Journal of Applied Science, Engineering and Technology
e-ISSN: 2722-8363 p-ISSN: 2722-8371
DOI: https://doi.org/10.47355/aset

J. ASET

Implementation of the Support Vector Machine Algorithm in the Student Thesis Subject Classification System at the University of Lampung

Rizky Hadi [1,*], Meizano Ardhi Muhammad, S.T., M.T. [1], Puput Budi Wintoro, S. Kom, M.T.I [1], Ing. Hery Dian Septama, S.T. [1], Rio Ariestia Pradipta, S.Kom.,M.T.I. [1] Electrical

[1] Engineering Department, Engineering Faculty, University of Lampung, Jl. Sumantri Brojonegoro No.1, Bandar Lampung 35145, Indonesia

[*] Correspondence: rizkyhadi942@gmail.com

Abstract: The Technical Implementation Unit (UPT) of the University of Lampung Library is responsible for archiving student scientific work (thesis). Filing is done manually, using the DDC or Dewey Decimal Classification system, thus allowing for the tendency of inaccurate Subject selection and long grouping durations. This study aims to apply the Support Vector Machine (SVM) algorithm in classifying thesis subjects stored in the Unila student scientific article repository. The application of the SVM algorithm uses the machine learning life cycle method which includes the data collection process, data pre-processing, data splitting, model training, to the model evaluation process. The data used are Unila student thesis titles totaling 1707 data. The results of this study are a thesis subject classification model based on the title with the SVM model where the accuracy of the training data in the model training process is 0.95, and the evaluation model process with an accuracy rate of 0.65.

Keywords: Artificial Intelligence, Classification, Dewey Decimal Classification, Machine Learning, Supervised Learning, Support Vector Machine.

1. Introduction

Artificial Intelligence (AI) or what is commonly called artificial intelligence is a branch of computer science that studies the field of smart machines to solve complex problems quickly to help human needs [1]. Some of the elements that characterize artificial intelligence include: there is an expert system, cultivate natural language, recognize human and machine language, having reason with robotics and sensors for logic, can interpreting images and objects via computer, can guide or tutor humans, and lastly facilitate deep learning.

Machine learning is a computer science that studies how to create computer programs that learn through experience [1]. Machine learning attempts to solve problems by establishing general rules for various problems, through the use of statistical techniques such as artificial neural networks. In making machine learning, there are two types of approaches, namely supervised learning and unsupervised learning. Supervised Learning is a technique that is commonly used to solve classification problems, generally learning the expected output from inputs that have not been identified by using learning from datasets that are already labeled in it [2].

Classification is a form of data analysis that extracts a model to describe the classification or category of data [3]. Text classification has two ways, namely text clustering and text classification. Text clustering aims to find an unsupervised group structure from a group of sentences in a document. Meanwhile, text classification aims to form groups of documents based on a previously known (supervised) group structure. Based on this explanation, it can be seen that there are several ways to perform text classification automatically, namely pre-processing, feature extraction/selection, choosing modeling using machine learning techniques, as well as training and testing on the classifier [4].

Research on classification based on supervised learning has done previously. Classification and comparison using six different machine learning algorithm with SVM resulted with highest accuracy followed by Random Forest [5]. Another research used SVM with RBF kernel to classified text with three attribute and resulted with 93,9% accuracy [6]. Based on the value of the results obtained, classification in previous studies proved effective in solving prediction problem.

Therefore, this study aims to use SVM for subject classification multi-label data from essay using Support Vector Machine (SVM). SVM is widely used because it has good performance in text classification that use multi-dimensional data. SVM is a learning system that uses a hypothetical space in the form of a linear function in a high-dimensional feature space, which is trained with a learning algorithm based on optimization theory by applying machine learning derived from statistical learning theory. The concept of classification with SVM is to find the best hyperplane to separate the two data classes and use a support vector approach. [7] There are three key ideas needed to understand SVM: maximizing margins, dual formulation, and kernels.

