ISSN 2278-3091
Volume 9, No.5, September - October 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse29952020.pdf
https://doi.org/10.30534/ijatcse/2020/29952020

An Evaluation of Preprocessing Steps and Tree-based Ensemble Machine Learning for Analysing Sentiment on Indonesian YouTube Comments

A. S. Aribowo[1,2], H. Basiron[1], N. S. Herman[1], S. Khomsah[3]
[1] Center of Advanced Computing Technology (C-ACT), Fakulti Teknologi Maklumat dan Komunikasi Universiti Teknikal Malaysia Melaka (UTeM), Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia, sasmito.skom@upnyk.ac.id, halizah@utem.edu.my, nsuryana@utem.edu.my
[2] Department of Informatics, Universitas Pembangunan Nasional Veteran Yogyakarta, Indonesia
[3] Department of Data Science, Institut Teknologi TELKOM Purwokerto, Indonesia, siti@ittp-pwt.ac.id

ABSTRACT

This study aims to find the best model for the sentiment analysis of Indonesian YouTube video comments. Datasets crawled from YouTube video comments about government services related to COVID-19 pandemic in Indonesia. There are two opinion datasets obtained from two different domains, different characteristics, and errors. The problem is that comments from YouTube videos are very unstructured, containing spelling, diction, and slang word errors. The scenario for the solution of the problem is to test several preprocessing techniques, including standard preprocessing such as stop word removal, slang word, emoticon conversion, and stemming. Feature extraction using count vectorizer and TF-IDF method. For the development of the model, five types of models were tested, namely Naive Bayes (NB), Support Vector Machine (SVM), Decision Tree, Random Forest, and Extra Tree classifier. The result is a model with a maximum accuracy of 89.68% using a combination of standard preprocessing (converting emoticons and handling unstructured words), the count vectorizer feature extraction, and Extra Tree model classifier.

Key words: Feature extraction, machine learning, preprocessing, sentiment analysis.

1. INTRODUCTION

The use of YouTube as a video-based social media is growing rapidly in Indonesia, reaching 88% of 150 million users [1]. Youtube is used for political, economic, legal and human rights, social, education, and health news. YouTube provides collaboration between content creators (YouTubers), content consumers (viewers), and viewer interactions such as sharing videos, comments, and replies to video comments [2]. This study aims to find the model for the supervised learning sentiment analysis process on Indonesian YouTube video comment. This research is a continuation of previous research about text preprocessing model of Indonesian YouTube comments that only uses one machine learning [3]. The challenge for the comments is not a formal language. Users may express various spellings, dialects, and emotions in text form. Several types of errors are found such as spelling, diction, and grammatical errors.

This study uses two different datasets that are obtained from video topics in different domains. The first dataset is taken from the comments of Indonesian-language videos related to government programs about large-scale social restrictions (lockdown). The second dataset is taken from the comments of Indonesian videos related to the free electricity program. Both datasets are related to government programs to serve the public during the COVID-19 pandemic in Indonesia. The PSBB dataset consists of 4844 comments and 4862 comments for the free electricity program dataset. After being counted, there were 59% of opinion sentences in the first dataset and 70.9% in the second dataset contained slang words and incorrect spellings. Emoticons were found in 20% of comments, and there were lots of spelling, diction, and grammar mistakes too [4].

Sentiment analysis is a part of data mining for text processing which to be the blend of artificial intelligence, statistics, and machine learning [5]. Sentiment analysis uses supervised machine learning methods generally will go through four stages: the preprocessing stage, feature extraction, model development, and performance evaluation [6]. The selection of preprocessing methods, feature extraction, and the right machine learning is the main process to get a high-accuracy Indonesian YouTube dataset classifier model.

2. LITERATURE REVIEW

Some research on sentiment analysis and emotion analysis with YouTube video commentary datasets in Indonesian are experiments on emotion classification on Indonesian YouTube comments. The best performance is achieved by using the word embedding with the convolutional neural networks (CNN) method and show an accuracy rate of 76.2% [7]. Other studies are sentiment analysis on YouTube comments using SVM and show an accuracy rate of 84%, precision 91%, recall 80% [8]. The next research is the use of SVM for classify cyberbullying comments. This research shows the result of accuracy of 79.412% [9]. Other studies using SVM using the linear kernel function, show an accuracy of 62.76% [10], sentiment analysis using a combination of K-Nearest Neighbor and Levenshtein Distance, show the accuracy of 65.625% [11]. Other research on sentiment analysis uses Naive Bayes Classifier and Decision Tree Classifier and preprocessing using emoji deletion of punctuation removal, number correction of non-standard words, and POS-Tagging [12]. Similar studies using Multinomial Naive Bayes and show an accuracy of 95% on product review dataset [13], and research about combination of machine learning methods Naive Bayes and SVM, show an accuracy of 91%, F1 score of 87%, and recall of 83% [14]. Similar research on YouTube Movie Trailer uses Naive Bayes achieves results 81%, 74.83%, and 75.22% for accuracy, precision, and recall respectively [15]. It is hoped that the Naive Bayes method can achieve higher accuracy such as research for Twitter data [16]. The last study was emotion analysis for categorizing fanaticism using random forest, show an accuracy of 91.8% [17].

