Essay Answer Classification with SMOTE Random Forest and AdaBoost in Automated Essay Scoring

Wilia Satria¹, Mardhani Riasetiawan*²

¹Master Program of Computer Science, FMIPA UGM, Yogyakarta, Indonesia
²Department of Computer Science and Electronics, FMIPA UGM, Yogyakarta, Indonesia

e-mail: ¹wiliasatria@mail.ugm.ac.id, *²mardhani@ugm.ac.id

Abstract

Automated essay scoring (AES) is used to evaluate and assessment student essays are written based on the questions given. However, there are difficulties in conducting automatic assessments carried out by the system, these difficulties occur due to typing errors (typos), the use of regional languages, or incorrect punctuation. These errors make the assessment less consistent and accurate. Based on the dataset analysis that has been carried out, there is an imbalance between the number of right and wrong answers, so a technique is needed to overcome the data imbalance. Based on the literature, to overcome these problems, the Random Forest and AdaBoost classification algorithms can be used to improve the consistency of classification accuracy and the SMOTE method to overcome data imbalances.

The Random Forest method using SMOTE can achieve an F1 measure of 99%, which means that the hybrid method can overcome the problem of imbalanced datasets that are limited to AES. The AdaBoost model with SMOTE produces the highest F1 measure reaching 99% of the entire dataset. The structure of the dataset is something that also affects the performance of the model. So the best model obtained in this study is the Random Forest model with SMOTE.

Keywords— SMOTE, Random Forest, AdaBoost, Automated Essay Scoring

Kata kunci— SMOTE, Random Forest, AdaBoost, Automated Essay Scoring

Received February 24th, 2023; Revised April 30th, 2023; Accepted October 31st, 2023
1. INTRODUCTION

The implementation of the National Examination (UN) for the 2018/2019 academic year refers to the minister of education and culture regulation. In its performance, the implementation of the UN refers to Law Number: 0047/P/BSNP/XI/2018 concerning standard operating procedures for implementing the National Examination for the 2018/2019 academic year. In 2020 the National Examination was held by having the characteristics of each question. Several types of questions, namely Numeration and Literacy. Each type of question has another characteristic: multiple choice questions, questions whose answers are in the form of checkboxes, questions with essay answers, and questions with answers in the appropriate order, namely true or false. On multiple choice questions, check boxes, or true and wrong choices, as well as answers that pay attention to the order, the assessment can be done in a clear, namely automatic scoring based on the answer key. However, it is difficult to carry out automatic assessments in the system, not on the structure of questions that requires answers with descriptions written by students themselves. These difficulties occur due to typing errors (typos), the use of regional languages, or wrong punctuation. This error results in a lack of consistency in the classification of wrong and right.

For essay answers, both the answers contained in the text and the answers that are not in the text become a problem because of the free nature of the answers it creates opportunities for students to answer with spelling errors, standard words, punctuation marks and many more errors that make the assessment less consistent and accurate. The method is applied to machine learning so that the assessment of student answers is more accurate with an answer key according to the guideline. That is, the classification between humans and machines is the same. In machine learning, there are many algorithms for classifying. In previous studies, machine learning in the application of automated essay scoring. Automated essay scoring is used to evaluate and assess student essays based on the questions given. The answers are classified first based on the true or false label to get an essay assessment.

Approaches machine learning including Random Forest, XGBoost, and Adaboost. According to [1] Random Forest is an algorithm that is suitable for classification with a high degree of accuracy for classifying types of disease. According to [2] Adaboost focuses on improving the interpretation and optimization of classifications. The research [15] classified the oil palm dataset to measure its maturity, using the SMOTE and Adaboost methods, from the experimental results the combination of these methods is better in terms of performance and efficiency than the previous method mentioned in the paper. In addition, another problem is that there is an imbalance of data on right and wrong answers which can affect the model's performance. Therefore, a special method is needed to deal with the data imbalance. According to [3] experimental results using the Synthetic Minority Oversampling Technique (SMOTE), greatly affect the classification performance and can increase the level of accuracy.

