Implementation Of The Support Vector Machine Method In Predicting Student Graduation

Syakira Abdillah[1], Gomal Juni Yanris[2*], Volvo Sihombing[3]

[1,2,3] Faculty of Science and Technology, Universitas Labuhanbatu**,** Sumatera Utara Indonesia.
[*]Corresponding Author:
Email: pasaribusyakira@gmail.com

Abstract.

Student graduation is an important indicator of the quality of education at a higher education institution. A high graduation rate not only indicates success in the learning process but also has a positive impact on the reputation of the institution. Conversely, a low graduation rate can be a signal of problems that require special attention. Various factors, both academic and non-academic, influence higher education institutions in ensuring timely student graduation. Therefore, we need a method that can accurately predict student graduation to carry out early intervention. This study aims to apply the support vector machine method in predicting student graduation. We chose this method due to its capacity to classify complex data. We use historical student data, such as Semester Achievement Index scores, as input variables to build a prediction model. We evaluate the model using precision, recall, and f1-score metrics. According to the study's findings, the support vector machine model's accuracy level is 71.20%. This method is good at predicting students who graduate with a precision of 95%, recall of 72%, and f1-score of 82%. However, the model's performance in predicting students who failed was less than satisfactory, with a precision of only 17%, a recall of 62%, and an f1-score of 26%. The imbalance in data between passed and failed students contributed to this result. The Support Vector Machine method effectively predicts student graduation for the majority class (passed), but requires special handling of the data imbalance to enhance the accuracy of predictions for the minority class (failed). Universities expect to use the results of this study to carry out early intervention and increase student graduation rates.

Keywords: *Confusion Matrix, Graduation, Prediction, Student,and SVM.*

I. INTRODUCTION

One of the important components in the accreditation process of a university by BAN-PT is student graduation [1]. Student graduation is one of the important indicators in evaluating the success of the education process in universities. A high graduation rate reflects the effectiveness of the curriculum, the quality of teaching, and the readiness of students to face academic challenges. On the other hand, a low graduation rate may indicate issues within the education system that require immediate attention. Student graduation is a benchmark for the success of universities in providing quality education [2]. A high graduation rate can also improve a university's reputation and competitiveness in the eyes of the public and the industrial world [3]. However, a low graduation rate can raise doubts about the quality of education provided by the university. Universities should make every effort to reduce low graduation rates. This can be done by predicting graduation rates [4]. For educational institutions, predicting student graduation is critical. By having the ability to predict student graduation, institutions can take strategic steps to increase graduation rates and ensure the academic success of their students [5]. Institutions can identify students at risk of not graduating on time through graduation prediction, providing them with appropriate support and intervention [6]. Furthermore, graduation prediction can assist institutions in resource planning, such as determining the number of lecturers, classrooms, and other facilities required to support the learning process.Researchers and practitioners in the field of education have widely used machine learning methods to predict student graduation [7].

Researchers and practitioners in the field of education have applied various machine learning algorithms to predict student graduation with a fairly good level of accuracy [8], [9], [10], each with its own advantages and disadvantages.The composition of training data influences the accuracy of the K-Nearest Neighbor (KNN) algorithm, as a balanced composition distributes the training data evenly across each class

of test data [11]. Based on the results of research [2], it was found that the Naïve Bayes method can make a prediction regarding student graduation on time by taking into account the attributes of the college database used. Research by [12] suggests that the neural network algorithm is the most effective method for predicting student graduation. Based on research [13], the Random Forest algorithm tends to be more balanced in making predictions and is considered a "fair classification" [14]. Research by [15] indicates that the Support Vector Machine (SVM) algorithm can effectively manage the overfitting issue that arises when the model becomes excessively complex. SVM also performs well on high-dimensional data, which may be necessary when there are a large number of attributes or variables in student data.This study will apply the SVM algorithm to predict student graduation. Several previous studies have applied the SVM algorithm to predict student graduation. Based on the results of the study obtained by [16], the SVM algorithm is still better than the Decision Tree and Naïve Bayes algorithms with an accuracy of 83.64%. According to the results of the study conducted by [17], the SVM algorithm is superior in terms of accuracy compared to the Naïve Bayes algorithm with an accuracy score of 69.15%.

