Semi-supervised Learning Models for Sentiment Analysis on Marketplace Dataset

Wisnalmawati[1], Agus Sasmito Aribowo[2]*, Yunie Herawati[3]
[1]Faculty of Economics and Business, Universitas Pembangunan Nasional Veteran Yogyakarta, Indonesia
[2]Informatics Department, Universitas Pembangunan Nasional Veteran Yogyakarta, Indonesia
[3]Faculty of Mineral Technology, Universitas Pembangunan Nasional Veteran Yogyakarta, Indonesia
[1]wisnalmawati@upnyk.ac.id; [2]sasmito.skom@upnyk.ac.id*; [3]yunie_herawati@upnyk.ac.id
*corresponding author

ABSTRACT

Sentiment analysis aims to categorize opinions using an annotated corpus to train the model. However, building a high-quality, fully annotated corpus takes a lot of effort, time, and expense. The semi-supervised learning technique efficiently adds training data automatically from unlabeled data. The labeling process, which requires human expertise and requires time, can be helped by an SSL approach. This study aims to develop an SSL-Model for sentiment analysis and to compare the learning capabilities of Naive Bayes (NB) and Random Forest (RF) in the SSL. Our model attempts to annotate opinion documents in Indonesian. We use an ensemble multi-classifier that works on unigrams, bigrams, and trigrams vectors. Our model test uses a marketplace dataset containing rating comments scrapping from Shopee for smartphone products in the Indonesian Language. The research started with data preparation, vectorization using TF-IDF, feature extraction, modeling using Random Forest (RF) and Naïve Bayes (NB), and evaluation using Accuracy and F1-score. The performance of the NB model outperformed previous research, increasing by 5,5%. The conclusion is that SSL performance highly depends on the number of training data and the compatibility of the features or patterns in the document with machine learning. On our marketplace dataset, better to use Random Forest.

Keywords : Semi-supervised Learning; Sentiment Analysis; Marketplace; Performance Accuracy.

Article History

Received : Nov, 06th 2022	Accepted: Dec, 01st 2022	Published : Dec, 03rd 2022

I. INTRODUCTION

Sentiment analysis is part of Natural Language Processing (NLP) which aims to categorize opinions into positive, negative, or neutral sentiments. The benefits of sentiment analysis are widely felt, for example, obtaining sentiment information related to hotels [1], airlines [2], films[3], political events[4], and so on. The results of sentiment classification in a set of documents can be summarized to measure customer satisfaction with the services provided. For example, in the sentence, "The plot of this film is not surprising... The actors are not able to reflect the figure of Superman!!". The terms "not surprising" and "not able" reflect negative sentiments. In supervised sentiment analysis, classification into positive or negative is the main task of machine learning. In supervised sentiment analysis, machine learning will process a training dataset D is equal to $\{d1, d2, \ldots, dn\}$ and its associated label Y is equal to $\{y1, y2, \ldots, yn\}$ and learn the function $f(D; p1, p2, \ldots) \rightarrow Y$, where p1 and p2 are model parameters. This method is effective for analyzing sentiment, but it requires a huge amount of data that has been categorized. In order to develop high-quality datasets, it is necessary for professionals to gather and assign labels to the data. This dataset is going to be read by machine learning in order to train a classification model.

Most sentiment analysis study requires a fully labeled corpus to prepare the model. The expert determines the label in the corpus. However, building a fully labeled corpus with high quality takes a lot of effort, time, and expense, but manually labeling the data can be a strenuous task. Several studies explain that semi-supervised learning (SSL) can be a method that is faster, cheaper, and has high performance for labeling opinion datasets, such as [5]–[7] have solved the difficulty of manual labeling using semi-supervised learning (SSL). Semi-supervised learning study using IMDB datasets is [8]. In [8], a semi-supervised algorithm using deep neural networks with different settings divided the IMDB dataset into 4000 training data and 36000 unlabeled data. Their trials obtained accuracy ranging from 81%-82%, not much different from the baseline (82%). Various types of semi-supervised learning provide better accuracy in research [9][5]. AraSenCorpus in [5] is a semi-supervised framework to annotate a large Arabic text corpus using small manually annotated tweets. This model used the FastText and LSTM deep learning classifier to expand the annotated corpus. In English documents, Balakrishnan proposes SSL uses a Support Vector Machine, Random Forest, and the Naïve Bayes method. In their research, Random Forest reaches F1-score equal to 73.8%, Cohen's Kappa is equal to 52.2% for

sentiment analysis, F1-score equal to 58.8%, and Cohen's Kappa is equal to 44.7% for emotion analysis [6]. Alahmary proposes a semi-automatic approach to annotating the Saudi dialect tweets dataset. Their model's accuracy achieved by the Naïve Bayes classifier was 83%. Their model also uses three deep learning classifiers: convolutional neural network (CNN), long short-term memory (LSTM), and bidirectional long short-term memory (Bi-LSTM). In their study SVM was used as the baseline for comparison. Overall, the performance of the deep learning classifiers, especially CNN exceeded SVM. CNN outperformed the other classifiers with the highest accuracy of 87% [10].

