Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.11894
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Comparison of the Sentiment Analysis Model's Code Complexity and Processing Time

(Sentiment analysis for tolerance and religious moderation in Indonesia: A Case Study)

Luh Gede Surya Kartika[1)*], Putu Kussa Laksana Utama[2)], I Dewa Gede Budiastawa[3)], Komang Rinartha[4)]
[1,2,3)]Universitas Hindu Negeri I Gusti Bagus Sugriwa Denpasar, Indonesia
[4)]institut Teknologi dan Bisnis STIKOM Bali, Indonesia
[1)]suryakartika@uhnsugriwa.ac.id, [2)] kussa_laksana@uhnsugriwa.ac.id, [3)] budiastawa@uhnsugriwa.ac.id,
[4)]komangrinartha@gmail.com

Submitted : Nov 1, 2022 | **Accepted** : Nov 10, 2022 | **Published** : Jan 1, 2023

Abstract: Text processing, which includes sentiment analysis, obviously demands a lot of resources. The extent to which resources are used to promote environmental conservation is directly impacted by code complexity or computational time. Twitter users' responses on the Indonesian Language of religious moderation and tolerance are used to test the model. The phases of doing the research include determining the research's needs, gathering data, text preprocessing (case folding, tokenizing, stopword removal, and stemming), word weighting with TF/IDF, and classification with Multinomial Naive Bayes (MNB) and Support Vector Machine (SVM), validation, and the last step was calculation of the code complexity and computation time. The validation results showed that the performance of the two models was still low, with an average accuracy of 75.5%. Based on computation time, SVM has a faster computational time than MNB. However, when compared in terms of the code complexity, the Cyclomatic Complexity in both models was the same because both models used existing libraries in Python Interpreter, and the complexity of the libraries cannot be calculated directly. Based on the Raw Metrics, it can be seen that MNB and SVM not significantly different in LOC, LLOC, and SLOC. It was evident that SVM has a greater Halstead Complexity than SVM in all measures when comparing the program code of MNB and SVM. The programming effort metric revealed that the amount of mental effort needed to convert the SVM algorithm into a program is also 13.123 times greater than the MNB, and the results of the volume metric revealed that the number of bits needed to execute the SVM program is 10.75 times greater than the MNB.

Keywords: Algorithm Comparison, Sentiment Analysis, Program Code Complexity, Computing Time

INTRODUCTION

Analysis of the sentiment is the use of text analysis techniques to understand and classify emotions (positive, negative, and neutral) contained in a text (S. Lestari & Saepudin, 2021) with the help of a computer. The text data analyzed generally come from natural languages, so sentiment analysis can also be said to be part of natural languages processing (NLP) or natural language processing by computers. Data sources commonly used for sentiment analysis are collected through comments on digital spaces such as social media, digital marketplaces, and so on. The data can then be used by companies, governments, and other entities for decision making, market analysis, product reviews, product feedback, quality of public service, and other decision-making needs.

Big data that includes text is used for sentiment analysis. As a result, it is necessary to evaluate resource utilization during the categorization step. Writing efficient program code helps minimize memory use and CPU cycles. Green Software Development is the idea (Calero & Piattini, 2015). One goal of the green software development approach is to create software that, when used, uses a tiny amount of memory, bandwidth, and CPU power and takes up little space when installed (Mala & Ganesan, 2013).

Research on green software development in the case of sentiment analysis has not yet been found. Similar studies that have been carried out by previous researchers generally discuss the negative class, positive class,

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.11894
e-ISSN : 2541-2019
p-ISSN : 2541-044X

recall, and accuracy generated by the model. No research has been found that discusses the complexity of the program code or the computational time of the sentiment classification model. Both of these directly affect the level of use of resources that support environmental conservation. According to Malhotra (2015) a study on Cyclomatic Complexity which is related to the number of defects that appear in software, this metric shows the level of complexity of the software. While the computational time shows the time it takes a computer program to execute the program code.

The development of sentiment analysis studies is holistic and fluctuating because of the dependence on the intensity of technology use. Based on Singgalen (2021), a study on sentiment analysis on social media in 2021 discusses the topic of the Covid-19 pandemic, restrictions on social activities in various countries during the pandemic, as well as the conversion of social, cultural, economic, political and environmental issues that are being digitally campaigned as a result of the pandemic. presence of Covid-19. Other problems such as religious issues, and education are still not widely carried out by researchers (Singgalen, 2021). Furthermore, social media is both an agent and a medium for fighting between various perceptions of the truth. Social media produces dangerous narratives that can lead to violence, both physical and symbolic. The violence was exacerbated by making ethnicity, religion, race, inter-group a commodity in electoral politics so that there was polarization in society.

One of the efforts made to overcome the problem of intolerance is to promote religious moderation as an approach to understanding and practicing religion. The urgency of religious moderation is getting higher in the era of digital society where everything is connected via the internet. In addition, Indonesia is also a global model of religious moderation this is the basis for selecting cases in this study, namely the issue of religious moderation in Indonesia which consists of issues regarding facilities , quality, policy, and so on.