2. Materials and Methods

In this study, the subject classification process used the machine learning life cycle. This method has several stage of the research flow, as shown in figure 1

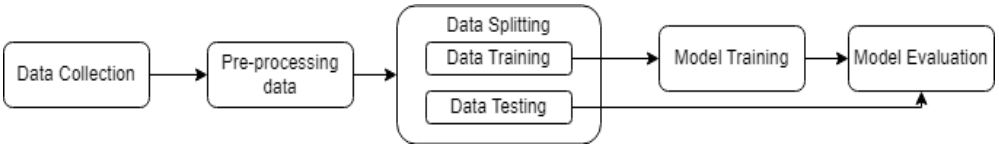

Figure 1. Research Flow

Based on figure 1, the thesis and subject is collected first from digilib.unila site. After the dataset is obtained, the following process is pre-processing data which includes data labelling, data cleaning, and feature extraction. After data is processed, the following process is data splitting. The SVM model is used in model training process by applying several architectures. In the final stage, SVM model is evaluated with confusion matrix to define model performance. A complete explanation will be given in the following sub-chapters.

2.1 Data collection

The dataset in this study is Lampung university student thesis title and it's subject. The subject is limited in DDC nine main category. The dataset was taken at July 30, 2022. The dataset has 4 attribute: nama, npm, judul, subject. The dataset has 1707 data and is used to predict subject. The sample of dataset can be seen in figure 2

Name	NPM	Tittle	Subject
GALEH WICAKSONO	1517031083	APLIKASI METODE DEKOMPOSISI ADOMIAN PADA PENYELESAIAN SISTEM PERSAMAAN DIFERENSIAL PARSIAL LI	natural sciences and mathematics
Lolyta Mutiara Putri	1817021045	GAMBARAN KERUSAKAN HISTOPATOLOGI JANTUNG PADA MENCIT YANG DIINDUKSI ALOKSAN DAN DENGAN P	natural sciences and mathematics
M Irfan Pratama	1817041035	PENGARUH VARIASI UKURAN BUTIR, KOMPOSISI SLAG SEBAGAI SUBTITUSI SEMEN, DAN BOTTOM ASH SEBAGA	natural sciences and mathematics
NABILA CITRA RAMADHANI	1513054047	Hubungan Pola Asuh Orang Tua dengan Keterampilan Sosial Anak Usia 5-6 Tahun.	natural sciences and mathematics

Figure 2 Dataset

This study uses nine subject based on amount of data in digilib.unila site. The distribution of the dataset is shown in table 1.

Table 1. Data usage of each subject

Digit	Category	Amount of Data
000	general works	310
100	philosophy and psychology	70
200	religion	12
300	social sciences	424
400	language	117
500	natural sciences and mathematics	240
600	technology	416
700	history, biography, and geography	96
900	the arts	22

2.2 Preprocessing Data

The pre-processing data is important because it improves accuracy and reliability. Data preprocessing is the process of processing raw data or raw data that is not useful as preparation for use in data processing [8]. This process involves activities such as data cleaning with the process being carried out is removing punctuation marks, stop-words, case folding, and the final process of data pre-processing is feature extraction.

Data cleaning works by selecting relevant data, redundant and noisy data will usually be deleted which will improve the quality of the input data. Data cleaning process for case folding is shown in figure 3, for removing punctuation marks and stop-words is shown in figure 4

```
In [11]: j=data['Judul'].apply(lambda i:i.lower())
         j
Out[11]: 0       aplikasi metode dekomposisi adomian pada penye...
```

Figure 3. case folding

Figure 3 is process of matching the title of all data into lower case letters. This needs to be done to reduce the amount of vocabulary that the computer needs to recognize.

