Institut Riset dan Publikasi Indonesia (IRPI)
MALCOM: Indonesian Journal of Machine Learning and Computer Science
Journal Homepage: https://journal.irpi.or.id/index.php/malcom
Vol. 3 Iss. 2 October 2023, pp: 188-198
ISSN(P): 2797-2313 | ISSN(E): 2775-8575

Determining the Final Project Topic Based on the Courses Taken by Using Machine Learning Techniques

Vicky Salsadilla[1*], Inggih Permana[2], Muhammad Jazman[3] , M.Afdal[4]

[1,2,3,4]Programa Studi Sistem Informasi, Fakultas Sains dan Teknologi,
Universitas Islam Negeri Sultan Syarif Kasim Riau, Indonesia

E-Mail: [1]11950321586@students.uin-suska.ac.id, inggihpermana@uin-suska.ac.id,
muhammadjazman@uin-suska.ac.id, m.afdal@uin-suska.ac.id

Received Aug 04th 2023; Revised Sept 25th 2023; Accepted Oct 10th 2023
Corresponding Author: Vicky Salsadilla

Abstract

A thesis (TA) is a scientific paper based on a problem. TA must be completed by students who wish to complete their studies. During this time, students often experience difficulties in determining the TA topic they want to research. To fix it, this research tries to determine TA topics using Machine Learning (ML) techniques based on the elective courses that students have taken. Elective courses are one form of academic data that can be used to consider TA topics. The ML algorithms used are KNN, NBC, ANN, SVM, C4.5, Random Forest, and Logistic Regression. The dataset used in this research is imbalanced data. This research balances the data using the Random Oversampling method and the Random Undersampling method. The results of experiments show that datasets balanced using ROS produce much higher ML performance, but tend to over-fit due to data duplication in the dataset. If the dataset is not balanced at all then the ML performance will be very low. Therefore, for unbalanced data, it is recommended to use the RUS method as data balance. The highest accuracy results for algorithms balanced using ROS are ANN=69.7%, RF=66.7%, SVM=57.6%, LR=57.6%, NBC=42.4%, C4.5=42.4%, and KNN=33.3%

Keywords: Machine Learning, Random Oversampling, Random Undersampling, Thesis

1. INTRODUCTION

As a student, completing a Thesis or Final Assignment (TA) is a crucial step towards finishing studies [1]. It's a form of scientific writing that requires me to thoroughly investigate an existing problem or phenomenon and test its validity using data that has been collected and processed. The aim is to produce reference material that can used in the future [2]. TA also includes research results in the field or based on literature studies [3]. By conducting research, it's hoped that students will be able to solve the problems by scientific, and can develop their insights [4].

Before preparing a TA, of course, students can pass the process of determining the topic or what they want to research [5]. The large amount of discussion and material that has been studied during lectures makes it difficult for students to determine how research topic they should take to make research into their thesis [6]. The Topic is an idea that underlies a TA. The topic is usually a benchmark for the discussion written by a writer [7]. Due to this phenomenon, some students felt they made the wrong choice of research topic when it went and ended up changing the research TA topics [8].

Apart from the lecture material that has been studied, they are usually also chosen according to their abilities [9], such as through analysis of academic data in the form of grades from study results during the lecture process fromthe beginning to the end semester [10]. As expected to help students determine appropriate TA topics. Along with that, students usually also choose TA topics through specialization in elective courses as a form of support in determining what they want to research [11]. By preffering the right topic, students can maximize the TA process and then complete the study on time [12].

Based on the previous explanation, this research uses machine learning (ML) to classify TA topics based on the elective courses that have been taken. It's hoped this classification can help students determine TA topics. There are 7 machine learning algorithms used, named (1) K-*Nearest Neighbor* (KNN); (2) *Naive Bayes Classifier* (NBC); (3) Artificial Neural Network (ANN); (4) *Support Vector Machines* (SVM); (5) C4.5; (6) Random Forest (RF); (7) Logistic Regression (LR). In this study, the KNN, NBC, SVM, and C4.5 algorithms were used, because these algorithms are included in the most frequently used algorithms [13] [14]. Meanwhile, the LR algorithm is used because this algorithm can calculate data probabilities; able to update linear models

with new data; can learn the data analysis process that can be applied in carrying out target classification; Apart from that, the LR results are not affected by small noise in the data [13]. Then, ANN is used because this algorithm can predict with very high accuracy [15]. Apart from that, other algorithms such as KNN, NBC, SVM, and C4.5 are also used because they have high accuracy which is following this research [15]. Last but not least, RF is used because it is a combination of several decision trees, each of which is created by a random subset and each node is selected from that random subset of features [14].

The aim of using machine learning in this research is so that future students will be able to learn from the data themselves. A lot of research has been done on how to understand machine learning without being explicitly programmed [16]. Whore in the dataset used, data imbalance or data imbalance occurs, which is one of the problems that can occur in ML [17]. This causes the resulting model to have poor performance [18]. The imbalance of ML towards majority class instances can be overcome by balancing using data-level techniques. This data-level technique aims to modify the dataset directly before ML reaches the measurement stage [19]. Because of this can balance the unequal class distribution. This process is divided into two categories, namely Random Oversampling and Random Undersampling which are applied in this research.

2. MATERIALS AND METHOD

In general, this research is divided into 4 phase, namely: (1) the data collection phase; (2) the data pre-processing phase; (3) the data balancing phase; and (4) the phase of implementing machine learning. These phase are shown in Figure 1.

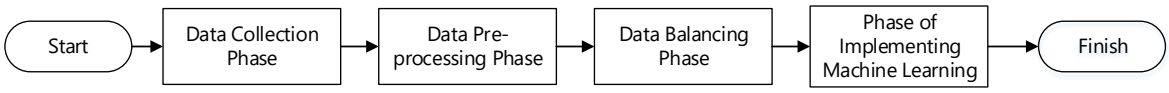


Figure 1 . Research Methodology

2.1 The Data Collection Stage

First of all, questionnaires are distributed via Google Forms. The respondents in this research were students of the Information Systems Study Major Class of 2019. The questions asked were: (1) the elective courses the students had taken; (2) TA topics taken; and (3) whether students feel they have taken the correct TA topic or not. For more details, see Table 1. The selected TA topics will then be used as classes in the dataset.

