Analysis of the Impact of Vectorization Methods on Machine Learning-Based Sentiment Analysis of Tweets Regarding Readiness for Offline Learning

Yesi Novaria Kunang[1*], Widya Putri Mentari[2]

[1]*Intelligent Systems Research Group, Faculty of Science and Technology, Universitas Bina Darma, Indonesia*
[1,2]*Information System, Faculty of Science and Technology, Universitas Bina Darma, Indonesia*
*corr_author: yesinovariakunang@binadarma.ac.id

Abstract - Twitter users use social media to express emotions about something, whether it is criticism or praise. Analyzing the opinions or sentiments in the tweets that Twitter users send can identify their emotions for a particular topic. This study aims to determine the impact of vectorization methods on public sentiment analysis regarding the readiness for offline learning in Indonesia during the Covid-19 pandemic. The authors labeled sentiment using two different approaches: manually and automatically using the NLP TextBlob library. We compared the vectorization method used by employing count vectorization, TF-IDF, and a combination of both. The feature vectors were then classified using three classification methods: naïve Bayes, logistic regression, and k-nearest neighbor, for both manual and automatic labeling. To assess the performance of sentiment analysis models, we used accuracy, precision, recall, and F1-score for performance metrics. The best results showed that the Logistic regression classifier with the feature extraction technique that combines count vectorization and TF-IDF provided the best performance for both data with manual and automatic labeling.

Keywords: Naïve Bayes, k-nearest neighbor, logistic regression, sentiment analysis, offline learning

I. INTRODUCTION

In late December 2019, researchers detected the Covid-19 virus in Wuhan City, Hubei Province, China [1]- [2]. The virus's spread has been enormous, global, and massive, affecting public health levels in general and economic, social, psychological, cultural, political, governmental, educational, sports, religious, and other activities [3]. The increasing number of people suffering from Covid-19 is a concern for everyone. As a result, educational institutions ranging from Kindergartens to Universities have implemented online learning due to the Covid-19 pandemic. With the decreasing number of positive Covid-19 cases, the government has implemented a policy for limited face-to-face learning starting in July 2021 [4]. However, this limited offline learning is not the same as regular offline learning, as students and teachers spend limited time.

This rapid digitization era of Industry 5.0 has driven a significant increase in social media users. Twitter is one of the most popular social media platforms, especially in Indonesia [5]. The public often uses Twitter to express opinions on various trending topics [6]-[7]. One of the topics the public has widely discussed is the government's policy regarding limited in-offline learning. During the Covid-19 pandemic, controversy arose due to the enforcement of limited offline learning, and the reactions are visible on various online platforms. The subjective nature of the vast number of public opinions on readiness for offline learning on Twitter makes it very interesting to conduct a sentiment analysis of the public's views on the topic. Therefore, the present study will analyze tweets collected using some keyword variations. In the labeling stage, this study will perform labeling using two methods: manual and using the Python TextBlob library [8], which will categorize sentiment into positive, negative, and neutral. From the labeled data based on these two labeling techniques, they will be used as training data for a classifier model based on naive Bayes, logistic regression, and k-nearest neighbor (kNN) algorithms.

Currently, many researchers conduct studies related to sentiment analysis. Sentiment analysis or opinion mining is the process of automatically understanding, extracting, and processing textual data to obtain sentiment information in an opinion sentence [9]-[10]. We can use sentiment analysis to measure public opinion about issues such as the resumption of offline learning. Recent research conducts a related study investigating public opinion in Indonesia regarding the new normal period of Covid-19 [11]. This study analyzed 1000 tweets classified using the kNN method. The highest accuracy score was 94.5%. They simply categorized

positive and negative categories and applied automatic labeling methods referring to sentiment scores from previous lexical studies. The weakness of this labeling method is that the lexicon utilized, usually derived from translated lexical dictionaries of other languages, may not be appropriate for the context being studied [12]. Another study used the kNN algorithm combined with BM25 weighting [13] and TF-IDF [14]. In sentiment analysis cases, the kNN algorithm is widely used because it has the advantage of not being affected by the distribution of each class. The kNN algorithm determines the sentiment classifications by determining the nearest distance between the test and training samples [15].

The sentiment analysis for Covid tweets has also been conducted by several other researchers [2], [7], [10], [16]. Recent research collects and categorizes Covid-19 tweets into five categories: positive, negative, extremely positive, extremely negative, and neutral [7]. They apply pre-processing and text vectorization processes before predicting the sentiment using naïve Bayes and logistic regression algorithms. The results showed that the logistic regression algorithm performed better. Another study [16] analyzes sentiment in Covid-19 tweets using deep neural networks. They improve the accuracy using two-word embedding techniques: count vectorizer (CV) and term frequency-inverse document frequency (TF-IDF). Support vector machine (SVM), Bernoulli naïve Bayes, single-layer perceptron (SLP), multi-layer Perceptron (MLP), and logistic regression (LR) are used as classifiers. They conclude that TF-IDF is more efficient than CV for larger datasets.

Researchers have widely used the naïve Bayes algorithm for sentiment analysis, such as [5], who conducted sentiment analysis regarding implementing community activity restrictions for negative and positive classes. This study has limited data and only focuses on sentiment polarity. The labeling process occurs after pre-processing based on a dictionary of sensational words. The weaknesses of the labeling process after the pre-processing eliminate the emotional detection of tweet comments [12]. Another study [17] also uses naive Bayes to classify public perceptions towards the government. The labeling process is conducted manually to classify two classes. However, this research's limitation lies in its minimal data utilization.