Based on the study's findings using data applied to three algorithms, the SVM algorithm achieved the highest accuracy level of 90.55% compared to the decision tree and artificial neural network algorithms [18]. In the study by [19], the SVM algorithm yielded a lower error rate of 16.84%, while the logistic regression algorithm produced an error rate of 19.3%. In the study by [20], the linear SVM algorithm yielded an accuracy value of 90%. In [21]'s research, the SVM algorithm outperformed the other two algorithms, KNN and Decision Tree, with a 95% accuracy rate in predicting students who can graduate on time.This study aims to apply the SVM algorithm to predicting student graduation. The main objective of this study is to develop a student graduation prediction model using the SVM algorithm. We will evaluate the SVM prediction model's performance using performance evaluation metrics like accuracy, precision, recall, and f1-score. We expect the results of this study to provide deeper insight into the key factors that influence student graduation. Universities can also use the resulting prediction model as a tool for decision-making. Universities, for example, can provide additional guidance, academic consultation, or other appropriate interventions to help students who have the potential to fail to graduate on time by identifying them. Thus, the application of the SVM method in predicting student graduation has enormous potential to increase the overall graduation rate, improve the quality of education, and improve the reputation of educational institutions in the eyes of the public.

II. METHODS

This study uses a quantitative approach with the Support Vector Machine (SVM) method to predict student graduation. The secondary data used in this study were obtained from Kaggle [22]. Figure 1 shows the stages of the research carried out.

Fig 1. Research Stage

This study is a collection of cumulative grade point average (gpa) data from semester 1 to semester 4. The data will undergo a preprocessing stage, which includes handling missing values and checking for data duplication, before the prediction process begins. Next, we will separate the data into two datasets: 70% for training and 30% for testing. This study employs a linear kernel function for SVM. Choosing the right kernel function will affect the performance of the SVM prediction. Once we obtain the test results, we proceed to evaluate the model by measuring performance metrics like accuracy, precision, recall, and f1-score.

III. RESULT AND DISCUSSION

This section presents the results of applying the SVM algorithm to predict student graduation. We systematically describe the study's results based on the research steps explained in the previous methodology section. This study uses the Python programming language, with the help of the Google Colab text editor.

	gps1	gps2	gps3	gps4	graduation
0	2.30	1.97	1.80	1.56	No
1	1.81	1.68	1.57	1.86	No
2	3.07	3.00	2.75	3.21	No
3	2.71	2.33	2.61	1.98	No
4	3.17	3.02	3.28	2.96	No
...	...	...	...	...	...
1682	3.07	3.04	3.39	3.55	Yes
1683	3.29	3.22	3.33	3.68	Yes
1684	3.31	3.25	3.44	3.52	Yes
1685	3.44	3.35	3.50	3.50	Yes
1686	3.18	3.05	3.05	3.27	Yes

1687 rows × 5 columns

Fig 2. Graduation dataset

Figure 2 shows part of the dataset used in this study. This study's dataset consists of 1687 rows, representing data from 1687 students. There are 5 columns in this dataset, namely 4 columns of IPS scores per semester and 1 column of graduation status. Columns gps1 to gps4 represent the values of students' grade point semester (GPS) from the first semester to the fourth semester. The graduation column represents the student's graduation status, with "Yes" meaning graduated and "No" meaning failed. The first row displays a student who received consecutive gps scores of 2.30, 1.97, 1.80, and 1.56 in semesters 1 to 4, resulting in a declaration of failure (graduation = no). The final row displays a student who achieved consecutive gps scores of 3.18, 3.05, 3.05, and 3.27, indicating graduation (graduation = Yes). This dataset serves as a database for training and testing the SVM model to predict student graduation based on their gps scores in the first four semesters.