Based on the literature, it can be concluded that most of the previous studies used Naive Bayes and SVM [18]. Usually, SVM provides high accuracy [19]. So this study will use the Naive Bayes and SVM methods as a comparison of tree-based methods such as Decision Tree, Random Forest, and Extra Tree Classifier. We suspect these tree-based methods, especially ensemble methods such as Random Forest and Extra Tree can achieve high accuracy as resulted in the sentiment analysis research on the three Central Government Schemes in India [20].

The preprocessing method in previous research studies is case folding (lowercase conversion), removal of punctuation, and converting URLs, emoticon conversion, number removal, convert password, duplicate characters, and one character removal [7]. Other preprocessing methods used are tokenization, stopwords removal, and stemming which are carried out sequentially [14] [21]. Stopword removal and stemming have been investigated but not satisfactory [22]. So if the preprocessing method is improved, it needs to be re-examined the effect of each step of the preprocessing method and its contribution to the accuracy of the final classification. This study will examine in detail the effect of each preprocessing method on increasing the accuracy of classification results. The results are new models for sentiment analysis of Indonesian YouTube video comments with higher accuracy

3. METHODS

The stages of the study follow the standard steps of the process in sentiment analysis as in Figure 1.

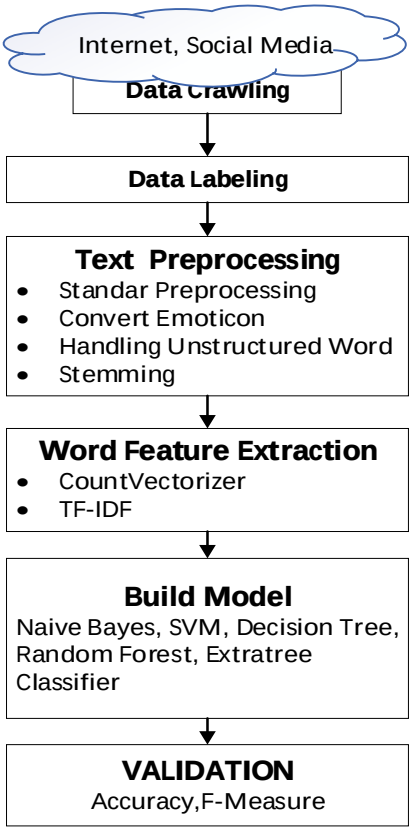

Figure 1: Research Steps

3.1 Crawling Data

The data crawled from YouTube video comments about community services related to Covid-19 in Indonesia. The dataset 1 contains the results of crawling netizens' comments on videos related to the PSBB (Large-scale Social Limitation policy), which was downloaded from March 2020 to April 2020. The dataset 2 contains comments on the videos relating to the provision of free electricity subsidies to the community also downloaded from March 2020 to April 2020. Both datasets have different characters so that the model provides more objective conclusions in processing various datasets.

3.2 Dataset Profile

From the two opinion datasets, the dataset profile is in Table 1 and Table 2. Both datasets are crawled from different sources

so they have different characteristics. In dataset 1, the number of comments containing slang word was 59.3%, in dataset 2 the number of comments containing slang word was 70.82%. Dataset 2 is more unstructured than dataset 1.

Table 1: Information of Dataset 1

Number of comments	4844
Number of comments with slang word	2873 (59.3%)
Number of words	77380
Number of Slangword	10521
Number of StopWord	16897

Table 2: Information of Dataset 2

Number of comments	4862
Number of comments with slang word	3444 (70.82%)
Number of words	83004
Number of Slangword	14191
Number of StopWord	17080

3.3 Labeling

In supervised learning, the label of datasets given by the experts [23]. The opinions in dataset 1 and dataset 2 are manually labeled by several experienced annotators. The two polarity labels are positive and negative sentiments. Labeling is done by observing the context of the whole sentence in the comment. If an opinion contains both positive and negative elements then the label is the dominant sentiment. From the results of the labeling process, dataset 1 contains 2462 positive comments and 2382 negative comments.

Dataset 2 contains 3211 negative comments and 1651 positive comments. In both datasets there is imbalanced data, a comparison of the amount of positive and negative data is not balanced. Imbalanced data can affect the results of classification. Oversampling or undersampling can be used to balance the number of positive and negative samples to be equal [24][25]. The ensemble algorithm can be operated in the imbalance dataset, for example the Random Forest algorithm and the extremely randomize tree (Extra tree). Naive Bayes, SVM, and Decision Trees will also be tested in classifying imbalanced data.

3.4 Preprocessing

Unstructured data in dataset 1 and dataset 2 needs to be preprocessed. There are several preprocessing steps referring to previous studies, namely:

A. Standard preprocessing (SP)

SP including case folding, remove number, remove non-alphabetic symbol, URL, and remove single char. Generally, this step is carried out in sentiment analysis and text mining in general.

