority class and the outliers [40]. This can lead to increased bias in the synthetic samples and, ultimately, a decrease in the overall model performance. To overcome the limitation of SMOTE, Borderline SMOTE has been suggested by [41], which generates synthetic minority class samples along the borderline between the minority and majority classes to increase the samples where the samples are harder to classify.

Nevertheless, Borderline-SMOTE does not create a discriminative model. Instead, samples in the minority class are assigned weights according to their density and use the SMOTE algorithm to generate synthetic examples for those with the lowest density. ADASYN, on the other hand, uses a weighted distribution of the minority class to create the new synthetic data, which is weighted according to how difficult the observations are to be learned so that more synthetic data is generated for the minority class [42]. This helps reduce the bias introduced by class imbalance and adaptively shifts the classification decision boundary toward the difficult examples.

Therefore, our research compares three resampling algorithms: Synthetic Minority Oversampling Technique (SMOTE), Borderline SMOTE, and Adaptive Synthetic (ADASYN) algorithm that addresses class imbalance to obtain a more representative measure of a model's performance. Because the effect of imbalanced data is seldomly addressed in academic performance prediction problems, this article aims to:

1) Compare the performance of resampling algorithms on imbalanced data to predict students' performance to determine whether resampling methods solve all class imbalance problems.
2) To assess the performance of different ML algorithms for multi-class classification problems involving academic performance prediction considering imbalanced data with cognitive, non-cognitive, and demographic input features.

Our study contributes to the literature on the use of resampling methods for improving classifier performance in imbalanced datasets and emphasizes the importance of selecting an appropriate resampling method for a specific dataset and classifier.

The rest of the paper is organized into four sections. The next section begins with a review of the extant literature on algorithms used to overcome the imbalanced dataset problem to predict academic performance. Section 3 discusses the research methodology implemented to achieve the objective of this work, while the fourth section offers the results and a detailed discussion of the findings once the imbalanced data are addressed and the impact on model performance. The final section concludes the current work and suggestions for future work.

II. Literature Review

Machine learning algorithms have recently been actively used to predict students' grades, attrition rate, or graduating on time (GOT). Even though academic performance prediction data almost always involve some form of imbalance, it is insufficiently discussed in the literature. This is evident when a search of the terms ("classification" OR "multi-classification" OR "multi-class") AND ("academic" AND "student") AND ("performance" OR "achievement" OR "success") in the English language in the SCOPUS database resulted in 1477 results. However, when the term "imbalance*" was included along with the previous terms, a stark reduction of only 27 articles was found. Next, the titles and abstracts of all 27 articles were checked to remove articles that do not explicitly investigate resampling methods to combat imbalance in the academic dataset, of which eight articles were removed. An example of removed articles included A Novel Stress State Assessment Method for College Students Based on EEG. Eight of the remaining 19 articles could not be included because they are inaccessible via the university library's subscription. The summary of the 11 articles is presented in Table I.

Table I
Summary of articles that have used resampling methods to address the class imbalance in the educational domain

	Input	Resampling Methods	ML Algorithm	Instances
[43]	A	S, U	DT, LR., RF, SVM	4396
[44]	A, D, SB	S, BS, Ada	DT, LR, KNN, NB, RF, SVM	550
[45]	A	S	NB, NN, SVM	44
[46]	A	S, Ada, ROS, SENN	DNN, DT, GB, KNN, LR, RF SVM	4266
[25]	A	S	DT, KNN, LR, NB, RF, SVM	1282
[47]	A, D, SE	S, RUS, ROS	DT, NB, NN, KNN, SVM	350
[48]	A	S	RF	2406
[49]	A	U	DT, E, NN, RF,	6882
[50]	A, D	O, U, S	DT, LR, NN, SVM	21,654
[51]	A, D	O, U	DT, NB, NN	117
[52]	NA	S	DT, NB, SVM	Set 1: 20492 Set 2: 936 Set 3: 151 Set 4: 1024

*A= Academic, D= Demographics, SB = Social-Behavioral, SE = Socio-economic, S = SMOTE, U= Undersampling, BS= Borderline SMOTE, Ada = AdaBoost, ROS = Random Oversampling, SENN = SMOTE-ENN, RUS= Random Undersampling, O = Oversampling, DT = Decision Tree, LR= Logistic Regression, RF = Random Forest, SVM = Support Vector Machine, , KNN = K-Nearest Neighbor, NB= Naïve Bayes,, NN = Neural Network, DNN= Deep Neural Network, GB= Gradient Boosting, E = Ensemble, NA = Not available.

Referring to Table I, it can be deduced that SMOTE is among the more popular methods of addressing the imbalance problem in the educational domain. In [43], an early detection system using 4396 academic data instances, taken only from the first four semesters to predict students with a high tendency to fail

was accomplished by comparing SMOTE with the undersampling method on four classifiers and found that the combination of SMOTE using RBF kernel with SVM resulted in the best performance with accuracy: 94.37%; precision: 94.37%; recall: 84.93%; F-measure: 74.18%. Similar trends can be found in [44] and [46], where both works compared SMOTE, SMOTE variations, and ADASYN and found that SMOTE consistently resulted in the best performance. Therefore, we conclude that it is imperative to include SMOTE as one of the candidates for the resampling method in our work. Borderline SMOTE is also considered since it only oversamples in the minority class that is close to the decision boundary, as opposed to SMOTE, which increases the size of the minority class without any discrimination, which suggests that Borderline SMOTE is more targeted in its approach and can lead to more accurate models [41]. Although [44] reported that the best performance in terms of the highest accuracy was achieved with SMOTE with an accuracy of 94.54%, a closer look at the recall values from Table IV of their article showed that ADASYN had comparatively similar recall performance as that of SMOTE. Authors in [46] also concluded that the best resampling method was SMOTE, but when zooming in on Table 9 in their article, it showed that ADASYN had comparable recall performance as SMOTE. Hence why we included ADASYN in our resampling methods to be investigated.

III. METHODOLOGY

This paper aims to compare the performance of different oversampling methods, such as SMOTE, Borderline SMOTE, and ADASYN, to overcome the challenges of class imbalance. Each oversampling method is tested on five different classifiers to find the combination that results in the best performance measure in terms of accuracy and recall. All models were executed in Python with the Anaconda platform. The methodology used to attain these goals is depicted in Fig. 1with the pseudocode presented in Algorithm 1, where the subsequent sections elaborate in detail the steps entailed in each phase.