	Total of missing	Percentage of missing
gps1	0	0.0%
gps2	0	0.0%
gps3	0	0.0%
gps4	0	0.0%
graduation	0	0.0%

Fig 3. The sum and percentage of missing values

Figure 3 The figure shows the results of handling missing values in the dataset used. The figure provides information about the amount and percentage of missing data in each column. The “Total of missing” column shows the amount of missing data in each column. According to the figure, all columns have 0 missing values, indicating that there is no missing data in these columns. The “Percentage of missing” column shows the percentage of missing data in each column. With the number of missing values all being 0, the percentage of missing data for each column is 0.0%. Based on the results of handling missing values displayed, all columns in the dataset have no missing data. Each row in the dataset contains complete values for each column, indicating that no additional action is required to handle missing values. The absence of

missing values is an indication that the dataset has good quality in terms of data completeness. This is crucial to prevent the lack of data from affecting the analysis and prediction model that the SVM will develop. Other data preprocessing steps can concentrate on cleaning duplicate data in the absence of missing values. Overall, these results indicate that the dataset is ready for further analysis and model building without the need for special handling of missing values.

```
# Check for duplicate rows in the dataset
duplicate = dataset.duplicated()

# Filter out and display the duplicate rows
duplicate_row = dataset[duplicate]
print("Duplicate rows:")
duplicate_row = pd.DataFrame(duplicate_row)
duplicate_row
```

```
Duplicate rows:
   gps1  gps2  gps3  gps4  graduation
```

Fig 4. Dataset Duplication Check

As part of data preprocessing, Figure 4 shows the steps for checking and displaying duplicate rows in the dataset. This process is important to ensure that the data used does not contain repeated entries, which can affect the accuracy of the prediction model. Duplication of data can cause the model to be biased and affect the prediction results. Therefore, identifying and removing duplicate rows is a crucial step in data preprocessing. This process is part of an effort to ensure that the dataset used to train the SVM model is free from repeated data entries. The end result is cleaner data that is ready for analysis, as well as more accurate model building. The data duplication check results indicate the absence of duplicate rows.

```
[ ] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)

[ ] print('x_train :',X_train.shape)
    print('x_test :',X_test.shape)
    print('y_train :',y_train.shape)
    print('y_test :',y_test.shape)

    x_train : (1180, 4)
    x_test : (507, 4)
    y_train : (1180,)
    y_test : (507,)
```

Fig 5. Data split

Figure 5 shows the process of separating the dataset into training data and test data, which is an important part of the implementation stage of the support vector machine algorithm in predicting student graduation. This step guarantees accurate evaluation and prevents overfitting of the built model. We divide the dataset into two parts: training data and test data, using the train_test_split function from the scikit-learn library. X represents independent features (GPS scores in various semesters), and y represents the label or target (graduation status). The test_size=0.3 parameter specifies that we will use 30% of the total data as test data and the remaining 70% as training data. We use Random_state=0 to ensure reproducible data separation, guaranteeing identical separation results if we run this code again. According to the division results, there are 1180 samples in the training data and 507 samples in the test data. This is in accordance with the 70%–30% division specified by the `test_size` parameter. We need this separation to train the model using training data and evaluate it independently using test data. This aids in assessing the model's generalization ability, ensuring that it does not overfit to the training data. Overall, this step is a critical part of the machine learning pipeline. Proper separation enhances the reliability of performance evaluation for models like support vector machines, as it evaluates the model on data it never encountered during training.

```
[ ] klasifier = svm.SVC(kernel='linear', cache_size=1000, class_weight='balanced')

[ ] klasifierfit = OneVsOneClassifier(klasifier).fit(X_train, y_train)
```

Fig 6. SVM Modeling

Figure 6 shows the SVM model's implementation in predicting student graduation. This process includes two main steps, namely, defining the SVM model and training it using training data. svm.SVC is a function in the scikit-learn library that defines the SVM model. The kernel='linear' parameter determines the type of kernel used in the SVM. When a linear hyperplane can effectively separate the data, we use the linear kernel. A linear kernel selection is generally simple and suitable for cases where the number of features is greater than or equal to the number of samples. This process provides 1000 MB of memory for temporary storage during model training.

This can speed up the training process, especially on large datasets. Class_weight='balanced' functions to automatically adjust class weights based on class frequencies. If the number of graduates is much higher or lower than the number of non-graduates, this is crucial. By balancing the weights, the model can handle class imbalances better. The OneVsOneClassifier learning strategy builds a classifier for each pair of classes in a multi-class case. Despite the graduation problem containing only two classes (pass and fail), this approach serves to validate the results. This method trains the model using training data (X_train for features and y_train for labels). The training process involves finding the best hyperplane that separates the classes with the largest margin. Overall, this process shows a satisfactory implementation of the SVM method in an attempt to predict student graduation, with attention to handling imbalanced classes and choosing the appropriate kernel.