B. Convert Emoticons (CE)

Convert emoticons symbol into words that represent them.

C. Handling Unstructured Words (HUW).

HUW consists of remove stopword, and convert slang word. If processes find a word that is in the stop words list then it is deleted. List of stop words using literary stopwords**.** Convert slangword to standard Indonesian words requires an Indonesian slang word dictionary compiled by researchers, containing 5436 slang words. The process begins with deleting repeated characters in sequence into one single character. Followed by removing nouns that are the subject or object in the sentence, for example, the names of figures, institutions, or names of objects that do not have sentiment elements such as: "jokowi", "pln", "COVID", "PSB", "maruf amin", "electricity","watt", etc. This process requires a subject dictionary that is compiled by researchers which contain 51 subjects that often appear in datasets in the domain being studied.

D. Stemming (ST)

ST is the process of converting affixed words into basic words. The process of stemming can eliminate meaning because a word can have different meanings after being affixed. So this study will also find out whether stemming is needed or not.

3.5 Feature Extraction

Preprocessing phase will change the unstructured data into semi-structured data so that the pattern can be extracted more easily. The dominant feature extraction in the positive and negative sentiment groups is carried out by two methods namely Count vectorizer and TF-IDF. These methods are familiar to machine learning and are proven to produce high accuracy up to 93% [26]. Count-Vectorizer (CV) used to convert a collection of sentences to a vector of terms / token counts. Every word in a sentiment group will be counted [27]. The dominant word that is leaning towards one type of sentiment will be a member of the sentiment group. The Count vectorizer method only considers word counts as feature values. This method does not reflect the dominance of the word in a sentence. The dominance of a word in a sentence is not only calculated on the number of occurrences in the sentence, but also the number of appearances on all opinions in a dataset [23]. This idea is referred to as "term-frequency inverse document frequency" (TF-IDF). Term frequency (TF) is the ratio of the number of occurrences of words in a sentence and IDF is the frequency of words to the whole opinion in the dataset [28].

3.6 Build The Model

There are five types of learning machines that will be used in model development. These five methods were chosen based

on the results of a literature review of previous studies. The ensemble method was chosen because it is believed to provide high accuracy as in previous studies [29].

A. *Naive Bayes (NB)*

NB used in previous studies [12][14][15]. Naive Bayes is already known as machine learning which is widely used in sentiment analysis and produces high accuracy. Bayes' rule is presented in (1).

$$P(c|X) = \frac{P(X|c)P(c)}{P(X)} \quad (1)$$

P (c | X) = Probability of c to be true if X is true
P (X | c) = Probability of X is true if c is true
P (c) = Probability c is correct
P (X) = Probability of X is true

The Bayes theorem is based on the statistics of probability and cost generated from the decision of the classification. NB is one of the simple implementations of Bayes theorem.

B. *Support Vector Machine*

SVM is a popular technique for classification. SVM used in this research is linear kernel. This technique attempts to find the most optimum separation function (hyperplane) to separate opinion data from different classes (positive and negative), or in this case called binary classes. The illustration of a hyperplane in SVM can be seen in Figure 2. SVM has the separation function that separates Class 1 and Class 2 effectively. The question is how does SVM find the optimum hyperplane. The trick is to find the outermost data in the two classes that are on the border, then find the optimum hyperplane considering the outer data.

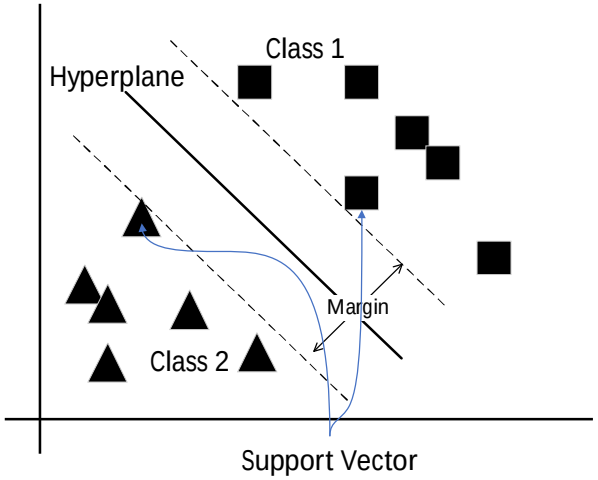

Figure 2: Support Vector Machine

C. *Decision Tree.*

Decision Tree is an algorithm that will arrange the features of the training data into a Decision Tree model for classification as in Figure 3. Decision tree branches are a classification question. The arrows are the values of the answers and the leaves are the classes. There are several algorithms used to construct the tree structure, namely CART, ID3, and C4.5. C45 algorithm is an algorithm that is developed from ID3 algorithm [30]. The C4.5 algorithm step in building a Decision Tree is to choose the main attribute as root, create a branch for each value, then divide the data into the branches that are created using entropy formula [31]. The process is repeated in all branches until all data in the branch has a homogeneous class. Decision Trees usually overfits the data it is learning from because it learns from only one pathway of decisions. Predictions from a single Decision Tree usually don't make accurate predictions on new data.