Table 1. A list of Question

No	Question	Information
1	What are the elective courses you have taken?	Answers are in the form of multiple choice: 1. Data Mining (DM), kode: A1; 2. Sistem Informasi Intelijen (SII), kode: A2; 3. Customer Relation Management (CRM), kode: A3; 4. Business Inteligence (BI), kode: A4; 5. Knowledge Management (KM), kode: A5; 6. E-business (E-biz), kode: A6; 7. IT Audit, kode: A7; 8. ERP M1, kode: A8; 9. ERP M2, kode: A9; 10. Geographic Information System (GIS), kode: A10.
2	What is the topic of your chosen thesis?	Answers are in the form of multiple choice: 1. Analisa Proses Bisnis (APB); 2. Evaluasi SI (ESI); 3. Data Mining (DM); 4. Customer Relation Management (CRM); 5. Rekayasa Perangkat Lunak (RPL); 6. Knowledge Management (KM); 7. Manajemen Risiko (MR).
3	Is your current thesis topic the right one?	Answers are in the form of multiple choice: 1. Yes 2. No

2.2 Data Pre-processing Phase

In data pre-processing, data selection and data transformation are carried out as follows:

1. Selection Data
   Based on the results of data collection, a dataset was obtained consisting of 70 rows of data. The dataset was selected by selecting the rows of data where the answer to question number 3 (see Table 1) was Yes so the remaining 64 rows of data.
2. Transformation Data
   At the data transformation stage, the shape of the dataset changed so that it looks like in Table 2. In this table, for columns A1 to A2, if the value is 1.0 then the student is taking the elective course that corresponds to the name of that column, otherwise, if it is 0.0 then the student does not take the course that corresponds to the column name.

Table 2. Data Transformation

No	Topik TA	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11
D1	APB	1,0	1,0	0,0	1,0	1,0	0,0	0,0	1,0	0,0	0,0	0,0
D2	APB	1,0	0,0	1,0	1,0	0,0	1,0	1,0	0,0	0,0	0,0	0,0
D3	APB	1,0	1,0	0,0	1,0	1,0	0,0	1,0	0,0	0,0	0,0	0,0
D4	APB	1,0	1,0	0,0	0,0	1,0	0,0	1,0	0,0	0,0	0,0	1,0
D5	APB	1,0	1,0	0,0	1,0	1,0	0,0	1,0	0,0	0,0	0,0	0,0
D6	KM	1,0	0,0	0,0	1,0	1,0	1,0	1,0	0,0	0,0	0,0	0,0
D7	KM	1,0	1,0	1,0	1,0	1,0	0,0	0,0	0,0	0,0	0,0	0,0
…	…	…	…	…	…	…	…	…	…	…	…	…
D64	MR	1,0	0,0	0,0	0,0	1,0	1,0	1,0	1,0	1,0	0,0	0,0

2.3 Data Balancing Phase

Data balancing is carried out which functions to balance the amount of data in each class. The balancing technique used is the Random Oversampling (ROS) technique and the Random Undersampling (RUS) technique. The ROS technique will create synthetic data from the minority class by randomizing the existing data. Meanwhile, the RUS technique will reduce the majority class by selecting random existing data [20]. The data balancing process is carried out using Orange Data Mining software. The results of data balancing can be seen in Table 3.

Table 3. The amount of data

TA Topics	Amount of Data		
	Without *Balancing*	ROS	RUS
APB	8	20	3
ESI	15	20	3
DM	20	20	3
CRM	10	20	3
RPL	3	20	3
KM	5	20	3
MR	3	20	3

2.4 Application of machine learning

This research uses 7 ML algorithms, namely: KNN, NBC, SVM, ANN, C4.5, RF, and LR. The parameters used for each algorithm can be seen in Table 4. Each combination of experimental parameters was carried out on 3 types of datasets, namely datasets that were balanced using ROS, datasets that were balanced using RUS, and datasets that did not use data balancing. Meanwhile, for performance measurement metrics, this research uses accuracy, precision, and recall. At this stage of implementing ML, Orange Data Mining Software is used.

Table 4. Algorithm Parameters

No	Algorithm	Information	
		Parameters	Mark / Number / Symbol
1	KNN	K	3, 5, 7, 9, 11
2	NBC	-	-
3	SVM	Kernels	Linear
		Kernels	Polynomial
		Gamma	Auto
		C = Cost ; D = Degree	[C=1,00 D=1,0]; [C=1,00 D=2,0]; [C=1,00 D=3,0]; [C=2,00 D=1,0]; [C=2,00 D=2,0]; [C=2,00 D=3,0]; [C=3,00 D=1,0]; [C=3,00 D=2,0]; [C=3,00 D=3,0]

No	Algorithm	Information	
		Parameters	Mark / Number / Symbol
		Kernels Gamma	*Radial Basis Function* (RBF) Auto
		Kernels C = Cost	Sigmoid [(c) = 1; (c) = 2; (c) = 3]
		Iterasi	1000
4	ANN	Hidden Layer	[100,100,100]; [100,100,200]; [100,100,300]; [100,200,100]; [100,200,200]; [100,200,300]; [100,300,100]; [100,300,200]; [100,300,300]; [200,100,100]; [200,100,200]; [200,100,300]; [200,200,100]; [200,200,200]; [200,200,300]; [200,300,100]; [200,300,200]; [200,300,300]; [300,100,100]; [300,100,200]; [300,100,300]; [300,200,100]; [300,200,200]; [300,200,300]; [300,300,100]; [300,300,200]; [300,300,300].
		Activation	ReLu
		Solver	Adam
		Learning Rate	0.0001
		Maximal Number of Iteration	1000
5	C4.5	Min Leaves	2, 3, 5, 7
6	RF	Min Trees	3, 5, 7, 9, 11
7	LR	Regularization type	Lasso (L1) Ridge (L2)
		C	7

3. RESULTS AND DISCUSSION

The overall results of the experiments carried out in this research can be seen in Table 5.