Our research aims to create an SSL model for sentiment classification with a slight decrease in accuracy and F1-score between baseline conditions and convergent (final) conditions. So, we used several strategies to find the SSL-Model. Continuing our previous research in [11][12], we introduce an SSL model for annotating corpus using Naïve Bayes and Random Forest for the classifier model. In our SSL, we use several classifiers that work together but independently to expand the annotated corpus. Each classifier works in one type of tokenization. The first classifier works on unigrams, the second classifier works on bigrams, and the third classifier works on trigrams. The research question is whether the combination of TF-IDF and Random Forest can maintain their accuracy when used in the SSL model, compared to the baseline model. We also compared the Random Forest with Naïve Bayes as a machine learning in SSL. The next question is whether the number of annotated datasets for training in semi-supervised learning significantly affects the model's accuracy. We used the Marketplace dataset (in Indonesian Languages) to test the model.

This paper contains: section 1 presents an introduction, research objectives, and related works; section 2 describes the data collection, pre-processing, vectorization, modeling, and validation methods. Section 3 contains results and discussion, and section 4 contains conclusions.

II. METHOD

In this section, we will go through the data preparation methods, vectorization, feature extraction, modeling with Random Forest, model validation, model architecture, and pseudocode for the model.

A. Data Collection

For experiments, we used two marketplace datasets in Indonesian languages. The datasets containing scrapped shop rating comments from Shopee for smartphone products: MarketData1 and MarketData2, consist of 8523 and 5421 document reviews. MarketData1 is a data set for sentiment classification that has been manually labeled positive, neutral, and negative. MarketData2 is a data set for binary sentiment classification that has been manually labeled positive and negative.

B. Data Cleaning and Preprocessing

Marketplace datasets need to be analyzed consisting of words, numbers, and special symbols. Some processes for structuring the data go through several stages, such as tokenizing (unigram, bigram, and trigram), converting to a small case, removing a number, removing stop words, removing all non-alphabetic characters and punctuation, and stemming.

C. Vectorization

TF-IDF is used to calculate the weight of each word in the corpus. A document's term frequency can be calculated by taking the total number of terms in the document and dividing that total by the total number of terms in the document. IDF is the notation used to distribute the terms throughout document D. The TF value increases in proportion to the frequency of a word's appearances in the document; conversely, the IDF value increases in proportion to the decreasing frequency of the word's appearances. The term weights resulting from the TF-IDF weighting are converted into vector data. In very large documents, the features form a large dimensional matrix because each word that appears in the document is represented by its score [13]. TF-IDF Vectorizer used for sentiment analysis in research [14]–[16].

D. Ensemble Multi Classifier

We use Random Forest (RF) to build the SSL model. Random Forest creates multiple trees based on bootstrapped data samples and splitting nodes using the best split among a random subset of features selected at every node, then combines the predictions in Fig. 1. Random Forest used for sentiment classification in [18]–[20]. In this research, the parameter of Random Forest was set using some estimators=200.

Naive Bayes is used in many sentiment analysis studies in Indonesian [20]–[22] and in movie commentary datasets in [23]. Naive Bayes is already known as machine learning which is widely used in sentiment analysis and produces high accuracy. Bayes' rule is presented in Equation (1).

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

Where, the $P(y)$ variable is a probability y is true, the $P(X)$ variable is a probability of the X variable is true, the $P(y|X)$ variable is a probability of the y to be true if X variable is true, and the $P(X|y)$ variable is a probability of the X is true if y variable is true.

DOI: http://dx.doi.org/10.25139/ijair.v4i2.5267

Naive Bayes is a suitable method for binary and multiclass classification. This method applies a supervised classification technique by assigning class labels to instances using conditional probabilities. Conditional probability is the probability of an event occurring when another event has already occurred.