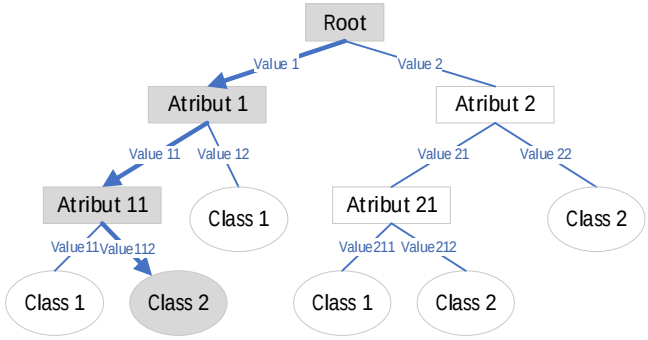

Figure 3: Decision Tree

D. *Random Forest (RF)*

RF is the development of a Decision Tree. Random Forest builds many trees in the same way as a Decision Tree. Random Forest reduces the risk of overfitting by introducing randomness by building multiple trees, bootstrapping, and splitting nodes using the best split among a random subset of features selected at every node. An example of the process of making many trees is in Figure 4. Each tree provides the results of classification. The final classification is the most classes produced from these trees (majority class).

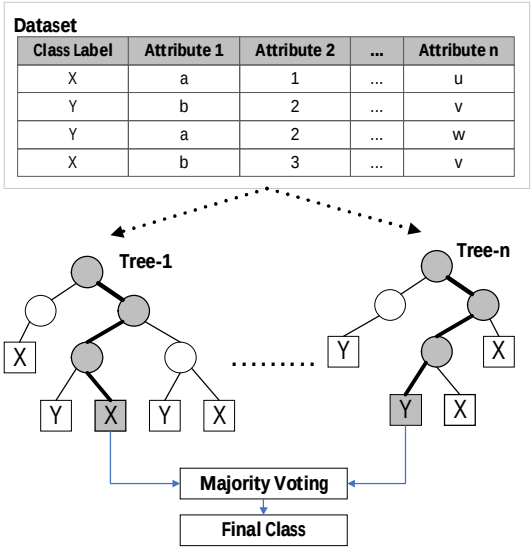

Figure 4: Random Forest

E. Extra Tree Classifier

Extra tree is almost like a Random Forest. The main process is to divide the dataset into clusters to build lots of trees and split nodes randomly, but with two differences when compared to Random Forest that is not using bootstrap (sample without replacement), and nodes are divided using random splits, not by the best split. In Extra Trees, randomness is not derived from bootstrap data, but rather comes from the random separation of all observations. The final classification is the majority class that results from these trees. Extra Trees is named for Extremely Randomized Trees.

3.7 Evaluation and Validation

For testing, the dataset is divided into 80% as training data and 20% as testing data. Training data will be used for modeling, then testing data for testing models. The model for each experiment will be tested for performance using the confusion matrix in Table 3. The confusion matrix compares the predicted results and the actual conditions of the machine learning model results. The label given by the annotator is the actual class, and the classification given by the model is the predicted class.

Table 3: Confusion Matrix

		Actual Class	
		Positive	**Negative**
Predicted Class	**Positive**	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

From the confusion matrix, accuracy and F-measure values will be obtained. Accuracy is a measure of how many actual class values are the same as the predicted class, the number of true positive (TP), and true-negative (TN). Accuracy is calculated using (2).

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \quad (2)$$

The second performance measure is the F-Measure. To calculate the F-Measure, you must calculate precision and recall. Precision is a measure of the accuracy of the classifier model in predicting positive sentiment values, what percentage of actual positive sentiment data is from all data predicted positive sentiment. Precision to be high if a few false positives (FP), and low if many false positives (FP) (3)

$$Precision = \frac{TP}{(TP+FP)} \quad (3)$$

Recall measures the sensitivity of the classifier model. The increase in recall value shows the ability of the model to map the actual value of positive sentiment, which is what percentage of data predicted positive sentiment from all data that is actually positive sentiment. Recall to be high if slightly false negative (FN) and low if more false negative (FN) (4).

$$Recall = \frac{TP}{(TP+FN)} \quad (4)$$

F-Measure is calculated by combining the value of Precision and Recall. The F-measure formula is in (5). A high f-Measure value indicates a high Precision and Recall value too.

$$f1 - measure = \frac{2.(Precision\ x\ Recall)}{(Precision + Recall)} \quad (5)$$

4. RESULT AND DISCUSSION

This research aims to find an efficient and high accuracy model. Testing on the four sentiment analysis processes begins with preprocessing, feature extraction, model development and evaluation. The preprocessing stage is tested in four stages, namely the preprocessing standard (SP), the preprocessing standard coupled with convert emoticons (CE), handling unstructured words (HUW), and stemming (ST). The test scenarios are shown in Table 4 for machine learning Naive Bayes, Table 5 for SVM, Table 6 for Decision Tree, Table 7 for Random Forest, and Table 8 for Extra Tree classifier.