A. Dataset curation.

To ascertain whether resampling methods impacted the recall values, especially for the poor-performing students, this study utilizes three datasets: original data collected from the UiTM students and another dataset from Kaggle. Our research differs from most of the research in Table I because, on top of academic and demographic data, we also consider students' motivational tendencies and how they study. This is true for datasets 1 and 2, which consider students' motivational inclination via the Motivated Strategies for Learning Questionnaire (MSLQ) [53]. It is a self-report questionnaire designed to identify students' motivational tendencies and learning methods. Students rated themselves using a seven-point Likert scale ranging from 'not at all true of me' to 'very true of me'. Two elements comprise the MSLQ: motivation

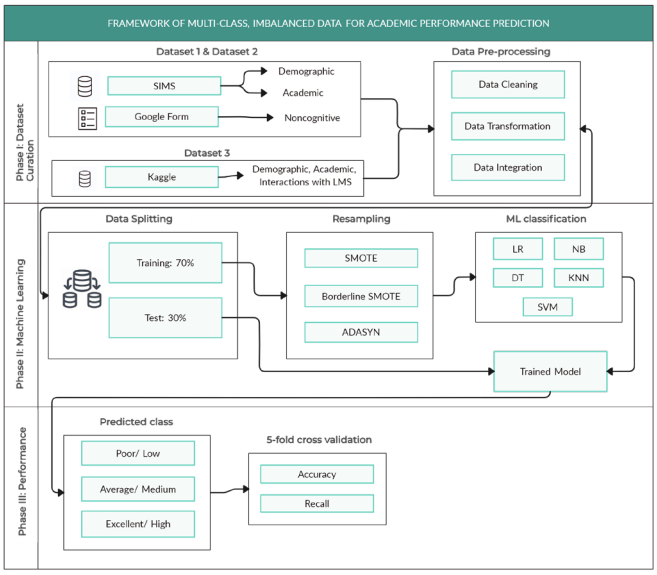


Fig.1.Framework of multi-class imbalanced data for academic performance prediction

Algorithm 1: Multi-class Imbalanced Data for Academic Performance Prediction
Inputs: .csv file containing data from SIMS and Google form
Outputs: mean accuracy and recall scores for each model and resampling method

```
// Phase 1: Dataset curation
BEGIN
  // Get data from 2 separate sources: SIMS and Google Form
  // Dataset Pre-processing: Remove redundant data, transform categorical to numerical, combine data sources.
END

// Phase 2: Machine learning
BEGIN
  // Data splitting: Training 70% Test 30%
  X_train, X_test, y_train, y_test = split_data(X, y)

  // Perform SMOTE, Borderline SMOTE, and ADASYN resampling on the training data.
  resampling_methods = [SMOTE, BorderlineSMOTE, ADASYN]
  for each resampling_method in resampling_methods do
    X_resampled, y_resampled = resample(resampling_method, X_train, y_train)
    // Feed the original and resampled training data to 5 different models: LR, NB, DT, KNN, SVM
    // to predict whether students belong to Poor/ Average/ Excellent class.
    models = [LogisticRegression(), GaussianNB(), DecisionTreeClassifier(), KNeighborsClassifier(), SVC()]
    for each model in models do
      model.fit(X_resampled, y_resampled)
      // Store all the confusion matrix values including precision, recall, accuracy ad f1-score for each model.
      y_pred = model.predict(X_test)
      confusion_matrix_values = compute_confusion_matrix(y_test, y_pred)
      compute precision_recall_accuracy_f1_score(confusion_matrix_values)
      store_results (resampling_method, model_name (model), precision _recall_accuracy_f1_score_values)
    END FOR
  END FOR
END

# Phase 3: Performance

# Perform 5-fold cross-validation
for model in models:
  accuracy, recall = perform_cross_validation(model, transformed_dataset, 5)
```

and learning techniques. The motivational scale has six subscales (intrinsic goal orientation, extrinsic goal orientation, task value, control of learning beliefs, self-efficacy, and test anxiety). The motivation scale consists of 31 items that measure students' goals and values, opinions of their ability to succeed in the course, and anxiety around course exams. The learning techniques consist of nine subscales: rehearsal, elaboration, organization, critical thinking, metacognitive self-regulation, time and study environment, effort regulation, peer learning, and help-seeking. Thirty-one items examine students' perceptions of their cognitive and metacognitive techniques, while the remaining 19 questions deal with students'

management of diverse educational materials. There are 81 items in the 1991 version of the MSLQ. Following [53] and [54], scales were created by averaging scale elements. For example, the composite score for metacognitive self-regulation (MSR) was determined by adding the scores of all twelve questions and dividing them by twelve. The following paragraphs briefly describe each dataset, and the summary of all three datasets used in our work is provided in Table II, while Table III denotes the input features and predicted outcomes for all three datasets.

TABLE II
SUMMARY OF THE DATASETS USED

Set	*Total*	*Training Data*			*Test Data*		
		P	A	E	P	A	E
1	188	13	68	50	6	30	21
2	702	56	270	165	24	116	71
3	480	89	148	99	38	63	43

*P= Poor, A = Average, E= Excellent

B. Data Collection

Dataset 1

One hundred eighty-eight final-year students from one engineering school in UiTM participated in the study. They were all from the same cohort, March 2018 intake. Out of 188, 86 (46%) are female, and 102 (54%) are male. Their CGPA upon graduation was acquired from the Students Information and Management System (SIMS) and is utilized as the predicted output, designated as CGPA8. This is a multi-class classification since CGPA8 is divided into three classes: 'Poor' ($CGPA < 3$), 'Average' ($3 \leq CGPA < 3.5$), and 'Excellent' ($CGPA \geq 3.5$).

Dataset 2

In dataset two, 702 final-year students from four engineering schools in UiTM participated in the study. Among the 702, 410 are male students (58.40%), and the remaining 292 (41.60%) are female. Their final semester CGPA, which is the predicted output in this research, is separated into three classes: 'Poor' ($CGPA < 3$), 'Average' ($3 \leq CGPA < 3.5$) and 'Excellent' ($CGPA \geq 3.5$) making this a multi-class classification problem.

Dataset 3

This dataset is obtained from Kaggle [55]. It is a set of educational data gathered by the learning management system (LMS) Kalboard 360, a multi-agent learning management system (LMS) developed to promote learning with cutting-edge technologies. The data comprises 480 student records and 16 features. The characteristics are divided into three primary groups: (1) Demographic, (2) Academic background, and (3) Behavioural characteristics such as raising a hand in class, using school resources, responding to a parent survey, and school satisfaction. There are 305 males and 175 females in the sample from various countries. The dataset was collected over two academic semesters: 245 student records were gathered during the first semester, and 235 were gathered during the second semester. The predicted output is the cumulative marks, where the students were divided into three classes: low-level: marks between 0 to 69, middle-level: marks between 70 to 89; and high-level: marks between 90 to 100. A detailed explanation of the dataset can be found in [56].

