Dysgraphia Identification from Handwriting with Support Vector Machine Method

Sari Widya Sihwi, Khoirul Fikri, Abdul Aziz

Informatics Department of Mathematics and Natural Science Faculty of Universitas Sebelas Maret, Indonesia
E-mail: sariwidya@staff.uns.ac.id, khoirulfikri@student.uns.ac.id, aaziz@staff.uns.ac.id

Abstract. Dysgraphia, a handwriting disorder in which a person has difficulty in writing at any level such as slow writing or unreadable letter. Many research has done to study the characteristics and to diagnose it for early prevention in children. In this study, we try to identify dysgraphia among children and divide it into 4 class, normal, light, moderate, and severe. Therefore an android application with embedding a handwriting recognition tool was created to collect the data from elementary school students that have dysgraphia and those who don't. We use Support Vector Machine in classifying the data to identify dysgraphia because SVM has the ability to learn well with limited data compared to ANN on many occasions. The result, after using three different kernels in SVM such as Linear, Polynomial, and Radial Base Function kernel (RBF), shows that the RBF kernel produces better average accuracy and Cohen's kappa value compared to Linear and Polynomial kernels, where the average accuracy of each kernel is 78.56% for Linear, 81.40% for Polynomial, and 82.51% for RBF.

1. Introduction

Writing is one of the basics in learning other than reading. As a means in the learning process, writing is taught at early stage along with reading. Many teachers misinterpret that their students are lazy in learning to write, thus forcing the students in a wrong manner, even though they may be affected by a disorder called Dysgraphia. Dysgraphia is a disorder in which a person has difficulty writing at any level, including an unreadable letter, slow writing, difficulty spelling, and syntactic problems and composition [1]. So that a teacher should better understand the condition of his students and be able to deal with the right actions.

In the research by [2] studied the differences in handwriting and writing letters, both from and not dysgraphic sufferers, to obtain characteristics that could be used as predictive factors. From results of the research, there are significant differences between those who suffer from dysgraphia and those who do not, there are at time and content of the writing. While the research by [3] focuses on difficulties in writing in children with Developmental Coordination Disorder (DCD) by analyzing the nervous system in children, it is expected to provide more explanation about what happened. From results of the research, it was found that Dysgraphia became comorbidity in Developmental Coordination Disorder (DCD) and may reflect the severity of DCD, and Dysgraphia must be considered different from bad handwriting. Also, in the research by [4], which centered on the development of an expert system for the detection of Dysgraphia, dividing the levels in Dysgraphia patients into three categories, light, moderate, and severe. The division is based on the characteristics found in children diagnosed as Dysgraphia such as if the children wrong in holding a pencil, the size of letter is unideal, and the letter writing is thin, therefore the children is categorized into light
Dysgraphia. While, if the children also slow at writing and hands quickly feel tired, therefore the children will be categorized into moderate Dysgraphia.

In the research by [2] they used Wacom tablets in recording writing and analyzed with the ComPET system, so it obtained data in the form of time, pressure, and font size. With this information, it was compared between the writings of dysgraphic sufferers and those who did not. Likewise in the research conducted by [5] using tablets to record writing, obtained the information that was divided into 4 categories: Static Features, Kinematic Features, Pressure Features, and Tilt Features. Whereas in the research by [6] create a mobile application in diagnosing children with dysgraphia. With the help of Handwriting-Recognition software, it is possible to know what is written by the child through the mobile application that has been made and analyzes the results of the writing that has been obtained by comparing it with the actual text.

To classify the results of writing whether dysgraphia or not, in this study proposed using Support Vector Machine (SVM) because some studies showed how good performance produced by SVM is and gain global solution on many occasions compared to others. In the research by [7], classifying the performance efficiency of marine diesel engines using Artificial Neural Network (ANN) and Support Vector Machine (SVM). The results are obtained with only using 25 data samples, that SVM is superior to ANN both in efficiency, BSFC, Max-P, Soot, and Nox which are the parameters in the measurement of the two methods. Whereas in the research by [8], using the Naive Bayes method, Linear SVM, Polynomial SVM, and Sigmoid SVM for detection of attacks on computer networks, that all 4 methods produce accuracy percentages of 85.055%, 99.995%, 99.999%, and 99.995% respectively, which is accuracy The biggest is obtained by the Polynomial SVM method.