Table 4 shows that preprocessing in the Naive Bayes model in dataset 1 increase performance from 73,3% to 76,13% or approximately 2.83% (F-Measure), and in dataset 2 increase from 79,23% to 83,67% or approximately 4.44% (F-Measure). The preprocessing step that give the most significant increase the accuracy is handling unstructured words (HUW), from 72,34% to 75,33% or approximately 2,9%. Convert emoticons (CE) decrease accuracy. Feature extractor and stemming not affect accuracy. Maximum of model accuracy is 76.13% for dataset 1 and 83.67% for dataset 2. Naive Bayes method is not affected by imbalance and noise which is more on dataset 2.

Table 4: The Performance of Naive Bayes Model (%)

No	**Testing Scenarios**		**Dataset 1**		**Dataset 2**	
	Pre-processing	**Feature Extraction**	**Accuracy**	**Fmeasure**	**Accuracy**	**Fmeasure**
1	SP	CV	72,44	**73,30**	79,24	**79,23**
2	SP+CE	CV	72,34	73,21	80,26	80,34
3	SP+CE+HUW	CV	75,33	75,99	82,94	**83,67**
4	SP+CE+HUW+ST	CV	75,64	**76,13**	82,73	83,26

No	Testing Scenarios		Dataset 1		Dataset 2	
	Pre-processing	Feature Extraction	Accu-racy	Fmea-sure	Accu-racy	Fmea-sure
5	SP	TF-IDF	72,45	73,30	79,24	79,23
6	SP+CE	TF-IDF	72,34	73,21	80,26	80,34
7	SP+CE+HUW	TF-IDF	75,33	75,99	82,94	83,67
8	SP+CE+HUW+ST	TF-IDF	75,64	76,13	82,73	83,26
		Average	73,94	74,66	81,29	81,63

Table 5 shows that the Support Vector Machine (SVM) model give maximum accuracy 86.27% for dataset 1 and 85.9% for dataset 2. The preprocessing increase the performance from 82,46% to 86,27% or around 3,81% in dataset 1 (accuracy), and from 81,65% to 85,9% or around 4.25% in dataset 2 (f-measure). The type of preprocessing that give the most significant increase the accuracy is handling unstructured words (HUW), from 81,22% to 85,86% or around 4,64%. Convert emoticons reduce accuracy. Stemming does not affect accuracy. A better feature extractor is Count Vectorizer. The SVM model works better on dataset 1 which is more balanced and less noise than dataset 2.

Table 5: The Performance of SVM Model (%)

No	Testing Scenarios		Dataset 1		Dataset 2	
	Pre- processing	Feature Extraction	Accu-racy	Fmea-sure	Accu-racy	Fmea-sure
1	SP	CV	**81,73**	**81,74**	**81,20**	**81,65**
2	SP+CE	CV	81,01	81,03	80,88	81,50
3	SP+CE+HUW	CV	84,93	84,93	83,76	83,99
4	SP+CE+HUW+ST	CV	83,79	83,80	**84,69**	**85,90**
5	SP	TF-IDF	**82,46**	**82,48**	78,72	80,69
6	SP+CE	TF-IDF	81,22	81,23	78,52	80,43
7	SP+CE+HUW	TF-IDF	85,86	85,86	81,70	82,98
8	SP+CE+HUW+ST	TF-IDF	**86,27**	**86,27**	81,60	82,76
		Average	83,41	83,42	81,38	82,49

Table 6 shows that the Decision Tree model achieves maximum accuracy 85.34% for dataset 1 and 82.63% for dataset 2. The preprocessing increase the performance from 82,05% to 85,34% or around 3.3% in dataset 1 (accuracy and F-Measure), and from 76,77% to 82,63% or around 5,86% in dataset 2 (accuracy).

Table 6: The Performance of Decision Tree Model (%)

No	Testing Scenarios		Dataset 1		Dataset 2	
	Pre- processing	Feature Extraction	Accu-racy	Fmea-sure	Accu-racy	Fmea-sure
1	SP	CV	81,42	81,42	76,57	76,58
2	SP+CE	CV	82,77	82,77	77,08	77,09
3	SP+CE+HUW	CV	84,41	84,41	81,09	81,14
4	SP+CE+HUW+ST	CV	84,82	84,83	82,43	82,52
5	SP	TF-IDF	82,05	82,04	76,77	76,80
6	SP+CE	TF-IDF	82,87	82,87	75,85	75,72
7	SP+CE+HUW	TF-IDF	85,34	85,34	81,91	81,82
8	SP+CE+HUW+ST	TF-IDF	84,93	84,94	82,63	82,55
		Average	83,58	83,58	79,29	79,28

For the preprocessing step, the highest accuracy is handling of unstructured word (HUW) which increase the accuracy from 75,85% to 81,91%, or 6,6%. Convert emoticons slightly increase accuracy, but stemming does not significantly increase accuracy, even in dataset 1 decreases accuracy. A good extraction feature is TF-IDF. The imbalance and noisy dataset 2 condition reduces accuracy.