Previously, research was conducted on Automated Essay Scoring (AES) to help graders get accurate results. This study uses the Adaptive Boosting (AdaBoost) algorithm by calculating five times the accuracy of the cross-validation method and evaluating the model using the F1 score. The datasets used are three types of student answer datasets from the Program for International Student Assessments (PISA). Based on the experimental results, the AES system received an F1 score in the bicycle dataset of 71.74%, in the Jacket dataset, 67.20%, and in the Machu Picchu dataset, 97.69% [9]. Then research from [10] researched essay answers aimed at clustering and classification. The dataset used is the answers of Indonesian-speaking school students from the Ministry of Education and Culture. For grouping answers using the K-means clustering algorithm and Convolutional Neural Network (CNN) classification. The accuracy value produced by the model is 85% to 89.03%. This study will focus on analyzing the classification model by comparing the Random Forest and AdaBoost methods to get the best model by applying the data imbalance technique using SMOTE. The main purpose of using SMOTE is to reduce the error rate in the classification and to increase the accuracy of the
model's performance, which means that the classification between humans, namely PUSMENJAR, is the same as the classification generated by machines.

2. METHODS

This study aims to produce a classification model of right and wrong answers that can be used in automated essay scoring classifications, by analyzing two algorithms Random Forest and AdaBoost by applying the SMOTE method. The first is data collection and data analysis. The data in this study is a type of primary data. After getting a set of datasets, then do preprocessing and word representation. The technique oversampling using the SMOTE method to get balanced technique oversampling, synthetic data will be formed which makes the data balanced, then each classification is carried out using Random Forest and AdaBoost, then testing is carried out using k-fold cross-validation for the two algorithms. The results of the two algorithms are compared. The following is an overview of the research flow in the figure below.

![Research Flow Diagram](image)

2.1 Dataset Description

The data obtained is based on questions and answer keys that can be seen on the Indonesian Student Competency Assessment (AKSI) website page, namely at the address Aksi.puspendik.kemendikbud.go.id. The names of the datasets are the names of the titles of the reading illustrations for each question in each question of the essay answer. The dataset code is the code used to write the identity for each dataset in the system implementation. Description of the dataset, the number of total answers, right and wrong answers can be seen in the table below.

---

*Essay Answer Classification with SMOTE Random Forest and AdaBoost in ... (Wilia Satria)*
2.2 Preprocessing

Preprocessing is the first process carried out to process input data so that it is ready to be processed to the next stage [17]. Preprocessing is a step that needs to be done before further analysis. This is because the form of the text obtained is still unstructured which needs to be changed first to be structured[16]. Preprocessing is the initial data processing to transform unstructured textual data into structured data. Data preprocessing can improve data quality, accuracy and efficiency of the mining process [18]. Several stages are included in preprocessing, namely:

2.2.1. Lowercase(case folding)

Lowercase is the process of converting all input data into a single letter form, which is lowercase and removing all non-alphanumeric characters such as period punctuation, commas and exclamation points or empty characters.

2.2.2. Filtering

Filtering is the process of removing special characters such as signs ($, %, *, and so on). This process also removes words that do not match the parsed results for example emoticons. The sign/symbol is omitted because it does not have much effect on the determination of the label.

2.2.3. Tokenization

Tokenization is the process of breaking sentences into words. Tokenization is the process of recognizing documents into smaller units or cutting every sentence in the text into words, which produces words that stand alone or are not tied to other words. The tokenization stage separates words in sentences based on spaces, enter, tabulation, commas and periods. From this separation, tokenization will produce a term [7].

2.3 Word Representation

Term Frequency-Inverse Document Frequency (TF-IDF) is a feature extraction method in a document where the term frequency in the document indicates how important the word is. While the document frequency value (inverse document frequency) shows how common or important the word is in the whole document. The TF-IDF method is used in text classification to determine features or words that affect the classified document.