As noted in [2], [16], [18]-[19], several researchers focus on the impact of word-to-vector techniques in natural language processing (NLP). For example, researchers compare TF-IDF and count vectorization as feature extraction techniques [18]. They used LR, SVM, NN, and decision trees as classifiers and found that the extraction of TF-IDF features is slightly better. Similarly, the analysis of the Bangla text is done in this way [19]. They use count vectorization, TF-IDF, and various combinations and then classify using LR, NB, and MLP. The findings indicate that LR performs better when coupled with TF-IDF.

A more comprehensive analysis approach examines sentiment in Covid-19 vaccine hesitancy tweets [2]. They labeled the tweets using three computational methods (Azure machine learning, VADER, and TextBlob). The analyzed vectorization techniques include Doc2Vec, CV, TF-IDF, and various combinations. They classify the sentiment using random forest, logistic regression, decision tree, linear SVC, and naive Bayes. Results show that combining TextBlob's emotions with TF-IDF vectorization and linear SVC has surpassed other methods. However, automatic feelings scores also depend significantly on discussion and linguistic diversity.

Based on previous research, this study focuses on the following gaps: 1) the performance of model extraction methods in sentiment analysis models using the Indonesian dataset; 2) the availability of detailed tests that compare manual and automatic labeling techniques with various vectorization methods, especially for the Indonesian dataset.

In this study, we want to investigate the effect of vectorization methods for pre-processing Indonesian language tweet datasets related to readiness for offline learning in the Indonesian language. The focus of this research is to compare feature extraction pre-processing techniques using three-word encoding techniques to convert text data into numbers, namely count vectorizer (CV) or also known as bag of words, term frequency-inverse document frequency (TF-IDF), and the combination of CV and TF-IDF (CV+TF-IDF). The classification algorithms used are naïve Bayes, logistic regression, and kNN for sentiment analysis. Furthermore, this study will analyze the variant method with two different sentiment labeling techniques to evaluate their impact on performances. Two labeling methods are used to investigate the emotional aspects of the labeling process: manual labeling and sentiment scores from a lexical dictionary. The research will provide valuable information about the effectiveness and performance of each method by analyzing the performance of three classification algorithms with different vectorization variants.

II. METHOD

Fig. 1 shows the research flow for sentiment analysis. In this case, we conduct several stages, beginning with

collecting data from the social media platform Twitter. Data used in this study consists of the opinions of Twitter users regarding the government's policy to resume offline learning. From 15 May 2022 to 26 May 2022, we collected public responses to the offline learning policy during the Covid-19 pandemic. The collected tweets amounted to 14,298 tweets. The data collected were tweets with predetermined keywords related to offline learning during Covid-19 in the Indonesian language. In this study, data crawling was carried out using the keywords "belajar tatap muka," "kuliah offline," "sekolah offline," and "luring." The crawling process on Twitter was performed using the Twitter API using the tweepy library [20]. There was no limit on the amount of data collected. In the pre-processing stage, we only utilized the comment/tweet data after crawling and did not include other information such as user, geolocation, and others. Thus, we ensure that the dataset used in the study protects the individual privacy of the analyzed tweets. We combined the collected data into a CSV file. The next step was to perform data cleaning to remove duplicate data. The tweet data has many duplicate data due to the retweeting process. Therefore, after cleaning, the original data of 14,298 was reduced to 8,731.

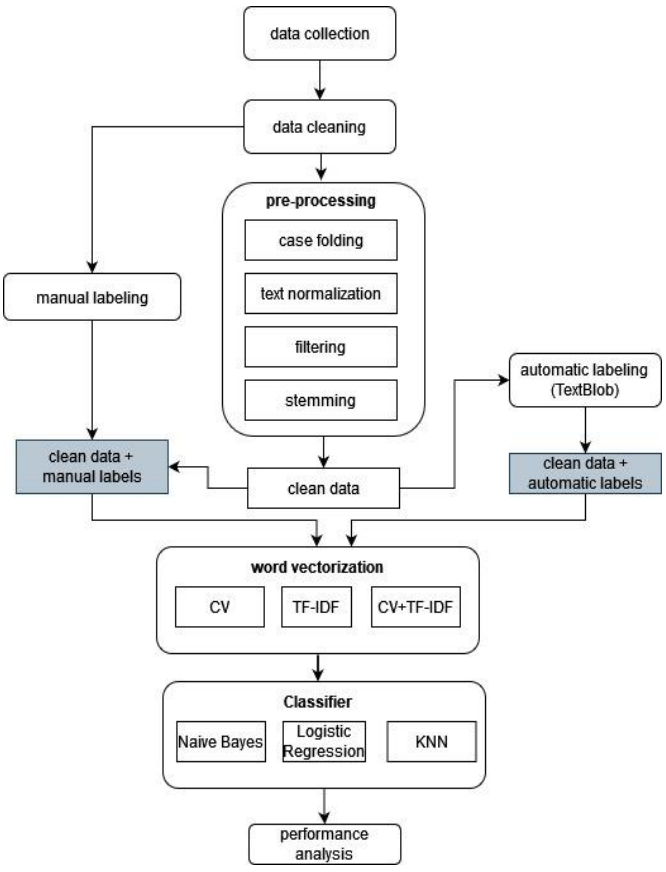

Fig. 1 Flow process in sentiment analysis

The most critical stage in NLP was data pre-processing. Data pre-processing was crucial because Indonesian tweets contain non-standard words and symbols that would not be processed in the next stage. We conducted pre-processing using Python-provided libraries. The steps performed in pre-processing using Python were case folding, text normalization, filtering, and stemming.