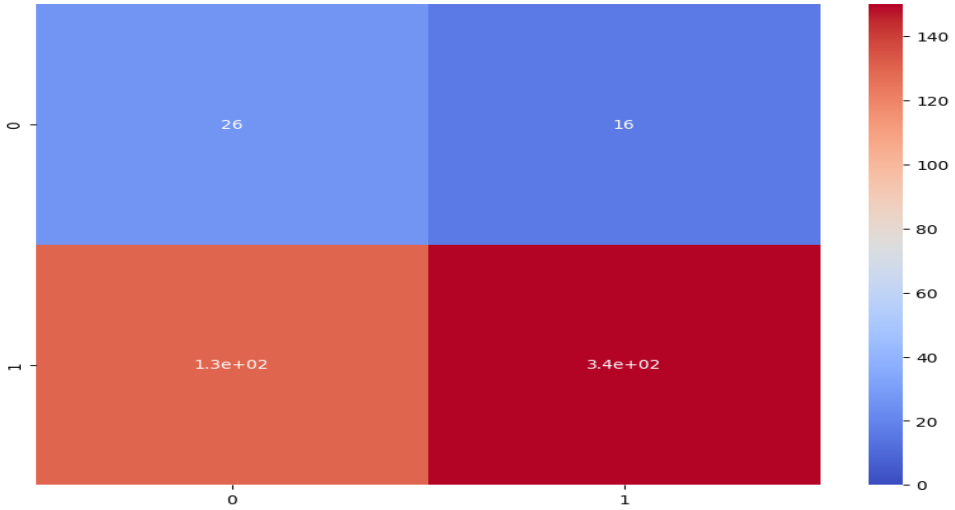

Fig 7. Confusion matrix

We use Figure 7 as a confusion matrix to assess how well the SVM model predicts student graduation. A confusion matrix is a useful tool for understanding how a classification model predicts on a test dataset, showing the number of correct and incorrect predictions made by the model for each class. The confusion matrix in the figure is divided into two classes: rows and columns. Row 0 is the negative class (0) or "fail" while row 1 is the positive class (1) or "pass". Column 0 is the model's prediction for the negative class (0), while row 1 is the model's prediction for the positive class (1). The model generated a True Negative (TN) of 26, indicating that both the model's prediction and the actual class were negative. Here, the model predicted "fail" and the student actually failed. The false positives (FP) generated were 16: the model's prediction was positive, but the actual class was negative. The model predicted a "pass" but the student actually received a "fail" The false negatives (FN) generated were 130: the model's prediction was negative, but the actual class was positive. The model predicted a "fail" but the student actually passed. True Positive (TP) generated as many as 335: the model prediction is positive, and the actual class is also positive. The model predicts "pass" and the student actually passes. The high FN value reveals the imbalance between the number of predictions in classes 0 and 1. Although the `class_weight='balanced'` setting in SVM has been applied, it seems that there is still an imbalance that affects the results.

This confusion matrix reveals that the model correctly classifies the "pass" case (high TP) more frequently than the "fail" case (relatively low TN). The model also has a lower FP number compared to FN, indicating that it tends not to overfit the "Pass" class. Overall, this confusion matrix provides a clear view of

how the SVM model functions in predicting student graduation and is a basis for further evaluation and improvement.The results show that the SVM model produces an accuracy of 71.20% in predicting student graduation. With an accuracy of 71.20%, the model successfully classifies about 71.2% of the total test data correctly. Although this is the majority, about 28.8% of the predictions, both FP and FN, are still wrong. In the case of imbalanced classification, where one class (e.g., "pass") is much more dominant than the other class ("fail"), accuracy alone may not be enough to evaluate the performance of the model. High accuracy can be misleading if the model tends to always predict the majority class. From the previous confusion matrix, it appears that there is an imbalance between predictions for classes 0 and 1. A high FN value (130) indicates that the model quite often incorrectly predicts students who should have passed as failing. This can be a serious problem, especially in an academic context where graduation decisions have major implications. A lower FP (16) indicates that the model is better at avoiding incorrectly predicting students who should have failed as passing.