TF-IDF is commonly used to calculate the weight of a given keyword [5]. TF-IDF consists of two component values, namely term-frequency and inverse document frequency. Feature extraction TF-IDF assigns weight to term t in document d which is indicated by equation (1).
The value of $TFT_d$ is the weight of term $t$ in document $d$, namely the frequency of occurrence of term $t$ in document $d$. While $IDFT$ is the inverse document frequency of term $t$. Equation (2) is used to find the IDFT value.

$$IDFT = \log \frac{N}{dft}$$

The IDFT was obtained from the result of the logarithm of $N$ divided by DFT. $N$ is the total number of documents while DFT is the number of documents containing the term $t$.

### 2.4 SMOTE

**Synthetic Minority Oversampling Technique (SMOTE)** is one of oversampling good and effective overfitting in the oversampling process and deals with imbalances in the minority class. The SMOTE algorithm starts by finding the $k$ nearest neighbors for each instance in the minority class, and then for each neighbor, randomly selecting a point from the line connecting the neighbor and instance. To generate an instance, choose one of the $k$ neighbors at random, then calculate the difference between the instance and the $k$ nearest neighbors, then multiply by a random number between 0 to 1. Finally, the data at that point is entered as an instance. The equation used in the SMOTE method is as follows:

$$k_{\text{syn}} = x_i + (x_{\text{nn}} - x_i) \cdot \text{rand}[0,1]$$

$k_{\text{syn}}$ is instance generated in the oversampling, while $x_i$ is the instance from the minority class that will be used as a reference for creating instances and $x_{\text{nn}}$ artificial nearest neighbor instances of. Smote will share synthetic data on the line connecting minority class data with the $k$-minority nearest neighbor. In an imbalanced dataset, an imbalance between the minority class and the majority class.

### 2.5 Random Forest

Random forest is a classification method derived from a decision tree. Random forest is a development of the CART method by specifying the bootstrap method and random feature selection. The flexibility of random forest makes this method very useful as a data exploration method. Random forest is also known as the ensemble method or combined method. Called the combined method because it is formed from a small model. However, the prediction results are determined by combining all outputs in the small model or what is commonly called a submodel. To determine the classification in Random Forest, it is taken based on the most votes from each tree. Random forest uses the Gini value to determine the split to be used as a node, for which the formula is:

$$Gini(s) = 1 - \sum_{i=1}^{k} p_i^2$$

Where $p_i$ is the probability $s$ belonging to class $i$. Then after the Gini is obtained, the next step is to calculate the Gini Gain value using the formula:

$$Gini \text{Gain}(S) = Gini(s) - Gini(A,s) = Gini(s) - \sum_{i=1}^{k} \frac{|s_i|}{|s|} Gini(s_i)$$

Where is the partition $S$ caused by attribute $A$.

### 2.6 AdaBoost

**AdaBoost (Adaptive boosting)** is a machine-learning algorithm formulated by Yoav Freund and Robert Schapire. The AdaBoost algorithm is an algorithm that builds strong classifiers by combining several simple (weak) classifiers [2]. The AdaBoost equation is:
\[ F(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \]  

Which can be defined as:

- \( h_t(x) \): Basic classification (weak)
- \( \alpha_t \): Learning rate (learning rate)
- \( F(x) \): final result (strong classification).

3. RESULTS AND DISCUSSION

Results *automated essay scoring* can be seen in the table below. Input is the initial answer data that is inputted before the process is carried out preprocessing, while the output is the final result after doing *automated essay scoring*. The output is in the form of a number because the vectorization process has been carried out and the label is the result *scoring*, 0 means false and 1 means true.