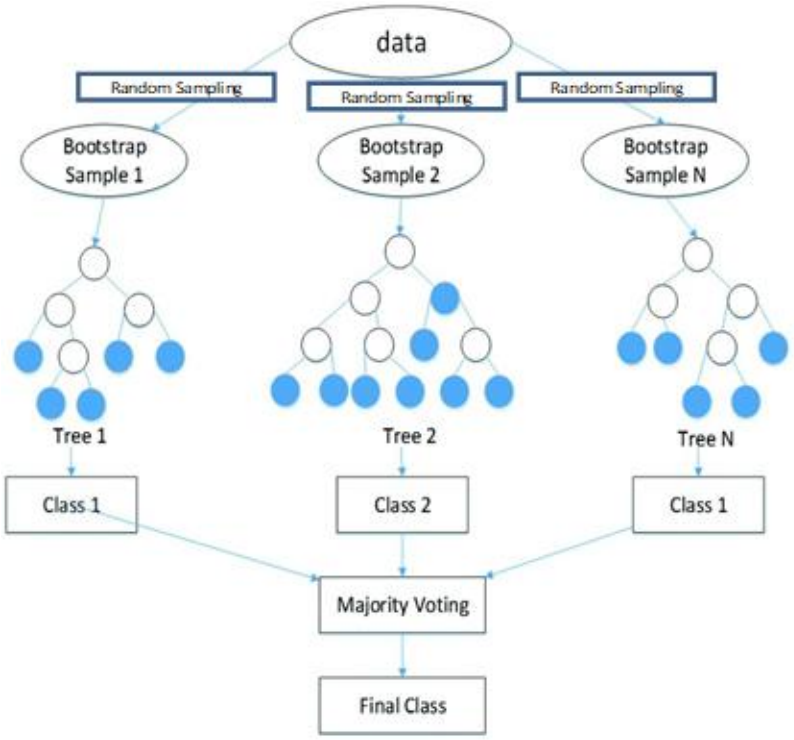

Fig 1. Random Forest Architecture

E. Validation

Performance measurement for SSL model tested using a confusion matrix. The confusion matrix compares the actual and prediction results (Table 1).

TABLE 1
CONFUSION MATRIX

		Actual	
		Positive	**Negative**
Predicted	**Positive**	True Positive / TP	False Positive / FP
	Negative	False Negative / FN	True Negative / TN

This study uses two measurements to validate the model: Accuracy and F1-score. Accuracy in Equation (2) is a great measure but only for symmetric datasets where values of false positives and false negatives are almost the same.

$$Accuracy = (TP + TN) / (TP + FP + FN + TN) \quad (2)$$

F1-score is the weighted average of Precision and Recall in Equation (3). In unequal class distributions, the F1 score is usually more useful than the accuracy

$$F1 - score = 2 * (Recall * Precision) / (Recall + Precision) \quad (3)$$

Precision is the degree of match between the information requested by the user and the answers given by the system. Precision-formulated in (4) is the ratio of correctly predicted positive to the total predicted positive.

$$Precision = TP / TP + FP \quad (4)$$

Recall (Sensitivity) is the system's success rate in retrieving information. Recall presented in Equation (5).

$$Recall = TP/TP+FN \quad (5)$$

DOI: http://dx.doi.org/10.25139/ijair.v4i2.5267

F. Semi-Supervised Learning Architecture

The proposed SSL model was developed from previous research in [11][12]. The difference is in the type of machine learning, the datasets, the voting mechanism to determine the class for the data, and the more varied threshold values. The architectural model shown in Figure 1 starts by reading the annotated input dataset. The proposed SSL model shown in Figure 1 began with reading the annotated input dataset. The annotated dataset is clean after pre-processing and divided into unlabelled data, data training, and data testing. TF-IDF vectorization processes the data training into three vectors: unigram, bigram, and trigram vector. The vectors used to build models using Random Forest (RF) and Naïve Bayes (NB) (in the next experiment). The result is three models that work separately (using the ensemble stacking mechanism). The three models were used to annotate Unlabeled Data. TF-IDF also vectorizes unlabeled data. Unlabeled data annotated by each model. The resulting Pseudo Labels are three classified documents. A label is considered high confidence if it is supported by the sum of weight divided by the total weight of several models and higher than a threshold. Threshold numbers are used to select whether the annotated data (with pseudo-labels) is worthy of being training data. The high-confidence document will be integrated with the Training Data. The document will be re-labeled in the next iteration if categorized as low confidence.