TABLE III
LIST OF INPUT FEATURES AND PREDICTED OUTCOMES FOR ALL THREE DATASETS.

Category/ Set	***Set 1***	***Set 2***	***Set 3***
Demographic	Gender, Household Income	Gender, Household Income	Nationality, Gender, Place of Birth, Parent responsible for student
Academic	Academic Program, grades for five common subjects, GPA and CGPA for Sem 1 and Sem 2, GPA for Sem 8	Diploma CGPA, GPA for semesters 3 to 8, the grade for Calculus subject	Educational Stages (school levels), Grade Levels, Section ID, Semester, Course Topic, Student Absence Days
Other	**Non-cognitive:** Intrinsic Goal Orientation, Extrinsic Goal Orientation, Task Value, Control Of Learning Beliefs, Self-Efficacy for Learning and Performance, Test Anxiety, Rehearsal, Elaboration, Organization, Critical Thinking, Metacognitive Self-Regulation, Time and Study Environment, Effort Regulation, Peer Learning, Help Seeking	**Non-cognitive:** Intrinsic Goal Orientation, Extrinsic Goal Orientation, Task Value, Control Of Learning Beliefs, Self-Efficacy for Learning and Performance, Test Anxiety, Rehearsal, Elaboration, Organization, Critical Thinking, Metacognitive Self-Regulation, Time and Study Environment, Effort Regulation, Peer Learning, Help Seeking	**Behavioral Features on Kalboard 360:** Discussion groups, visited resources, raised hand in class, viewing announcements. **Parents Participation:** Parent Answering Survey, Parent School Satisfaction
Predicted Outcome	3 classes: Excellent, Average, Poor performance	3 classes: Excellent, Average, Poor performance	3 classes: Low, Middle, High

C. Data Pre-processing

Data pre-processing applies analytical techniques to convert an incomprehensible dataset into a meaningful and quality format that can be used for further processing [57]. This is necessary so that the programming language, Python, can read the information smoothly and ensure that the data is deprived of inconsistencies in training and testing the models in proceeding steps. Among the steps taken in our work:

1. Data Cleaning: removes or corrects inaccurate or incomplete records from the dataset. It also involves dealing with missing values, outliers, and other inconsistencies. It is necessary to remove duplicate data because a student answered

the MSLQ survey more than once. Constant features that do not aid in learning are also removed.

2. Data Transformation: transforming the data into a format compatible with Python. This includes normalizing data, transforming categorical variables into numerical variables, and scaling data.

3. Data Integration: Combines multiple datasets into a single dataset. For datasets 1 and 2, the data are obtained from two sources (SIMS and Google form); hence it is necessary to identify, extract, and combine them into one .csv file.

D. Data Splitting

Data splitting is a common method used in machine learning to divide a dataset into two or more subsets. The purpose of data splitting is to provide separate sets of data for training and testing a machine learning model. We implemented a 70/30 split, where the dataset is divided into two subsets: a training set that contains 70% of the data and a testing set that contains the remaining 30%. The training set is used to train the machine learning model, while the testing set is used to evaluate the model's performance. The data splitting ensures that the model is not overfitted to the training data. By using a separate testing set to evaluate the model's performance, we can ensure that the model has learned to make predictions that are accurate on the training data and new data.

E. Resampling

SMOTE

According to [32], the SMOTE algorithm starts by setting the amount of oversampling, N, usually set to achieve an approximate 1:1 class distribution. First, it selects a positive class instance randomly from the training set. Then, it obtains the K-nearest neighbors of that instance. Finally, it randomly selects N of these neighbors and generates new samples by interpolating their feature vectors. To do this, it calculates the difference between the feature vector of the instance under consideration and each neighbor, multiplies this difference by a random number between 0 and 1, and adds it to the previous feature vector. This randomly creates a new synthetic sample at a point along the line segment between the original instance and its neighbor.

Borderline SMOTE

Borderline SMOTE is a variation of the SMOTE algorithm, which can generate new, synthetic data to increase the samples of the minority class in a dataset [41].The algorithm works by applying the k-nearest neighbors to the entire dataset. It would find and ignore the samples from the minority class if most neighbors are also from the minority class because these samples are easy to classify. It also ignores samples from the minority class if all its neighbors are from the majority class treating it as noise. It then finds the samples from the minority class located on the border between the classes, treating them as the danger group. These are called borderline samples. Once the borderline samples are identified, the algorithm creates new synthetic observations by interpolating between them and their nearest neighbors. This synthetic data is then added to the original dataset, increasing the number of minority samples.

ADASYN

The ADASYN algorithm uses a density distribution to automatically decide the number of synthetic samples that need to be generated for each minority data sample in an imbalanced dataset [42]. The density distribution measures the distribution of weights for different minority class samples based on how difficult they are to learn. By using the density distribution as a criterion, the ADASYN algorithm can generate a more balanced representation of the data distribution according to the desired balance level. This differs from the SMOTE algorithm, which produces equal numbers of synthetic samples for each minority data example. After resampling with ADASYN, most observations are created closer to the interface between the classes because more samples are generated from those samples that are harder to classify. And the situations that are harder to classify are those at the boundary between the two classes.

F. Learning Algorithms

Decision Tree (DT)

The learning strategy for decision trees involves creating decision rules that partition the data into separate classes [58]. The decision tree algorithm begins by selecting an attribute for the root node and then recursively splits the data based on the selected attribute. The algorithm then uses an impurity measure to determine the best split. The tree grows until each node is completely pure, containing only one class. One of the challenges of using decision trees with imbalanced data is that it can lead to overfitting.

Logistic regression (LR)

Logistic regression uses a logistic function to estimate the probability of a given data point belonging to a given class [59]. The logistic regression model is trained using a maximum likelihood estimation, which finds the weights that maximize the probability of correctly predicting the class labels of the training data. The main difficulty with logistic regression when dealing with imbalanced data is that it tends to be biased toward the majority class. This means the model is more likely to predict the majority class than the minority class.

Support Vector Machines (SVM)

The goal of an SVM is to find the optimal boundary between different classes. To do this, SVMs use the maximum margin classifiers to separate the classes by finding the maximum distance between them [60]. SVM first takes the training data and converts it into a high-dimensional feature space. This is done using a kernel function, transforming the data into a higher-dimensional space. Next, the SVM finds the optimal boundary between the different classes. This boundary, called the maximum margin hyperplane, maximizes the distance between the classes. Once the maximum margin hyperplane has been found, the SVM uses it to predict new samples.