**Table 2 Automated Essay Scoring Results**

<table>
<thead>
<tr>
<th>Action</th>
<th>Essay Answer</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>tidak Setuju: KARENA PEMBELI SANGAT MENYUKAI ANEKA RASA YG BERBEDA</td>
<td>0</td>
</tr>
<tr>
<td>Output</td>
<td>[0.000, 0.000, 0.000, ..., 0.000, ....]</td>
<td>0</td>
</tr>
<tr>
<td>Input</td>
<td>IYA SETUJU KARENA MARTABAK DENGAN ISI COKLAT DAN KEJU MERUPAKAN FAVORIT DI TOKO TERSEBUT</td>
<td>0</td>
</tr>
<tr>
<td>Output</td>
<td>[0.1399922, 0.28484065, 0.000000, ..., 0.000000]</td>
<td>0</td>
</tr>
</tbody>
</table>

3.1 Results of the SMOTE Random Forest Model Test

Testing in the SMOTE Random Forest model, a Random Forest test was carried out first, for the next step a test was carried out on the SMOTE Random Forest. For Random Forest performance results, the results can be seen in the graph below.

**Table 3 Random Forest Performance**

<table>
<thead>
<tr>
<th>No</th>
<th>Dataset</th>
<th>Precision</th>
<th>Recall</th>
<th>accuracy</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dataset_1: Sampah Plastik</td>
<td>0.81</td>
<td>0.77</td>
<td>0.81</td>
<td>0.81</td>
</tr>
<tr>
<td>2</td>
<td>Dataset_2: Binge watching</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
</tr>
<tr>
<td>3</td>
<td>Dataset_3: mengelola keuangan</td>
<td>0.8</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
</tr>
<tr>
<td>4</td>
<td>Dataset_4: Ayo melangkah</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
</tr>
<tr>
<td>5</td>
<td>Dataset_5: waktu dekomposisi</td>
<td>0.92</td>
<td>0.92</td>
<td>0.92</td>
<td>0.92</td>
</tr>
<tr>
<td>6</td>
<td>Dataset_6: pendapatan penduduk</td>
<td>0.96</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
</tr>
<tr>
<td>7</td>
<td>Dataset_7: Martabak Manis</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
</tr>
</tbody>
</table>

Based on the tables and figures on datasets 2, 5 and 7 show the same results for the parameters *precision, recall, accuracy dan F1 score*. The least percentage is in dataset 1, and the highest percentage is in dataset 6, which reaches 97% for the parameter *recall, f1-score dan accuracy*. 

IJCCS Vol. 17, No. 4, October 2023 : 359 – 370
Table 4 Random Forest with SMOTE Performance

<table>
<thead>
<tr>
<th>No</th>
<th>Dataset</th>
<th>Precision</th>
<th>Recall</th>
<th>accuracy</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dataset_1: Sampah Plastik</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
</tr>
<tr>
<td>2</td>
<td>Dataset_2: Binge watching</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
</tr>
<tr>
<td>3</td>
<td>Dataset_3: mengelola keuangan</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
</tr>
<tr>
<td>4</td>
<td>Dataset_4: Ayo melangkah</td>
<td>0.93</td>
<td>0.92</td>
<td>0.92</td>
<td>0.92</td>
</tr>
<tr>
<td>5</td>
<td>Dataset_5: waktu dekomposisi</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>6</td>
<td>Dataset_6: pendapatan</td>
<td>0.58</td>
<td>0.77</td>
<td>0.58</td>
<td>0.49</td>
</tr>
<tr>
<td>7</td>
<td>Dataset_7: Martabak Manis</td>
<td>0.96</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
</tr>
</tbody>
</table>

Based on Table 4.2 and the graph in Figure 4.3 below, the performance of the Random Forest by adding the SMOTE method produces the same values for each parameter in the dataset except for the 6th dataset, which produces different precision, recall, accuracy and F1-score. It can be seen that dataset_6 produces 58% precision, 77% recall, 49% F1-score, and 58% accuracy.

3.2 Adaboost SMOTE Model Test Results

The implementation of SMOTE Adaboost and the results of SMOTE testing with Adaboost and Adaboost without SMOTE were obtained. The results of Adaboost performance without SMOTE can be seen in the following table.

