International Journal of Information System & Technology
Akreditasi No. 158/E/KPT/2021 | Vol. 6, No. 3, (2022), pp. 325-337

COVID-19 Vaccination Sentiment Analysis on Twitter Using Random Forest and Information Gain

Andi Nur Rachman[1], Husni Mubarok[2], Euis Nur Fitriani Dewi[3], Mitha Maharani[4]
[1,2,3,4]Department of Information System, Universitas Siliwangi, Tasikmalaya, Indonesia
Email: andy.rachman@unsil.ac.id, husni.mubarok@unsil.ac.id, euis.fitrianidewi@unsil.ac.id, mitha@student.unsil.ac.id

Abstract

Covid-19 in Indonesia has increased from January 2021 unti February 2021 there were 1,217,468 people who were confirmed positive for the corona virus. As a result the increase in the number, the government has taken preventive measures, one of which is the distribution of vaccines or vaccinating the Indonesian people, which has been started since January 13,2021. The government's covid-19 vaccination efforts had a broad influence on the community through social media (especially Twitter) which then led to pros and cons. Therefore, sentiment analysis is needed to predict the tendency of public opinion regarding the Covid-19 vaccination policy which is classified into positive opinions, neutral opinions, and negative opinions. Random Forest Classifier has high performance compared to other machine learning methods. But the Random Forest Classifier is weak in the level of accuracy and stability of data, so it requires a selection feature to increase its accuracy by applying Information Gain which can increase accuracy by optimizing data features. Measurement of accuracy and sentiment prediction is measured by confusion matrix and classification report. The results show that the application of Information Gain can improve accuracy with the highest accuracy obtained in experiment 1 of 0.00747, that is 0.94776 from 0.94029 with a precision value of 0.65, recall 0.43 and f1-score 0.47 and have a tendency to have a neutral opinion on public tweets about the Covid-19 vaccination on Twitter.

Keywords: *Covid-19, Information Gain, Random Forest Classifier, Sentiment Analysis, Twitter, Vaccination.*

1. Introduction

COVID-19 in Indonesia has increased again, from January 2021 until February 2021 there were 1,217,468 people who were confirmed positive for the corona virus based on data from the Covid-19 Handling Committee and National Economic Recovery. Due to this increase in numbers, the government has taken preventive measures, one of which is the distribution of vaccines or vaccinating the Indonesian people, which has been started since January 13, 2021 [1]. The vaccine that will be distributed to the public will be carried out in phases from January 2021 - March 2022. The initial distribution stage is in the form of the Sinovac vaccine. Regarding the Sinovac Vaccine clinical trial itself, BPOM has released the evaluation results from the Phases III interim clinical trial report which showed the efficacy or efficacy level of the Sinovac Covid-19 Vaccine of 65.3 percent which was in accordance with the standards and efficacy thresholds set by the World Human Organization. (WHO) which is at least 50 percent based on the statement of the head of BPOM Penny K Lukito as reported by cnnindonesia.com [2].

The Covid-19 vaccination effort carried out by the government has a broad influence on the community through social media (especially Twitter) which then raises the pros and cons in the community. Although BPOM has released the results of clinical trials, there are still people who doubt the effectiveness and efficacy of the covid-19 vaccine, considering that there are still some COVID-19 vaccines that

ISSN : 2580-7250
Copyright © 2022 IJISTECH

are still in the research and trial phase. Therefore, sentiment analysis is needed to see how successful the government's COVID-19 vaccination efforts are.

The Random Forest Classifier Algorithm is a classification method in the Machine Learning field that has high performance compared to other classification methods and has weaknesses in terms of the relatively low level of accuracy and stability of the data. This is related to the random function that is generated to perform the selection of data rows and the selection of candidate attribute solvers randomly [3][4]. So, need a selection feature in the form of Information Gain to improve accuracy and improve data stability in the algorithm.

There are several studies on sentiment analysis with several algorithm models that support this research, such as the research conducted by Januarsyah et al, 2019 and Fitri, 2020 which compares the Random Forest algorithm with the Naïve Bayes algorithm, Support Vector Machine, Decision Stump, Naïve Bayes, Bayesian Network, and C4.5 which produces the highest level of accuracy is the Random Forest algorithm with an accuracy rate of 97.16% and 74.2863%, as well as research conducted by Wijaya, 2017 and Somantri et al, 2018 regarding the application of selection features to the SVM and Naïve Bayes algorithm using Information gain and Chi square and the results obtained the highest accuracy increase of 3.08% and 1.0075% from the application of Information gain [5][6][7][8].

Based on the description above, sentiment analysis will be carried out using the Random Forest Classifier algorithm by selecting a feature, namely Information Gain to increase accuracy and optimize attributes in the data by taking the title "Sentiment Analysis on Public Opinion Regarding Covid-19 Vaccination on Twitter with the Random Forest Algorithm. Classifier and Information Gain".