This problem is the background of this research, which is to compare two algorithms in terms of Cyclomatic Complexity and computational time to analyze public sentiment regarding the issue of religious moderation in Indonesia. The algorithm chosen is Multinomial Naive Bayes and Support Vector Machine. Both algorithms are commonly used in the analysis of public sentiment(Arsi & Waluyo, 2021; Huda Ovirianti et al., 2022; Laurensz & Sediyono, 2021; Sodik & Kharisudin, 2021; Yusliani et al., 2022). Both algorithms have been proven to be able to perform sentiment analysis with good accuracy. This is the reason for choosing the algorithm to be compared.

LITERATURE REVIEW

Big data is used in Natural Language Processing, which includes sentiment analysis. Large amounts of resources (CPU and RAM) are undoubtedly needed for big data processing, and less computer code will execute faster. The code Complexity and its computation time show the resource requirements of a computer program.Numerous earlier scholars have conducted research on sentiment analysis, demonstrating how significant and beneficial this study is to society. Unstructured data can be converted into much more structured data using sentiment analysis to understand public sentiment toward many issues, including politics, companies, brands, and services. This serves as the foundation for choosing study subjects. The findings from the earlier studies are summarized as follows:

Tabel 1 Similar research by other researchers

Source	Case	Algorithm	Testing topics
(Laurensz & Sediyono, 2021)	Twitter users' sentiments on vaccination.	Support Vector Machine And Naïve Bayes	**Accuracy** of both algorithms on multiple keywords
(Sodik & Kharisudin, 2021)	Indonesian Twitter users' responses to the Covid-19 pandemic	K-nearest Neighbour, Support Vector Machine, and Naïve Bayes	The comparison of those algorithm
(Apriani & Gustian, 2019)	Comments on the Tokopedia app	Naïve Bayes	The ***class negative, class positive, recall,*** *dan* ***Accuracy*** on sentiment analysis
(K. F. Lestari & Lazuardi, 2018)	Twitter user sentiment regarding the MR immunization campaign in Indonesia.	K-Nearest Neighbour Using Rapidminer 8.1	The balance of training data and test data
(Gunawan et al., 2018)	Indonesian online product reviews	Naïve Bayes	The ***class negative, class positive, recall,*** *dan* ***Accuracy*** on sentiment analysis
(Sari &	Jd.Id Online Store Customers	Naïve Bayes	The comparison of ***class***

*name of corresponding author

Source	Case	Algorithm	Testing topics
Wibowo, 2019)			***negative, class positive, recall,*** *dan* ***Accuracy*** between comment with or without icon
(Fitri et al., 2020)	Sentiment on the Ruangguru app	Naive Bayes, Random Forest, and Support Vector Machine	The comparison of those algorithm
(S. Lestari & Saepudin, 2021)	Sinovac vaccine sentiment on Twitter	Algoritma Naïve Bayes	The **accuracy** of result.

Based on Table 1, it is clear that studies of a comparable nature conducted in the past have typically included the negative class, positive class, recall, and accuracy produced by the test. There is no literature on code complexity or computation time for model of sentiment analysis. These two factors contribute to the idea of "green software development" and have an impact on how much resources are used. In terms of Information and Communication Technology, the development of green program codes will aid in environmental preservation (Calero & Piattini, 2015).

METHOD

Requirement analysis was completed initially, then the stage of Natural Language Processing. Case selection is a crucial factor that will impact the reliability of the findings(Singgalen, 2021). Public perceptions of religious moderation on the social media site Twitter served as the research's data source. The words "*moderasi beragama*," and "*toleransi beragama*" were utilized as keywords in Bahasa Indonesia.

First step – The data collection. The first stage was data collection. This stage consists of two processes: (a) The crawling process, was the process where we take data from Twitter to be used as training data and test data. (b) Labeling, the process of giving a positive, negative, or neutral label to the data that has been obtained from the crawling process.

Second – Cleaning the data. To be processed, the data needs to be cleaned first so that good and consistent results are obtained. (a) The first process was case folding. All letters contained in the document converted to lowercase, as well as removing punctuation characters such as "?;/.," and so on. (b) The next process is stopword removal. In this process, all words that have no meaning, such as the words in Indonesian Languanges‘dan’, ‘di’, ‘oleh’, ‘yang’ will be removed from the document, leaving only meaningful words in the document. (c) Next, the last process in the text preprocessing stage is the stemming process. All words contained in the document converted into its basic form by removing the affixes and suffixes that are closest to the word. Stemming was done to overcome the presence of unusual words as well as to group other words that have the same basic form.

Third - After going through the entire process of cleaning the data, then the number of occurrences in the document is calculated. In the process, the Term Frequency-Inverse Document Frequency (TF-IDF) weighting method will be used.