In building the model for classification, training data is the main key, and the problem that often occurs is when the training data that is owned is imbalanced so that the constructed model is not good. In most case, real-world data is imbalanced [9], and in this study, the data that have been obtained is imbalanced, where the amount of data in one of the categories is too low. Therefore we use one method of oversampling in handle unbalanced data that is Synthetic Minority Over-sampling Technique (SMOTE). In some research, such as [10] combining oversampling and undersampling methods to stabilize the number of instances between classes, where oversampling using SMOTE and undersampling using Tomek links. Then in the research by [9] also use SMOTE to overcome the negative effects of the model formed due to the imbalance of instances in the class. Both in the paper above show quality enhancement of the model that is formed, be evidenced by the result of increased accuracy.

Therefore in this study, we try how to identifying the writing of children both that dysgraphia and those not, which divided into four classes, that are normal, light, moderate, and severe based on a study conducted by [4]. Using one of the classification methods, that is Support Vector Machine with Linear, Polynomial, and Radial Basis Function (RBF) kernels for doing the identification. In collecting the data we use a mobile application, as done by [6] because of its ease in retrieving written data and in its development, as well as the use of Handwriting-Recognition tool to detect the writing result. In dealing with unbalanced data, we use SMOTE, one of the oversampling methods. Because of the data we have collected is difficult to obtain, it is necessary to avoid data reduction. Therefore, we not try to use the undersampling method for balancing the data.

2. METHODOLOGY

![Figure 1. Research steps that used in this paper](image)
2.1 Data Collection

2.1.1 Application for Collecting The Data

As shown in Figure 1. First step in this paper is data collection, but to be able to get the writing data from children who suffer from dysgraphia or those who are not, an Android-based application with embedding the WritePad® Handwriting Recognition tool is needed (https://github.com/phatware/WritePadSDK), which can recognize letters written on the smartphone screen using the stylus and some other information about writing. In this research, the smartphone that we use for collecting the data is Lenovo Vibe K4 Note.

![Figure 2. Android Layout for handwriting data collection](image)

Android application, as shown in Figure 1., is set only to accept alphabet characters. Because, after doing some experiments it was found that it is not uncommon to detect errors when writing several letters such as 'm', 'c', 'g', 's' which are mistaken for numbers or other characters. The data that can be taken from this application is in the form of time, pressure, the distance between letters, ideal or not the size of a letter, the position of letters (up or down), consistency of the boundary line, and text from the writing result, which will later be used as attribute in classification. They are chosen base on the research conducted by [4] and [5]. In the research by [5] said that the speed of writing, pressure, and the distance between words are in the most important features that have been discovered. While ideal size, position, consistency of the boundary line, and the result of writing are characteristics of dysgraphia that mention it in the research conducted by [4] that use for devide the dysgraphia into three categories, shown in Table 1.

2.1.2 Data Gathering

Data was obtained from the students of Al-Firdaus and Lazuardi Kamila Elementary School in Surakarta using the Android application that was created. The targets are students in 3rd to 6th grade, divided into 4 categories; normal, light, moderate, and severe. The obtained data are 135 records from 32 students, showed in Table 2., with 14 from Al-Firdaus and 18 from Lazuardi Kamila students, where each student was asked to write random sentences as many as 3-7 times in android application that was created. To determine whether each student is in the normal, light, moderate or severe category, we tested the children using the concepts in the research conducted by [4]. Some of the students seem not comfortable while writing on the smartphone screen due to the texture of the smartphone screen is slippery, so the writing result looks a bit different from writing on paper.
Table 1. The characteristic of dysgraphia and its allocation to light, moderate, and severe. [4]