2. Reseach Methodology

The research to be carried out is related to previous studies. Where the Random Forest algorithm has a higher level of accuracy than other classification algorithms. It can be proven by research conducted by Fitri, et al 2020 in their research using the Naïve Bayes algorithm, Random Forest, and Support Vector Machine to find out which classification algorithm has the highest performance and the results of the Random Forest algorithm are obtained with an accuracy of 97.16% and a value of 97.16%. AUC 0.996. Support Vector Machine algorithm that produces an accuracy of 96.01% with an AUC value of 0.543. Naive Bayes algorithm with an accuracy value of 94.16% and an AUC value of 0.999. The results of this study prove that the Random Forest algorithm has the highest accuracy value performance from the other two algorithms with an increase in accuracy of 7.16% [6].

Then strengthened in the research of Januarsyah, et al 2019 in comparing the performance of the Random Forest, Decision Stump, Naïve Bayes, Bayesian Network and C4.5 algorithms algorithms to find out which algorithm has the highest accuracy and the results of the C4.5 algorithm are obtained with an accuracy of 57,537%. The Decision Stump algorithm produces an accuracy of 49.952%. Naïve Bayes algorithm with an accuracy value of 49.0644%. Bayesian Network Algorithm with an accuracy value of 48.0764%. Random Forest Algorithm is 74.2863%. The results of this study prove that the Random Forest algorithm has the highest accuracy value than other algorithms [5].

Information Gain is one of the feature selection methods, can improve the performance of the algorithm used. As in the research conducted by Oman Somantri, et al 2018 to determine the performance of the Suppot Vector Machine algorithm after and before Information Gain was applied and the best accuracy rate was 72.45. % experience an increase of about 3.08% which was initially only 69.36% [8].

Then, it was strengthened in Wijaya's 2017 research in comparing the performance of the SVM and Naïve Bayes algorithms after and before using Information Gain and the

results of the SVM algorithm showed an increase in accuracy of 0.3333% from 98% to 98.3333% and the Naïve Bayes algorithm showed an increase of 1.0075% from 98.4925% to 99.5% [7].

The research methodology is a step to obtain data that is processed into more accurate information so that it can be a guide in conducting research so that the results do not deviate from the review to be achieved. This research methodology aims to describe all activities that will be carried out during the research. There are four main processes in this research, that is the process in the early phases, data processing, text classification, and analysis of results (see Figure. 1).

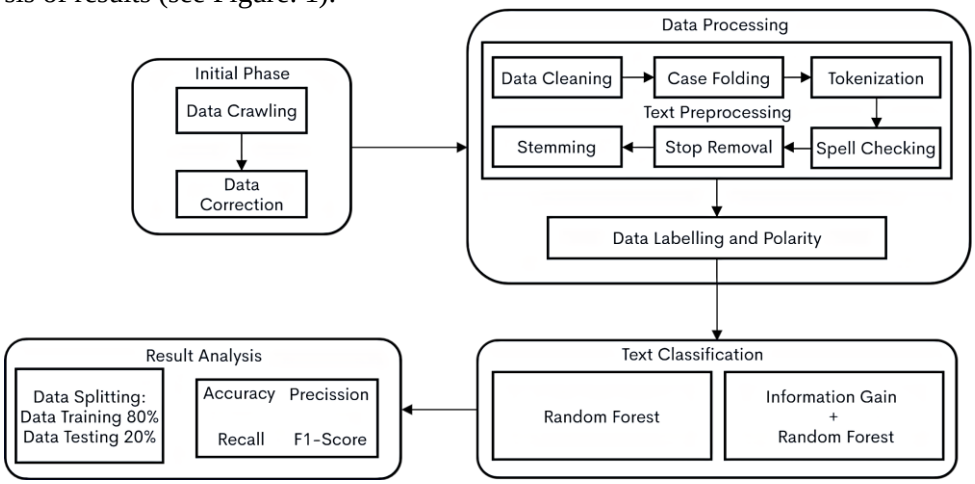

Figure 1. Research Methodology

3. Result And Discussion

3.1.Initial Phase

a. Data Crawling

This research uses a crawling method for data retrieval on Twitter using the Python programming language. Where the data retrieval process requires an token and key from Twitter by entering the consumer key, consumer secret, access token and also access token.

Tweet data collection was taken using keywords in the form of hashtags, that is #VaksinasiNasional and #VaksinCOVID19. The result of data crawling can be seen in table 1.