- The case folding process is an automated method for tweet data conversion using the library re in Python to convert tweet data to lowercase and remove non-letter characters such as numbers and punctuation marks.
- The next stage was the text normalization function, replacing typos, abbreviations, foreign terms, and slang with standard words. This process is done by creating a word list of words to be normalized.
- Filtering or stopword removal is the process that removes meaningless or unnecessary words. In this study, the stop word filtering process for the Indonesian language was done using the NLTK (Natural Language ToolKit) library.
- The last step was the Stemming process, which converts words to their primary form by removing affixes. The stemming process uses the Sastrawi library [21].

These pre-processing steps were likely chosen for this study to ensure the text data's quality and consistency. By applying these steps, we aimed to improve the performance of the sentiment analysis model by reducing noise and increasing the model's ability to recognize underlying patterns and semantics in the text.

In this research, we conducted the labeling process using two methods: manual labeling and automatic labeling. The first method, manual labeling, was carried out before the data was pre-processed (raw). This process involved respondents reading the entire content of the comments. The study utilized three informants for manual labeling, assessing the sufficiency of information using informant determination techniques in qualitative research [22]. Respondents provided labels based on the context of the comments. With human perceptual capabilities, respondents were expected to capture the emotions in the tweeted comments, especially in the context of the Indonesian language, which may have unique linguistic and cultural nuances. Then the dominant comment was taken as the label. We expected this to yield a relatively high precision result. The process of labeling raw data by respondents had the advantage that humans would pay attention to the aspects of sentiment analysis [23] and emotion detection [24] in the comments.

The automatic labeling process was assisted by the TextBlob Python library [8]. On pre-processed data, we used the automatic labeling process. This automatic labeling process reviewed the polarity and subjectivity in the tweet data. Based on the polarity, tweets were then categorized into three classes: positive sentiment, negative sentiment, and neutral sentiment.

This research aims to achieve the best accuracy by combining word-to-vector techniques and several classifiers. Three approaches were tested in the vectorization method: count vectorization (CV), word weighting using term frequency-inverse document frequency (TF-IDF), and combining both CV+TF-IDF. CV and TF-IDF and combination are commonly used in natural language processing (NLP) tasks like text classification and information retrieval [16], [18], [19]. The three methods work, and the rationale behind selecting each method is explained as follows:

- Count Vectorizer, also known as bag of words, is a technique that transforms words into vectors by counting the frequency of each word in each tweet. The CV approach converts a textual document collection into a matrix containing word frequency. The frequency of occurrence indicates the relevance of the corresponding word, with higher frequency indicating high significance towards the sentiment of a class. In this research, we used the CountVectorizer from scikit-learn [25].
- On the other hand, TF-IDF is a technique used to calculate the relative frequency of a word in a data or group of data. This technique considers a word's inverse frequency of occurrence to reduce the domination of frequently appearing words. TF-IDF is a strategy considering high frequency may not provide substantial information benefits. In other words, uncommon words may give additional weight to the model. We used the TF-IDF from scikit-learn [25] in our research. Frequency refers to how often a specific word appears in a tweet comment. In contrast, the inverse document frequency considers all tweets containing that word. Mathematically, TF-IDF can be calculated using the following (1).

$$tf - idf\ (t,d,D) = idf\ (t,D) * tf\ (t,d) \quad (1)$$

Where $idf\ (t,D)$ indicates how common or rare a word t is in all documents or all tweet comments D, while $tf\ (t,d)$ is the frequency of the word t in a single tweet comment d. The relevance of a term increases proportionally with its frequency in the same tweet comment but decreases proportionally with the summarization of words in the entire corpus data set.

- Combination of Count Vectorizer and TF-IDF: In some cases, a combination of CV and TF-IDF can be beneficial. This approach involves using Count Vectorizer to convert the text into a matrix of word frequencies and then applying TF-IDF to that matrix.

The choice of vectorization method depends on the specific task and the characteristics of the text data. CV is a simple baseline method, TF-IDF considers word importance, and combining both can capture both frequency and importance aspects. Experimenting with different vectorization techniques in sentiment analysis allows us to find the most suitable method to achieve optimal performance in sentiment analysis.

The three data sets that have been transformed into vectors using the three-word vectorization methods were then classified using three algorithms: naïve Bayes (NB), logistic regression (LR), and k-nearest neighbors (kNN). NB, LR, and kNN are classification algorithms for sentiment analysis based on their characteristics, performance, and suitability for text classification tasks. Naïve Bayes is a probabilistic algorithm that works well for text data and is computationally efficient [17], [26]. Logistic regression is a simple yet powerful linear classification algorithm used in binary classification tasks, particularly sentiment analysis [26]. KNN is a versatile non-parametric instance-based classification algorithm that captures complex relationships in data and is effective when sentiment is influenced by proximity to similar texts [27]. Through the conducted experiments, we aim to assess how each of these algorithms contributes to the accuracy of sentiment analysis. This evaluation was contingent upon various factors, including data characteristics, particularly concerning the Indonesian tweets dataset, and the complexity of the individual algorithms. The evaluation of these algorithms depends on data characteristics and the complexity of the individual algorithms.