Table 5. Experiment Result

No	Algorithm	Parameters	ROS			RUS			Without Balancing		
			Acc	Prec	Recall	Acc	Prec	Recall	Acc	Prec	Recall
1	KNN	K=3	0.764	0.826	0.736	0.333	0.329	0.333	0.391	0.319	0.391
		K=5	0.757	0.824	0.757	0.242	0.202	0.242	0.422	0.291	0.422
		K=7	0.743	0.778	0.743	0.182	0.229	0.182	0.375	0.239	0.375
		K=9	0.571	0.630	0.671	0.182	0.159	0.182	0.375	0.218	0.375
		K=11	0.557	0.662	0.557	0.212	0.152	0.212	0.375	0.208	0.375
2	NBC	-	0.536	0.500	0.536	0.424	0.562	0.424	0.391	0.427	0.391
3	SVM	Kernels : Linear	0.750	0.772	0.750	0.576	0.605	0.576	0.375	0.208	0.375
		Kernels : Polynomial									
		C=1,00 D=1,0	0.700	0.701	0.700	0.455	0.448	0.455	0.391	0.276	0.391
		C=1,00 D=2,0	0.764	0.781	0.764	0.424	0.455	0.424	0.391	0.295	0.391
		C=1,00 D=3,0	0.750	0.765	0.750	0.455	0.457	0.455	0.375	0.278	0.375
		C=2,00 D=1,0	0.700	0.701	0.700	0.455	0.448	0.455	0.391	0.276	0.391
		C=2,00 D=2,0	0.736	0.759	0.736	0.424	0.439	0.424	0.406	0.308	0.406
		C=2,00 D=3,0	0.750	0.765	0.750	0.424	0.470	0.424	0.406	0.299	0.406
		C=3,00 D=1,0	0.700	0.701	0.700	0.455	0.448	0.455	0.391	0.276	0.391
		C=3,00 D=2,0	0.736	0.759	0.736	0.424	0.459	0.424	0.406	0.313	0.406
		C=3,00 D=3,0	0.750	0.767	0.750	0.424	0.470	0.424	0.406	0.306	0.406
		Kernels : RBF	0.793	0.808	0.793	0.576	0.516	0.576	0.375	0.261	0.375
		Kernels : Sigmoid									
		(c) = 1	0.321	0.345	0.321	0.273	0.152	0.273	0.266	0.137	0.266
		(c) = 2	0.257	0.219	0.257	0.273	0.074	0.273	0.312	0.098	0.312

No	Algorithm	Parameters	ROS			RUS			Without Balancing		
			Acc	Prec	Recall	Acc	Prec	Recall	Acc	Prec	Recall
		(c) = 3	0.264	0.231	0.264	0.273	0.074	0.273	0.312	0.098	0.312
4	ANN	Hidden Layer									
		[100,100,100]	0.779	0.791	0.779	0.576	0.600	0.576	0.391	0.327	0.391
		[100,100,200]	0.779	0.787	0.779	0.576	0.574	0.576	0.391	0.327	0.391
		[100,100,300]	0.779	0.786	0.779	0.606	0.590	0.606	0.344	0.264	0.344
		[100,200,100]	0.786	0.796	0.786	0.697	0.697	0.697	0.375	0.290	0.375
		[100,200,200]	0.786	0.803	0.786	0.697	0.697	0.697	0.391	0.299	0.391
		[100,200,300]	0.786	0.790	0.786	0.515	0.494	0.515	0.359	0.311	0.359
		[100,300,100]	0.793	0.804	0.793	0.697	0.712	0.697	0.375	0.283	0.375
		[100,300,200]	0.771	0.785	0.771	0.636	0.614	0.636	0.375	0.283	0.375
		[100,300,300]	0.779	0.794	0.779	0.606	0.606	0.606	0.375	0.283	0.375
		[200,100,100]	0.771	0.782	0.771	0.606	0.600	0.606	0.406	0.340	0.406
		[200,100,200]	0.793	0.802	0.793	0.697	0.682	0.697	0.391	0.322	0.391
		[200,100,300]	0.779	0.791	0.779	0.636	0.534	0.636	0.391	0.337	0.391
		[200,200,100]	0.771	0.786	0.771	0.606	0.615	0.606	0.359	0.291	0.359
		[200,200,200]	0.786	0.794	0.786	0.606	0.602	0.606	0.359	0.294	0.359
		[200,300,300]	0.786	0.795	0.786	0.606	0.621	0.606	0.359	0.311	0.359
		[200,300,100]	0.786	0.795	0.786	0.576	0.559	0.576	0.375	0.327	0.375
		[200,300,200]	0.779	0788	0.779	0.697	0.701	0.697	0.391	0.331	0.391
		[200,300,300]	0.786	0.795	0.786	0.576	0.590	0.576	0.359	0.311	0.359
		[300,100,100]	0.786	0.799	0.786	0.576	0.611	0.576	0.391	0.296	0.391
		[300,100,200]	0.793	0.800	0.793	0.697	0.732	0.697	0.359	0.276	0.359
		[300,100,300]	0.779	0.788	0.779	0.667	0.583	0.667	0.375	0.284	0.375
		[300,200,100]	0.786	0.803	0.786	0.576	0.615	0.576	0.406	0.307	0.406
		[300,200,200]	0.771	0.788	0.771	0.697	0.686	0.697	0.344	0.281	0.344
		[300,200,300]	0.779	0.787	0.779	0.667	0.577	0.667	0.344	0.274	0.344
		[300,300,100]	0.793	0.804	0.793	0.576	0.611	0.576	0.391	0.309	0.391
		[300,300,200]	0.771	0.784	0.771	0.576	0.636	0.576	0.391	0.312	0.391
		[300,300,300]	0.779	0.794	0.779	0.576	0.611	0.576	0.359	0.298	0.359
5	C4.5	Min Leaves									
		2	0.771	0.773	0.771	0.424	0.459	0.424	0.359	0.302	0.359
		3	0.764	0.772	0.764	0.394	0.350	0.394	0.406	0.306	0.406
		5	0.643	0.666	0.643	0.424	0.318	0.424	0.422	0.308	0.422
		7	0.514	0.494	0.514	0.212	0.148	0.212	0.422	0.315	0.422
6	RF	Min Trees									
		3	0.779	0.783	0.779	0.545	0.567	0.545	0.375	0.299	0.375
		5	0.771	0.781	0.771	0.667	0.695	0.667	0.359	0.286	0.359
		7	0.786	0.792	0.786	0.576	0.586	0.576	0.375	0.292	0.375
		9	0.771	0.777	0.771	0.606	0.676	0.606	0.344	0.285	0.344
		11	0.807	0.810	0.807	0.606	0.630	0.606	0.406	0.319	0.406
7	LR	Lasso (C7)	0.679	0.657	0.679	0.576	0.574	0.576	0.406	0.333	0.406
		Ridge (C7)	0.757	0.765	0.757	0.576	0.576	0.576	0.406	0.336	0.406
		None	0.736	0.743	0.736	0.545	0.623	0.545	0.375	0.333	0.375