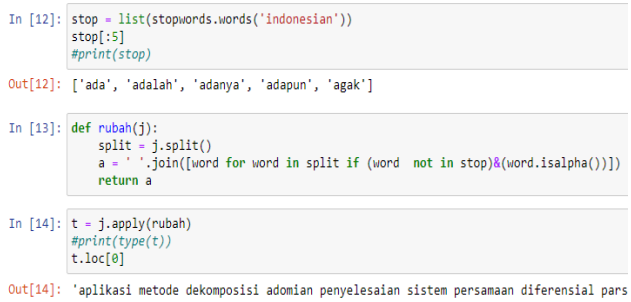

Figure 4. removing punctuation marks and stop-words

Figure 4 is the process of removing stop-words in Indonesian and removing numeric, so that any punctuation and numbers in the thesis title will be removed. Stop-words are conjunctions that appear frequently in textual language. Conjunctions cannot stand alone in a sentence because they have no meaning (for example, "and", "or", and so on in Indonesian), so they are not useful for classification [9]. This is done so that the computer will only focus on the words contained in each category. Comparison before and after the data cleaning process can be seen in the table 2

Table 2. Comparison before and after the data cleaning process

Before Lower Case Process	**After Lower Case Process**
APLIKASI METODE DEKOMPOSISI ADOMIAN PADA PENYELESAIAN SISTEM PERSAMAAN DIFERENSIAL PARSIAL LINIER HOMOGEN ORDE 1.	aplikasi metode dekomposisi adomian pada penyel esaian sistem persamaan diferensial parsial linier ho mogen orde 1.
Before Stopwords Removal	**After Stopwords Removal**
aplikasi metode dekomposisi adomian pada penyel esaian sistem persamaan diferensial parsial linier ho mogen orde 1.	aplikasi metode dekomposisi adomian penyelesaian sistem persamaan diferensial parsial linier homogen orde 1.
Before numeric removal	**After numeric removal**
aplikasi metode dekomposisi adomian penyelesaian sistem persamaan diferensial parsial linier homogen orde 1.	aplikasi metode dekomposisi adomian penyelesaian sistem persamaan diferensial parsial linier homogen orde

The final process of data pre-processing is feature extraction. Feature extraction refers to the process of transferring attribute operations, selecting attributes, subsets of attributes can be combined or can contribute to creating replacement attributes from data sets. To select attributes from data in the form of text, a method is used using TF-IDF (term frequency-inverse document frequency). TF-IDF is a method that integrates term frequency (TF) and inverse document frequency (IDF) which is useful for calculating the weight of each word used in a document so that you can find out how often a word appears in a document. TF is a value that indicates how often a word appears in a document. The more often the word appears the more important it is in the document. However, if a word appears too often in the corpus then the word is too common in that document. This is likened to the word "atom" which does not usually appear in normal documents, but this

word usually appears in physics documents. This situation will be resolved by the IDF [10]. To calculate the IDF, the commonly used formula is n/df, where n is the total number of documents that appear in the corpus. In the TF-IDF used by scikit considers n as n+1, which calculates to ((n+1))/df, and adds 1 as the final result [11].

$$TF = \frac{f_d(i)}{\max f_d(j)}$$

$$IDF = \log\left(\frac{N}{df(i)+1}\right)$$

Where:

fd (i) = number of terms (i) in document (j)

fd (j) = number of terms in the document (j)

N = number of documents in the corpus

df(i) = number of documents containing term (i), one is added if df(i) is not found in the corpus

The first stage of pre-processing data is starts with labeling the data subject using the Label-Encoder which will change the subject to a numeric. This stage is shown in figure 5.

```
In [16]: le = LabelEncoder()
         le.fit(list(data['Subject'].values))
         data['Subject'] = le.transform(list(data['Subject']))
```

Figure 5. Label-Encoder data subject

The second stage is Tf-IDF process, all sentences in the corpus or one data set will be split into words which are divided into columns. These words will then be calculated by weight or how important the word is in one corpus. This calculation will be done automatically by Tf-IDF. The results of this calculation are in numerical form which will be read by the program. This stage is shown in figure 6.

```
In [34]: vectorizer = TfidfVectorizer()
         vectors = vectorizer.fit_transform(t)
         #print(vectorizer.vocabulary_)
         vectors.shape
Out[34]: (1707, 4724)
```