The fourth stage was the classification stage. Clean data will be processed using Multinomial Naive Bayes (MNB) and Support Vector Machine (SVM) models. In this stage the machine will be taught to recognize the existing text data patterns and then be able to classify the data into two classes, namely the positive class and the negative class. The model is evaluated through the results of precision, accuracy, recall, f1-measure, and support.

Accuracy is how close the predicted results (observations or readings) to the true values. Recall is the ratio of TP / (TP + FN) where recall represents the ratio of the number of true positives (TP) to the number of false negatives (FN) and the number of true positives (TP). Intuitively, recall is the ability of SVM and MNB to find all positive samples. (Sklearn.Metrics.Recall_score — Scikit-Learn 1.1.2 Documentation, n.d.). f1 score is the average between precision and recall. In F1, both recall and precision made the same relative contribution. The formula for the f1 score is: f1 = 2 * (precision * recall) / (precision + recall). Support is the number of each class in y_true (y-true is the target value of ground truth). The best value of accuracy, precission, recall, f1-measure, and support is represented by 1 and the worst value is 0 (*Sklearn.Metrics.Precision_recall_fscore_support — Scikit-Learn 1.1.2 Documentation*, n.d.)

The final stage of this research was the comparison of the complexity of the source code. The metrics used in the analysis of the program code are Raw, Cyclometic Complexity, Halstead, and Maintainability Index. Radon has been designed to take into account all of these characteristics. Radon is a library that can compute numerous metrics from the source code of an application.

The tools used for this research are as follows: Python 3 interpreter for making all instructions (source code) in all stages of research, the developer's twitter account will be used for crawling data on Twitter, Microsoft Excel 2016, used for labeling text data, and Computer with CPU 2.00GHz and 4GB RAM used for compute time testing.

*name of corresponding author

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.11894
e-ISSN : 2541-2019
p-ISSN : 2541-044X

RESULT

The results of this study will be presented sequentially, starting with data collection and text preprocessing and ending with the classification and performance findings, to make it easy to see the findings.

Data Collection and Text Preprocessing Process

The first stage after the requirements are known, was data collection which is also known as data crawling. The Twitter data crawl process was carried out on April 30, 2022 with 3 topic: moderation, religion, and tolerance. From the keywords, 533 data/sentences of Twitter users' tweets were obtained. Labeling is done manually by analyzing the meaning of the sentence, and devided data into three categories: positive emotions, negative emotions, and neutral. The challenge when doing labeling is determining the emotion in the sarcasm sentence. In addition, most of the sentences that were crawled by Twitter data were truncated or incomplete, so that the meaning of the sentences was difficult to identify. To ensure accuracy, the labeling results have been verified by Indonesian language experts. In summary, the recapitulation of labeling results is shown in Table 2.

Tabel 2 Recapitulation of Labeling Results

Sentences containing emotions	Numbers	Percentage
Positive	255	47,842%
Negative	231	43,340%
Neutral	47	8,818%
Total	533	

Wordcloud in Fig. 1 and Fig. 2 help us to recognize the words that appear in the data. then presented a visual image displayed on the worldcloud allowed to quickly and practically capture the important essence of the data to be analyzed(Qeis, 2015). Figure 1 and Figure 2 show that the same words appear in the negative and positive classes. That is, for sentences with negative emotional labels also contain positive words (sarcasm). Furthermore, data with a neutral label will not be used for further analysis because it can reduce the accuracy of the results.

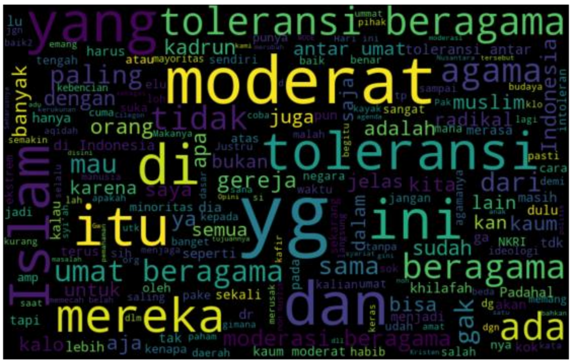

Fig. 1 Words that appear in sentences with a negative label

Fig. 2. Words that appear in sentences labeled positive

The next process was text preprocessing which consists of processes, notably case folding, stopword removal, and stemming. The stemming process used the literary Sastrawi.Stemmer.StemmerFactory and the stopword removal process used the literary Sastrawi.StopWordRemover.StopWordRemoverFactory. From the results of research by Rosid et al.(2020), the Sastrawi library were able to reduce over stemming and under stemming that often occur in Indonesian texts. The Sastrawai library also required a faster processing time than using a Tala stemmer. To adjust to the results of crawling Twitter data regarding religious moderation, the Stopword removal dictionary from the the Sastrawi library has been added with a new list of words: ['yg', 'di', 'nih', 'dgn', 'itu', 'gak', 'yang', 'lah', 'klo', 'lgi'].