In Table 5, it can be seen that the best performance for KNN+ROS is when the K value = 3, with an accuracy value = 76.4%, a precision value = 82.6%, and a recall value = 73.6%. Meanwhile, the best performance for KNN+RUS is when K = 3, with an accuracy value = 33.3%, a precision value = 32.9%, and a recall value = 33.3%. Then the best performance for KNN without balancing is when K = 5, namely accuracy value = 42.2%, precision value = 29.1%, and recall value = 42.2%. A comparison of the performance of KNN+ROS, KNN+RUS, and KNN without balancing can be seen in Figure 2. In this figure, it can be seen that the performance of KNN+ROS is better than KNN+RUS and KNN without balancing, both in terms of accuracy, precision, and recall.

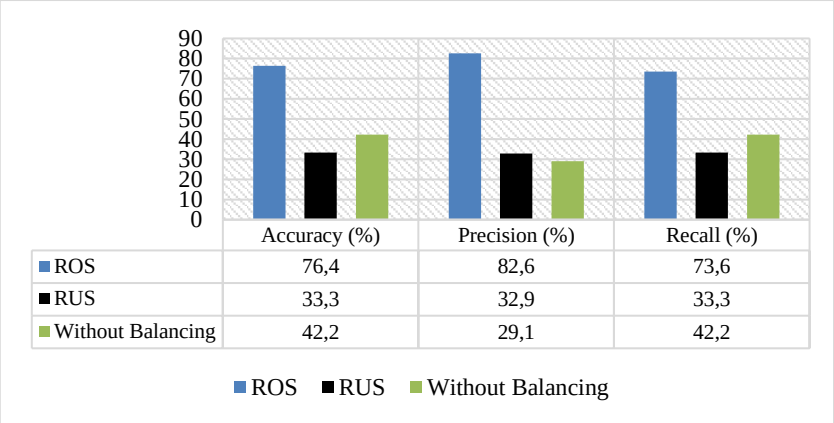


Figure 2. KNN Performance comparison

In the NBC algorithm, as seen in Table 5, NBC+ROS gets an accuracy value = 53.6%, a precision value = 50.0%, and a recall value = 53.6%. Meanwhile, NBC+RUS obtained an accuracy value = 42.4%, a precision value = 56.2%, and a recall value = 42.4%. Then for NBC without balancing, the accuracy value = 39.1%, precision value = 42.7%, and recall value = 39.1%. A comparison of the performance of NBC+ROS, NBC+RUS, and NBC without balancing can be seen in Figure 3. In this figure, it can be seen that NBC+ROS is better than NBC+RUS and NBC without balancing, both in terms of accuracy, precision, and recall.

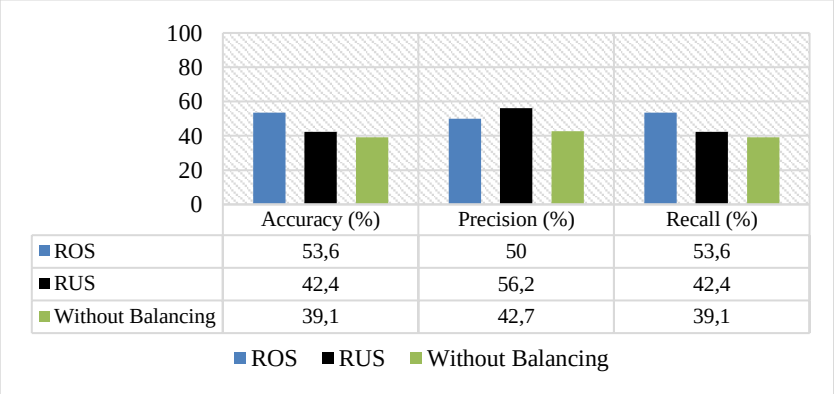


Figure 3. NBC Performance comparison

In the SVM algorithm, as seen in Table 5, the best performance for SVM+ROS is when using the RBF kernel, with an accuracy value = 79.3%, a precision value = 80.8%, and a recall value = 79.3%. Meanwhile, the best performance for SVM+RUS is when using the Linear kernel, with an accuracy value = 57.6%, a precision value = 60.5%, and a recall value = 57.6%. Then the best performance for SVM without balancing is when using the Polynomial kernel (g = auto, c = 3.00 and d = 2.0) with an accuracy value = 40.6%, a precision value = 31.3%, and a recall value = 40.6%. A comparison of the performance of SVM+ROS, SVM+RUS, and SVM without balancing can be seen in Figure 4. In this figure, it can be seen that the performance of SVM+ROS is better than SVM+RUS and SVM without balancing, both in terms of accuracy, precision, and recall.