Figure 6. feature extraction process with TF-IDF on title data

2.3 Data Splitting

The data that has become numeric after the previous feature extraction stage will then be split into two subsets, training data and test data. The training data is used for model training while the testing data is used to evaluate the model from the training model being trained. Most of the data will be used in the training data, this is done so that the computer gets a lot of data when carrying out the training process which will increase the accuracy of data predictions. To do a split, the data is divided by a ratio of 80:20, where 80% of the data is for training data while the rest is for testing data. The process and result are shown in figure 7

```
In [16]: X_train, X_test, y_train, y_test = train_test_split(vectors, data['Subject'], test_size=0.2)

In [36]: print (X_train.shape, y_train.shape, X_test.shape, y_test.shape)
         (1365, 4724) (1365,) (342, 4724) (342,)
```

Figure 7. data splitting and result

2.4 Model Training

With the data training and data testing has been obtained, the last step in making this classification model is to enter the model and do the testing. This model uses SVM from the Sklearn library with the kernel used is "rbf". The RBF kernel or also called Gaussian kernel is the most widely used kernel concept to solve linearly inseparable data classification problems. The RBF kernel locates a radial basis function centered at each point, then performs linear manipulation to map the points to higher-dimensional spaces that are more easily separable. The RBF kernel function equation is:

$$K(x, xi) = \exp\left(-\frac{\|x1 - x2\|^2}{2\sigma^2}\right)$$

Where σ is the variance of the hyperparameter, and $\|x1 - x2\|$ is the Euclidean rule which is the distance from x1 to x2. Exp is an exponential function in mathematics and has a value of 2.71828183.

The rbf kernel is used when more than two data labels are used, because this kernel is able to separate data that exceeds two labels, by repeating 1500 times, and the error tolerance is 1/10000, with a training duration of 0.6 seconds, as can be seen in Figure 8.

```
In [26]: svm=SVC(kernel='rbf',gamma='scale',max_iter=1500, tol=1e-4, verbose=True)
         start = time.time()
         svm.fit(X_train, y_train)
         stop = time.time()
         print(f"Training time: {stop - start}s")
         pred = svm.predict(X_test)
         #print (metrics.accuracy_score(y_test, pred))
         #print (metrics.precision_score(y_test, pred, average='macro'))
         #print (metrics.recall_score(y_test, pred, average='macro'))
         #print(svm.score)

         [LibSVM]Training time: 0.6052701473236084s
```

Figure 8. SVM model

2.5 Model Evaluation

Testing is done by evaluating the training model to determine the value of accuracy, precision, and recall through the confusion matrix. This process can also be done with the sklearn library, by importing the confusion matrix from sklearn.metrics like shown in figure 9.

```
In [35]: confusion_matrix(y_test, pred)

Out[35]: array([[16,  0,  0,  0,  6,  7,  0,  0, 18],
                [ 0,  0,  0,  0,  0,  2,  0,  0,  0],
                [ 0,  0, 20,  0,  1,  2,  0,  0,  1],
                [ 0,  0,  0, 16,  0,  0,  0,  0,  0],
                [ 5,  0,  2,  0, 29, 16,  0,  0, 18],
                [ 2,  0,  0,  1,  0, 63,  0,  0,  9],
                [ 0,  0,  0,  0,  0, 14,  4,  0,  2],
                [ 0,  0,  1,  0,  0,  6,  0,  0,  0],
                [ 0,  0,  1,  0,  2,  8,  0,  0, 70]], dtype=int64)
```

Figure 9. confusion matrix

To read the results of the confusion matrix with labels form the data title, we will use a heat map. On the heat map, there are already various thesis subjects that have previously been converted to integers with lable.encoder. This heat map can be seen in figure 10.