Naïve Bayes (NB)

The learning strategy for Naive Bayes uses the Bayes theorem to calculate the probability of an event occurring based on prior knowledge of certain events [61]. This is done by calculating individual probabilities for each event and multiplying them to get the overall probability. The learning difficulty with imbalanced data is that the data may not be representative of the population, which can lead to inaccurate predictions.

K-nearest neighbors (KNN)

The K-nearest neighbors learning strategy classify data points based on their similarity to other data points in the training dataset [62]. KNN works by finding the k-nearest neighbors of a given data point (where k is a user-defined parameter), then assigning the data point to the most common class among those k-nearest neighbors. KNN has the advantage of being easy to understand and implement, and it can work with both continuous and categorical data. However, it can be difficult to use when dealing with imbalanced data, where one class is much more common than others. This can lead to overfitting, where the model performs well on the training dataset but poorly on unseen data.

G. Model Performance

The confusion metric in Table IV visualizes the number of correctly and incorrectly classified instances. In dataset 1 and 2, the model's output was classified into one of three classes; excellent (A), average (B), and poor (C) performance. To make the labels uniform, the predicted outcomes in Dataset 3 were also changed from high to excellent, medium to average, and low to poor.

TABLE IV
CONFUSION MATRIX FOR MULTI-CLASS ACADEMIC PERFORMANCE CLASSIFICATION

		Actual/ True Class		
		A	B	C
Predicted Class	A	AA	AB	AC
	B	BA	BB	BC
	C	CA	CB	CC

From Table IV, the accuracy is obtained using (1):

$$Accuracy = \frac{AA+BB+CC}{Numberofsamples} \quad (1)$$

For imbalanced datasets, accuracy is not an appropriate metric because it does not distinguish between the numbers of correctly classified examples of the different classes [20]. The minority class has very little impact on the overall accuracy value because there are fewer samples in the minority class [63]. For an imbalanced dataset, the accuracy is not indicative of the performance of an algorithm since it always has a high value, regardless of what algorithm is used. The most important limitation is that it cannot classify the minority class, which is the class of interest, hence, using accuracy for the imbalanced dataset is unsuitable [63][64].

Recall, also known as sensitivity, indicates the number of correctly identified observations from the considered class. For example, the recall value for the poor-performing class is the number of correctly predicted poor-performing students out of the actual poor-performing students. It can be translated to (2)

$$Recall = \frac{CC}{AC + BC + CC} \quad (1)$$

In our work, predicting the minority class, which is predicting poor academic performance, is more important, hence, recall is the better metric to assess the effect of resampling algorithm performance on the recall values. An increase in recall decreases the probability of misclassifying a sample from a particular class. Therefore, it is desirable to have a high recall rate.

H. Model Validation

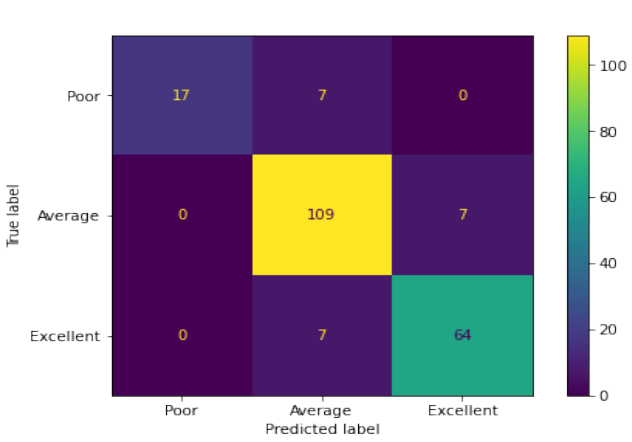
We used the 5-fold cross-validation, a common technique used to assess the performance of a machine learning model. It involves randomly splitting the dataset into five equal-sized subsamples [65]. One subsample is kept as the validation set, while the other four are used to train the model. The process is repeated five times using a different validation set. The results are then averaged to produce an overall estimate of model performance. This method helps to reduce the variance in the model's performance, as it is tested on multiple datasets and subsets of the data.

IV. RESULTS AND DISCUSSION

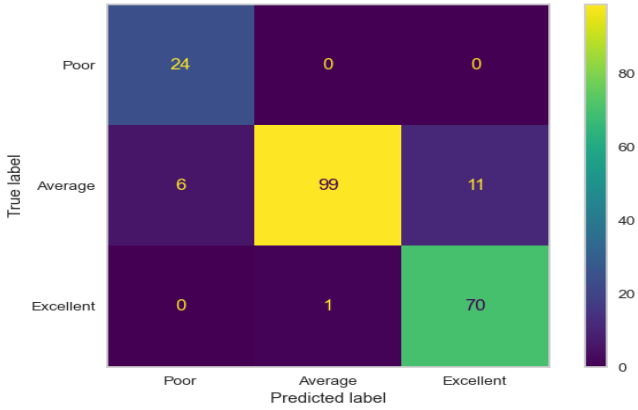
The results for all experiments are tabulated in Table V. From Table V, it is interesting to note that for Dataset 2, when no resampling was applied, the LR obtained a high accuracy of 97%, with the recall value for the Poor class only 88%. However, the inclusion of resampling methods, regardless of the type used, increased the recall value for the Poor class to 92%. A similar trend was observed for the KNN classifier, which obtained a high accuracy of 90% but only a 71% recall value for the Poor class before resampling. Incorporating resampling methods saw the recall values for the Poor class increase to 88% (SMOTE), 100% (ADASYN), and 88% (Borderline SMOTE). For clarity, Fig.2 compares the confusion matrix before and after ADASYN.

To analyze the general performance across three datasets, the recall values of all three resampling methods (SMOTE, ADASYN, Borderline SMOTE) in Table V are compared against the recall values when no resampling was applied for each of the five classifiers (DT, LR, KNN, SVM, NB) across all three datasets summing a total of 45 observations. The result is then divided into three categories: Recall increase, Recall does not change, and Recall decrease, which is presented in Table VI. It can be seen from Table VI that applying ADASYN leads to improved recall values in 37.78% of the experiment. In contrast, Borderline SMOTE performed the worst when 40% of the experiment resulted in recall deterioration.

There are several possible reasons why Borderline SMOTE may perform poorly in terms of recall. This could be because the method may not suit the datasets under consideration. Different resampling methods can have varying effectiveness depending on the characteristics of the data, such as the distribution of classes, the amount of noise, and the number of samples [42]. It could also be due to the method being too aggressive in generating synthetic samples [41]. This means that the Borderline SMOTE tends to produce excessive synthetic samples in regions where minority class examples are concentrated. This overemphasis on regions with a high concentration of minority class examples can result in the training data being overfitted, where the model becomes too closely fitted to the training data and cannot generalize to new data [66].