In Table 5, it can be seen that the best ANN+ROS performance is when using a hidden layer structure = [300, 300, 100], with an accuracy value = 79.3%, a precision value = 80.4%, and a recall value = 79.3%. Meanwhile, the best performance for ANN+RUS is when using a hidden layer structure = [300, 100, 200], with an accuracy value = 69.7%, a precision value = 73.2%, and a recall value = 69.7%. Then the best performance for ANN without balancing is when using a hidden layer structure = [200,100,100] with an accuracy value = 40.6%, a precision value = 34.0%, and a recall value = 40.6%. A comparison of the performance of ANN+ROS, ANN+RUS, and ANN without balancing can be seen in Figure 5. In this figure, it can be seen that the performance of ANN+ROS is better than ANN+RUS and ANN without balancing, both in terms of accuracy, precision, and recall.

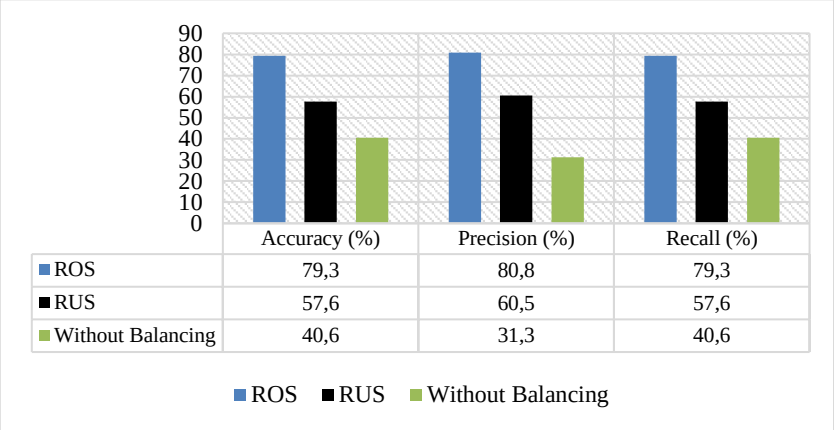


Figure 4. SVM Performance comparison

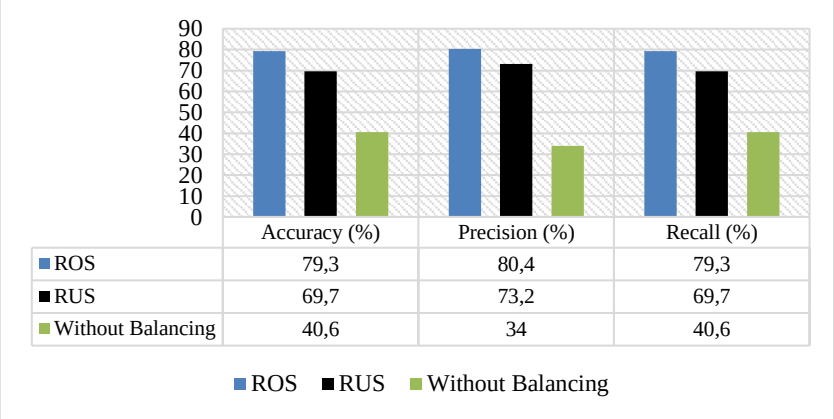


Figure 5. ANN Performance comparison

In the C4.5 Algorithm, as seen in Table 5, the best performance of C4.5+ROS was obtained when using a minimum number of leaves = 2, with accuracy = 77.1%, precision = 77.3%, and recall = 77.1%. Meanwhile, the best performance of C.45+RUS is also using a minimum number of leaves = 2, namely accuracy = 42.4%, precision = 45.9%, and recall = 42.4%. Then the best performance of C4.5 without balancing is when using a minimum number of leaves = 7, with accuracy = 42.2%, precision = 31.5%, and recall = 42.2%. A comparison of the performance of C4.5+ROS, C4.5+RUS, and C.45 without balancing can be seen in Figure 6. In this figure, it can be seen that the performance of C4.5+ROS is better than C4.5+RUS and C4.5 without balancing, both in terms of accuracy, precision, and recall.

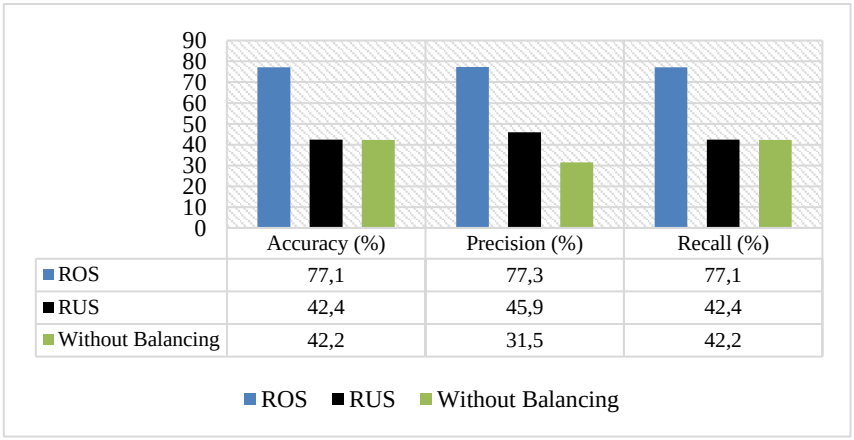
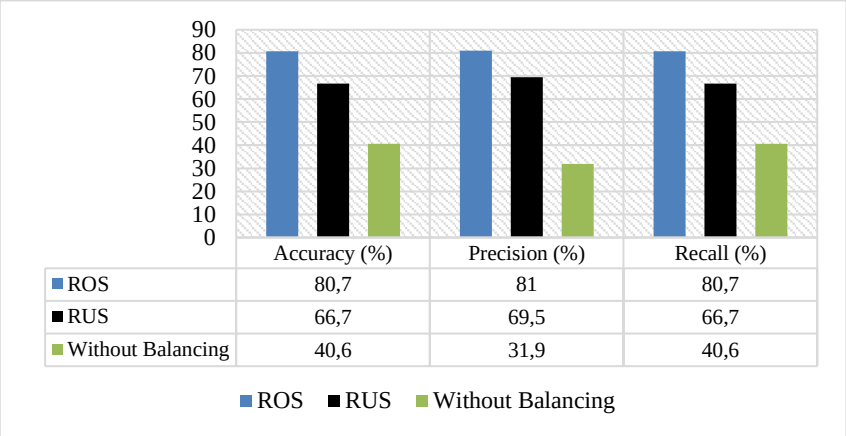