The results of the stopword removal and stemming processes are shown in Fig. 3 and Fig. 4. In both images, the stopwords removal and stemming processes have taken place and have produced the expected results.

*name of corresponding author

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.11894
e-ISSN : 2541-2019
p-ISSN : 2541-044X

```
print (df['stopword'])

0      islam lengkap sempurna  justru islam mengkotak...
1      gw pernah disuruh nerangin terkait moderasi be...
2      moderasi beragama public policy apa kekuatan k...
3      hari terdapat banyak sekali serangan menimpa u...
4      menjelaskan moderasi beragama beliau menjelask...
                             ...
528    genderang perang kaum ngakunya nesyenelis mayo...
529    orang hebat orang  hebat islam  moderat cuma b...
530    para kadrun istilah kadrun kan dipake utk memb...
531                                moderatmode melarat
532    sikap moderat segala perkara merupakan jalan m...
Name: stopword, Length: 533, dtype: object
```

Fig. 3 Results of Stopword Removal using the Sastrawi library

```
print (df['stemming'])

0      islam lengkap sempurna justru islam kotak kota...
1      gw pernah suruh nerangin kait moderasi agama d...
2      moderasi agama public policy apa kuat lemah te...
3      hari dapat banyak sekali serang timpa umat isl...
4      jelas moderasi agama beliau jelas hakikat wuju...
                             ...
528    genderang perang kaum ngakunya nesyenelis mayo...
529    orang hebat orang hebat islam moderat cuma bed...
530    para kadrun istilah kadrun kan dipake utk beda...
531                                moderatmode melarat
532    sikap moderat segala perkara rupa jalan tuju b...
Name: stemming, Length: 533, dtype: object
```

Fig. 4 The results of Stemming using the Sastrawi library

The next process is word weighting with TF/IDF or Term Frequency-Inverse Document Frequency. The TF/IDF method gives the weight of the relationship of a word (term) to the document(Nurjannah et al., 2013). For the weight calculation process, this method combines two concepts for weight calculation: The number of a word in a certain sentence and inverse the number of sentences containing that word. The importance of a word is indicated by the frequency of its occurrence in the sentence. If the frequency of sentences containing the word is high, it means that the word is general. So if the frequency of the word is high in the sentence and the frequency of the entire sentence containing the word is low, then the weight of the relationship between a word and a sentence will be high.

Classification Model Performance

Classification is supervised learning, which is a predictive model, where the prediction results are discrete. In this research, the MNB and SVM model for classification was built in python interpreter. Although we no longer need to manually calculate the performance of the classifier model, it is also important that we know how the performance is calculated. The performance of the MNB dan SVM is measured by comparing the actual values with the predicted values represented in the confusion matrix. The matrix consists of different combinations of predicted and actual values: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). To determine the confusion matrix, the MNB and SVM model was tested in three iterations: The first iteration was on the amount of training data and test data compared to 80: 20, the second iteration was on the amount of training data and test data compared to 70: 30, and the third iteration was on the amount of training data and test data compared to 75:25.

After that, the accuracy, recall, precision, and f1-score of the two models may be determined using the confusion matrix. Sklearn.metrics was the module that was used to determine how well MNB and SVM performed. A prediction of the likelihood of a positive class, confidence value, or binary decision value is necessary for some metrics. Most implementations use the sample weight argument to let each sample contribute in a weighted manner to the final score. The results of accuracy, recall, precision, and f1-score of the three iterations in the MNB and SVM model are shown in Fig 5, Fig 6, and Fig 8.

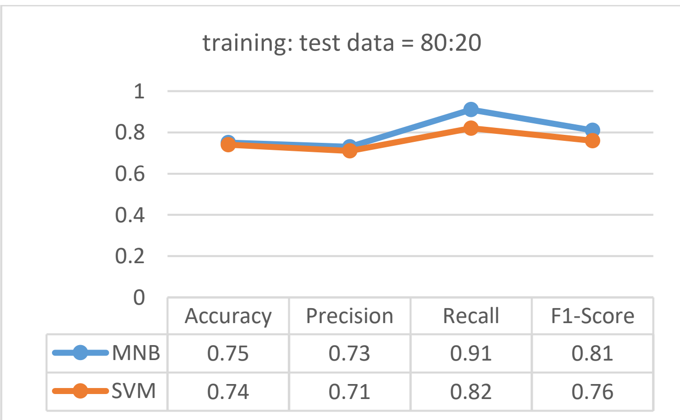

Fig. 5 Comparison of Classification Performance of MNB and SVM on the amount of training data and test data as much as 80:20

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.11894
e-ISSN : 2541-2019
p-ISSN : 2541-044X