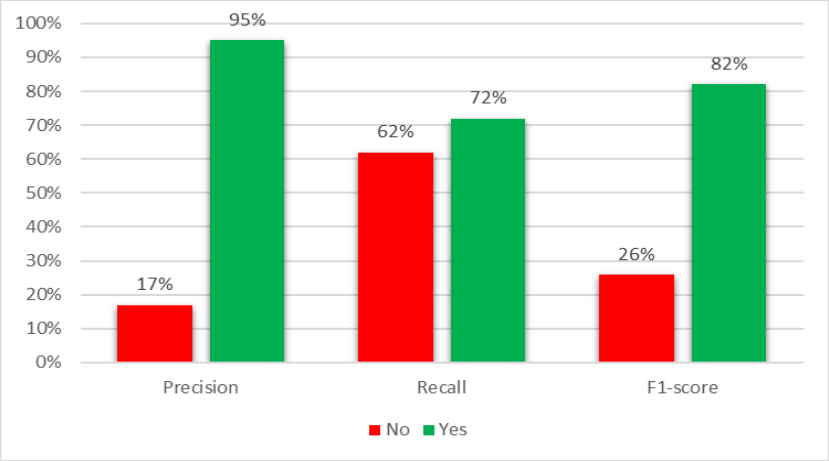

Fig 8. Matrix Evaluation

Figure 8 shows the results of the SVM evaluation matrix used to predict student graduation. This evaluation displays the precision, recall, and f1-score for two classes: "no" (students who did not graduate) and "yes" (students who passed). The precision for the "no" class is very low (17%). This implies that only 17% of the "no" predictions made by the model actually corresponded to "no" in the real data. This shows that the model tends to make a lot of mistakes when predicting students who did not graduate. The precision for the "yes" class is very high (95%). This means that almost all the "yes" predictions made by the model are correct. The model is extremely good at predicting graduating students. The recall for the "No" class is 62%, indicating that out of all the samples that were actually "No" the model successfully found 62% of them. Although not perfect, this shows that the model is quite sensitive in detecting students who did not graduate. The recall for the "yes" class is 72%, indicating that out of all the samples that were actually "yes," the model successfully found 72% of them. Although quite good, there is still room for improvement in capturing all passing samples. The F1-score for the “no” class is very low (26%), indicating that the model is unbalanced in predicting failing students, as both precision and recall are low. The F1-score for the “Yes” class is high (82%), indicating that the model is better at predicting passing students, with a good balance between precision and recall.This study's SVM model demonstrated excellent ability in predicting student graduation ("Yes" class), but it was less accurate in predicting failure ("No" class). This is evident from the high precision and f1-score for the “yes” class and very low for the “no” class. Significant class imbalance causes the model to focus more on predicting the majority class (“yes”).

As a result, the model’s performance for the minority class (“no” class) is very low. This is a common problem in machine learning when working with imbalanced data. Balancing the data is one of the first steps to improve model performance. We can use techniques like oversampling the minority class ("No" class) or undersampling the majority class ("Yes" class") to create a more balanced class distribution. Resetting the decision thresholds for the “No” and “Yes” classes can help improve recall and precision for the underrepresented class. Collecting more data for the “no” class can help improve the model’s representation and performance in detecting failure students. This can involve collecting more or more

detailed historical data. Using metrics that are more sensitive to class imbalance, such as the Matthews Correlation Coefficient (MCC) or balanced accuracy, can provide a clearer picture of the model's performance. Despite the limitations of the current model in predicting failure, universities can still use the results to identify students at risk of failure. This can allow universities to intervene early, such as by providing additional tutoring, academic counseling, or other support to help these students. We can also use the findings of this study to evaluate and improve existing educational policies, and to design more effective learning strategies to improve overall graduation rates.