We aimed to achieve the highest possible accuracy of machine learning algorithms after input of word vectors into a combination of classifiers. To evaluate the consistency of the models, we conducted experiments by modifying the composition of the training and testing datasets. Then, the pre-processed data will be trained using vectorization methods to recognize sentiment analysis with variations in training and test datasets. We conducted this testing variation to ascertain the vectorization method's stability with each algorithm for the sentiment analysis dataset.

We evaluated the sentiment analysis model using accuracy, precision, recall, and F1-score performance metrics. Accuracy measures the proportion of correctly predicted instances but may not be optimal in

imbalanced classes like our case (see Figures 2a and 2b). Precision measures the number of predicted instances for a class, recall measures the model's capture of instances, and F1-score balances these metrics. Precision is essential when the cost of misclassifying a specific sentiment is high, and recall is crucial when missing instances of a specific sentiment have a high cost. It is beneficial when classes are imbalanced.

III. RESULT AND DISCUSSION

A. Data Labeling

We conducted the labeling process using two methods, manual labeling and automatic labeling using TextBlob, to observe how the labeling process affects the classification results. Three respondents conducted the manual labeling process to determine whether the tweet comments were positive, negative, or neutral. Sentiment determination was based on the majority choice of the respondents. From 8,731 raw data, we can see the manual labeling results in Table I and Fig. 2(a), where 5,946 were positive sentiment, 2,070 as negative sentiment, and 715 as neutral sentiment.

Different results were obtained by utilizing the Python TextBlob library. This automatic labeling process calculated the polarity and subjectivity values of the words in a tweet's text. The sentiment was considered positive if the polarity value tends toward 1, but if the polarity value tends toward -1, the sentiment is classified as negative, and if the polarity value is around 0, the sentiment is considered neutral. The results of automatic labeling in Table I and Fig. 2(b) show that there were 246 positive sentiments, 155 negative sentiments, and 8330 neutral sentiments.

Automatic labeling results show that the opinions generated tend toward neutrality. The result is reasonable, as TextBlob itself will ignore unknowable words. It will consider words and phrases that can be assigned polarity values and average them to obtain the final score. TextBlob calculates polarity based on a lexicon dictionary that contains word rules and weight dictionaries. This method had weaknesses because words not in the rules would be ignored. In addition, the weight of a word itself depends heavily on the topic and case that may not necessarily be covered in the lexicon dictionary. Therefore, this TextBlob labeling method produced a dominant neutral label, particularly in sentiment analysis cases in Indonesian language research, such as in the following studies [28]–[30]. The difference in the number of labeled data between manual and automatic labeling methods affected the number of words in each sentiment class.

B. Classification and Evaluation

In this section, we investigate the effectiveness of three vectorization methods used with three machine learning algorithms, namely naive Bayes (NB), logistic regression (LR), and k-nearest neighbor (kNN), for distinguishing tweet sentiments. We used accuracy value, precision score, recall score, and F1-measure to determine the model's performance. In order to investigate three vectorization methods, we utilized count vectorization (CV), TF-IDF, and a combination of CV and TF-IDF for the pre-processing stage, which was then evaluated with classifiers to obtain the best results. The evaluation was gradually carried out with the combination of training and test data, i.e., 90%, 80%, 70%, 60%, and 50%, to evaluate the impact of the number of training data.

TABLE 1
NUMBER OF TWEETS FOR EACH SENTIMENT IN THE DATASET AFTER LABELING

Sentiment	Manual Labeling	TextBlob Labeling
positive	5946	246
neutral	2070	8330
negative	715	155

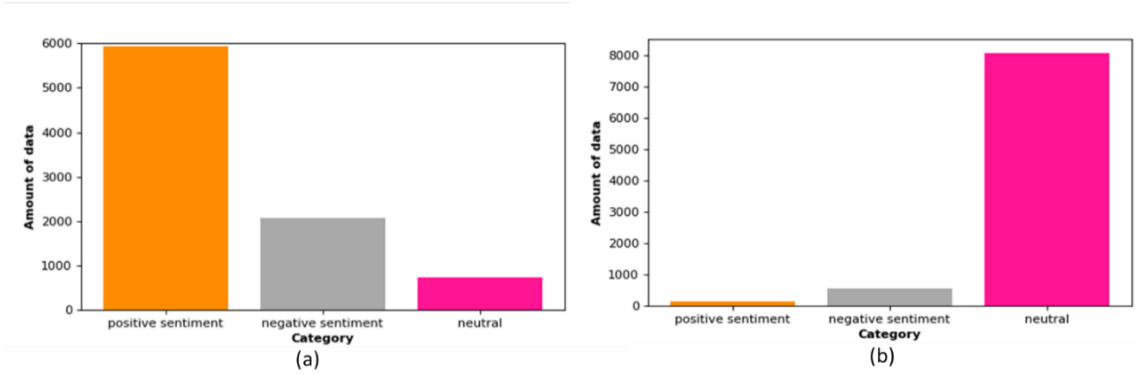

Fig. 2 Representation of sentiment labeling results: (a) manual labeling, (b) labeling with TextBlob

Table II compares the effectiveness of CV, TF-IDF, and CV+TF-IDF for the three classifiers NB, LR, and kNN for data with manual labeling. Parameter tuning of k (from k = 1 to k = 7) was explicitly performed for kNN algorithm to obtain the best results. We differentiate text with a bold font for the best performance of each training data. For data with manual labeling, the LR classifier provided better performance for sentiment analysis. Moreover, kNN classifier had poorer performance than NB and LR. The combination of CV + TFIDF indicates that the vectorization method gave almost consistently better performance than the CV and TF-IDF pre-processing methods for NB and LR classifiers. However, different results for kNN algorithm, where kNN and TF-IDF were combined, gave better results.