| Characteristic                     | Light | Moderate | Severe |
|------------------------------------|-------|----------|--------|
| M                                  | NM    | M        | NM     |
| Wrong in holding a pencil          | V     | V        | V      |
| Unideal font size                  | V     | V        | V      |
| Thin writing                       | V     | V        | V      |
| Slow writing                       | V     | V        |        |
| “uphill” and “downhill” writing    | V     | V        |        |
| Irregular distance between letters | V     | V        |        |
| Unideal shape of the letter        | V     | V        |        |
| The hands quickly feel tired       | V     | V        |        |
| Capital letters and lowercase letters are still mixed | V |
| Incomplete writing                 | V     |        |        |
| Letter misperception               | V     |        |        |
| Unable to finish writing test      | V     |        |        |

*M = mandatory, *NM = Non-Mandatory

Table 2. Raw data that obtained from school

| Category | Total |
|----------|-------|
| Normal   | 55    |
| Light    | 55    |
| Moderate | 16    |
| Severe   | 8     |

Table 3. Raw data after deleting same corrupted data

| Category | Total |
|----------|-------|
| Normal   | 43    |
| Light    | 29    |
| Moderate | 13    |
| Severe   | 6     |
Table 4. Exemple of raw data that have been obtained

| Feature                          | Value                                                                 |
|----------------------------------|----------------------------------------------------------------------|
| Text from writing result (Actual text : Salah satu tarian yang terkenal di Indonesia adalah Tari Lilin) | 0.185,0.3465,0.3591,0.3734,0.3802,0.3999,0.3703,0.3836,0.387,0.3799,  
 |                                  | 0.4795,0.4533,0.3248,0.3863,0.3044,0.4798,0.44,0.3945,0.4458,0.4791,  
 |                                  | 0.5196,0.3322,0.3867,0.3999,0.3435,0.319,0.3465,0.3737,NaN,0.4662,    
 |                                  | 0.3724,0.3867,0.3465,0.3731,0.4135,0.3986,0.3027,0.2153,0.267,        
 |                                  | 0.4264,0.3465,0.3707,0.3197,0.4268,0.506,0.3758,0.4003,               
 |                                  | 0.4788,0.252,0.3734,0.2802,0.3462,0.3843,0.4533,                      |
| Pressure                         | 5,0,2,1,10,1,0,6,0,10,3,4,0,0,5,0,0,20,3,6,19,0,0,1,9,0,24,35,16,7,6,7,0,  
 |                                  | 0,0,2,0,0,2,8,10,0,1,26,9,0,10,0,19,0,21,15,11                      |
| Distance each letter             | 73                                                                  |
| Time (in second)                 | 1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,  
 | Ideal or not a                   | 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,  
 | Position (is in parallel or not) | 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,  
 | Consistency of the boundary line | 0,1,0,1,0,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,  
 |                                                   | 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, |

\* I mean yes, 0 mean no.

2.2 Data Pre-Processing

After browsing each data, it turns out that there are incomplete data due to a bug in the application during the data retrieval process, then we remove the incomplete data and as a result leaving 91 records, shown in Table 3.

The data obtained still cannot be used for classification, because each attribute has different values, some are in the form of numbers and text as shown in the Table 4. Then it will be normalized so that all data is in the form of numbers and within the same scope and ensures that no data has an empty attribute value. Therefore, the similarity between the text from writing result and the original sentence will be sought using Cosine Similarity with \( k = 2 \) for k-shingles. For the pressure, distance and the time use Min-Max Scaling so that the data range is only 0-1. And for the rest like ideal or not the size of a letter, the position of letters (up or down), consistency of the boundary line, it will be changed to an average, because the data is binary (0 or 1).

\[
\text{Cosine similarity} = \frac{A \cdot B}{\|A\| \|B\|} 
\]

\[
x' = \frac{(x - \min(x))}{(\max(x) - \min(x))} 
\]