IV. CONCLUSION

This study applies the Support Vector Machine (SVM) method to predict student graduation. The accuracy of 71.20% indicates that the SVM model has a decent ability to predict student graduation. However, there is room for improvement, especially in overcoming the high false negatives case. We can optimize the model's performance to provide more accurate and reliable predictions with the right improvement steps. We obtained several key findings from the evaluation results using precision, recall, and f1-score metrics: the model performed well for the "yes" class (pass) and poorly for the "no" class (fail). Precision reached 95%, indicating that the model is very accurate in predicting graduating students. A recall of 72% indicates that the model is able to detect most students who graduate, although not all. The high F1-score (82%) confirms that the model has a good balance between precision and recall for the "yes" class.

The model's very low precision (17%) indicates that many "no" predictions turned out to be wrong. The recall of 62% suggests that while the model can identify most failing students, it fails to identify many others. The low F1-score (26%) indicates a significant imbalance in the model's ability to predict the "no" class. The significantly better performance for the "Yes" class compared to the "No" class reflects the imbalanced data, where the number of passing students is much greater than the failing students (465 vs. 42). This study successfully demonstrates the application of the SVM method to predict student graduation with varying results. The model's good performance in predicting graduation and poor performance in predicting failure highlight the importance of addressing data imbalance. We can improve the prediction model to provide more accurate and useful results for decision-making in academic environments by improving the approach through data balancing, threshold adjustment, data augmentation, and the use of more appropriate metrics.

REFERENCES

[1] Y. Apridiansyah, N. D. M. Veronika, and E. D. Putra, "Prediksi Kelulusan Mahasiswa Fakultas Teknik Informatika Universitas Muhammadiyah Bengkulu Menggunakan Metode Naive Bayes," ***JSAI J. Sci. Appl. Informatics***, vol. 4, no. 2, pp. 236–247, 2021, doi: 10.36085/jsai.v4i2.1701.

[2] L. Setiyani, M. Wahidin, D. Awaludin, and S. Purwani, "Analisis Prediksi Kelulusan Mahasiswa Tepat Waktu Menggunakan Metode Data Mining Naïve Bayes : Systematic Review," ***Fakt. Exacta***, vol. 13, no. 1, p. 35, 2020.

[3] F. S. D. Alfia, "**Determinan Good University Governance dalam Persaingan Kompetitif Berkelanjutan pada Perguruan Tinggi Negeri di Surabaya**," Universitas Pembangunan Nasional "Veteran," 2024. [Online]. Available: https://repository.upnjatim.ac.id/26741/

[4] C. N. Dengen, K. Kusrini, and E. T. Luthfi, "Implementasi Decision Tree Untuk Prediksi Kelulusan Mahasiswa Tepat Waktu," ***SISFOTENIKA***, vol. 10, no. 1, p. 1, 2020, doi: 10.30700/jst.v10i1.484.

[5] S. S. Muliani, V. Sihombing, and I. R. Munthe, "Implementation of Exploratory Data Analysis and Artificial Neural Networks to Predict Student Graduation on-Time," ***Sink. J. dan Penelit. Tek. Inform.***, vol. 8, no. 2, pp. 1188–1199, 2024, doi: 10.33395/sinkron.v8i2.13658.

[6] L. R. Pelima, Y. Sukmana, and Y. Rosmansyah, "Predicting University Student Graduation Using Academic Performance and Machine Learning: A Systematic Literature Review," ***IEEE Access***, vol. 12, no. February, pp. 23451–23465, 2024, doi: 10.1109/ACCESS.2024.3361479.

[7] E. Alyahyan and D. Düştegör, "Predicting academic success in higher education: literature review and best practices," ***Int. J. Educ. Technol. High. Educ.***, vol. 17, no. 1, 2020, doi: 10.1186/s41239-020-0177-7.

[8] R. Asif, A. Merceron, S. A. Ali, and N. G. Haider, "Analyzing undergraduate students' performance using educational data mining," ***Comput. Educ.***, vol. 113, pp. 177–194, 2017, doi: https://doi.org/10.1016/j.compedu.2017.05.007.

[9] N. Mduma, K. Kalegele, and D. Machuve, "A Survey of Machine Learning Approaches and Techniques for Student Dropout Prediction," ***Data Sci. J.***, vol. 18, no. 1, Apr. 2019, doi: 10.5334/dsj-2019-014.

[10] M. Vaarma and H. Li, "Predicting student dropouts with machine learning: An empirical study in Finnish higher education," ***Technol. Soc.***, vol. 76, p. 102474, 2024, doi: https://doi.org/10.1016/j.techsoc.2024.102474.