We achieved the best performance for manually labeled data with precision at 80.498%, recall at 81.007%, and F1-score at 80.181%. The best result using the CV + TF-IDF vectorization method was the LR classifier with a training data ratio of 90% and a testing data ratio of 10%.

Table III summarizes the comparison of combined vectorization methods and machine learning algorithms for data labeled with the TextBlob method. Combining the LR algorithm with the CV vectorization method and TF-IDF, we obtained the best performance with a training data ratio of 80% and a testing data ratio of 20%. The best performance achieved was 97.229% for precision, 97.195% for recall, and 96.605% for F1-score. The LR algorithm consistently outperformed NB and kNN as in the manually labeled data. The TF-IDF+ CV vectorization method resulted in better performance when combined with the LR algorithm, while the NB algorithm always performed better with the CV vectorization method. In contrast, kNN algorithm showed better performance with the TF-IDF vectorization method, especially with small training data.

Fig. 6 and 7 show the accuracy values or weighted average accuracy or recall of all models for data with manual labeling and labeling using TextBlob, respectively. From both graphs, the LR algorithm in orange consistently gives the best accuracy performance for both datasets. Moreover, the LR and NB classifiers with the CV and CV+TF-IDF methods show better performance impact. On the other hand, the kNN algorithm with the TF-IDF method shows the most suitable performance impact for kNN classifier.

TABLE II
PERFORMANCE COMPARISON OF COMBINED MODELS FOR DATA WITH MANUAL LABELING

Training Data	Method	Precision	Recall	F1-Score	Method	Precision	Recall	F1-Score	Method	k	Precision	Recall	F1-Score
	NB+CV	0.78748	0.79176	0.76759	LR+CV	0.80079	0.80549	0.79506	KNN+CV	k=4	0.78462	0.76628	0.77534
90%	NB+TFIDF	0.81898	0.754	0.6877	LR+TFIDF	0.79111	0.79405	0.77968	KNN+TFIDF	k=7	0.76125	0.77117	0.75863
	NB+TFIDF+CV	0.80084	0.79863	0.77331	LR+TFIDF+CV	**0.80498**	**0.81007**	**0.80181**	KNN+TFIDF+CV	k=7	0.75148	0.75515	0.7318
	NB+CV	0.79995	0.79222	0.76158	LR+CV	0.7949	**0.79908**	**0.78554**	KNN+CV	k=3	0.72834	0.72868	0.71365
80%	NB+TFIDF	0.78785	0.7344	0.66088	LR+TFIDF	**0.79953**	0.79508	0.77506	KNN+TFIDF	k=5	0.76482	0.77218	0.76457
	NB+TFIDF+CV	0.79401	0.78935	0.75973	LR+TFIDF+CV	0.79189	0.79737	0.78473	KNN+TFIDF+CV	k=5	0.73314	0.73784	0.71783
	NB+CV	0.78333	0.78626	0.75835	LR+CV	**0.80299**	**0.80725**	**0.79594**	KNN+CV	k=3	0.71854	0.70725	0.70156
70%	NB+TFIDF	0.8098	0.74198	0.66997	LR+TFIDF	0.79031	0.79008	0.77195	KNN+TFIDF	k=7	0.75277	0.76031	0.7511
	NB+TFIDF+CV	0.78713	0.78893	0.76198	LR+TFIDF+CV	0.79866	0.80382	0.79548	KNN+TFIDF+CV	k=5	0.72731	0.73206	0.71176
	NB+CV	0.79094	0.79158	0.76434	LR+CV	0.79622	0.80074	0.78782	KNN+CV	k=7	0.6947	0.69997	0.67847
60%	NB+TFIDF	0.80535	0.74005	0.66591	LR+TFIDF	0.78385	0.78528	0.76604	KNN+TFIDF	k=7	0.75428	0.76381	0.75248
	NB+TFIDF+CV	0.79334	0.79216	0.76500	LR+TFIDF+CV	**0.79967**	**0.80418**	**0.79341**	KNN+TFIDF+CV	k=7	0.72677	0.73375	0.7029
	NB+CV	0.78140	0.77989	0.74829	LR+CV	0.78557	0.78882	0.77386	KNN+CV	k=7	0.6832	0.69469	0.66128
50%	NB+TFIDF	0.78432	0.72767	0.64951	LR+TFIDF	0.77414	0.77325	0.75039	KNN+TFIDF	k=7	0.74429	0.75378	0.74082
	NB+TFIDF+CV	0.78641	0.78218	0.75087	LR+TFIDF+CV	**0.78827**	**0.79180**	**0.77830**	KNN+TFIDF+CV	k=7	0.72438	0.72904	0.69361