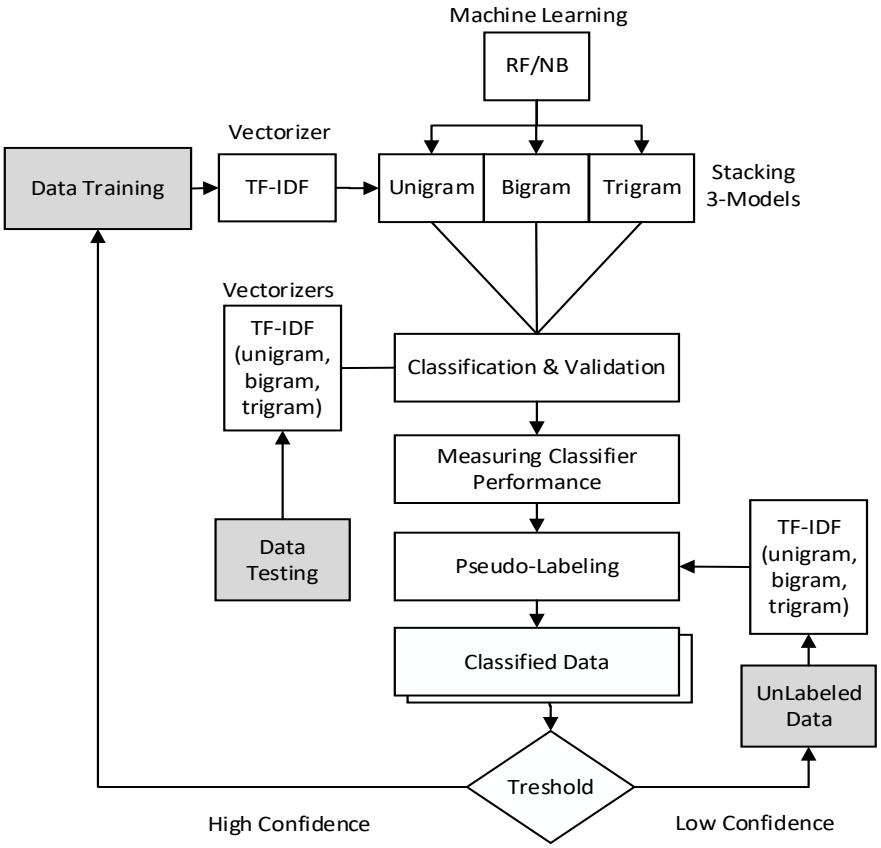

Fig 1. Proposed SSL-Model Architecture

Iterations in our SSL model run ten times or until the Unlabeled Data runs out. The model's output is Data Training (DT) which humans and machines have labeled.

G. Semi-Supervised Learning Pseudocode

Fig 2 is the pseudocode of our model. The pseudocode begins with setting the threshold number. Lines 2-4 are about input training data (DT), testing data (DTest), and unlabeled data (UN). DataTraining, Data testing, and Unlabeled dataset tokenized to unigram, bigram, and trigram using TF-IDF methods (lines 6-8). The classifier models were formed using three training sets and machine learning (RF or NB) on lines 10-12.

```
Function SSL(Treshld, ML)
1       Threshold=Treshld
2       READ DT       //Data Training(X,y)
3       READ DTest    //Data Testing(X,y)
4       READ UN       //Unlabeled Data(X)
5       VTestUnigram, VTestBigram, VTestTrigram =TFIDF (DTest, ngram=1,2,3)
```

```
6       Loop Until Convergence:
7           VTrainingUnigram, VTrainingBigram, VTrainingTrigram = TFIDF(DT, ngram=1,2,3)
8           VUnlabeledUnigram, VUnlabeledBigram, VUnlabeledTrigram =TFIDF(UN, ngram=1,2,3)
9
10          Mod1 = ML.Training(VTrainingUnigram)
11          Mod2 = ML.Training(VTrainingBigram)
12          Mod3 = ML.Training(VTrainingTrigram)
13
14          Labeling[1]=Mod1.Predict(VUnlabeled_Unigram)
15          Labeling[2]=Mod2.Predict(VUnlabeled_Bigram)
16          Labeling[3]=Mod3.Predict(VUnlabeled_Trigram)
17
18          For J = 1 to LEN(UN):
19              Pos=0; Neu=0; Neg=0; Total=0
20              For Mod=1,3:
21                  Predicted= Labeling[Mod].RecordNo[J]
22                  If Predicted=="Positive" Then Pos++]
23                  If Predicted=="Neutral" Then Neu++
24                  If Predicted=="Negative" Then Neg++
25                  Total++
26              If Pos/Total >= Threshold: Append(UN[J] as Positive) to DT and Remove(UN[J]) from UN
27              If Neu/Total >= Threshold: Append(UN[J] as Neutral) to DT and Remove(UN[J]) from UN
28              If Neg/Total >= Threshold: Append(UN[J] as Negative) to DT and Remove(UN[J]) from UN
29      Output(DT)
30      Validate(DT) with Accuracy, F1Score

Main:
31      START
32      ML=['RF', 'NB']
33      Treshold=[60%,70%,80%,90%]
34      For X in ML :
          For Y in Treshold:
            DO SSL(Y, X)
35      END
```