Table 7 shows that the Random Forest model gives maximum accuracy of 89.27 for dataset 1 and 88.71 for dataset 2. The preprocessing performance is 4,55%, increase from 84,72% to 89,27% in dataset 1 (accuracy and F-Measure), and from 81,28% to 88,71% or around 7,43% in dataset 2 (accuracy). Handling unstructured word (HUW) increase the accuracy from 84,11% to 89,6%. Convert emoticons (CE) reduce accuracy but stemming slightly increases accuracy. Feature extraction does not affect accuracy results. Imbalance and noise in dataset 2 reduce accuracy.

Table 7: Random Forest Model Performance (%)

No	Testing Scenarios		Dataset 1		Dataset 2	
	Pre- processing	Feature Extraction	Accu-racy	Fmea-sure	Accu-racy	Fmea-sure
1	SP	CV	**84,72**	**84,72**	81,29	81,84
2	SP+CE	CV	84,31	84,31	81,19	81,69
3	SP+CE+HUW	CV	88,75	88,75	85,10	85,16
4	SP+CE+HUW+ST	CV	**89,27**	**89,27**	86,33	86,51
5	SP	TF-IDF	85,14	85,14	81,29	81,86
6	SP+CE	TF-IDF	84,11	84,11	79,75	80,49
7	SP+CE+HUW	TF-IDF	89,06	89,06	83,14	83,41
8	SP+CE+HUW+ST	TF-IDF	88,75	88,75	**88,71**	**85,90**
		Average	86,76	86,76	83,35	83,36

Table 8 shows that the Extra Tree model gives a maximum accuracy of 89.68% for dataset 1 and 86.2% for dataset 2

Table 8: The Performance of Extra Tree Model (%)

No	Testing Scenarios		Dataset 1		Dataset 2	
	Pre- processing	Feature Extraction	Accu-racy	Fmea-sure	Accu-racy	Fmea-sure
1	SP	CV	**86,40**	**86,40**	**82,20**	**82,70**
2	SP+CE	CV	87,31	87,31	83,14	83,53
3	SP+CE+HUW	CV	**89,68**	**89,69**	86,20	86,17
4	SP+CE+HUW+ST	CV	89,06	89,09	**86,43**	**86,67**
5	SP	TF-IDF	**85,60**	**85,60**	**81,50**	**82,10**
6	SP+CE	TF-IDF	**85,66**	85,67	81,81	82,45
7	SP+CE+HUW	TF-IDF	**88,75**	88,77	84,69	84,96
8	SP+CE+HUW+ST	TF-IDF	88,65	88,67	84,99	85,28
		Average	88,19	88,20	84,54	84,84

Preprocessing steps improve performance from 86,4% to 89,68% or around 3.2% in dataset 1 (accuracy and F-Measure), and from 82,2 to 86,4 or around 4% on Dataset 2 (accuracy). Handling unstructured word (HUW) increase the accuracy form 85,66% to 88,75% or around 3,09%. Convert emoticons slightly increase accuracy. Stemming does not affect accuracy, tends to reduce accuracy in dataset 1. The right type of feature extraction is the count vectorizer and the accuracy results are influenced by the condition of dataset 2 which is noisier.

The accuracy summary (average) of each type of machine learning (from the five tables above) is shown in Table 9. In dataset 1 and dataset 2 the machine learning method that provides the best classification is Extra Tree Classifier, then followed by Random Forest. In dataset 1, the worst method is Naive Bayes, and dataset 2 is the Decision Tree. For more details, the comparison of accuracy among five machine learning classifiers is shown in Figure 5.

Table 9: Average of Machine Learning Performance (%)

Machine Learning	Dataset 1		Dataset 2	
	Accuracy	FMeasure	Accuracy	FMeasure
Naive Bayes	**73,94**	**74,66**	81,29	81,63
SVM	83,41	83,42	81,38	82,49
Decision Tree	83,58	83,58	**79,29**	**79,28**
Random Forest	86,76	86,76	83,35	83,36
Extra Tree	**88,19**	**88,20**	**84,54**	**84,84**