TABLE III
PERFORMANCE COMPARISON OF COMBINED MODELS FOR DATA USING TEXTBLOB LABELING

Training Data	Method	Precision	Recall	F1-Score	Method	Precision	Recall	F1-Score	Method	k	Precision	Recall	F1-Score
	NB+CV	0.91962	0.93478	0.90982	LR+CV	0.96904	0.96796	0.96030	KNN+CV	k=3	0.91962	0.93478	0.90982
90%	NB+TFIDF	0.85679	0.92563	0.88988	LR+TFIDF	0.93902	0.95652	0.94393	KNN+TFIDF	k=7	0.91565	0.93021	0.90045
	NB+TFIDF+CV	0.91565	0.93021	0.90045	LR+TFIDF+CV	**0.96904**	**0.96796**	**0.96030**	KNN+TFIDF+CV	k=7	0.91863	0.93364	0.90758
	NB+CV	0.91885	0.92387	0.89522	LR+CV	0.97173	0.97138	0.96542	KNN+CV	k=1	0.91207	0.92845	0.90552
80%	NB+TFIDF	0.90806	0.91586	0.87620	LR+TFIDF	0.94292	0.95707	0.94662	KNN+TFIDF	k=7	0.91883	0.93188	0.90635
	NB+TFIDF+CV	0.92223	0.92330	0.89389	LR+TFIDF+CV	**0.97229**	**0.97195**	**0.96605**	KNN+TFIDF+CV	k=3	0.91497	0.93246	0.90872
	NB+CV	0.93148	0.93092	0.90494	LR+CV	0.96687	0.96603	0.95829	KNN+CV	k=2	0.92116	0.92443	0.89352
70%	NB+TFIDF	0.91173	0.92252	0.88572	LR+TFIDF	0.93991	0.95496	0.94286	KNN+TFIDF	k=7	0.93639	0.93359	0.91000
	NB+TFIDF+CV	0.93214	0.92977	0.90254	LR+TFIDF+CV	**0.96869**	**0.96794**	**0.96051**	KNN+TFIDF+CV	k=5	0.92126	0.93359	0.90937
	NB+CV	0.93102	0.93100	0.90498	LR+CV	0.96561	0.96479	0.95720	KNN+CV	k=1	0.91281	0.92814	0.90939
60%	NB+TFIDF	0.85138	0.92270	0.88561	LR+TFIDF	0.93813	0.95305	0.94022	KNN+TFIDF	k=5	0.93469	0.93444	0.91210
	NB+TFIDF+CV	0.93055	0.92957	0.90202	LR+TFIDF+CV	**0.96658**	**0.96565**	**0.95823**	KNN+TFIDF+CV	k=3	0.91476	0.93330	0.91019
	NB+CV	0.93103	0.93083	0.90388	LR+CV	0.96499	0.96404	0.95554	KNN+CV	k=1	0.91558	0.93014	0.91281
50%	NB+TFIDF	0.85327	0.92373	0.88711	LR+TFIDF	0.93820	0.95190	0.93874	KNN+TFIDF	k=3	0.92540	0.93610	0.91795
	NB+TFIDF+CV	0.93015	0.92923	0.90047	LR+TFIDF+CV	**0.96650**	**0.96564**	**0.95836**	KNN+TFIDF+CV	k=3	0.91973	0.93541	0.91313

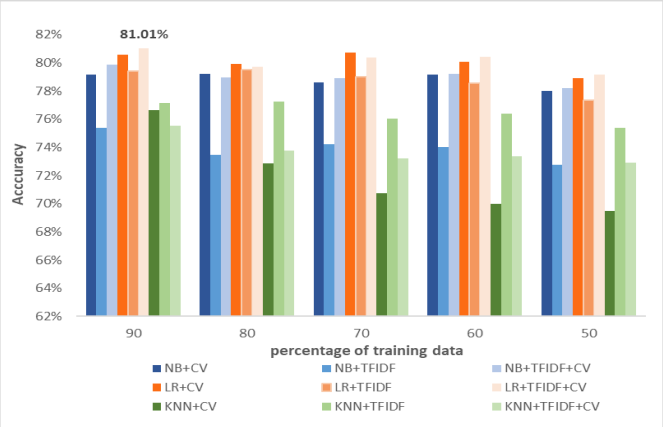

Fig. 6 Comparison of weighted average accuracy values for various model combinations for manual labeling

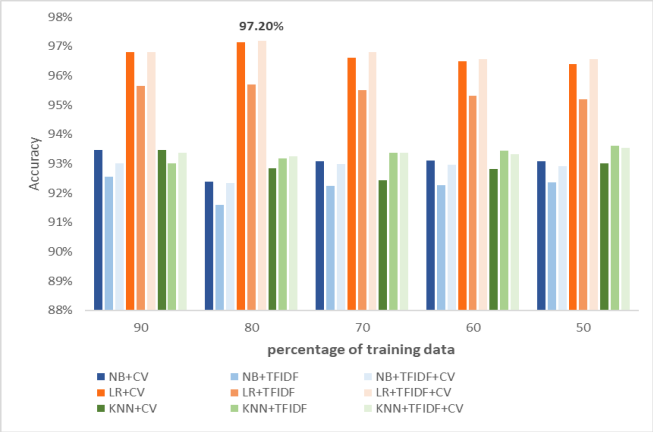

Fig. 7 Comparison of weighted average accuracy values for various model combinations for labeling with TextBlob

Tables IV and V show the detailed performance for each sentiment class of the best LR and CV+TFIDF models for data with manual and TextBlob labeling, respectively. For data with manual labeling, the positive class, which accounts for around 68% of all tweet data (Table 1), shows good performance values, especially for recall. This value indicates that the model effectively identifies positive sentiment class and can provide accurate predictions for the majority class. However, the recall value for the negative sentiment class was only 55.122%. Several factors, including the complexity of sentiment analysis tasks, may cause low recall values for sentiment analysis. Sentiment analysis involves identifying feelings or emotions in tweet texts that are sometimes lengthy and diverse. The emotions create a challenge and complexity of sentiment analysis, as many factors influence interpreting one's feelings or emotions in a text. Another factor that affects the results with manual labeling is the quality of the dataset used. In its ineffective recognition of some different sentiment types, the model can be affected by an unbalanced or inadequately representative dataset of the text variations in the language used.