Fig 2. Pseudocode of Proposed Semi-Supervised Model

The annotation process was on lines 14-16. Lines 18-28 are the test process for each new annotated data whether it meets to become training data. The process begins by checking whether the new annotated data tends to be positive, negative, or neutral based on the pseudo-label weight (lines 26-28). If it is more than the threshold, then it is feasible to become training data. If not, it will be retested in the next iteration.

III. RESULT AND DISCUSSION

A. Testing the SSL Model using Market Dataset 1.

For an experiment, data are coded to *D1, D2, D3*, and *D4*. We randomly divided the dataset into training data and test data in a 9:1 ratio. The number of **labeled test data** for each D1, D2, D3, and D4 is 850 (approximately 10% of all documents). The number of **labeled training data** (annotated dataset) in D1, D2, D3, and D4 are 1700, 850, 425, and 212, respectively. The leftover training data was used as the unlabeled data set. The baseline model in *D1, D2, D3,* and *D4* was built with training data only. The baseline model was tested using labeled test data. In Table 2, we display the accuracy and F1-score of the baseline and semi-supervised learning (SSL) model in D1, D2, D3, and D4 under different numbers of thresholds, respectively.

There is some knowledge gained from 64 SSL models. First, the baseline classification results show that the accuracy score and F1 score are directly proportional to the number of training data instances. The accuracy and F1-score at the baseline of the Random Forest models are higher than that of Naïve Bayes. Second, the results of semi-supervised learning classification show that accuracy and F1-score also tend to be linear with the number of training data instances but inversely proportional to the threshold. The threshold strongly influences the SSL accuracy rate. A low threshold provides high accuracy and a high F1 score. The reason is that a low threshold will produce more pseudo-labeled datasets than a high threshold, so the classifier model formed in the next iteration will be smarter than the model formed by a few pseudo-labeled datasets.

TABLE 2
ACCURACY AND F1-SCORE OF SSL MODELS ON MARKET DATASET 1

Experiment		Accuracy						F1-score					
		Naïve Bayes			Random Forest			Naïve Bayes			Random Forest		
No	Threshold	Baseline	SSL	Diff	Baseline	SSL	Diff	Baseline	SSL	Diff	Baseline	SSL	Diff
D1	60%	0.69	0.67	0.02	0.74	0.71	0.03	0.7	0.68	0.02	0.73	0.72	0.01
	70%	0.69	0.65	0.04	0.73	0.71	0.02	0.7	0.67	0.03	0.73	0.71	0.02
	80%	0.69	0.66	0.03	0.73	0.7	0.03	0.7	0.67	0.03	0.73	0.71	0.02
	90%	0.69	0.61	0.08	0.73	0.68	0.05	0.7	0.64	0.06	0.73	0.68	0.05
D2	60%	0.67	0.64	0.03	0.7	0.69	0.01	0.68	0.66	0.02	0.69	0.69	0