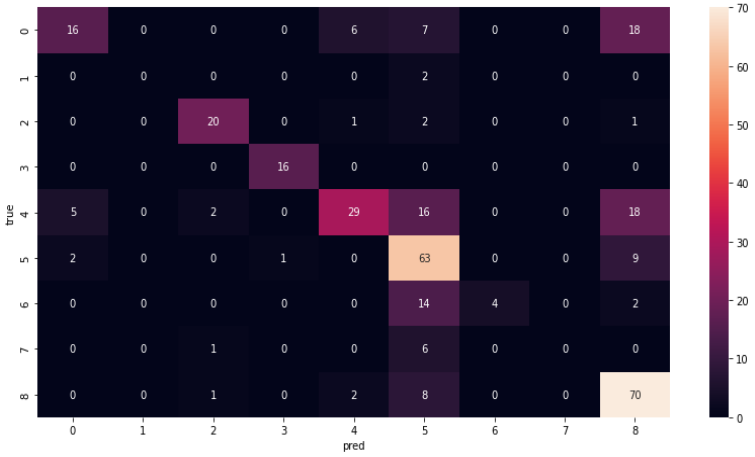

Figure 10. confusion matrix in heat map

3. Result

A total of 342 data from 20% of total dataset used to evaluate and with the confusion matrix accuracy, precision, recall, can be obtained in the form of a classification report in table 3 below.

Table 3. classification report

	precision	**recall**	**support**
natural sciences and mathematics	0.70	0.34	47
religion	0.00	0.00	2
language	0.83	0.83	24
philosophy and psychology	0.94	1.00	16
general works	0.76	0.41	70
social sciences	0.53	0.84	75
history, biography, and geography	1.00	0.20	20
the arts	0.00	0.00	7
technology	0.59	0.86	81
accuracy			0.64
avg / total	0.67	0.64	342

4. Discussion

The result from evaluating model with confusion matrix in table 3 is average precision 0,67, recall 0,64 and accuracy 0,64. Based on this result the amount of data used is very influential on the value of the model. As already seen in categories with a small number of support columns or units that do not have precision and recall values.

5. Conclusion

From the result of this study, it can be concluded that the Support Vector Machine algorithm can be used to classify text in the form of thesis titles. The final result of the model made at the model training stage using training data has an accuracy of 0.95, whereas at the model evaluation stage using data testing it obtains an accuracy of 0.65, a precision of 0.54, and a recall of 0.45. The amount of data used is very influential on the value of the model. The future research will propose preprocessing data using stop words that can recognize at least two languages, Indonesian and English, because among the data quite a lot of languages are used besides Indonesian also using algorithms other than SVM for classification text for comparison of the resulting accuracy levels.

References

[1] Mitchell, T.M., 1997. Machine learning. McGraw-hill. New York.
[2] Nasteski, V., 2017. An overview of the supervised machine learning methods. Horiz. B 4, 51–62.
[3] Han, J., Pei, J., Tong, H., 2022. Data mining: concepts and techniques. Morgan kaufmann.
[4] Dalal, M.K. and Zaveri, M.A., 2011. Automatic text classification: a technical review. International Journal of Computer Applications, 28(2), 37-40.
[5] Osisanwo, F.Y., Akinsola, J.E.T., Awodele, O., Hinmikaiye, J.O., Olakanmi, O. and Akinjobi, J., 2017. Supervised machine learning algorithms: classification and comparison. *International Journal of Computer Trends and Technology (IJCTT)*, *48*(3), 128-138.
[6] Octaviani, P.A., Wilandari, Y. and Ispriyanti, D., 2014. Penerapan Metode Klasifikasi Support Vector Machine (SVM) pada Data Akreditasi Sekolah Dasar (SD) di Kabupaten Magelang. Jurnal Gaussian, 3(4),811-820.
[7] Cristianini, N., Shawe-Taylor, J., 2000. An introduction to support vector machines and other kernel-based learning methods. Cambridge university press.
[8] García, S., Luengo, J. and Herrera, F., Data Preprocessing in Data Mining. Intelligent Systems Reference Library. 2015. doi, 10, 978-3.
[9] Dalal, M.K. and Zaveri, M.A., 2011. Automatic text classification: a technical review. International Journal of Computer Applications, 28(2), 37-40.
[10] Yoo, J.-Y., Yang, D., 2015. Classification Scheme of Unstructured Text Document using TF-IDF and Naive Bayes Classifier. Presented at the Computer and Computing Science 2015, 263–266
[11] Lavin, M., 2019. Analyzing documents with TF-IDF.