After pre-processing, the data will be oversampling using the SMOTE to create a better model in classification. As shown in Table 3, the amount of data is imbalanced, especially in the Severe class that has an amount which too far from the other classes. Synthetic Minority Over-sampling Technique or SMOTE is creating extra data which depend on nearest neighbors they have [11], in this research using 3 nearest neighbors for creating extra data. The minority classes was over-sampled at 50%, 200%, and 600% of its original size to get the same amount of data for each class. The formula is
shown in equation (3). Where for each instance $x_i$ in minority class, SMOTE searches its k nearest neighbors and one neighbor is randomly selected as $x'$, then a random number between [0, 1] $\delta$ is generated [12].

$$x_{new} = x_i + (x' - x_i)\delta$$

(3)

### 2.3 SVM Classification

As noted earlier, in this research will use Support Vector Machine to perform dysgraphia classification with normal, light, moderate, and severe classes. Therefore, in the training and testing process, the data will be tested with several kernel models on SVM like Linear, Polynomial, and RBF. As noted in [13] that Support Vector Machine is basically designed only for binary classification. Therefore, multiclass problems must be parsed into several binary classification tasks that can be resolved efficiently using binary classifiers [14], and the easiest ones to use are One-vs-One (OvO) or One-vs-All (OvA). So, in this research will also compare both OvO and OvA for handling multiclass problems. Also as noted earlier, SMOTE are used for oversampling the minority data, we will compare both before and after using SMOTE in classification using SVM. All of the experiments above will be tested in 10-Cross-Validation case.

### 2.4 Result Evaluation

For the evaluation of the final results of each experiment will be evaluated using a confusion matrix to calculate accuracy and Cohen's Kappa value to measure how well each SVM kernel model has been formed [15].

#### Table 5. Confusion Matrix

| Predicted | Actual | Positive | Negative |
|-----------|--------|----------|----------|
| Positive  | TP     | FP       |          |
| Negative  | FN     | TN       |          |

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)}$$

(4)

$$k(\text{Cohen's Kappa}) = \frac{(p_e - \bar{p_e})}{(1 - \bar{p_e})}$$

(5)

$$p_e = \text{Accuracy}$$

(6)

$$p_e = \frac{(A+B)}{(TP+TN+FP+FN)}$$

(7)

$$A = (TP + FP)(TP + FN)$$

(8)

$$B = (TN + FN)(TN + FP)$$

(9)

### 3. RESULT AND DISCUSSION

#### 3.1 SVM Classification

For parameters each kernel, we use GridSearch to obtain the best parameters that will result in high accuracy. For linear kernel use $C = 16$. Polynomial kernel use $C = 0.0001$, degree = 4, $\text{coef0} = 3.5$, and gamma = 10. RBF kernel use $C = 16$ and gamma = 1. All experiments are tested using sklearn.SVC [16].
3.1.1 Before using SMOTE

The experiment in this section, using data that shown in Table 3. The results are shown using comparison table. Table 6.1 and Table 6.2 show comparison using One vs One for handling multiclass problems, where Table 6.1 is based on accuracy and Table 6.2 is based on kappa value from each kernel. While, Table 7.1 and Table 7.2 show comparison using One vs All for handling multiclass problems, where Table 7.1 is based on accuracy and Table 7.2 is based on kappa value from each kernel.

### Table 6.1
**Comparison based on accuracy with OvO**

| k-fold | Linear | Polynomial | RBF |
|--------|--------|------------|-----|
| 1      | 0.50   | 0.60       | 0.40|
| 2      | 0.7778 | 0.7778     | 0.7778|
| 3      | 0.7778 | 0.7778     | 0.7778|
| 4      | 0.7778 | 0.7778     | 0.8889|
| 5      | 0.6667 | 0.7778     | 0.7778|
| 6      | 0.5556 | 0.5556     | 0.5556|
| 7      | 0.6667 | 0.7778     | 0.7778|
| 8      | 0.7778 | 0.7778     | 0.7778|
| 9      | 0.6667 | 0.5556     | 0.6667|
| 10     | 0.8889 | 0.6667     | 0.7778|