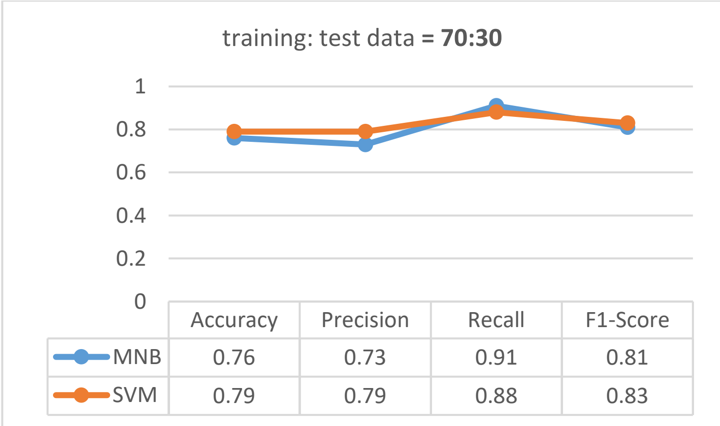

Fig. 6 Comparison of Classification Performance of MNB and SVM on the amount of training data and test data as much as 70:30

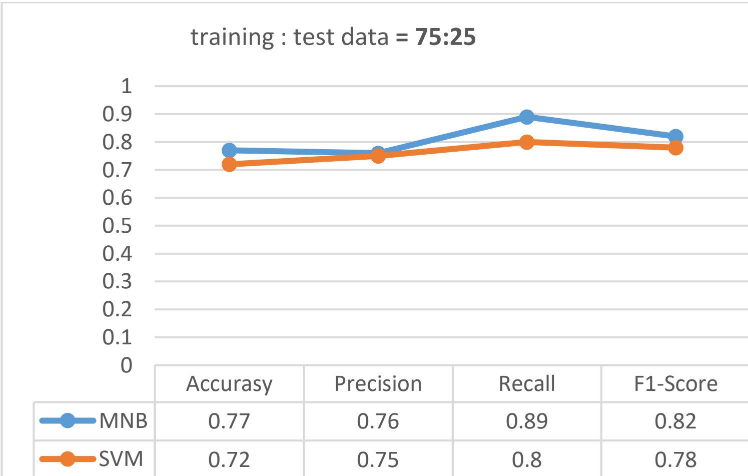

Fig. 7 Comparison of Classification Performance of MNB and SVM on the amount of training data and test data as much as 75:25

Computing time

The pre-processing process is carried out for 433.44 Seconds or 7.2241 Minutes using an CPU 2.00GHz and 4GB RAM as much as 533 text data, for 319.86 Seconds or 5.3311 Minutes on the second try, and for 148,811 Seconds on the third try. The average time used for pre-processing is 300.703 seconds or 5.011 minutes. While the time used for the classification process is different for each proportion of train data and test data. Table 2 shows the computational time of Preprocessing, MNB and SVM models in three iterations.

Table 1 Preprocessing Computing Time Results, MNB and SVM Model

Computing time	First	Second	Third	Average
PreProcessing	433.44 second	319.86 second	148.811 second	300,703 second
MNB	0,621 second	0,316 second	0,763 second	
SVM	0,135 second	0,196 second	0,531 second	

Comparison of Source Code Complexity

The metrics used in the analysis of the program code are Raw, Cyclometic Complexity, Halstead, and Maintainability Index. All of these metrics have been accommodated by Radon. Radon is a tool in the Python programming language that can measure the characteristics of the program code using various metrics(Kamil & Kurniastuti, 2020).

Cyclomatic Complexity: calculated according to the number of iterations containing the code block or the number of linear independent paths in the code, then added 1. This metric can be used when testing conditional logic in blocks. To calculate Cyclomatic Complexity, Radon analyzes the Abstract Syntax Tree of the program in the Python interpreter (*Welcome to Radon's Documentation! — Radon 4.1.0 Documentation*, n.d.).

Raw metrics has metric as follows:

a. LOC: or Lines of Code The total number of lines of program code.

*name of corresponding author

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.11894
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Formula: *LOC = SLOC + Multi + Single comments + Blank*

b. LLOC: or Logical Lines Of Code. Each line of code logic contains exactly one statement.
c. SLOC: Amount of source code, not directly related to LLOC.
d. Comments: Number of comment lines. The number of strings that are more than one line is not counted as comments, because for the python interpreter it is a whole string.
e. Multi: The number of lines representing a string consisting of more than one line.
f. Blanks: Number of blank lines
g. C % L: Ratio (percentage) between the number of lines that are comments and LOC
h. C % S: Ratio (percentage) between the number of lines that are comments and SLOC
i. C + M % L: Ratio (percentage) between the number of comment lines, strings with more than one line, and LOC

Halstead metrics. The Halstead metric consists of several metrics that can be used to count lines of software code. A computer program consists of instructions that are considered as a collection of tokens. By calculating tokens and specifying operators and operands. The operand is the value used in the operation, and the command given so that the result can be obtained is called the operator (*Welcome to Radon's Documentation! — Radon 4.1.0 Documentation*, n.d.).