Figure 6. C4.5 Performance comparison

Then in the RF Algorithm, as seen in Table 5, the best RF+ROS performance is when using the number of attributes considered in each separation value = 11, with an accuracy value = 80.7%, a precision value =

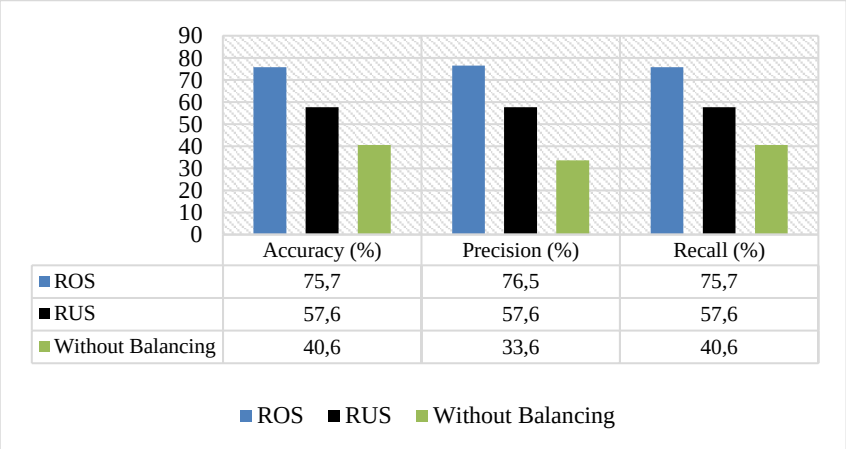
81.0%, and a recall value = 80.7%. Meanwhile, the best RF+RUS performance is when using the number of attributes considered for each separation with a value of = 5, with an accuracy value = 66.7%, a precision value = 69.5%, and a recall value = 66.7%. Then the best RF performance without balancing is when using the number of attributes considered in each separation value = 11, with an accuracy value = 40.6%, a precision value = 31.9%, and a recall value = 40.6%. A comparison of RF+ROS, RF+RUS, and RF without balancing can be seen in Figure 7. In this figure, it can be seen that the performance of RF+ROS is better than RF+RUS and RF without balancing, both in terms of accuracy, precision, and recall.



	Accuracy (%)	Precision (%)	Recall (%)
ROS	80,7	81	80,7
RUS	66,7	69,5	66,7
Without Balancing	40,6	31,9	40,6

Figure 7. RF Performance comparison

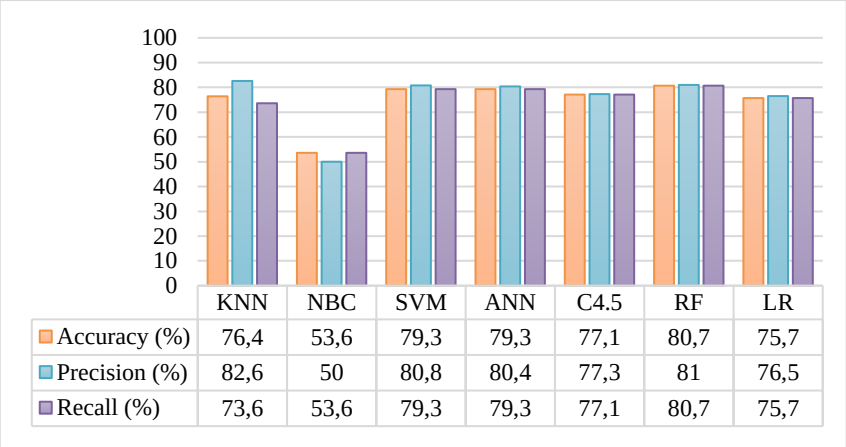
In the LR algorithm, as seen in Table 5, the best performance for LR+ROS is when using the Ridge kernel [C = 7], with an accuracy value = 75.7%, a precision value = 76.5%, and a recall value = 75.7%. Meanwhile, the best performance for LR+RUS is also when using the Ridge kernel [C=7], with an accuracy value = 57.6%, a precision value = 57.5%, and a recall value = 57.6%. Then the best performance for LR without balancing is also when using the Ridge kernel [C=7], with an accuracy value = 40.6%, a precision value = 33.6%, and a recall value = 40.6%. A comparison of LR+ROS, LR+RUS, and LR without balancing can be seen in Figure 8. In this figure, it can be seen that the performance of LR+ROS is better than LR+RUS and LR without balancing, both in terms of accuracy, precision, and recall.