(a)



(b)

Fig. 2. Confusion Matrix of the KNN classifier with (a) no resampling and (b) ADASYN in Dataset 2

TABLE V
ACCURACY AND RECALL RESULTS FOR ALL THREE DATASETS

Resample	Model	Dataset 1			Dataset 2			Dataset 3		
		Accuracy	Class	Recall	Accuracy	Class	Recall	Accuracy	Class	Recall
No	DT	0.7	Poor	0.67	0.89	Poor	0.79	0.64	Low	0.74
			Average	0.7		Average	0.9		Medium	0.56
			Excellent	0.71		Excellent	0.9		High	0.67
	LR	0.77	Poor	0.83	0.97	Poor	0.88	0.74	Low	0.89
			Average	0.77		Average	**0.97**		Medium	0.7
			Excellent	0.76		Excellent	**1**		High	0.67
	KNN	0.81	Poor	0.83	0.9	Poor	0.71	0.63	Low	0.79
			Average	**0.9**		Average	0.94		Medium	0.62
			Excellent	0.67		Excellent	0.9		High	0.51
	SVM	0.74	Poor	0.5	**0.98**	Poor	0.96	0.63	Low	0.76
			Average	0.77		Average	**0.97**		Medium	0.63
			Excellent	0.76		Excellent	**1**		High	0.51
	NB	**0.84**	Poor	**1**	0.89	Poor	0.88	0.47	Low	0.87
			Average	0.9		Average	0.88		Medium	0.13
			Excellent	0.71		Excellent	0.92		High	0.63
SMOTE	DT	0.72	Poor	0.67	0.85	Poor	0.75	0.61	Low	0.79
			Average	0.7		Average	0.87		Medium	0.52
			Excellent	0.76		Excellent	0.86		High	0.58
	LR	0.77	Poor	0.83	0.97	Poor	0.92	0.75	Low	0.89
			Average	0.77		Average	**0.97**		Medium	0.68
			Excellent	0.76		Excellent	**1**		High	0.72
	KNN	0.81	Poor	0.83	0.89	Poor	0.88	0.59	Low	0.84
			Average	0.87		Average	0.85		Medium	0.46
			Excellent	0.71		Excellent	0.94		High	0.56
	SVM	0.74	Poor	0.5	**0.98**	Poor	0.96	0.61	Low	0.84
			Average	0.77		Average	**0.97**		Medium	0.43
			Excellent	0.76		Excellent	**1**		High	0.67
	NB	0.82	Poor	0.83	0.83	Poor	0.58	0.47	Low	0.84

Resample	Model		Dataset 1			Dataset 2			Dataset 3	
			Average	**0.9**		Average	0.79		Medium	0.16
			Excellent	0.71		Excellent	0.97		High	0.58
ADASYN	DT	0.72	Poor	0.67	0.89	Poor	0.79	0.72	Low	0.87
			Average	0.83		Average	0.89		Medium	0.65
			Excellent	0.57		Excellent	0.92		High	0.7
	LR	0.77	Poor	0.83	0.97	Poor	0.92	**0.76**	Low	0.89
			Average	0.77		Average	**0.97**		Medium	0.71
			Excellent	0.76		Excellent	**1**		High	**0.72**
	KNN	0.72	Poor	0.83	0.91	Poor	**1**	0.6	Low	0.87
			Average	0.57		Average	0.85		Medium	0.41
			Excellent	0.9		Excellent	0.99		High	0.65
	SVM	0.74	Poor	0.5	**0.98**	Poor	0.96	0.59	Low	0.84
			Average	0.77		Average	**0.97**		Medium	0.4
			Excellent	0.76		Excellent	**1**		High	0.65
	NB	0.77	Poor	0.67	0.85	Poor	0.79	0.47	Low	0.84
			Average	0.8		Average	0.78		Medium	0.16
			Excellent	0.76		Excellent	0.99		High	0.58
Borderline SMOTE	DT	0.79	Poor	0.67	0.88	Poor	0.71	0.63	Low	0.82
			Average	0.83		Average	0.91		Medium	0.52
			Excellent	0.76		Excellent	0.9		High	0.63
	LR	0.77	Poor	0.83	0.96	Poor	0.92	0.75	Low	**0.89**
			Average	0.77		Average	0.96		Medium	0.68
			Excellent	0.76		Excellent	0.99		High	**0.72**
	KNN	0.77	Poor	0.83	0.88	Poor	0.88	0.59	Low	0.84
			Average	0.8		Average	0.84		Medium	0.46
			Excellent	0.71		Excellent	0.96		High	0.56
	SVM	0.74	Poor	0.5	0.97	Poor	0.92	0.61	Low	0.84
			Average	0.77		Average	**0.97**		Medium	0.43
			Excellent	0.76		Excellent	0.99		High	0.67
	NB	0.79	Poor	0.83	0.92	Poor	0.75	0.47	Low	0.84
			Average	0.83		Average	**0.97**		Medium	0.16
			Excellent	0.71		Excellent	0.9		High	0.58

TABLE VI
COMPARISON OF RECALL VALUES FOR DIFFERENT RESAMPLING METHODS

	Recall Increase (%)	Recall does not change (%)	Recall Decrease (%)
SMOTE	28.89	**37.78**	33.33
ADASYN	**37.78**	33.33	28.89
Borderline SMOTE	33.33	26.67	**40.00**

Table VII compares the change in all five classifiers' recall values. A total of 27 experiments consisted of three classes (Poor, Average, and Excellent) across three datasets for all three resampling methods. As can be seen, LR and SVM are more immune to the resampling methods when more than 50% of the experiments resulted in no variation of the recall after resampling was introduced. This could be attributed to how SVMs learn by finding the optimal separating hyperplanes between classes, which is less affected by the class imbalance, making it more robust. On the other hand, KNN consistently showed improved recall values when 55.56% of the experiments resulted in recall increment. This is because KNN is more effective when the training data is balanced, as it allows the algorithm to determine the true neighbors of each data point and make more accurate predictions.

Because we are more interested in predicting the poor-performing students, who all belong to the minority class in all three datasets, so that necessary intervention can be applied, we zoom in on the recall performance of all resampling methods across all three datasets.

TABLE VII
COMPARISON OF RECALL VALUES FOR DIFFERENT CLASSIFIERS

	Recall Increase (%)	Recall does not change (%)	Recall Decrease (%)
DT	40.74	22.22	**37.04**
LR	22.22	**59.26**	18.52
KNN	**55.56**	11.11	33.33
SVM	22.22	**59.26**	18.52
NB	25.93	11.11	62.96

According to Table V, LR repeatedly produced the best recall in all three datasets, as all their recall values exceeded 80% for the Poor group. It is worth noting that all resampling methods have deteriorated recall values for the NB classifier. On the other hand, resampling methods seem to have no impact on LR and SVM since their recall values are similar before and after resampling was applied across all three datasets.