j. $\eta 1$ = number of distinct operators
k. $\eta 2$ = number of distinct operands
l. N1 = number of operators overall
m. N2 = number of operands overall
n. Program vocabulary: $\eta=\eta 1+\eta 2$
o. Program length: $N=N1+N2$
p. Calculated program length: $N\hat{}=\eta 1 \log 2\ \eta 1+\eta 2 \log 2\ \eta 2$
q. Volume: $V=N \log 2\ \eta$
   Program volume (V) has been defined as the sum of the "*number of bits necessary to run the program*" and the "*number of mental comparisons required to write the program*," two independent units of measurement.
r. Difficulty: $D=\eta 1/2 * N2/\eta 2$
s. Effort: $E=D*V$
   Programming effort (PE): defined as the mental activity or effort required to convert an algorithm into a program code.
t. Time required to program: *T=E/18 seconds*
u. Number of delivered bugs: *B=V/3000.*

Maintainability Index: is a software metric that determines how maintainable or modifiable the program code is. The Maintainability Index is calculated using SLOC (Source Lines Of Code), Cyclomatic Complexity and Halstead complexity.The results of the comparison between MNB and SVM in terms of Raw code, Halstead Complexity, Maintainability Index, and Cyclomatic Complexity are shown in Table 3

Table 2 Comparison of MNB and SVM in Raw code, Halstead Complexity, Maintainability Index, and Cyclomatic Complexity

Pembanding	MNB	SVM
Cyclomatic Complexity	1	1
Raw	LOC: 51 LLOC: 28 SLOC: 28 Comments: 10 Single comments: 10 Multi: 0 Blank: 13 - Comment Stats (C % L): 20% (C % S): 36% (C + M % L): 20%	LOC: 56 LLOC: 36 SLOC: 36 Comments: 9 Single comments: 9 Multi: 0 Blank: 11 - Comment Stats (C % L): 16% (C % S): 25% (C + M % L): 16%
Halstead	η 1: 1	η 1: 2

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.11894
e-ISSN : 2541-2019
p-ISSN : 2541-044X

complexity	η 2: 2 N1: 1 N2: 2 vocabulary: 3 length: 3 calculated_length: 2.0 volume: 4.754887502163469 difficulty: 0.5 effort: 2.3774437510817346 time: 0.1320802083934297 bugs: 0.0015849625007211565	η 2: 4 N1: 2 N2: 4 vocabulary: 6 length: 6 calculated_length: 10.0 volume: 15.509775004326936 difficulty: 1.0 effort: 15.509775004326936 time: 0.861654166907052 bugs: 0.005169925001442312
Maintain-ability index	MI score: 100 - 20 Rank: A Maintainability: Very high	MI score: 100 - 20 Rank: A Maintainability: Very high

Meanwhile, Figure 9 shows Halstead Complexity Comparison of MNB and SVM

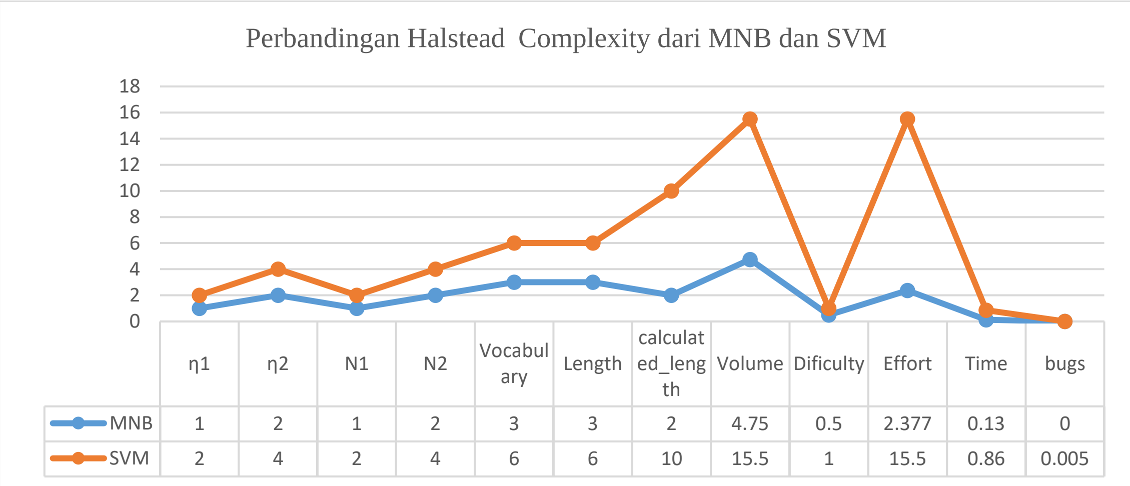

Figure 1. Halstead Complexity Comparison of MNB and SVM

DISCUSSIONS

The Support Vector Machine (SVM) classification model has a faster computation time than Multinomial Naive Bayes (MNB). The average time difference between the two classification models is 0.279 seconds, or it can be said that SVM is 0.279 seconds faster when classifying sentiments than MNB. However, for processing text data, this computing time is not excessive. The short processing time is due to the small amount of text data (533). The learning phase may become slower as the dataset size grows, as confirmed by .Nayak et al. (2015)