DOI: http://dx.doi.org/10.25139/ijair.v4i2.5267

Experiment		Accuracy						F1-score					
		Naïve Bayes			Random Forest			Naïve Bayes			Random Forest		
No	Threshold	Baseline	SSL	Diff	Baseline	SSL	Diff	Baseline	SSL	Diff	Baseline	SSL	Diff
	70%	0.67	0.64	0.03	0.68	0.68	0	0.68	0.66	0.02	0.68	0.68	0
	80%	0.67	0.63	0.04	0.69	0.64	0.05	0.68	0.65	0.03	0.68	0.65	0.03
	90%	0.67	0.53	0.14	0.71	0.64	0.07	0.68	0.56	0.12	0.7	0.66	0.04
D3	60%	0.62	0.6	0.02	0.72	0.68	0.04	0.65	0.63	0.02	0.68	0.69	0.01
	70%	0.62	0.61	0.01	0.7	0.65	0.05	0.65	0.63	0.02	0.68	0.66	0.02
	80%	0.62	0.59	0.03	0.72	0.6	0.12	0.65	0.62	0.03	0.69	0.62	0.07
	90%	0.62	0.48	0.14	0.71	0.45	0.26	0.65	0.52	0.13	0.68	0.47	0.21
D4	60%	0.63	0.62	0.01	0.7	0.67	0.03	0.65	0.65	0	0.65	0.67	0.02
	70%	0.63	0.52	0.11	0.65	0.7	0.05	0.63	0.56	0.07	0.67	0.63	0.04
	80%	0.63	0.52	0.11	0.7	0.56	0.14	0.63	0.56	0.07	0.66	0.57	0.09
	90%	0.66	0.52	0.14	0.67	0.57	0.1	0.68	0.55	0.13	0.67	0.59	0.08
	Average	0.65	0.59	**0.06**	0.70	0.65	**0.05**	0.67	0.62	**0.05**	0.69	0.65	**0.04**

In general, the accuracy and F1-score of the SSL Random Forest model are higher than that of Naïve Bayes. Third, the difference between the baseline and the SSL model's average accuracy in Random Forest is 0.05, more significant than the Naive Bayes model, 0.06. The difference between the baseline F1-score and the average F1-score of the SSL model in Random Forest is 0.04, which is better than the Naive Bayes SSL model, which is 0.05. This means that Random Forest is better at maintaining the accuracy of the SSL process than Naive Bayes. There is even some accuracy, and the F1 score of the SSL-Random Forest model is higher than the baseline (highlighted).

B. Testing the SSL Model using Market Dataset 2

As same as the previous experiment, four conditions of the Market Dataset 2 are coded *D1*, *D2*, *D3*, and *D4*. We also divided the dataset into training data and test data in a 9:1 ratio. The number of **labeled test data** for each *D1*, *D2*, *D3*, and *D4* is 540 (10% of all Market Dataset 2). The number of **labeled training data** in D1, D2, D3, and D4 are 1080, 540, 270, and 135, respectively. The leftover training data is used as the unlabeled data set.

TABLE 3
ACCURACY AND F1-SCORE OF SSL MODELS ON MARKET DATASET 2

Experiment		Accuracy						F1-score					
		Naïve Bayes			Random Forest			Naïve Bayes			Random Forest		
No	Threshold	Baseline	SSL	Diff	Baseline	SSL	Diff	Baseline	SSL	Diff	Baseline	SSL	Diff
D1	60%	0.87	0.82	0.05	0.85	0.82	0.03	0.87	0.82	0.05	0.85	0.82	0.03
	70%	0.87	0.82	0.05	0.85	0.83	0.02	0.87	0.82	0.05	0.85	0.83	0.02
	80%	0.85	0.75	0.1	0.83	0.77	0.06	0.85	0.75	0.1	0.83	0.76	0.07
	90%	0.85	0.75	0.1	0.83	0.76	0.07	0.85	0.75	0.1	0.83	0.76	0.07
D2	60%	0.81	0.8	0.01	0.81	0.77	0.04	0.81	0.79	0.02	0.81	0.76	0.05
	70%	0.81	0.75	0.06	0.82	0.79	0.03	0.81	0.75	0.06	0.82	0.79	0.03
	80%	0.81	0.77	0.04	0.82	0.71	0.11	0.81	0.77	0.04	0.82	0.69	0.13
	90%	0.81	0.77	0.04	0.82	0.71	0.11	0.81	0.77	0.04	0.82	0.71	0.11
D3	60%	0.83	0.76	0.07	0.81	0.77	0.04	0.83	0.75	0.08	0.81	0.77	0.04
	70%	0.83	0.75	0.08	0.81	0.79	0.02	0.83	0.75	0.08	0.81	0.79	0.02
	80%	0.83	0.7	0.13	0.82	0.67	0.15	0.83	0.69	0.14	0.82	0.67	0.15
	90%	0.83	0.7	0.13	0.81	0.67	0.14	0.83	0.7	0.13	0.81	0.64	0.17
D4	60%	0.82	0.79	0.03	0.79	0.8	0.01	0.82	0.79	0.03	0.79	0.8	0.01
	70%	0.82	0.79	0.03	0.79	0.8	0.01	0.82	0.79	0.03	0.79	0.8	0.01
	80%	0.82	0.71	0.11	0.79	0.52	0.27	0.82	0.71	0.11	0.79	0.38	0.41
	90%	0.82	0.71	0.11	0.78	0.52	0.26	0.82	0.7	0.12	0.78	0.39	0.39
	Average	0.83	0.76	**0.07**	0.81	0.73	**0.08**	0.83	0.76	**0.07**	0.81	0.71	**0.10**