[11] Hamdani and T. Nasution, "Implementasi Algoritma K-Nearest Neighbor untuk Penentuan Kelulusan Mahasiswa Tepat Waktu," ***J. Perangkat Lunak***, vol. 2, no. 1, pp. 1–14, 2020, doi: 10.32520/jupel.v2i1.944.

[12] A. Rohman and M. Rochcham, "Komparasi Metode Klasifikasi Data Mining untuk Prediksi Kelulusan Mahasiswa," ***Neo Tek.***, vol. 5, no. 1, Jun. 2019, doi: 10.37760/neoteknika.v5i1.1379.

[13] G. Suwardika and I. K. P. Suniantara, "Analisis Random Forest pada Klasifikasi Cart Ketidaktepatan Waktu Kelulusan Mahasiswa Universitas Terbuka," ***BAREKENG J. Ilmu Mat. dan Terap.***, vol. 13, no. 3, pp. 179–186, 2019, doi: 10.30598/barekengvol13iss3pp177-184ar910.

[14] J. Zeniarja, A. Salam, and F. A. Ma'ruf, "Seleksi Fitur dan Perbandingan Algoritma Klasifikasi untuk Prediksi Kelulusan Mahasiswa," ***J. Rekayasa Elektr.***, vol. 18, no. 2, pp. 102–108, 2022, doi: 10.17529/jre.v18i2.24047.

[15] V. K. S. Que, A. Iriani, and H. D. Purnomo, "Analisis Sentimen Transportasi Online Menggunakan Support Vector Machine Berbasis Particle Swarm Optimization," ***J. Nas. Tek. Elektro dan Teknol. Inf.***, vol. 9, no. 2, pp. 162–170, 2020, doi: 10.22146/jnteti.v9i2.102.

[16] T. Handhayani and L. Hiryanto, "Predicting and Analyzing the Length of Study-Time Case Study: Computer Science Students," ***ComTech Comput. Math. Eng. Appl.***, vol. 8, no. 2, p. 107, 2017, doi: 10.21512/comtech.v8i2.3756.

[17] A. Kesumawati and D. T. Utari, "Predicting patterns of student graduation rates using Naïve bayes classifier and support vector machine," ***AIP Conf. Proc.***, vol. 2021, no. 1, p. 60005, Oct. 2018, doi: 10.1063/1.5062769.

[18] A. Fadli, M. I. Zulfa, and Y. Ramadhani, "Performance Comparison of Data Mining Classification Algorithms for Early Warning System of Students Graduation Timeliness," ***J. Teknol. dan Sist. Komput.***, vol. 6, no. 4, pp. 158–163, 2018, doi: 10.14710/jtsiskom.6.4.2018.158-163.

[19] I. T. Utami, "Perbandingan Kinerja Klasifikasi Support Vector Machine (SVM) dan Regresi Logistik Biner Dalam Mengklasifikasikan Ketepatan Waktu Kelulusan Mahasiswa FMIPA UNTAD," ***J. Ilm. Mat. Dan Terap.***, vol. 15, no. 2, pp. 256–267, 2018, doi: 10.22487/2540766x.2018.v15.i2.11361.

[20] O. Bangun, H. Mawengkang, and S. Efendi, "Metode Algoritma Support Vector Machine (SVM) Linier Dalam Memprediksi Kelulusan Mahasiswa," ***J. Media Inform. Budidarma***, vol. 6, no. 4, p. 2006, 2022, doi: 10.30865/mib.v6i4.4572.

[21] S. Wiyono and T. Abidin, "Comparative Study of Machine Learning KNN, SVM, and Decision Tree Algorithm to Predict Student'S Performance," ***Int. J. Res. - GRANTHAALAYAH,*** vol. 7, no. 1, pp. 190–196, 2019, doi: 10.29121/granthaalayah.v7.i1.2019.1048.

[22] Kaggle, "**A GPA dataset gathered from a private university in Indonesia**," kaggle.com. Accessed: Jun. 13, 2024. [Online]. Available: https://www.kaggle.com/datasets/oddyvirgantara/on-time-graduation-classification