Table 1. Data Crawling Result

Crawling Results	
Experiment 1	1970 tweet
Experiment 2	499 tweet
Experiment 3	1448 tweet

The number of tweet data that was successfully retrieved with the hashtag, that is experiment 1, obtained 1970 tweets of tweet data, experiment 2 obtained as many as 499 tweets, and in experiment 3 obtained as many as 1448 tweets.

b. Data Correction

This correction process is carried out in MS.Excel by importing a CSV file, with this the data format will be changed to xslx from csv format to facilitate data correction. The correction process is carried out by removing unnecessary attributes and deleting data that is detected as duplication.The data used in this study was tweet data containing Indonesian text data. The result of data correction can be seen in table 2.

Table 2. Data Correction Result

	Crawling Results	Results of Data Correction
Experiment 1	1970 tweet	759 tweet
Experiment 2	499 tweet	333 tweet
Experiment 3	1448 tweet	834 tweet

From the data taken in experiment 1, which was 1970 tweet data, it was reduced to 759 tweet data. Then from the data taken in experiment 2 as many as 499 tweet data, it was reduced to 333 data. Meanwhile, from the data taken in experiment 3 as many as 1448 tweet data, it was reduced to 834 tweet data remaining after the data correction process was carried out.

3.2. Data Processing

a) Data Cleaning

Data cleaning is carried out to remove web links, usernames, taggers, delete numbers including numbers in strings and nonASCII chars, and remove symbols, numbers, and strange characters in text [9]. The result of data cleaning can be seen in table 3.

Table 3. Data Cleaning Result.

Before Data Cleaning	After Data Cleaning
Update perkembangan vaksinasi COVID-19 di Indonesia, per tanggal 29 Mei 2021 pukul 18.00 WIB. #VaksinasiNasional https://t.co/aDMx6quvFp"	Update perkembangan vaksinasi COVID di Indonesia per tanggal Mei pukul WIB
RT @iamnyssapotter: Tolonglah ambil vaksin tidak kira apa jenis vaksin sekali pun. Mana yang dapat dulu just go for it! Jgn memilih. Kesianâ€¦	Tolonglah ambil vaksin tidak kira apa jenis vaksin sekali pun Mana yang dapat dulu just go for it Jgn memilih Kesian

b) Case Folding

Case folding is carried out to convert the entire text in the document into a standard form or change each word to the same, that is by changing the letters to lowercase using the lowercase function [9]. The result of case folding can be seen in table 4.

Table 4. Case Folding Result

Before Case Folding	After Case Folding
Update perkembangan vaksinasi COVID di Indonesia per tanggal Mei pukul WIB	update perkembangan vaksinasi covid di indonesia per tanggal mei pukul wib
Tolonglah ambil vaksin tidak kira apa jenis vaksin sekali pun Mana yang dapat dulu just go for it Jgn memilih Kesian	tolonglah ambil vaksin tidak kira apa jenis vaksin sekali pun mana yang dapat dulu just go for it jgn memilih kesian

c) Tokenization

Tokenization is carried out to separate word for word in a text document into independent words and also to get tokens (word pieces) which will become valuable entities in the preparation of the document matrix in the next process which can facilitate the process of calculating the presence of words in the document or calculating frequency of occurrence of words in the corpus [9]. The result of tokenization can be seen in table 5.

Table 5. Tokenization Result

Text	Tokens
update perkembangan vaksinasi covid di indonesia per tanggal mei pukul wib	['update', 'perkembangan', 'vaksinasi', 'covid', 'di', 'indonesia', 'per', 'tanggal', 'mei', 'pukul', 'wib']

Text	Tokens
tolonglah ambil vaksin tidak kira apa jenis vaksin sekali pun mana yang dapat dulu just go for it jgn memilih kesian	['tolonglah', 'ambil', 'vaksin', 'tidak', 'kira', 'apa', 'jenis', 'vaksin', 'sekali', 'pun', 'mana', 'yang', 'dapat', 'dulu', 'just', 'go', 'for', 'it', 'jgn', 'memilih', 'kesian']

d) Spell Checking

Spell checking is carried out to fix writing in the form of abbreviations or typos into writing that is in accordance with the KBBI, in this process using the file 'kbbi.txt' as a dictionary of words used during the checking process [10]. The result of spell checking can be seen in table 6.