The value of Cyclomatic Complexity in both models is 1, this happens because both models use existing libraries, and the complexity of the libraries is not calculated directly by Radon. Based on the Raw Metrics, it can be seen that MNB and SVM have LOC, LLOC, and SLOC which are not significantly different, as shown in Fig 8. Based on Figures 5, 6, and 7, it is clear that SVM only performs considerably better than MNB at a 70% training data percentage. Which is demonstrated by accuracy and precision values that are higher than the MNB. SVM, however, typically performs worse than MNB at different combination of training and test data. However, SVM and MNB might not do very well when identifying emotions in instances of religious moderation and religious tolerance in Indonesia, with their average accuracy being 0.755 or 75.5%. This could be because the MNB can have a zero probability, especially when the model finds words in the test data for a particular class that are not in the training data. While SVM has the characteristics of not functioning properly with overlapping classes. As seen at the initial data, notably Fig. 1 and Fig. 2, there are similar words both in sentences with positive and negative labels, this can lead the determination of margins and hyperlanes to become less than optimal in SVM.

In terms of precision, MNB is also better able to not give positive labels to negative samples. The precision value of MNB in two iterations on average is 0.015 or 1.5% greater than SVM. However, this difference can be called as insignificant. In terms of recall, in all iterations MNB is better than SVM in finding all positive samples, this is indicated by the recall value of MNB 7% better than the recall value of SVM.

*name of corresponding author

Looking at the dificulty results on Halstead Metrics, it is seen that SVM is twice as difficult as MNB. It also corresponds to the LOC, LLOC, and SLOC of the SVM which is greater than the MNB. It can be said that overall SVM is a bit more complex than MNB. Particularly on programming effort and volume metrics. The programming effort metric revealed that the amount of mental effort needed to convert the SVM algorithm into a program is also 13.123 times greater than the MNB, and the results of the volume metric revealed that the number of bits needed to execute the SVM program is 10.75 times greater than the MNB.

This research has used a library or libraries that can be easily installed on the Python interpreter, including for text preprocessing and SVM and MNB models. As a result, the program code analysis may not be as effective as it would be if the library had not been used at all. The utilization of the library can be avoided for the subsequent research that is identical. However, one result that may happen is that the algorithm was misunderstood or incorrectly translated into program code, which could have a devastating effect on the model that was created by using improper mathematical or logical calculations.

CONCLUSION

The things that have been concluded in this study are as follows:

First, the results of collecting tweets from Twitter social media users in cases of "religious moderation" and "religous tolerance" show that there are many sarcasm sentences which are expressions of social media users. These sarcasm sentences generally have a negative emotional label. However, if look at the composition of words in the sentence, it is similar to the composition of words in sentences with positive emotional labels. This is what causes the MNB and SVM models to be inappropriately used as models to classify community sentiment with the keywords "*moderasi beragama*" or religius moderation and "*toleransi beragama*" or religious tolerance. This is because the results of the classification of the test data show that the performance of the two models is still low, with an average accuracy of 75.5%.

Second, if the computation time is compared for the two classification models, then SVM has a faster computation time than MNB. However, when compared in terms of the complexity of the program code, the Cyclomatic Complexity in both models is the same, this happens because both models use existing libraries, and the complexity of the libraries cannot be calculated directly. Based on the Raw Metrics, it can be seen that MNB and SVM have LOC, LLOC, and SLOC which are not significantly different. In terms of Halstead complexity of the MNB and SVM program code, it can be seen that SVM has a higher Halstead Complexity than SVM in all metrics. Especially on volume metrics and programming effort metrics. The results of these two metrics show that the number of bits required to execute an SVM program is three times greater than that of an MNB, and the mental activity required to reduce a preformed SVM algorithm into a program is also six times greater than that of an MNB.

ACKNOWLEDGMENT

We would like to thank the Lembaga Penelitian dan Pengabdian Kepada masyarakat, Universitas Hindu Negeri I Gusti Bagus Sugriwa Denpasar, which has provided financial support to complete this research. Thanks are also conveyed to the Dharma Duta Faculty for facilitating this research.

REFERENCES

Apriani, R., & Gustian, D. (2019). Analisis Sentimen Dengan Naïve Bayes Terhadap Komentar Aplikasi Tokopedia. *Jurnal Rekayasa Teknologi Nusa Putra*, *6*(1), 54–62.