### Table 6.2
**Comparison based on kappa value with OvO**

| k-fold | Linear | Polynomial | RBF |
|--------|--------|------------|-----|
| 1      | 0.2753 | 0.4117     | 0.1176|
| 2      | 0.4375 | 0.55       | 0.55|
| 3      | 0.6667 | 0.6667     | 0.6667|
| 4      | 0.64   | 0.64       | 0.8125|
| 5      | 0.1818 | 0.4        | 0.4|
| 6      | 0.3076 | 0.3076     | 0.3076|
| 7      | 0.5263 | 0.6727     | 0.6727|
| 8      | 0.55   | 0.55       | 0.55|
| 9      | 0.4807 | 0.3207     | 0.4807|
| 10     | 0.7804 | 0.1562     | 0.5|

### Table 7.1
**Comparison on accuracy with OvA**

| k-fold | Linear | Polynomial | RBF |
|--------|--------|------------|-----|
| 1      | 0.6    | 0.5        | 0.6 |
| 2      | 0.6667 | 0.7778     | 0.7778|
| 3      | 0.7778 | 0.7778     | 0.7778|
| 4      | 0.7778 | 0.7778     | 0.8889|
| 5      | 0.7778 | 0.7778     | 0.7778|
| 6      | 0.5556 | 0.5556     | 0.5556|
| 7      | 0.6667 | 0.7778     | 0.7778|
| 8      | 0.5556 | 0.6667     | 0.5556|
| 9      | 0.6667 | 0.6667     | 0.6667|
| 10     | 0.7778 | 0.6667     | 0.7778|

### Table 7.2
**Comparison based on kappa value with OvA**

| k-fold | Linear | Polynomial | RBF |
|--------|--------|------------|-----|
| 1      | 0.4366 | 0.2753     | 0.4366|
| 2      | 0.3076 | 0.55       | 0.55|
| 3      | 0.6727 | 0.6727     | 0.6667|
| 4      | 0.64   | 0.64       | 0.8125|
| 5      | 0.4    | 0.4        | 0.4|
| 6      | 0.3076 | 0.2941     | 0.3076|
| 7      | 0.4905 | 0.6470     | 0.6470|
| 8      | 0.23   | 0.4255     | 0.3076|
| 9      | 0.4807 | 0.46       | 0.4807|
| 10     | 0.5681 | 0.1562     | 0.5|

3.1.2 After using SMOTE

The experiment in this section, using data shown in Table 3 that have been oversampled using SMOTE, so amount of instance become same, that 43 records of each class. The results are shown using comparison table. Table 8.1 and Table 8.2 show comparison using One vs One for handling multiclass problems, where Table 8.1 is based on accuracy and Table 8.2 is based on kappa value from each kernel. While, Table 9.1 and Table 9.2 show comparison using One vs All for handling multiclass problems, where Table 9.1 is based on accuracy and Table 9.2 is based on kappa value from each kernel.
Table 8.1
Comparison based on accuracy with OvO

| k-fold | Linear | Polynomial | RBF |
|--------|--------|------------|-----|
| 1      | 0.7222 | 0.8333     | 0.8889 |
| 2      | 0.7222 | 0.7778     | 0.8333 |
| 3      | 0.7647 | 0.7646     | 0.7647 |
| 4      | 0.8235 | 0.7647     | 0.7647 |
| 5      | 0.8235 | 0.8823     | 0.9411 |
| 6      | 0.8235 | 0.8235     | 0.8235 |
| 7      | 0.5882 | 0.5882     | 0.5294 |
| 8      | 0.9441 | 0.9411     | 0.8823 |
| 9      | 0.7647 | 0.8823     | 0.9411 |
| 10     | 0.8235 | 0.8823     | 0.8823 |