Table 6. Spell Checking Result

Word	Spell Check
['**yuk**', 'intip', 'komposisi', 'vaksin', 'covid', 'di', 'indonesia', 'apa', 'iya', 'punya', 'efek', 'magnet']	['**ayo**', 'intip', 'komposisi', 'vaksin', 'covid', 'di', 'indonesia', 'apa', 'iya', 'punya', 'efek', 'magnet']
['jangan', 'lengah', 'guys', 'pandemi', 'belum', 'berakhir']	['jangan', 'lengah', 'teman-teman', 'pandemi', 'belum', 'berakhir']
['vaksin', 'sangat', 'aman', 'bagi', 'masyarakat', 'indonesia', 'jadi', 'kalian', 'semua', 'jgn', 'takut', 'di', 'vaksin', 'ya']	['vaksin', 'sangat', 'aman', 'bagi', 'masyarakat', 'indonesia', 'jadi', 'kalian', 'semua', 'jangan', 'takut', 'di', 'vaksin', 'iya']

e) Stop Removal

Stop removal is carried out to remove words that are too general and less important which have a relatively large number of occurrences compared to other words. This stage also removes non-descriptive words [9]. The result of stop removel can be seen table 7.

Table 7. Stop Removel Result

Tokens	Stop Removal Results
['update', 'perkembangan', 'vaksinasi', 'covid', 'di', 'indonesia', 'per', 'tanggal', 'mei', 'pukul', 'wib']	['perkembangan', 'vaksinasi', 'covid', 'indonesia', 'pukul']
['masyarakat', 'sepatutnya', 'memilah', 'informasi', 'yang', 'beredar', 'di', 'jagat', 'maya', 'dengan', 'cara', 'berimbang', 'terkait', 'viralnya', 'vaksin', 'me']"	['masyarakat', 'sepatutnya', 'memilah', 'jagat', 'maya', 'berimbang', 'viralnya', 'vaksin']"

f) Stemming

Stemming is carried out to change the word into its basic form using the Indonesian literary python library. The stemming process itself is removing all word affixes including word prefixes, word insertions, word endings and or removing the prefix and suffix of words in derived words [9]. The result of stemming can be seen in table 8.

Table 8. Stemming Result

Tokens	Stemming Results
['perkembangan', 'vaksinasi', 'covid', 'indonesia', 'pukul']	[['kembang'], ['vaksinasi'], ['covid'], ['indonesia'], ['pukul']]
['janji', 'temu', 'pusat', 'pemberian', 'vaksin', 'ppv', 'beroperasi', 'lockdown', 'fasa', 'bermulai']	[['janji'], ['temu'], ['pusat'], ['beri'], ['vaksin'], ['ppv'], ['operasi'], ['lockdown'], ['fasa'], ['mulai']]

g) Data Labelling

This process extracts the sentiment value by utilizing the sentiment score in the python library, namely Textblob. Where does it work by determining the threshold for positive, negative and neutral labels. The result of data labelling can be seen in table 9.

Table 9. Data Labelling Result

Stemming	*Score*	Labels
[['update'], ['laksana'], ['vaksinasi'], ['covid'], ['bagi'], ['tenaga'], ['sehat'], ['palayan'], ['publik'], ['lansia'], ['provonsi'], ['papua'], ['barat']]	0.0	NEUTRAL
[['puji'], ['tuhan'], ['vaksin'], ['terima'], ['pak'], ['vaksinasi']]	0.136	POSITIVE
[['banyak'], ['kedaluwarsa'], ['dosis'], ['vaksin'], ['covid'], ['hong'], ['kong'], ['buang'], ['duh'], ['sayang'], ['banget'], ['lengkap']]	-0.3	NEGATIVE

h) Polarity

The method use in text classification is a machine learning method where the method cannot train text directly so that text data must be converted into numeric data. Then the polarity process is carried out to convert the label to the polarity. Positive has value 1, neutral has value 0, and negative has value -1. The result of polarity can be seen in table 10.

Table 10. Polarity Result

Stemming	Labels	Polarity
[['update'], ['laksana'], ['vaksinasi'], ['covid'], ['bagi'], ['tenaga'], ['sehat'], ['palayan'], ['publik'], ['lansia'], ['provonsi'], ['papua'], ['barat']]	NEUTRAL	0
[['puji'], ['tuhan'], ['vaksin'], ['terima'], ['pak'], ['vaksinasi']]	POSITIVE	1
[['banyak'], ['kedaluwarsa'], ['dosis'], ['vaksin'], ['covid'], ['hong'], ['kong'], ['buang'], ['duh'], ['sayang'], ['banget'], ['lengkap']]	NEGATIVE	-1

The results of percentage data labelling and polarity from experiments 1, 2, and 3 can be seen in table 11.

Table 11. Sentiment Class Percentage

	Sentiment Class percentage		
	Positive	Negative	Neutral
Experiment 1	7%	1%	92%
Experiment 2	7%	7%	85%
Experiment 3	6%	9%	85%

3.3.Text Classification

At this phase, data training and predicting sentiment will be carried out using the machine learning method, that is the random forest classifier which can only train or test data in numeric type, therefore it is necessary to convert text data into numeric data first by using a feature extraction library, namely the count vectorizer. and tf-idf transformers. Source code of count vectorizer process can be seen in table 12.