V. CONCLUSION AND IMPLICATIONS OF RESEARCH.

In conclusion, we evaluated the effectiveness of three popular resampling methods (SMOTE, ADASYN, Borderline SMOTE) in improving the recall values of a minority class, specifically the "Poor" class, in three datasets. To assess the performance of the resampling methods, we used five different classification algorithms (DT, LR, KNN, SVM, NB). Our results, as tabulated in Table V, revealed that incorporating resampling methods led to a significant improvement in the recall values of the "Poor" class, particularly for the LR and KNN classifiers. Notably, the ADASYN method resulted in improved recall values in 37.78% of the experiments. Conversely, we observed that the Borderline SMOTE method performed poorly in 40% of the experiments, resulting in a decline in recall values. This suggests that different resampling methods are not equally effective, and the choice of method depends on the dataset and classifier used. We also found that LR and SVM classifiers were more robust to the resampling methods, while KNN consistently showed improved recall values.

The implications of this study are various. The study supports the idea that resampling methods can effectively enhance the performance of classifiers for minority classes. The study found that resampling methods can increase recall values for the minority class, as demonstrated in this study.

The choice of resampling method is critical, as the study results indicate that different resampling methods can have varying effectiveness depending on the characteristics of the data. The poor performance of the Borderline SMOTE method in this study emphasizes the importance of selecting the appropriate resampling method for a specific dataset and classifier.

Additionally, the study found that KNN classifiers were consistently more effective when the training data was balanced, which can be achieved through resampling methods. The study's findings can be useful for practitioners in educational settings where early intervention can be applied to poor-performing students.

In future work, it would be interesting to explore other resampling methods, classifiers, and different evaluation metrics better suited for imbalanced datasets. It would also be interesting to study the effects of imbalanced classification on the performance of different learning algorithms, such as deep learning models, and to investigate the use of ensemble methods, such as XGBoost, for improving the performance on imbalanced datasets.

ACKNOWLEDGMENT& FUNDING

The authors would like to acknowledge and thank UniversitiTeknologi MARA (UiTM) for the financial grant 600-RMC/MyRA 5/3/LESTARI (118/2020) titled "Heterogeneous Ensemble Method To Predict First-Year Engineering Students Academic Performance Using Cognitive Non-Cognitive And Demographic Traits."

AUTHOR STATEMENT

The authors affirmed that there is no conflict of interest in this article. A'zraaAfhzan carried out the fieldwork, prepared the literature review, overlooked the whole article's writeup, wrote the research methodology, and did the data entry. A'zraaAfhzan also conducted the statistical analysis and interpretation of the results with suggestions from Norlida.

REFERENCES

[1] A. Alshanqiti and A. Namoun, "Predicting Student Performance and its Influential Factors using Hybrid Regression and Multi-label Classification," *IEEE Access*, vol. 8, pp. 203827–203844, 2020, doi: 10.1109/ACCESS.2020.3036572.

[2] J. Galopo Perez and E. S. Perez, "Predicting Student Program Completion Using Naïve Bayes Classification Algorithm," *International Journal of Modern Education and Computer Science*, vol. 13, no. 3, pp. 57–67, Jun. 2021, doi: 10.5815/ijmecs.2021.03.05.

[3] M. T. Sathe and A. C. Adamuthe, "Comparative study of supervised algorithms for prediction of students' performance," *International Journal of Modern Education and Computer Science*, vol. 13, no. 1, pp. 1–21, 2021, doi: 10.5815/ijmecs.2021.01.01.

[4] R. C. Deo, Z. M. Yaseen, N. Al-Ansari, T. Nguyen-Huy, T. A. M. P. Langlands, and L. Galligan, "Modern Artificial Intelligence Model Development for Undergraduate Student Performance Prediction: An Investigation on Engineering Mathematics Courses," *IEEE Access*, vol. 8, pp. 136697–136724, 2020, doi: 10.1109/ACCESS.2020.3010938.

[5] Ministry of Higher Education Malaysia, *Enhancing academic productivity and cost efficiency: University Transformation programme, Silver Book*. 2017.

[6] Economic Planning Unit, "Mid-Term Review of the Eleventh Malaysia Plan, 2016-2020: New Priorities and Emphases," 2018.

[7] R. Ghorbani and R. Ghousi, "Comparing Different Resampling Methods in Predicting Students' Performance Using Machine Learning Techniques," *IEEE Access*, vol. 8, pp. 67899–67911, 2020, doi: 10.1109/ACCESS.2020.2986809.

[8] N. Hutagaol and Suharjito, "Predictive Modelling of Student Dropout Using Ensemble Classifier Method in Higher Education," *Advances in Science, Technology and Engineering Systems*, vol. 4, no. 4, pp. 206–211, 2019, doi: 10.25046/aj040425.

[9] C. Mason, J. Twomey, D. Wright, and L. Whitman, "Predicting Engineering Student Attrition Risk Using a Probabilistic Neural Network and Comparing Results with a Backpropagation Neural Network and Logistic Regression," *Res High Educ*, vol. 59, no. 3, pp. 382–400, 2018, doi: 10.1007/s11162-017-9473-z.

[10] R. Alkhasawneh and R. H. Hargraves, "Developing a Hybrid Model to Predict Student First Year Retention and Academic Success in STEM Disciplines using Neural Network," *J STEM Educ*, vol. 15, no. 3, pp. 35–42, 2014.

[11] S. Huang and N. Fang, "Predicting Student Academic Performance in an Engineering Dynamics Course: A Comparison of Four Types of Predictive Mathematical Models," *Comput Educ*, vol. 61, pp. 133–145, 2013, doi: 10.1016/j.compedu.2012.08.015.

[12] J. -P. Vandamme, N. Meskens, and J. -F. Superby, "Predicting Academic Performance by Data Mining Methods," *Educ Econ*, vol. 15, no. 4, pp. 405–419, 2007, doi: 10.1080/09645290701409939.

[13] A. I. Adekitan and E. Noma-Osaghae, "Data Mining Approach to Predicting the Performance of First Year Student in a University Using the Admission Requirements," *Educ Inf Technol (Dordr)*, vol. 24, pp. 1527–1543, 2019, doi: 10.1007/s10639-018-9839-7.