In Table 3, we display the accuracy and F1-score of the baseline and semi-supervised learning (SSL) model in D1, D2, D3, and D4 under different numbers of thresholds, respectively. Table 3 describes 64 SSL-model operations using Market Dataset 2 and gives different results from Market Dataset 1. First, the baseline classification results show that the accuracy score and F1 score are not directly proportional to the number of training data instances. In D2, the accuracy and F1-score are lower than in D3 and D4. The accuracy and F1-score at baseline of the Naïve Bayes models are higher than that of Random Forest. Second, the results of semi-supervised learning classification show that accuracy and F1-score also tend to be linear with the number of training data instances but inversely proportional to the threshold. The threshold also influences the SSL accuracy rate. A low threshold provides high accuracy and a high F1 score because a low threshold will produce more pseudo-labeled datasets than a high threshold. In general, the accuracy and F1-score of the SSL Naïve Bayes model are higher than the Random Forest model. Third, the difference between the baseline and the SSL model's average accuracy in Naïve Bayes is the same as in Random Forest (0,07). The difference between the baseline F1-score and the average F1-score of the SSL model in Naïve Bayes is 0.08, which is better than the Random Forest SSL model, which is 0.1. This means that in the Market Dataset 2, Naïve Bayes is better at maintaining the accuracy of the

SSL process than Random Forest. However, several experiments show that the Random Forest model has more accuracy than the baseline (highlighted).

C. Comparison with Previous Research

We compare our SSL model with previous studies of the same type of machine learning (NB and RF). The performance of the NB model outperformed Balakrishnan et al.'s F1-score (70.5%). In this study, on the Market Dataset 2, the F1-score results reached 76% for NB. It also outperformed the accuracy from [24], whose F1-score results were 57,16 (NB) and 59,34 (RF). In this study, on the Market Dataset 2, the F1-score results reached 0,71 for RF and 0,76 for NB.

IV. CONCLUSION

This study presents an SSL model for sentiment analysis to label Market Data 1 and Market Data 2. In this study, on the Market Dataset 2, the F1-score results reached 0,76 for NB and 0,71 for RF. The results of this study provide several conclusions. The conclusion is that SSL performance highly depends on the number of training data and the compatibility of the features or patterns in the document with machine learning. On Market Data 1, a dataset with three classes, it is better to use Random Forest (F1-score of RF 0,65, and 0,62 for NB). In the Market Data 2 dataset, which consists of two classes, it is better to use Naïve Bayes (F1-score of RF 0,71, and 0,76 for NB). The future research is a sentiment analysis test using SSL on several other datasets and other types of machine learning.

ACKNOWLEDGMENTS

The authors would like to thank the Lembaga Penelitian dan Pengabdian Kepada Masyarakat (LPPM), Universitas Pembangunan Nasional "Veteran" Yogyakarta, Indonesia for their incredible support for this research.