Table 12. Count Vectorizer Code

```
    #Count Vectorizer
    bow_transformer = CountVectorizer(ngram_range=(1,3), strip_accents='unicode',
max_features=1000)
    print((dataset.Tweet.shape))
    rdf = bow_transformer.fit_transform(dataset.Tweet)
    print('Shape of Sparse Matrix: ', rdf.shape)
    print('Amount of Non Zero occurances: ', rdf.nnz)
```

From the process of converting text to numeric using the CounVectorizer library, information was obtained that in experiment 1 there were 668 data with 1000 features set in the CountVectorizer library and the number of occurrences of non-

zero words was 4678. Then in experiment 2 obtained 122 data with 1000 features and the number of occurrences non-zero words are 1537. While in experiment 3, 362 data were obtained with 1000 features and the number of occurrences of non-zero words was 2991.

A TF-IDF Transformer is required on the Count Vectorizer result to produce a higher value or weighting of a word in certain documents if the frequency of occurrence of the word is higher in certain documents but lower in other documents. Source code of TF-IDF transformer process can be seen in table 13.

Table 13. TF-IDF Transformer Code

```
#TF-IDF Transformer
tf_transform = TfidfTransformer(use_idf=False).fit(rdf)
rdf = tf_transform.transform(rdf)
print(rdf.shape)
filename = 'tfidf_transform.pkl'
pickle.dump(bow_transformer, open(filename, 'wb'))
```

The result of the TF-IDF Transformer process in experiments 1, 2, and 3, that it has a pattern (<row index>, <value of a word(term)>) <TF-IDF score> (see Figure.2,)

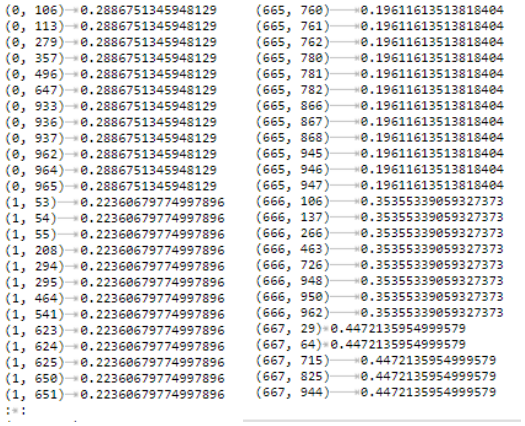

Figure 2. Result of TF-IDF Experiment

After the text data is converted into numeric data, the next stage is text classification modeling to conduct training on the model and predicting sentiment using the model described below.

a. Random Forest algorithm text classification without Information Gain

The definition of variable X uses the rdf.toarray() function which is intended to retrieve numeric data in the data converting process and variable Y uses the Polarity dataset as data that appears when the training process is introduced. Then it will be divided into training data in the form of X_train, Y_train with a percentage of 80% and testing data in the form of X_test, Y_test with a percentage of 20%.

The results of the data separation process with 1000 features are obtained as follows (see table 14).

Table 14. Data Splitting Result

	***Data Splitting* with 1000 feature**	
	Data Train	*Data Test*
Experiment 1	534	134
Experiment 2	97	25
Experiment 3	289	73

The next process is to train the classification text model and predict sentiment with the Random Forest Classifier model without Information Gain. Source code of random forest without information gain model can be seen in table 15.

Table 15. Random Forest Without Information Gain Model

```
#Training Text Classification Model and Predicting Sentiment
#Random Forest Classifier without Information Gain
classifier = RandomForestClassifier(criterion="entropy", n_estimators=1000)
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
```

To train the model using the Random Forest Classifier class from the sklearn.ensemble library by entering some important parameters in it. The first parameter is criterion which is a function to measure split quality. The criterion function is set with the entropy criterion so that the distribution data is homogeneous, where the more homogeneous the distribution, the better the accuracy. The second parameter used is n_estimators which is set to 1000, which means the author makes 1000 decision trees on the model used. Then the classifier.fit method will be used to train the model by passing the training data, namely X_train and the training target set, namely Y_train to the method and at the end predicting sentiment using the classifier.predict method for the dataset using test data, namely X_test.

b. Random Forest algorithm text classification with Information Gain

This second model uses a new X variable created by applying the selection feature in the form of Information Gain. The result of the feature selection process in experiments 1, 2, and 3 (see table 16, table 17, and table 18).