Table 8.2.
Comparison based on kappa value with OvO

| k-fold | Linear | Polynomial | RBF |
|--------|--------|------------|-----|
| 1      | 0.625  | 0.7672     | 0.8444 |
| 2      | 0.5945 | 0.6842     | 0.7621 |
| 3      | 0.6699 | 0.6690     | 0.6699 |
| 4      | 0.8425 | 0.6880     | 0.6880 |
| 5      | 0.7627 | 0.8418     | 0.9209 |
| 6      | 0.7462 | 0.7512     | 0.7462 |
| 7      | 0.4278 | 0.4687     | 0.4086 |
| 8      | 0.9205 | 0.9205     | 0.8411 |
| 9      | 0.6894 | 0.8365     | 0.9174 |
| 10     | 0.7605 | 0.8411     | 0.8411 |

Table 9.1
Comparison based on accuracy with OvA

| k-fold | Linear | Polynomial | RBF |
|--------|--------|------------|-----|
| 1      | 0.8333 | 0.8333     | 0.8889 |
| 2      | 0.7222 | 0.7778     | 0.7778 |
| 3      | 0.7058 | 0.7058     | 0.7058 |
| 4      | 0.6470 | 0.7646     | 0.7647 |
| 5      | 0.6470 | 0.9411     | 0.8823 |
| 6      | 0.8235 | 0.8235     | 0.8235 |
| 7      | 0.5882 | 0.5294     | 0.5294 |
| 8      | 0.9441 | 0.9411     | 0.9411 |
| 9      | 0.7647 | 0.8823     | 0.8823 |
| 10     | 0.7647 | 0.8823     | 0.8823 |

Table 9.2.
Comparison based on kappa value with OvA

| k-fold | Linear | Polynomial | RBF |
|--------|--------|------------|-----|
| 1      | 0.775  | 0.7589     | 0.8421 |
| 2      | 0.5890 | 0.6771     | 0.6842 |
| 3      | 0.5952 | 0.5933     | 0.5933 |
| 4      | 0.5363 | 0.6880     | 0.6880 |
| 5      | 0.5277 | 0.9209     | 0.8418 |
| 6      | 0.7462 | 0.7462     | 0.7462 |
| 7      | 0.4615 | 0.4082     | 0.4086 |
| 8      | 0.9212 | 0.9212     | 0.9205 |
| 9      | 0.6682 | 0.8365     | 0.8365 |
| 10     | 0.6880 | 0.8411     | 0.8411 |

3.2 Result
After doing a lot of experiments by comparing the results of each kernel using OvO and OvA methods and in the data before and after using SMOTE. It was found that oversampling showed an improvement in the model, shown in the increasing of kappa value in each experiment, this proves that dysgraphia identification needs balance data on each class to acquire a better model classification. Moreover, from the three kernels, the best results were shown in experiments using the RBF kernel followed by Polynomial and Linear kernel. Even though Polynomial kernel seems good too, but take more a lot of time in building the model compared to RBF and Linear, and overall in overcoming the multiclass that One vs One (OvO) showed better results.

4. Conclusion
4.1 Conclusion
This study succeeded in how to identifying the writing of children both that dysgraphia and those not, which divided into four classes, that are normal, light, moderate, and severe. The best results are shown in experiments using the RBF kernel with One vs One method on data that has been carried out by oversampling using SMOTE, with an average accuracy of using 10-cross-validation is 82.51%. Misclassification occurs because, in data collection some students are not accustomed to writing on the Android screen, so some students whose writing results on paper and on the Android screen experience significant differences.

4.2 Suggestion
From the results of the research, there are shortcomings in the problem of data retrieval where students are not accustomed to writing on an android screen because of the texture of the smartphone
screen is slippery, so the writing result looks a bit different from writing on paper. Therefore, we recommend introducing the application to students first and made them get used to it so that the data obtained are more accurate. Future work for this research, try to use a device that can record the writing on the paper, so the data that obtained are more natural — looking for another feature that can be taken to improve the accuracy of the identification of dysgraphia or use feature selection to get the most important feature. Use the results of identification, whether children are detected as light, moderate or severe dysgraphia to carry out appropriate therapy to cure dysgraphia in children.