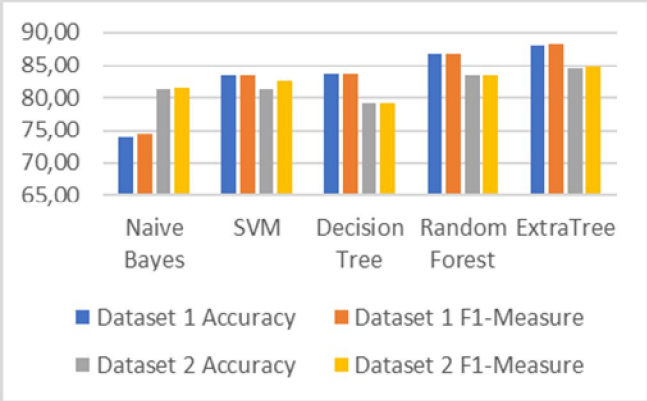

Figure 5: The Comparison of Machine Learning Model

5. CONCLUSION

Some conclusions from the results of the study are preprocessing methods such as handling unstructured words, clear stop words, convert slang words, and word corrections important in cleaning Indonesian YouTube opinions. The best machine learning model is the ensemble method, Extra Tree, and Random Forest. The best model can be developed using a combination of standard preprocessing, converting emoticons, handling unstructured words, then feature extractor using Count Vectorizer and machine learning using Extra Tree classifier.

ACKNOWLEDGEMENT

The authors would like to thank the Computational Intelligence and Technologies laboratory (CIT Lab) research group, the Center of Advanced Computing Technology (C-ACT), Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka (UTeM) for their incredible support for this research.

REFERENCES

[1] T. Redaksi, "**Youtube, medsos no. 1 di Indonesia,**" *Katadata.co.id*, 2019. https://katadata.co.id/infografik/2019/03/06/youtube-medsos-no-1-di-indonesia (accessed Jul. 17, 2020).

[2] E. H. Poche, "**Analyzing user comments on YouTube coding tutorial videos**," Louisiana State University and Agricultural and Mechanical College, 2017.

[3] S. Khomsah and A. S. Aribowo, "**Model text-preprocessing komentar Youtube dalam bahasa Indonesia**," *Rekayasa Sistem dan Teknologi Informasi*, *RESTI*, vol. 4, no. 4, pp. 648–654, 2020.

[4] S. K. H. Sebayang and A. S. Sofyan, "**Analisis kesalahan berbahasa pada sosial media Instagram dalam postingan, komentar, dan cerita singkat**," *Jurnal Serunai Bahasa Indonesia*, vol. 16, no. 1, pp. 49–57, 2019, doi: 10.37755/jsbi.v16i1.124.

[5] S. S. Khan, "**Soft computing- a journey from statistics,**" *International Journal of Theoretical &*

Applied Sciences, vol. 9, no. 2, pp. 260–268, 2017.

[6] P. Chauhan, N. Sharma, and G. Sikka, "**The emergence of social media data and sentiment analysis in election prediction,**" *Journal of Ambient Intelligence and Humanized Computing*, no. March, 2020, doi: 10.1007/s12652-020-02423-y.

[7] J. Savigny and A. Purwarianti, **"Emotion classification on Youtube comments using word embedding**," in *International Conference on Advanced Informatics: Concepts, Theory and Applications*, 2017, pp. 1–5, doi: 10.1109/ICAICTA.2017.8090986.

[8] F. I. Tanesab, I. Sembiring, and H. D. Purnomo, "**Sentiment analysis model based on Youtube comment using support vector machine**," *International Journal of Computer Science and Software Engineering (IJCSSE)*, vol. 6, no. 8, pp. 180–185, 2017, [Online]. Available: http://ijcsse.org/published/volume6/issue8/p2-V6I8.pdf .

[9] M. Andriansyah *et al.*, "**Cyberbullying comment classification on Indonesian selebgram using support vector machine method,**" in *The 2nd International Conference on Informatics and Computing*, 2018, vol. 2018-Janua, pp. 1–5, doi: 10.1109/IAC.2017.8280617.

[10] E. Rinaldi and A. Musdholifah, "**FVEC-SVM for opinion mining on Indonesian comments of youtube video,**" *Proceedings of 2017 International Conference on Data and Software Engineering, ICoDSE 2017*, vol. 2018-Janua, pp. 1–5, 2018, doi: 10.1109/ICODSE.2017.8285860.

[11] N. Anggraini and M. J. Tursina, "**Sentiment analysis of school zoning system on Youtube social media using the K-nearest neighbor with levenshtein distance algorithm,**" in *7th International Conference on Cyber and IT Service Management*, 2019, no. May, pp. 1–4, doi: 10.1109/CITSM47753.2019.8965407.

[12] R. A. Maisal, A. N. Hidayanto, N. F. Ayuning Budi, Z. Abidin, and A. Purbasari, "**Analysis of sentiments on Indonesian YouTube video comments: case study of the Indonesian government's plan to move the capital city**," in *1st International Conference on Informatics, Multimedia, Cyber and Information System*, 2019, pp. 121–124, doi: 10.1109/ICIMCIS48181. 2019.8985228.

[13] M. P. Abraham and K. R. Udaya Kumar Reddy, "**Feature based sentiment analysis of mobile product reviews using machine learning techniques**," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 2, pp. 2289–2296, 2020, doi: 10.30534/ijatcse/2020/210922020.