[14] A. I. Adekitan and O. Salau, "The Impact of Engineering Students' Performance in the First Three Years on their Graduation Result using Educational Data Mining," *Heliyon*, vol. 5, no. 2, p. e01250, 2019, doi: 10.1016/j.heliyon.2019.e01250.

[15] S. Al-Sudani and R. Palaniappan, "Predicting Students' Final Degree Classification using an Extended Profile," *Educ Inf Technol (Dordr)*, vol. 24, no. 4, pp. 2357–2369, 2019, doi: 10.1007/s10639-019-09873-8.

[16] T. M. Barros, P. A. S. Neto, I. Silva, and L. A. Guedes, "Predictive Models for Imbalanced Data: A School Dropout Perspective," *Educ Sci (Basel)*, vol. 9, no. 4, 2019, doi: 10.3390/educsci9040275.

[17] L. A. Jeni, J. F. Cohn, and F. de La Torre, "Facing Imbalanced Data - Recommendations for the Use of Performance Metrics," in *Proceedings - 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, ACII 2013*, 2013, pp. 245–251. doi: 10.1109/ACII.2013.47.

[18] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research* , vol. 16, pp. 321–357, 2002.

[19] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Editorial: Special Issue on Learning from Imbalanced Data Sets," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 1–6, 2004, doi: 10.1145/1007730.1007733.

[20] J. Brownlee, *Imbalanced Classification with Python - Choose Better Metrics, Balance Skewed Classes, and Apply Cost-Sensitive Learning*. 2021.

[21] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M. S. Hacid, and H. Zeineddine, "An Experimental Study With Imbalanced Classification Approaches for Credit Card Fraud Detection," *IEEE Access*, vol. 7, pp. 93010–93022, 2019, doi: 10.1109/ACCESS.2019.2927266.

[22] N. Farhana Hordri, S. SophiayatiYuhaniz, N. Firdaus Mohd Azmi, and S. Mariyam Shamsuddin, "Handling Class Imbalance in Credit Card Fraud using Resampling Methods," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 11, 2018, [Online]. Available: www.ijacsa.thesai.org

[23] S. Fotouhi, S. Asadi, and M. W. Kattan, "A comprehensive data level analysis for cancer diagnosis on imbalanced data," *J Biomed Inform*, vol. 90, Feb. 2019, doi: 10.1016/j.jbi.2018.12.003.

[24] L. Sha, M. Rakovic, A. Das, D. Gasevic, and G. Chen, "Leveraging Class Balancing Techniques to Alleviate Algorithmic Bias for Predictive Tasks in Education," *IEEE Transactions on Learning Technologies*, 2022, doi: 10.1109/TLT.2022.3196278.

[25] S. D. A. Bujang *et al.*, "Multiclass Prediction Model for Student Grade Prediction Using Machine Learning," *IEEE Access*, vol. 9, pp. 95608–95621, 2021, doi: 10.1109/ACCESS.2021.3093563.

[26] A. Kulkarni, D. Chong, and F. A. Batarseh, "Foundations of data imbalance and solutions for a data democracy," in *Data Democracy: At the Nexus of Artificial Intelligence, Software Development, and Knowledge Engineering*, Elsevier, 2020, pp. 83–106. doi: 10.1016/B978-0-12-818366-3.00005-8.

[27] Y. Sun, M. S. Kamel, A. K. C. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern Recognit*, vol. 40, no. 12, pp. 3358–3378, Dec. 2007, doi: 10.1016/j.patcog.2007.04.009.

[28] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent Data Analysis*, vol. 6, pp. 429–449, 2002.

[29] G. H. Nguyen, A. Bouzerdoum, and S. L. Phung, "Learning Pattern Classification Tasks with Imbalanced Data," in *Pattern Recognition*, P.-Y. Yin, Ed., 2009, pp. 193–208. [Online]. Available: www.intechopen.com

[30] K. Andrić, D. Kalpić, and Z. Bohaček, "An insight into the effects of class imbalance and sampling on classification accuracy in credit risk assessment," *Computer Science and Information Systems*, vol. 16, no. 1, pp. 155–178, Jan. 2019, doi: 10.2298/CSIS180110037A.

[31] A. Ali, S. M. Shamsuddin, and A. L. Ralescu, "Classification with class imbalance problem: A review," *Int. J. Advance Soft Compu. Appl*, vol. 5, no. 3, 2013, [Online]. Available: https://www.researchgate.net/publication/288228469

[32] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from Imbalanced Data Sets*. doi: 10.1007/978-3-319-98074-4.

[33] A. Fernández, S. García, and F. Herrera, "Addressing the Classification with Imbalanced Data: Open Problems and New Challenges on Class Distribution," in *Hybrid Artificial Intelligent Systems. HAIS 2011. Lecture Notes in Computer Science()*, E. Corchado, M. Kurzyński, and M. Woźniak, Eds., Berlin, Heidelberg: Springer, 2011.

[34] Y. Sun, A. K. C. Wong, and M. S. Kamel, "Classification of Imbalanced Data: A Review," *Intern J Pattern RecognitArtifIntell*, vol. 23, no. 4, 2009, [Online]. Available: www.worldscientific.com

[35] A. Estabrooks and N. Japkowicz, "A Multiple Resampling Method for Learning From Imbalanced Data Sets," *ComputIntell*, vol. 20, no. 1, 2004.

[36] Y. Feng, M. Zhou, and X. Tong, "Imbalanced classification: A paradigm-based review," *Stat Anal Data Min*, vol. 14, no. 5, pp. 383–406, Oct. 2021, doi: 10.1002/sam.11538.

[37] L. Liu, X. Wu, S. Li, Y. Li, S. Tan, and Y. Bai, "Solving the class imbalance problem using ensemble algorithm: application of screening for aortic dissection," *BMC Med Inform Decis Mak*, vol. 22, no. 1, Dec. 2022, doi: 10.1186/s12911-022-01821-w.

[38] J. Wu, Z. Zhao, C. Sun, R. Yan, and X. Chen, "Learning from Class-imbalanced Data with a Model-Agnostic Framework for Machine Intelligent Diagnosis," *Reliab Eng Syst Saf*, vol. 216, Dec. 2021, doi: 10.1016/j.ress.2021.107934.

[39] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Systems with Applications*, vol. 73. Elsevier Ltd, pp. 220–239, May 01, 2017. doi: 10.1016/j.eswa.2016.12.035.

[40] J. A. Sáez, J. Luengo, J. Stefanowski, and F. Herrera, "SMOTE-IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering," *Inf Sci (N Y)*, vol. 291, no. C, pp. 184–203, 2015, doi: 10.1016/j.ins.2014.08.051.

[41] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning," in *Advances in Intelligent Computing. ICIC 2005. Lecture Notes in Computer Science*, D. Huang, X. Zhang, and G. Huang, Eds., 2005, pp. 878–887.