Table 5 Adaboost Performance

<table>
<thead>
<tr>
<th>No</th>
<th>Dataset</th>
<th>Precision</th>
<th>Recall</th>
<th>accuracy</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dataset_1: Sampah Plastik</td>
<td>0.78</td>
<td>0.77</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>2</td>
<td>Dataset_2: Binge watching</td>
<td>0.72</td>
<td>0.72</td>
<td>0.72</td>
<td>0.71</td>
</tr>
<tr>
<td>3</td>
<td>Dataset_3: mengelola keuangan</td>
<td>0.77</td>
<td>0.82</td>
<td>0.82</td>
<td>0.75</td>
</tr>
<tr>
<td>4</td>
<td>Dataset_4: Ayo melangkah</td>
<td>0.91</td>
<td>0.92</td>
<td>0.92</td>
<td>0.91</td>
</tr>
<tr>
<td>5</td>
<td>Dataset_5: waktu dekomposisi</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
</tr>
<tr>
<td>6</td>
<td>Dataset_6: pendapatan</td>
<td>0.96</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
</tr>
<tr>
<td>7</td>
<td>Dataset_7: Martabak Manis</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
</tr>
</tbody>
</table>

Based on the results of Table 4.2 and Figure 4.3 above, the Adaboost algorithm without SMOTE produces almost the same parameter values for each dataset. For datasets 2, 5 and 7, the four parameters have the same values, namely precision, recall, accuracy and F1-score which has a value of 82% for dataset_2, 91% for dataset_5 and 96% for dataset_7. For dataset_1 it produces 78% precision, 77% recall, f1-score and 80% accuracy. Dataset_3 gets 77% precision, 82% recall, 75% F1-score and 82% accuracy. Dataset 4 produces a precision and F1-score of 91%, a recall and an accuracy of 92%. Then in dataset 6 it gets 96% precision, recall, accuracy, and 97% F1-score.

Table 6 Adaboost with SMOTE Performance

<table>
<thead>
<tr>
<th>No</th>
<th>Dataset</th>
<th>Precision</th>
<th>Recall</th>
<th>accuracy</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dataset_1: Sampah Plastik</td>
<td>0.97</td>
<td>0.82</td>
<td>0.82</td>
<td>0.82</td>
</tr>
<tr>
<td>2</td>
<td>Dataset_2: Binge watching</td>
<td>0.87</td>
<td>0.87</td>
<td>0.87</td>
<td>0.87</td>
</tr>
<tr>
<td>3</td>
<td>Dataset_3: mengelola keuangan</td>
<td>0.93</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
</tr>
<tr>
<td>4</td>
<td>Dataset_4: Ayo melangkah</td>
<td>0.99</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
</tr>
<tr>
<td>5</td>
<td>Dataset_5: waktu dekomposisi</td>
<td>0.58</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
</tr>
<tr>
<td>6</td>
<td>Dataset_6: pendapatan penduduk</td>
<td>0.96</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>7</td>
<td>Dataset_7: Martabak Manis</td>
<td>0.96</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
</tbody>
</table>
Based on Table 6.9 and graph 6.6, the results of the parameters recall, accuracy, and f1-score have the same value in the 1st, 4th, 5th, 6th, and 7th datasets. In dataset 1 the value is 82%, in dataset 4 is 95%, in dataset 5 is 96%, dataset 6 is 97% and in dataset 7 is 99%. For precision in dataset 1 has a value that is much different from the other three parameters, namely 97%. In dataset 2 precision has the same value as the other parameters, namely 87%. Then another significant difference is in dataset 6 with a precision value of 58% which differs greatly from the other three parameters.

3.3 Comparison of Classification Model Results

The graph shows the results of the F1 score obtained from each algorithm with SMOTE and without SMOTE. From the data above, an average increase that occurs from SMOTE with AdaBoost is 2% while SMOTE with Random Forest is as much as 7% of the entire dataset.