[14] A. N. Muhammad, S. Bukhori, and P. Pandunata, "**Sentiment analysis of positive and negative of YouTube comments using naïve bayes-support vector machine (NBSVM) classifier**," in *International Conference on Computer Science, Information Technology, and Electrical Engineering*, 2019, vol. 1, pp. 199–205, doi: 10.1109/ICOMITEE.2019.8920923.

[15] R. Novendri, A. S. Callista, D. N. Pratama, and C. E. Puspita, "**Sentiment analysis of YouTube movie trailer comments using naïve bayes,**" *Bulletin of Computer Science and Electrical Engineering*, vol. 1, no. 1, pp. 26–32, 2020, doi: 10.25008/bcsee.v1i1.5.

[16] E. Kannan and L. A. Kothamasu, "**A Pattern based approach for sentiment analysis using ternary classification on twitter data,**" *International Journal on Emerging Technologies*, vol. 11, no. 2, pp. 811–816, 2020.

[17] A. S. Aribowo, H. Basiron, N. S. Herman, and S. Khomsah, "**Fanaticism category generation using tree-based machine learning method**," *Journal of Physics: Conference Series*, vol. 1501, no. 1, 2020, doi: 10.1088/1742-6596/1501/1/012021.

[18] Z. Drus and H. Khalid, "**Sentiment analysis in social media and its application: systematic literature review**," *Procedia Computer Science*, vol. 161, pp. 707–714, 2019, doi: 10.1016/j.procs.2019.11.174.

[19] N. Sultana and M. M. Islam, "**Meta classifier-based ensemble learning for sentiment classification,**" in *Proceedings of International Joint Conference on Computational Intelligence, e, Algorithms for Intelligent Systems*, 2020, vol. 669, pp. 1–481, doi: 10.1007/978-981-13-7564-4.

[20] E. Sujatha and R. Radha, "**A sentiment classification on Indian government schemes using PySpark**," *International Journal on Emerging Technologies*, vol. 11, no. 2, pp. 25–30, 2020.

[21] M. Christianto, J. Andjarwirawan, and A. Tjondrowiguno, "**Aplikasi analisa sentimen pada komentar berbahasa Indonesia dalam objek video di website YouTube menggunakan metode Naïve Bayes classifier**," *Jurnal Infra*, vol. 8.1, pp. 255–259, 2020.

[22] A. W. Pradana and M. Hayaty, "**The effect of stemming and removal of stopwords on the accuracy of sentiment analysis on Indonesian-language texts**," *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, vol. 4, no. 4, pp. 375–380, 2019, doi: 10.22219/kinetik.v4i4.912.

[23] L. B. Shyamasundar and P. Jhansi Rani, "**A multiple-layer machine learning architecture for improved accuracy in sentiment analysis**," *The Computer Journal*, vol. 63, no. 3, pp. 395–409, 2019, doi: 10.1093/comjnl/bxz038.

[24] N. Cahyana, S. Khomsah, and A. S. Aribowo, "**Improving imbalanced dataset classification using oversampling and gradient boosting**," *5th International Conference on Science in Information Technology: Embracing Industry 4.0: Towards Innovation in Cyber Physical System, ICSITech*, pp. 217–222, 2019, doi: 10.1109/ICSITech46713.2019.8987499.

[25] D. Tiwari and N. Singh, "**Ensemble approach for twitter sentiment analysis,**" *I.J. Information Technology and Computer Science*, no. August, pp.

20–26, 2019, doi: 10.5815/ijitcs.2019.08.03.

[26] F. Ahmad and R. Lokeshkumar, "**A comparison of machine learning algorithms in fake news detection**," *International Journal on Emerging Technologies*, vol. 10, no. 4, pp. 177–183, 2019.

[27] S. Kaur, P. Kumar, and P. Kumaraguru, "**Automating fake news detection system using multi-level voting model,**" *Soft Computing*, vol. 24, no. 12, pp. 9049–9069, 2020, doi: 10.1007/s00500-019-04436-y.

[28] T. Wang, K. Lu, K. P. Chow, and Q. Zhu, "**COVID-19 sensing : negative sentiment analysis on social media in China via BERT model**," *IEEE Access*, vol. 4, 2020, doi: 10.1109/ACCESS.2020.3012595.

[29] Y. Sahu, G. S. Thakur, and S. Dhyani, "**Dynamic feature based computational model of sentiment analysis to improve teaching learning system**," *International Journal on Emerging Technologies*, vol. 10, no. 4, pp. 17–23, 2019.

[30] M. Akhtar and R. S. Parihar, "**An hybrid data mining approach to detection and classification of health care data**," *International Journal of Electrical, Electronics and Computer Engineering*, 2017.

[31] A. K. Mohamad, M. Jayakrishnan, and N. H. Nawi, "**Employ twitter data to perform sentiment analysis in the Malay language**," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 2, pp. 1404–1412, 2020, doi: 10.30534/ijatcse/ 2020/76922020.