	Accuracy (%)	Precision (%)	Recall (%)
ROS	75,7	76,5	75,7
RUS	57,6	57,6	57,6
Without Balancing	40,6	33,6	40,6

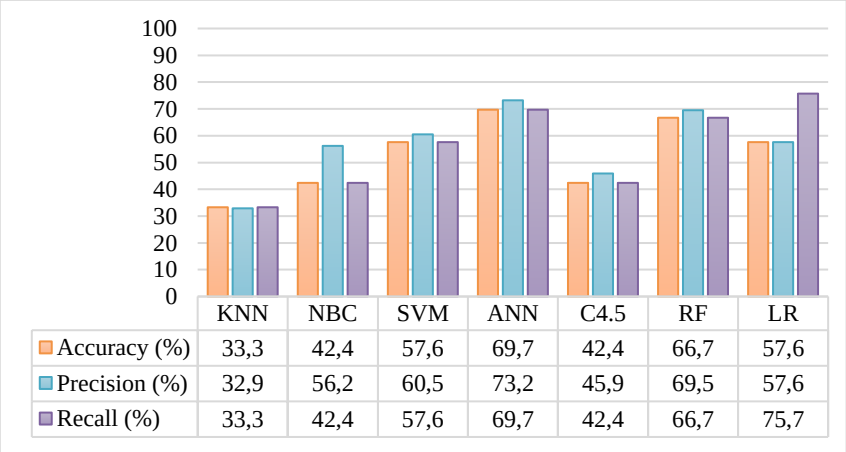
Figure 8. LR Performance comparison

Figure 9 is a comparison of the performance of ML algorithms on a dataset where ROS is applied as a balancing technique. It can be seen that the highest accuracy was obtained by the RF Algorithm, namely 80.7%. Meanwhile, the algorithm with the highest precision is the KNN algorithm, namely 82.6%. Even so, the resulting precision is not that far from RF which has a precision of 81.0%. The difference in KNN precision is only 1.6% higher when compared to RF. Meanwhile, the highest recall was obtained by the RF algorithm, namely 80.7%. With the existing results of accuracy, precision, and recall, it can be concluded that in datasets balanced using ROS, the ML algorithm that produces the best performance is the RF algorithm.



	KNN	NBC	SVM	ANN	C4.5	RF	LR
Accuracy (%)	76,4	53,6	79,3	79,3	77,1	80,7	75,7
Precision (%)	82,6	50	80,8	80,4	77,3	81	76,5
Recall (%)	73,6	53,6	79,3	79,3	77,1	80,7	75,7

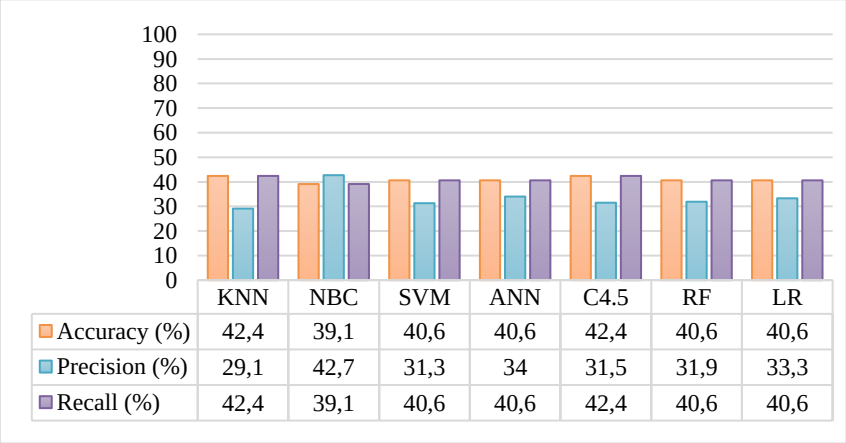
Figure 9. ML+ROS Performance comparison



	KNN	NBC	SVM	ANN	C4.5	RF	LR
Accuracy (%)	33,3	42,4	57,6	69,7	42,4	66,7	57,6
Precision (%)	32,9	56,2	60,5	73,2	45,9	69,5	57,6
Recall (%)	33,3	42,4	57,6	69,7	42,4	66,7	75,7

Figure 10. ML+RUS Performance comparison

Figure 10 is a comparison of the performance of ML algorithms on the dataset where RUS is applied as a data balancing technique. It can be seen that the highest accuracy was obtained by the ANN algorithm, namely 69.7%. The algorithm with the highest precision is also the ANN algorithm, namely 73.2%. Meanwhile, the highest recall was obtained by the LR algorithm, namely 75.7%. So it can be concluded that the best performance for accuracy and precision on datasets balanced using RUS is the ANN algorithm, while the best performance for recall on datasets balanced using RUS is the LR algorithm.



	KNN	NBC	SVM	ANN	C4.5	RF	LR
Accuracy (%)	42,4	39,1	40,6	40,6	42,4	40,6	40,6
Precision (%)	29,1	42,7	31,3	34	31,5	31,9	33,3
Recall (%)	42,4	39,1	40,6	40,6	42,4	40,6	40,6

Figure 11. ML+Without Balancing Performance comparison

Figure 11 is a comparison of the performance of ML algorithms whose datasets are not balanced. The highest accuracy value obtained was only 42.4%, namely the KNN and C4.5 algorithms. The highest precision value produced was only 42.7%, namely the NBC algorithm. The highest recall value produced was only 42.4%, namely the KNN and C4.5 algorithms. It can be concluded that for unbalanced datasets the performance produced by ML algorithms is very low.

Based on the experiments that have been carried out as in Table 5, it can be seen that the dataset balanced using ROS produces much better ML performance in terms of accuracy, precision, and recall. However, what is important to note is that a dataset that is balanced using the ROS method will produce ML training results that tend to be overfitting. On the other hand, if the dataset is not balanced at all then the ML performance will be very low. Therefore, based on the experiments that have been carried out, for imbalanced data it is recommended to balance it using the RUS method.

4. CONCLUSION

Based on the results of the experiments carried out, ML can be used to create a model for determining TA topics if the dataset used is balanced. This is proven by experiments carried out on datasets without balancing and datasets that are balanced. A dataset without balancing produces very low performance, whereas when the dataset is balanced the performance is much better. The data balancing method that gets the highest performance is ROS, but this method has a large risk of overfitting. Therefore, this research suggests using RUS as a data balancing method, even though the resulting performance is not as high as ROS.