From the graph, it can be seen that the precision results obtained for each model are different, according to the algorithm used. Then it is also influenced by different datasets. In dataset_1, the Random Forest algorithm with SMOTE has the highest precision value, reaching 97%. The Adaboost model produces a precision of 78% in dataset_1, while Adaboost with the addition of SMOTE achieves a precision of 83%. In dataset_2 the Random Forest model produces 83% precision, and the Random Forest model with the addition of SMOTE produces 86%. The AdaBoost model produces a precision of 72% and the addition of AdaBoost SMOTE produces a precision of 83%.
Based on the graph, it can be seen that the Recall results obtained by each model vary depending on the algorithm used and the differences in each dataset. In dataset_1 the highest recall results are in the Random Forest model with the addition of SMOTE which reaches 97%. In dataset_2 the Random Forest model with SMOTE is able to increase the recall to 86% from the 83% recall in the Random Forest model. The Adaboost model produces a 72% recall and the Adaboost model with the addition of SMOTE can increase the recall to 83%. In dataset_3 the best model is Random Forest with the addition of SMOTE which results in a 95% recall. In dataset_4 the best recall results are in the Adaboost model with the addition of SMOTE with a recall value of 95%. In dataset_5 the best model is Random Forest with SMOTE which achieves 99% recall. For dataset_6, the Random Forest, AdaBoost, and Adaboost models with the addition of SMOTE have the same recall result of 97%, while the Random Forest model with SMOTE has decreased with a 77% recall result. In dataset_7 the Random Forest model produces a 96% recall and the Random Forest model with SMOTE has decreased to 95%. For the Adaboost model, it produces a 96% recall and Adaboost with SMOTE increases, namely it produces a 99% recall.

In the graph below it can be seen that the accuracy value of each dataset is different, as well as each algorithm model. For dataset_1 the best model is Random Forest with the addition of SMOTE which produces 97% accuracy. In dataset_2 the Random Forest model with SMOTE produces 86%, an increase from the initial Random Forest model yielding 83%. For the Adaboost model, it produces a 72% recall, which increases with the addition of SMOTE to 83%. In dataset_3 the best model reaches 95%, namely the Random Forest algorithm with the addition of SMOTE. In dataset_4, the Adaboost model with the addition of SMOTE has the highest recall value of 95%. In dataset_5 the Random Forest model with the addition of SMOTE has increased to 99% and Adaboost with SMOTE to 96%. In dataset_6, the Random Forest, AdaBoost and Adaboost models with the addition of SMOTE have the same recall value of 97%, while the Random Forest model with SMOTE changes with a recall value of 60%. In dataset_7 the Adaboost model with the addition of SMOTE has increased with a recall value of 99%.

**Figure 4 Recall Comparison Results**

Essay Answer Classification with SMOTE Random Forest and AdaBoost in ... (Wilia Satria)
4. CONCLUSIONS

The use of SMOTE in Random Forest increases the F1-Score performance by 7%, 3% precision, 3% recall, and 0.14% accuracy. Using SMOTE on AdaBoost improves model performance by 0.2% F1-score, 6% precision, 7% recall, and 8% accuracy. In this study, Random Forest with SMOTE produced the highest classification performance among all datasets, namely F1-score 99% on the 5th and 6th datasets, recall 99% on dataset 5, precision 99% on dataset 5 and so on accuracy reached 99% on dataset 5. Adaboost with the addition of SMOTE in this study resulted in the highest score classification performance F1-score 99% on dataset 7, precision 99% on dataset 7, recall 99% on dataset 7, and anyway accuracy 99% on dataset 7. Based on these results, the use of SMOTE does not have a significant impact on increasing model performance. But on the F1-score side using the SMOTE Random Forest model it is more influential than the AdaBoost model with SMOTE. The AdaBoost model with SMOTE produces a lower boost than the Random Forest with SMOTE.

ACKNOWLEDGEMENTS

This research was supported by the provision of datasets by PUSMENJAR Kemendikbud RI. This paper is part of the Master's thesis in Magister Computer Science Gadjah Mada University's first author.

REFERENCES


Essay Answer Classification with SMOTE Random Forest and AdaBoost in … (Wilia Satria)