Table 16. Feature Selection Experiment 1

0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.176	0.176	0.176	0.176	0.353	0.353
0.169	0.169	0.169	0.169	0.3380	0.169
0.174	0.174	0.174	0.174	0.3481	0.3481

Table 17. Feature Selection Experiment 2

0.0	0.0	0.204	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.154	0.0
0.0	0.0	0.0	0.0	0.160	0.0
0.0	0.0	0.3162	0.0	0.0	0.0

Table 18. Feature Selection Experiment 3

0.0	0.0	0.0	0.0	0.188	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.192	0.0
0.0	0.0	0.0	0.408	0.0	0.0
0.208	0.208	0.0	0.0	0.0	0.208

From the table, information is obtained that the feature that has a value is the top feature of the search for the 10 best features in the SelectKBest class using the mutual_info_classif function.

After creating a new X variable in the form of the X_mic variable with the 10 best features from the feature selection process using Information Gain, it will be divided into training data in the form of X_mic_train, Y_train with a percentage of

80% and testing data in the form of X_mic_test, Y_test with a percentage of 20%. The result of data splitting can be seen in table 19.

Table 19. Data Splitting Result

	***Data Splitting* with 10 best feature**	
	Data Train	*Data Test*
Experiment 1	534	134
Experiment 2	97	25
Experiment 3	289	73

After data splitting is done, the next process is training the classification text model and predicting sentiment with the Random Forest Classifier model with the application of Information Gain. Source code of random forest with information gain model can be seen in table 20.

Table 20. Random Forest With Information Gain Model

```
# Random Forest Classifier menggunakan # Information Gain
classifierIG= RandomForestClassifier(criterion="entropy",  n_estimators=1000)
classifierIG.fit(x_mic_train, y_train) ypred = classifierIG.predict(x_mic_test)
```

To train the model using the Random Forest Classifier class from the sklearn.ensemble library by entering some important parameters in it. The first parameter is criterion which is a function to measure split quality. The criterion function is set with the entropy criterion so that the distribution data is homogeneous, where the more homogeneous the distribution, the better the accuracy. The second parameter used is n_estimators which is set to 1000, which means the author makes 1000 decision trees on the model used. Then the classifier.fit method will be used to train the model by passing the training data, namely X_mic_train and the training target set, namely Y_train to the method and at the end predicting sentiment using the classifier.predict method for the dataset using test data, namely X_mic_test.

3.4. Result Analysis

a. Evaluation of Random Forest Classifier Algorithm without Information Gain

Evaluation of Random Forest Classifier Algorithm Without Information Gain is a model evaluation in experiment 1, that is data training 534 data with 1000 features and data testing as many as 134 testing data with 1000 features that produce accuracy of 0.94030. The following is an explanation for the confusion matrix and classification report experiment 1 can be seen in table 21 and table 22.

Table 21. Confusion Matrix Experiment 1

		True Class		
		Positif	Negatif	Netral
Predicted Class	Positif	2	0	5
	Negatif	0	0	2
	Netral	1	0	124

Table 22. Classification Report Experiment 1

	Precision	*Recall*	*F1-Score*	*Support*
-1	0.00	0.00	0.00	2
0	0.95	0.99	0.97	124
1	0.67	0.29	0.40	7
Accuracy			0.94	134
Macro avg	0.54	0.43	0.46	134
Weighted avg	0.92	0.94	0.92	134

When viewed from the confusion matrix experiment 1 predicted there are 2 positive opinions, 0 negative opinions and 124 neutral opinions. The evaluation result of experiment 1, it has a 0.94 accuracy value with a precision value of 0.54, the recall of 0.43 and F1-score of 0.46.

In experiment 2 that is data training 97 data with 1000 features and data testing as many as 25 data with 1000 features that produce accuracy of 0.88.

The following is an explanation for the confusion matrix and classification report experiment 2 can be seen in table 23 and table 24.

Table 23. Confusion Matrix Experimen 2

		True Class		
		Positif	Negatif	Netral
Predicted Class	Positif	1	0	2
	Negatif	0	1	1
	Netral	0	0	20

Table 24. Classification Report Experiment 2

	Precision	*Recall*	*F1-Score*	*Support*
-1	1.00	0.50	0.67	2
0	0.87	1.00	0.93	20
1	1.00	0.33	0.50	3
Accuracy			0.88	25
Macro avg	0.96	0.61	0.70	25
Weighted avg	0.90	0.88	0.86	25

When viewed from the confusion matrix experiment 2 predicted there are 1 positive opinions, 1 negative opinions and 20 neutral opinions. The evaluation result of experiment 1, it has a 0.88 accuracy value with a precision value of 0.96, the recall of 0.61 and F1-score of 0.70.

Then in experiment 3, that is data training 289 data with 1000 features and data testing as many as 73 data with 1000 features that produce accuracy of 0.8356.

The following is an explanation for the confusion matrix and classification report experiment 3 can be seen in table 25 and table 26.