The prediction results for the data labeled using TextBlob show that the model was not very good at predicting the negative sentiment class. In this case, a precision of 100% for the negative sentiment class indicates that the model correctly predicted all examples in the negative sentiment class. In other words, all predictions made by the model as negative sentiments are truly negative sentiments. However, a recall of 0.11538 indicated that the model could only find 11.5% of all actual negative sentiment examples. This value means the model only found a small portion of the negative sentiment examples in the dataset, even though its predictions were always correct. In the context of the amount of data with negative sentiment, which was only 1.78% of the total data (Table I), a precision value of 1.00 and low recall indicate that the model could provide highly accurate predictions for the neutral class but is less effective in identifying rare cases like the negative sentiment class.

The results show a significant difference between manually labeled data and those labeled using TextBlob. It is reasonable, as shown in a previous study [31], which also showed that sentiment labeling with TextBlob produced some labels that differ from the actual labels. TextBlob has some limitations as a lexicon-based sentiment analysis library, such as some words that can mean positive or negative depending on the domain. On the other hand, in manual labeling, humans have different emotions to label and assess tweet sentiment.

TABLE IV
PRECISION, RECALL, AND THE F1-SCORE RESULT OF LR USING CV+TFIDF IN 10% MANUAL LABELING DATA TEST

Sentiment	Precision	Recall	F1-Score	Support
negative	0.72903	0.55122	0.62778	205
neutral	0.84906	0.64286	0.73171	70
positive	0.82583	0.9182	0.86957	599

TABLE V
PRECISION, RECALL, AND F1-SCORE RESULT OF LR USING CV+TFIDF IN 20% TEXTBLOB LABELING DATA TEST

Sentiment	Precision	Recall	F1-Score	support
negative	1.0000	0.11538	0.20690	26
neutral	0.97204	1.0000	0.98582	1599
positive	0.96970	0.78689	0.86878	122

Additionally, words containing sarcasm are challenging to analyze by machines [32]. However, overall, the performance obtained by combining vectorization methods and machine learning models for both labeling techniques gave the same performance pattern.

The evaluation shows that combining CV+TF-IDF produces better performance when using LR classifiers. Although the complexity of combining CV + TF-IDF resulted in longer computation time due to the doubled vector dimensionality in comparison with CV or TF-IDF alone. The results differ slightly from those obtained in [19] and [2], indicating that TF-IDF performs better than CV+TF-IDF. This difference may be due to the dataset used, especially the language and pre-processing libraries. This study's dataset was highly imbalanced, with manual labeling and TextBlob. The imbalances in the datasets impacted the results obtained, especially for the minor classes. However, the results show that LR is a machine-learning algorithm that works well in sentiment analysis.

Our research employs a more detailed approach than other sentiment analysis studies in the Indonesian context, which primarily rely on sentiment library methods based on lexicon dictionaries, as seen in studies [5]-[6], [9], [11], [28], [30]. The use of lexicon dictionaries in sentiment analysis is subjective and heavily dependent on the dictionary used, which is observed in the sentiment analysis of political election issues in Indonesia [30]. The labeling outcomes varied when three lexicon libraries were used –TextBlob, VaderSentiment, and SentiWordNet [30]. Our study evaluated sentiment analysis labeling using library methods and incorporated manual labeling that captures emotions and expressions in the comments. Additionally, while most sentiment analysis studies utilize the TF-IDF technique [6], [11] for vectorization, our research analyzes three techniques. The results show that combining the CV with TF-IDF could enhance sentiment classifier performance. By comparing three machine learning techniques, our methodology demonstrated that the logistic regression (LR) model offers robust performance for Indonesian language datasets. These conclusions can potentially guide upcoming sentiment analysis investigations concerning Indonesian language datasets.

The research findings can potentially inform educational policymakers about public perceptions of readiness for offline learning. Positive sentiments may indicate support for offline learning initiatives, while negative sentiments could highlight areas for improvement. Policymakers can use this data to make tailored decisions, allocate resources, and address concerns for effective implementation. Understanding public sentiment aids organizations and governments in making informed decisions, adjusting strategies, and engaging with audiences. Analyzing sentiment on platforms like Twitter provides real-time feedback on initiatives and policies. Educational institutions can monitor sentiment for issues, feedback, and improvements.

Furthermore, our study explores the impact of different vectorization methods on sentiment analysis accuracy. These findings can guide future sentiment analysis projects in various sectors. For instance, if specific vectorization methods consistently perform better, they can be recommended for sentiment analysis projects in areas such as brand reputation management, customer feedback analysis, and beyond.

However, there are several things to consider in this research. First, data limitations can affect the accuracy of the analysis results. Therefore, ensuring that the collected data is sufficiently representative and varied is essential. Second, using TextBlob for automatic labeling may not be accurate for certain cases, especially when dealing with complex languages or different domains and topics. Therefore, it must be further tested to ensure its accuracy in the desired context. Third, the testing results should be confirmed through cross-validation methods to ensure the accuracy and consistency of the results.