Arsi, P., & Waluyo, R. (2021). Analisis Sentimen Wacana Pemindahan Ibu Kota Indonesia Menggunakan Algoritma Support Vector Machine (SVM). *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *8*(1), 147. https://doi.org/10.25126/jtiik.0813944

Calero, C., & Piattini, M. (2015). *Introduction to Green in Software Engineering*. https://doi.org/10.1007/978-3-319-08581-4

Fitri, E., Yuliani, Y., Rosyida, S., & Gata, W. (2020). Analisis Sentimen Terhadap Aplikasi Ruangguru Menggunakan Algoritma Naive Bayes, Random Forest Dan Support Vector Machine. *Jurnal Transformatika*, *18*(1), 71. https://doi.org/10.26623/transformatika.v18i1.2317

Gunawan, B., Pratiwi, H. S., & Pratama, E. E. (2018). Sistem Analisis Sentimen pada Ulasan Produk Menggunakan Metode Naive Bayes. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, *4*(2), 113. https://doi.org/10.26418/jp.v4i2.27526

Huda Ovirianti, N., Zarlis, M., & Mawengkang, H. (2022). Support Vector Machine Using A Classification Algorithm. *Jurnal Dan Penelitian Teknik Informatika*, *7*(3). https://doi.org/10.33395/sinkron.v7i3

Kamil, A. S., & Kurniastuti, I. (2020). Evaluation of Interview Application based on Radon System. *Kresna Social Science and Humanities Research*, *1*, 1–7. https://doi.org/10.30874/ksshr.62

Laurensz, B., & Sediyono, E. (2021). Analisis Sentimen Masyarakat terhadap Tindakan Vaksinasi dalam Upaya Mengatasi Pandemi Covid-19. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, *10*(2), 118–123. https://doi.org/10.22146/jnteti.v10i2.1421

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.11894
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Lestari, K. F., & Lazuardi, L. (2018). *Analisis Sentimen pada Twitter Mengenai Kampanye Pemberian Imunisasi Measles Rubella di Indonesia*. *3*(3), 300.

Lestari, S., & Saepudin, S. (2021). Analisis Sentimen Vaksin Sinovac Pada Twitter Menggunakan Algoritma Naive Bayes. *SISMATIK (Seminar Nasional Sistem Informasi Dan Manajemen Informatika)*, 163–170.

Malhotra, M. P. (2015). Python Based Software for Calculating Cyclomatic Complexity. *International Journal of Innovative Science, Engineering & Technology*, *2*(3), 546–549.

Nayak, J., Naik, B., & Behera, H. S. (2015). A Comprehensive Survey on Support Vector Machine in Data Mining Tasks: Applications & Challenges. *International Journal of Database Theory and Application*, *8*(1), 169–186. https://doi.org/10.14257/ijdta.2015.8.1.18

Nurjannah, M., Hamdani, & Fitri Astuti, I. (2013). Penerapan Algoritma Term Frequency-Inverse Document Frequency (TF-IDF) Untuk Text Mining. *Jurnal Informatika Mulawarman*, *8*(3), 110–113.

Qeis, M. I. (2015). Aplikasi Wordcloud sebagai Alat Bantu Analisis Wacana. *International Conference on Language, Culture, and Society (ICLCS) - LIPI*. https://www.researchgate.net/publication/316736417

Rosid, M. A., Fitrani, A. S., Astutik, I. R. I., Mulloh, N. I., & Gozali, H. A. (2020). Improving Text Preprocessing for Student Complaint Document Classification Using Sastrawi. *IOP Conference Series: Materials Science and Engineering*, *874*(1). https://doi.org/10.1088/1757-899X/874/1/012017

Sari, F. V., & Wibowo, A. (2019). *Analisis Sentimen Pelanggan Toko Online Jd.Id Menggunakan Metode Naïve Bayes Classifier Berbasis Konversi Ikon Emosi*. *10*(2), 681–686.

Singgalen, Y. A. (2021). *Pemilihan Metode dan Algoritma dalam Analisis Sentimen di Media Sosial : Sistematic Literature Review*. *3*(2), 278–302.

sklearn.metrics.precision_recall_fscore_support — scikit-learn 1.1.2 documentation. (n.d.). Retrieved October 6, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html#sklearn.metrics.precision_recall_fscore_support

Sodik, F., & Kharisudin, I. (2021). Analisis Sentimen dengan SVM , Naive Bayes dan KNN untuk Studi Tanggapan Masyarakat Indonesia Terhadap Pandemi Covid-19 pada Media Sosial Twitter. *Prisma*, *4*, 628–634.

Welcome to Radon's documentation! — Radon 4.1.0 documentation. (n.d.). Retrieved September 18, 2022, from https://radon.readthedocs.io/en/latest/index.html#

Yusliani, N., Aruda, S. A. Q., Marieska, M. D., Saputra, D. M., & Abdiansah. (2022). The effect of Chi-Square Feature Selection on Question Classification using Multinomial Naïve Bayes. *Sinkron : Jurnal Dan Penelitian Teknik Informatika*, *7*(4), 2430–2436.

*name of corresponding author