REFERENCES

[1] A. Homaidi, "Perancangan dan implementasi E-Thesis untuk tugas akhir mahasiswa Universitas Ibrahimy Situbondo," *NJCA (Nusantara J. Comput. Its Appl.*, vol. 4, no. 1, pp. 15–26, 2019, doi: 10.36564/njca.v4i1.109.

[2] A. C. Siregar, "Pelatihan penulisan tugas akhir dengan menggunakan LaTeX bagi mahasiswa teknik informatika Universitas Muhammadiyah Pontianak," *J. Bul. Al-Ribaath*, vol. 18, no. 1, pp. 40–48, 2021, doi: 10.29406/br.v18i1.2555.

[3] M. R. Baharuddin, "Adaptasi Kurikulum Merdeka Belajar Kampus Merdeka (Fokus: Model MBKM Program Studi)," *J. Stud. Guru dan Pembelajaran*, vol. 4, no. 1, pp. 195–205, Apr. 2021, doi: 10.30605/jsgp.4.1.2021.591.

[4] B. Ahmad and M. S. Laha, "Penerapan studi lapangan dalam meningkatkan kemampuan analisis masalah (Studi Kasus pada mahasiswa Sosiologi IISIP YAPIS BIAK)," *J. NALAR Pendidik.*, vol. 8, no. 1, p. 63, Jun. 2020, doi: 10.26858/jnp.v8i1.13644.

[5] A. Salipolo, "Analisis kesulitan mahasiswa Pendidikan Matematika IAIN Palopo dalam menyusun skripsi selama Pandemi COVID-19," 2022.

[6] R. A. Kristian and I. Wahyuni, "Penentuan topik judul Tugas Akhir mahasiswa di STMIK Asia Malang menggunakan Fuzzy Inference System Tsukamoto," *J. Ilm. Teknol. Inf. Asia*, vol. 12, no. 01, pp. 33–47, 2018, doi: 10.32815/jitika.v12i1.223.

[7] A. Triawan and M. Della Lintang, "Penerapan Metode Naïve Bayes Untuk Rekomendasi Topik Tugas Akhir Berdasarkan Daftar Hasil Studi Mahasiswa di Perguruan Tinggi," *Teknois J. Ilm. Teknol. Inf. dan Sains*, vol. 10, no. 2, pp. 58–70, 2020, doi: 10.36350/jbs.v10i2.91.

[8] A. D. Adhi Putra and S. Juanita, "Analisis sentimen pada ulasan pengguna aplikasi Bibit dan Bareksa dengan algoritma KNN," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 8, no. 2, pp. 636–646, 2021, doi: 10.35957/jatisi.v8i2.962.

[9] A. B. Saputro, "Penerapan Machine Learning untuk mengidentifikasi faktor-faktor yang mempengaruhi kemampuan komunikasi matematis pada materi Program Linear," Universitas Islam Negeri Syarif Hidayatullah Jakarta, 2023.

[10] E. Irwandi, "Pengembangan sistem informasi pengelolaan Tugas Akhir Program Studi Sistem Informasi Uin Suska Riau," Universitas Islam Negeri Sultan Syarif Kasim Riau, 2020.

[11] A. D. T. Utomo, T. Andriyanto, and A. Ristyawan, "Implementasi metode Electre untuk menentukan topik skripsi," *Semin. Nas. Inov. Teknol. UN PGRI*, vol. 4, no. 3, pp. 23–30, 2020, doi: 10.29407/inotek.v4i3.27.

[12] H. A. Hermawan, "Identifikasi hambatan penyelesaian studi bagi mahasiswa PGSD PENJAS," *Jambura Heal. Sport J.*, vol. 4, no. 2, pp. 78–88, 2022, doi: 10.37311/jhsj.v4i2.15630.

[13] S. Ray, "A quick review of Machine Learning Algorithms," *Int. Conf. Mach. Learn. Big Data, Cloud Parallel Comput. (COM-IT-Con), India*, vol. 3, no. 2, pp. 35–39, 2019, doi: 10.1109/COMITCon.2019.8862451.

[14] P. P. Shinde and D. S. Shah, "A review of Machine Learning and Deep Learning Applications," in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 2019, pp. 1–6, doi: 10.1109/ICCUBEA.2018.8697857.

[15] R. Ghorbani and R. Ghousi, "Comparing different resampling methods in predicting students' performance using Machine Learning Techniques," in *IEEE Access*, 2020, vol. 8, pp. 67899–67911, doi: 10.1109/ACCESS.2020.2986809.

[16] B. Mahesh, "Machine learning algorithms - A review," *Int. J. Sci. Res.*, vol. 2, no. January 2019, pp. 1–6, 2020, doi: 10.21275/ART20203995.

[17] A. Syukron and A. Subekti, "Penerapan metode Random Over-Under Sampling dan Random Forest untuk klasifikasi penilaian kredit," *J. Inform.*, vol. 5, no. 2, pp. 175–185, 2018, doi: 10.31294/ji.v5i2.4158.

[18] S. Y. Bae, J. Lee, J. Jeong, C. Lim, and J. Choi, "Effective data-balancing methods for class-imbalanced genotoxicity datasets using machine learning algorithms and molecular fingerprints," *Comput. Toxicol.*, vol. 20, no. June, pp. 1–6, 2021, doi: 10.1016/j.comtox.2021.100178.

[19] N. Rodríguez, D. López, A. Fernández, S. García, and F. Herrera, "SOUL: Scala Oversampling and Undersampling library for imbalance classification," *SoftwareX*, vol. 15, no. July, pp. 1–8, 2021, doi: 10.1016/j.softx.2021.100767.

[20] S. Mutmainah, "Penanganan imbalance data pada klasifikasi kemungkinan penyakit Stroke," *SNATi*, vol. 1, no. 1, pp. 10–16, 2021, [Online]. Available: https://journal.uii.ac.id/jurnalsnati/article/view/20060.