Table 25. Confusion Matrix Experiment 3

		True Class		
		Positif	Negatif	Netral
Predicted Class	Positif	3	0	2
	Negatif	0	0	8
	Netral	2	0	58

Table 26. Classification Report Experiment 3

	Precision	*Recall*	*F1-Score*	*Support*
-1	0.00	0.00	0.00	8
0	0.85	0.97	0.91	60
1	0.60	0.60	0.60	5
Accuracy			0.84	73
Macro avg	0.48	0.52	0.50	73
Weighted avg	0.74	0.84	0.79	73

When viewed from the confusion matrix experiment 3 predicted there are 3 positive opinions, 0 negative opinions and 58 neutral opinions. The evaluation result of experiment 1, it has a 0.84 accuracy value with a precision value of 0.48, the recall of 0.52 and F1-score of 0.50.

b. Evaluation of Random Forest Classifier Algorithm with Information Gain

Evaluation of Random Forest Classifier Algorithm Without Information Gain is a model evaluation in experiment 1, that is data training 534 data with 10 best features and data testing as many as 134 testing data with 10 best features that produce accuracy of 0.94776.

The following is an explanation for the confusion matrix and classification report experiment 1 can be seen in table 27 and table 28.

Table 27. Confusion Matrix RF-IG Experiment 1

		True Class		
		Positif	Negatif	Netral
Predicted Class	Positif	2	0	5
	Negatif	0	0	2
	Netral	0	0	125

Table 28. Classification Report RF-IG Experiment 1

	Precision	*Recall*	*F1-Score*	*Support*
-1	0.00	0.00	0.00	2
0	0.95	1.00	0.97	125
1	1.00	0.29	0.44	7
Accuracy			0.95	134
Macro avg	0.65	0.43	0.47	134
Weighted avg	0.94	0.95	0.93	134

When viewed from the confusion matrix experiment 1 predicted there are 2 positive opinions, 0 negative opinions and 125 neutral opinions. The evaluation result of experiment 1, it has a 0.95 accuracy value with a precision value of 0.65, the recall of 0.43 and F1-score of 0.47.

In experiment 2 that is data training 97 data with 10 best features and data testing as many as 25 data with 10 best features that produce accuracy of 0.92.

The following is an explanation for the confusion matrix and classification report experiment 2 can be seen in table 29 and table 30.

Table 29. Confusion Matrix RF-IG Experiment 2

		True Class		
		Positif	Negatif	Netral
Predicted Class	Positif	3	0	0
	Negatif	0	0	2
	Netral	0	0	20

Table 30. Classification Report RF-IG Experiment 2

	Precision	*Recall*	*F1-Score*	*Support*
-1	0.00	0.00	0.00	2
0	0.91	1.00	0.95	20
1	1.00	1.00	1.00	3
Accuracy			0.92	25
Macro avg	0.64	0.67	0.65	25
Weighted avg	0.85	0.92	0.88	25

When viewed from the confusion matrix experiment 2 predicted there are 3 positive opinions, 0 negative opinions and 20 neutral opinions. The evaluation result of experiment 2, it has a 0.92 accuracy value with a precision value of 0.64, the recall of 0.67 and F1-score of 0.65.

Then in experiment 3, that is data training 289 data with 10 best features and data testing as many as 73 data with 10 best features that produce accuracy of 0.8493.

The following is an explanation for the confusion matrix and classification report experiment 3 can be seen in table 31 and table 32.

Table 31. Confusion Matrix RF-IG Experiment 3

		True Class		
		Positif	Negatif	Netral
Predicted Class	Positif	3	0	2
	Negatif	0	0	8
	Netral	1	0	59

Table 32. Classification Report RF-IG Experiment 3

	Precision	*Recall*	*F1-Score*	*Support*
-1	0.00	0.00	0.00	8
0	0.86	0.98	0.91	60
1	0.75	0.60	0.67	5
Accuracy			0.85	73
Macro avg	0.54	0.53	0.53	73
Weighted avg	0.75	0.85	0.80	73

When viewed from the confusion matrix experiment 3 predicted there are 3 positive opinions, 0 negative opinions and 59 neutral opinions. The evaluation result of experiment 2, it has a 0.85 accuracy value with a precision value of 0.54, the recall of 0.53 and F1-score of 0.53.