REFERENCES

[1] H. Imaduddin, Widyawan, and S. Fauziati, "Word Embedding Comparison For Indonesian Language Sentiment Analysis," *Proceeding - 2019 Int. Conf. Artif. Intell. Inf. Technol. ICAIIT 2019*, pp. 426–430, 2019.
[2] R. Monika, S. Deivalakshmi, and B. Janet, "Sentiment Analysis of US Airlines Tweets Using LSTM/RNN," *Proc. 2019 IEEE 9th Int. Conf. Adv. Comput. IACC 2019*, pp. 92–95, 2019.
[3] A. H. Abdulhafiz, "Novel opinion mining system for movie rviews in Turkish," *Int. J. Intell. Syst. Appl. Eng.*, vol. 8, no. 2, pp. 94–101, 2020.
[4] D. F. Budiono, A. S. Nugroho, and A. Doewes, "Twitter sentiment analysis of DKI Jakarta's gubernatorial election 2017 with predictive and descriptive approaches," *Proc. - 2017 Int. Conf. Comput. Control. Informatics its Appl. Emerg. Trends Comput. Sci. Eng. IC3INA 2017*, vol. 2018-Janua, pp. 89–94, 2017.
[5] A. Al-Laith, M. Shahbaz, H. F. Alaskar, and A. Rehmat, "Arasencorpus: A semi-supervised approach for sentiment annotation of a large arabic text corpus," *Appl. Sci.*, vol. 11, no. 5, 2021.
[6] V. Balakrishnan, P. Y. Lok, and H. Abdul Rahim, "A semi-supervised approach in detecting sentiment and emotion based on digital payment reviews," *J. Supercomput.*, vol. 77, no. 4, pp. 3795–3810, 2021.
[7] C. R. Aydln and T. Güngör, "Sentiment analysis in Turkish: Supervised, semi-supervised, and unsupervised techniques," *Nat. Lang. Eng.*, vol. 27, no. 4, pp. 455–483, 2021.
[8] V. L. Shan Lee, K. H. Gan, T. P. Tan, and R. Abdullah, "Semi-supervised Learning for Sentiment Classification using Small Number of Labeled Data," *Procedia Comput. Sci.*, vol. 161, pp. 577–584, 2019.
[9] V. L. Shan Lee, K. H. Gan, T. P. Tan, and R. Abdullah, "Semi-supervised Learning for Sentiment Classification Using Small Number of Labeled Data," *Procedia Comput. Sci.*, vol. 161, pp. 577–584, 2019.
[10] R. Alahmary and H. Al-Dossari, "A semiautomatic annotation approach for sentiment analysis," *J. Inf. Sci.*, 2021.
[11] A. Sasmito, H. Basiron, N. Fazilla, and A. Yusof, "Semi-supervised Learning for Sentiment Classification with Ensemble Multi-classifier Approach," *Int. J. Adv. Intell. Informatics*, vol. 8, no. 3, pp. 1–13, 2022.
[12] N. H. Cahyana, S. Saifullah, Y. Fauziah, A. S. Aribowo, and R. Drezewski, "Semi-supervised Text Annotation for Hate Speech Detection using K-Nearest Neighbors and Term Frequency-Inverse Document Frequency," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 10, pp. 147–151, 2022.
[13] S. Mitra and M. Jenamani, "SentiCon: A Concept Based Feature Set for Sentiment Analysis," in *2018 13th International Conference on Industrial and Information Systems, ICIIS 2018 - Proceedings*, 2018, no. 978, pp. 246–250.
[14] I. P. Windasari, F. N. Uzzi, and K. I. Satoto, "Sentiment analysis on Twitter posts: An analysis of positive or negative opinion on GoJek," *Proc. - 2017 4th Int. Conf. Inf. Technol. Comput. Electr. Eng. ICITACEE 2017*, vol. 2018-Janua, pp. 266–269, 2017.
[15] A. S. Aribowo, H. Basiron, N. S. Herman, and S. Khomsah, "An Evaluation of Preprocessing Steps and Tree-based Ensemble Machine Learning for Analysing Sentiment on Indonesian YouTube Comments," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 5, pp. 7078–7086, 2020.
[16] A. N. Farhan and M. L. Khodra, "Sentiment-specific word embedding for Indonesian sentiment analysis," *Proc. - 2017 Int.*

Conf. Adv. Informatics Concepts, Theory Appl. ICAICTA 2017, 2017.

[17] M. Aufar, R. Andreswari, and D. Pramesti, "Sentiment Analysis on Youtube Social Media Using Decision Tree and Random Forest Algorithm: A Case Study," *2020 Int. Conf. Data Sci. Its Appl. ICoDSA 2020*, 2020.

[18] M. A. Fauzi, "Random forest approach fo sentiment analysis in Indonesian language," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 12, no. 1, pp. 46–50, 2018.

[19] Y. Hedge and S. K. Padma, "Sentiment Analysis using Random Forest Ensemble for Mobile Product Review in Kannada," in *2017 IEEE 7th International Advance Computing Conference*, 2017.

[20] S. Khomsah, "Naive Bayes Classifier Optimization on Sentiment Analysis of Hotel Reviews," *J. Penelit. Pos dan Inform.*, vol. 10, no. 2, p. 157, 2020.

[21] 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.

[22] 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.

[23] R. Novendri, A. S. Callista, D. N. Pratama, and C. E. Puspita, "Sentiment analysis of YouTube movie trailer comments using naïve bayes," *Bull. Comput. Sci. Electr. Eng.*, vol. 1, no. 1, pp. 26–32, 2020.

[24] H. B. B. B and M. das G. V. Nunes, "Semi-supervised Sentiment Annotationof Large Corpora," *Comput. Process. Port. Lang.*, pp. 385–395, 2018.

DOI: http://dx.doi.org/10.25139/ijair.v4i2.5267