Acknowledgments
This research was funded by Hibah Penelitian Sosial, Humaniora dan Pendidikan (PSHP) Sebelas Maret University, 2018.

References
[1] P. Chung dan D. R. Patel, “questia,” 1 January 2015. [Online]. Available: https://www.questia.com/library/journal/1P3-3783245431/dysgraphia.
[2] L. Hen-Herbst and S. Rosenblum, "Which characteristics predict writing capabilities among adolescents with dysgraphia?,” Pattern Recognition Letters, 2018.
[3] C. Lopez, C. Hemimou, B. Golse dan L. Vaivre-Douret, “often associated with minor neurological dysfunction in children with developmental coordination disorder (DCD),” Neurophysiologie Clinique, vol. 48, no. 4, pp. 207-217, 2018.
[4] D. A. Kurniawan, S. W. Sihwi dan Gunarhadi, “An expert system for diagnosing dysgraphia,” 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE), vol. 5, no. 6, pp. 468-472, 2017.
[5] T. Asselborn, T. Gargot, Ł. Kidziński, W. Johal, D. Cohen, C. Jolly dan P. Dillenbourg, “Automated human-level diagnosis of dysgraphia using a consumer tablet,” npj Digital Medicine, vol. 1, no. 1, p. 42, 2018.
[6] T. F. Raza, H. Arif, S. H. Darvagheh dan H. Hajjdiab, “Interactive Mobile Application for Testing Children with Dysgraphia,” Proceedings of the 9th International Conference on Machine Learning and Computing - ICMLC 2017, pp. 432-436, 2017.
[7] X. Niu, C. Yang, H. Wang dan Y. Wang, “Investigation of ANN and SVM based on limited samples for performance and emissions prediction of a CRDI-assisted marine diesel engine,” Applied Thermal Engineering, vol. 111, pp. 1353-1364, 2017.
[8] M. F. Fibrianda dan A. Bhawiyuga, “Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer Dengan Metode Naïve Bayes Dan Support Vector Machine ( SVM ),” vol. 2, no. 9, pp. 3112-3123, 2018.
[9] M. Alghamdi, M. Al-Mallah, S. Keteyian, C. Brawner, J. Ehrman dan S. Sakr, “Predicting diabetes mellitus using SMOTE and ensemble machine learning approach: The Henry Ford ExercIsc Testing (FIT) project,” PLOS ONE, vol. 12, no. 7, 2017.
[10] H. Sain dan S. W. Purnami, “Combine Sampling Support Vector Machine for Imbalanced Data Classification,” Procedia Computer Science, vol. 72, pp. 59-66, 2015.
[11] N. V. Chawla, K. W. Bowyer, L. O. Hall dan W. P. Keegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” Journal of Artificial Intelligence Research, vol. 16, pp. 321-357, 2002.
[12] Z. Zheng dan Y. L. Yunpeng Cai, “OVERSAMPILING METHOD FOR IMBALANCED CLASSIFICATION,” Computing and Informatics, vol. 34, pp. 1017-1037, 2015.
[13] C.-J. Lin dan C.-W. Hsu, “A comparison of methods for multiclass support vector machines,” IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council, vol. 13, no. 4, pp. 1026-1027, 2002.
[14] M. Aly, “Survey on Multiclass Classification Methods,” 2005.
[15] I. Kumar, J. Virmani, H. S. Bhadauria, M. K. Panda dan Kriti, “Chapter 13 – Classification of Breast Density Patterns Using PNN, NFC, and SVM Classifiers,” Elsevier Inc., 2018, pp. 223-243.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, A. Müller, J. Nothman, G. Louppe, P. Prettenhofer, R. Weiss, V. Dubourg dan J. Vanderplas, “Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2012.