[42] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning," in *2008 International Joint Conference on Neural Networks (IJCNN 2008)*, 2008, pp. 1322–1328.

[43] P. Takaki, M. L. Dutra, G. de Araújo, and E. M. da S. Júnior, "A Proposed Framework for Evaluating the Academic-failure Prediction in Distance Learning," *Mobile Networks and Applications*, vol. 27, no. 5, pp. 1958–1966, Oct. 2022, doi: 10.1007/s11036-022-01965-z.

[44] S. Verma, R. Kumar Yadav, and K. Kholiya, "A Scalable Machine Learning-based Ensemble Approach to Enhance the Prediction Accuracy for Identifying Students at-Risk," *Int J Adv Comput Sci Appl*, vol. 13, no. 8, 2022, [Online]. Available: www.iijacsa.thesai.org

[45] N. Z. Salih and W. Khalaf, "Prediction of student's performance through educational data mining techniques," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 22, no. 3, pp. 1708–1715, Jun. 2021, doi: 10.11591/ijeecs.v22.i3.pp1708-1715.

[46] A. Nabil, M. Seyam, and A. Abou-Elfetouh, "Prediction of Students' Academic Performance Based on Courses' Grades Using Deep Neural Networks," *IEEE Access*, vol. 9, pp. 140731–140746, 2021, doi: 10.1109/ACCESS.2021.3119596.

[47] Y. K. Salal and S. M. Abdullaev, "Deep learning based ensemble approach to predict student academic performance: Case study," in *Proceedings of the 3rd International Conference on Intelligent Sustainable Systems, ICISS 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 191–198. doi: 10.1109/ICISS49785.2020.9316044.

[48] M. Utari, B. Warsito, and R. Kusumaningrum, "Implementation of Data Mining for Drop-Out Prediction using Random Forest Method," in *2020 8th International Conference on Information and Communication Technology, ICoICT*, 2020.

[49] W. Punlumjeak, S. Rugtanom, S. Jantarat, and N. Rachburee, "Improving classification of imbalanced student dataset using ensemble method of voting, bagging, and adaboost with under-

sampling technique," in *Lecture Notes in Electrical Engineering*, Springer Verlag, 2017, pp. 27–34. doi: 10.1007/978-981-10-6451-7_4.

[50] D. Thammasiri, D. Delen, P. Meesad, and N. Kasap, "A critical assessment of imbalanced class distribution problem: The case of predicting freshmen student attrition," *Expert Syst Appl*, vol. 41, no. 2, pp. 321–330, 2014, doi: 10.1016/j.eswa.2013.07.046.

[51] M. Koutina and K. L. Kermanidis, "Predicting Postgraduate Students' Performance Using Machine Learning Techniques," in *IFIP Advances in Information and Communication Technology*, 2011, pp. 159–168.

[52] N. Thai-Nghe, A. Busche, and L. Schmidt-Thieme, "Improving academic performance prediction by dealing with class imbalance," in *ISDA 2009 - 9th International Conference on Intelligent Systems Design and Applications*, 2009, pp. 878–883. doi: 10.1109/ISDA.2009.15.

[53] P. R. Pintrich, D. A. Smith, T. Garcia, and W. J. McKeachie, "A Manual for the Use of the Motivated Strategies for Learning Questionnaire (MSLQ)." 1991.

[54] S. Hamid and V. S. Singaram, "Motivated strategies for learning and their association with academic performance of a diverse group of 1st-year medical students," *Afr J Health Prof Educ*, vol. 8, no. 1, p. 104, Apr. 2016, doi: 10.7196/ajhpe.2016.v8i1.757.

[55] E. A. Amrieh, T. Hamtini, and I. Aljarah, "Student Academic Performance," *https://www.kaggle.com/c/student-academic-performance*, Nov. 08, 2016.

[56] E. A. Amrieh, T. Hamtini, and I. Aljarah, "Mining Educational Data to Predict Student's academic Performance using Ensemble Methods," *International Journal of Database Theory and Application*, vol. 9, no. 8, pp. 119–136, Aug. 2016, doi: 10.14257/ijdta.2016.9.8.13.

[57] M. Kuhn and K. Johnson, "Data pre-processing," in *Applied Predictive Modeling*, Springer New York, 2013, pp. 1–600. doi: 10.1007/978-1-4614-6849-3.

[58] Z. Ibrahim and D. Rusli, "Predicting Students' Academic Performance: Comparing Artificial Neural Network, Decision Tree and Linear Regression," in *Proceedings of the 21st Annual SAS Malaysia Forum*, 2007, pp. 1–6.

[59] M. E. Urrutia-Aguilar, R. Fuentes-García, V. D. Mirel Martínez, E. Beck, S. O. León, and R. Guevara-Guzmán, "Logistic Regression Model for the Academic Performance of First-Year Medical Students in the Biomedical Area," *Creat Educ*, vol. 07, no. 15, pp. 2202–2211, 2016, doi: 10.4236/ce.2016.715217.

[60] M. Awad and R. Khanna, "Support Vector Machines for Classification SVM from a Geometric Perspective," in *Efficient Learning Machines.*, Apress, Berkeley, CA, 2015.

[61] E. Osmanbegovi and M. Suljic, "Data Mining Approach for Predicting Student Performance," *Economic Review : Journal of Economics and Business*, vol. X, no. 1, pp. 3–12, 2012.

[62] G. Gray, C. Mcguinness, and P. Owende, "Non-Cognitive Factors of Learning as Early Indicators of Students At-Risk of Failing in Tertiary Education," in *Non-cognitive Skills and Factors in Educational Attainment*, Sense`Publishers, 2016, pp. 199–237.

[63] M. Grandini, E. Bagli, and G. Visani, "Metrics for Multi-Class Classification: an Overview," Aug. 2020. [Online]. Available: http://arxiv.org/abs/2008.05756

[64] S. Visa and A. Ralescu, "Issues in Mining Imbalanced Data Sets - A Review Paper," in *Mid West Artificial Intelligence and Cognitive Science Conference,* 2005, pp. 67–73. [Online]. Available: https://www.researchgate.net/publication/228386653

[65] R. Ghorbani and R. Ghousi, "Comparing Different Resampling Methods in Predicting Students' Performance Using Machine Learning Techniques," *IEEE Access*, vol. 8, pp. 67899–67911, 2020, doi: 10.1109/ACCESS.2020.2986809.

[66] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans Pattern Anal Mach Intell*, vol. 35, no. 8, pp. 1798–1828, 2013, doi: 10.1109/TPAMI.2013.50.