The following is a graph of increasing the accuracy of the Random Forest Classifier algorithm before and after the application of Information Gain (see Figure.3)

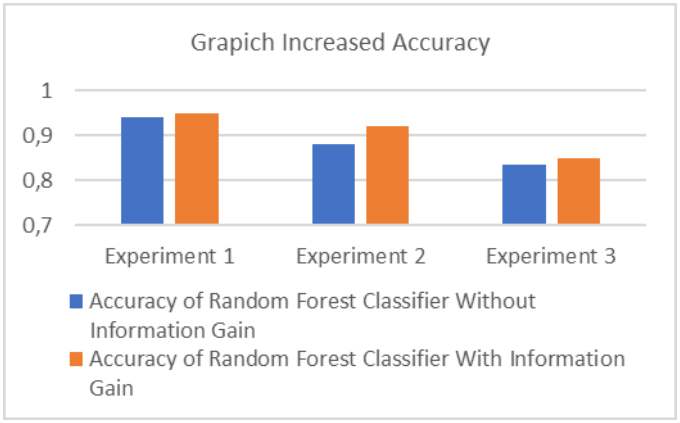

Figure 3. Graph increased accuracy

From the graph, information is obtained that Random Forest Classifier with the implementation of Information Gain has a high level of accuracy and the Information Gain method used has consistent results, where with the number of different data, the method can still increase the accuracy of the Random Forest Classifier algorithm.

4. Conclusion

From the results of the research can be said that the selection of features using information gain can improve the performance of the random forest classifier algorithm consistently with the number of varied data, where experiment 1 has the highest accuracy value than experiments 2 and 3, which is increased by 0.00747 with a value Accuracy of 0.94209 to 0.94776, as well as the precission value of 0.65, recall of 0.43, and F1-score of 0.47 which means it is quite good in research on the opinion of community relaxation related to Covid-19 vaccination on Twitter.

The tendency of community opinion related to Covid-19 vaccination on Twitter, seen from the results of testing the random forest classifier method with information gain is neutral. Where at least the people who have a negative culture are related to the Covid-19 vaccine and there are still people who receive and support the Covid-19 vaccination policy as a government effort to break the Covid-19 chain.

References

[1] F. Anwar, "Vaksinasi COVID-19 Indonesia Dimulai Hari ini, Menkes Juga Suntik," 2021. [Online]. Available: https://health.detik.com/berita-detikhealth/d-5331743/vaksinasi-covid-19-indonesia-dimulai-hari-ini-menkes-juga-disuntik. [Accessed: 10-Feb-2021].

[2] CNN Indonesia, "BPOM Umumkan Hasil Uji Klinis Sinovac, Efikasi 65,3 Persen," 2021. [Online]. Available: https://www.cnnindonesia.com/nasional/20210105163333-20-589783/bpom-umumkan-hasil-uji-klinis-sinovac-efikasi-653-persen. [Accessed: 15-Feb-2021].

[3] D. Y. Heryadi, Machine Learning Konsep dan Implementasi. Yogyakarta: Penerbit Gava Media, 2020.

[4] E. K. Adhitya, R. Satria, and H. Subagyo, "Komparasi Metode Machine Learning dan Metode Non Machine Learning untuk Estimasi Usaha Perangkat Lunak," IlmuKomputer.com J. Softw. Eng., vol. 1, no. 2, pp. 109–113, 2015.

[5] M. F. Januarsyah, E. Zuhairi, and ..., "Perbandingan Algoritma Random Forest, Decision Stump, Naïve Bayes, Bayesian Network dan Algoritma C4. 5 Untuk Prediksi Pola Kartu Poker," … Res. Semin. (ARS …, vol. 5, no. 1, pp. 978–979, 2020.

[6] E. Fitri, "Analisis Sentimen Terhadap Aplikasi Ruangguru Menggunakan Algoritma Naive Bayes, Random Forest Dan Support Vector Machine," J. Transform., vol. 18, no. 1, p. 71, 2020, doi: 10.26623/transformatika.v18i1.2317.

[7] J. I. Komputer, F. Matematika, D. A. N. Ilmu, and P. Alam, "Penerapan Information Gain Guna Support Vector Machine Dan Naïve Bayes," 2017.

[8] O. Somantri and D. Apriliani, "Support Vector Machine Berbasis Feature Selection Untuk Sentiment Analysis Kepuasan Pelanggan Terhadap Pelayanan Warung dan Restoran Kuliner Kota Tegal," J. Teknol. Inf. dan Ilmu Komput., vol. 5, no. 5, p. 537, 2018, doi: 10.25126/jtiik.201855867.

[9] I. F. Rozi, M. Hani'ah, and Y. D. Pradika, "Analisis Sentimen Terhadap Sistem Zonasi Berdasarkan Wilayah Menggunakan FK-NNC," Semin. Inform. Apl. Polinema, pp. 376–381, 2020.

[10] U. Chuzaimah Zulkifli, "Pengembangan Modul PreprocessingTeks untuk Kasus Formalisasi dan Pengecekan Ejaan Bahasa Indonesia pada Aplikasi Web Mining Simple Solution (WMSS)," J. Mat. Stat. dan Komputasi, vol. 15, no. 2, p. 95, 2018, doi: 10.20956/jmsk.v15i2.5718.