IV. CONCLUSION

This study evaluated count vectorization, TF-IDF, and their combination vectorization methods combined with machine learning algorithms, namely naive Bayes, logistic regression, and kNN. After conducting the study, it can be concluded that the vectorization method significantly used affects the performance achieved. The count vectorization method combined with TF-IDF improves performance for classifiers, especially for Logistic Regression classifiers, which perform an accuracy of 81.01% for manual labeling and 97.20% for labeling with TextBlob. Likewise, for other performance measures such as precision, recall, and F1-score, the combination of CV+TF-IDF and Logistic Regression delivers the best overall results. The performance obtained by the combined model for the vectorization method and machine learning models for both labeling techniques produces the same performance pattern.

Furthermore, to conduct further research, researchers can evaluate the vectorization method's suitability using several other Indonesian sentiment analysis datasets, as the classification algorithm significantly affects the

method's effectiveness. Additionally, researchers can develop deep learning approaches with various vectorization methods to produce more detailed NLP analysis, particularly for the Indonesian language.

REFERENCES

[1] A. Kumar *et al.*, "Wuhan to world: the COVID-19 pandemic," *Front. Cell. Infect. Microbiol.*, p. 242, 2021.

[2] M. Qorib, T. Oladunni, M. Denis, E. Ososanya, and P. Cotae, "Covid-19 vaccine hesitancy: Text mining, sentiment analysis and machine learning on COVID-19 vaccination Twitter dataset," *Expert Syst. Appl.*, vol. 212, p. 118715, Feb. 2023, doi: 10.1016/j.eswa.2022.118715.

[3] R. A. Utami, R. E. Mose, and M. Martini, "Pengetahuan, Sikap dan Keterampilan Masyarakat dalam Pencegahan COVID-19 di DKI Jakarta," *J. Kesehat. Holist.*, vol. 4, no. 2, pp. 68–77, Jul. 2020, doi: 10.33377/jkh.v4i2.85.

[4] Hendriyanto, "Pembelajaran Tatap Muka Dilaksanakan Secara Terbatas," *Direktorat Sekolah Dasar Direktorat Jenderal PAUD Dikdas dan Dikmen Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi*, Jun. 09, 2021. https://ditpsd.kemdikbud.go.id/public/artikel/detail/pembelajaran-tatap-muka-dilaksanakan-secara-terbatas (accessed Sep. 09, 2022).

[5] T. Krisdiyanto, "Analisis Sentimen Opini Masyarakat Indonesia Terhadap Kebijakan PPKM pada Media Sosial Twitter Menggunakan Naïve Bayes Clasifiers," *J. CoreIT J. Has. Penelit. Ilmu Komput. Dan Teknol. Inf.*, vol. 7, no. 1, p. 32, Jul. 2021, doi: 10.24014/coreit.v7i1.12945.

[6] N. D. Mentari, M. A. Fauzi, and L. Muflikhah, "Analisis Sentimen Kurikulum 2013 Pada Sosial Media Twitter Menggunakan Metode K-Nearest Neighbor dan Feature Selection Query Expansion Ranking," *J. Pengemb. Teknol. Inf. Dan Ilmu Komput. E-ISSN*, vol. 2548, p. 964X, 2018.

[7] D. Devarapalli, M. S. Sri, P. K. Sri, P. Charishma, and P. V. N. Mounika, "Sentiment Analysis of COVID-19 Tweets Using Classification Algorithms," in *Innovations in Computer Science and Engineering: Proceedings of the Ninth ICICSE, 2021*, Springer, 2022, pp. 395–405.

[8] Steven Loria., "TextBlob: Simplified Text Processing," *TextBlob: Simplified Text Processing*, 2020. https://textblob.readthedocs.io/en/dev/

[9] G. A. Buntoro, "Analisis sentimen hatespeech pada twitter dengan metode naïve bayes classifier dan support vector machine," *J. Din. Inform.*, vol. 5, no. 2, pp. 1–21, 2016.

[10] P. Balakesava Reddy, S. Ramasubbareddy, G. Viswanath, and K. Govinda, "Sentiment Analysis of Tweets Related to COVID-19," in *Innovations in Computer Science and Engineering: Proceedings of the Ninth ICICSE, 2021*, Springer, 2022, pp. 385–393.

[11] M. Furqan, S. Sriani, and S. M. Sari, "Analisis Sentimen Menggunakan K-Nearest Neighbor Terhadap New Normal Masa Covid-19 Di Indonesia," *Techno.Com*, vol. 21, no. 1, pp. 51–60, Feb. 2022, doi: 10.33633/tc.v21i1.5446.

[12] A. Hamzah, "Lexicon-based Emotion Detection for Academic Questionnaire Results," in *Seminar Nasional Informatika (SEMNASIF)*, 2021, pp. 37–49.

[13] I. D. Onantya, Indriati, and P. P. Adikara, "Analisis Sentimen Pada Ulasan Aplikasi BCA Mobile Menggunakan BM25 Dan Improved K-Nearest Neighbor," *J. Pengemb. Teknol. Inf. Dan Ilmu Komput.*, vol. 3, no. 3, pp. 2575–2580, 2019.

[14] J. A. Septian, T. M. Fachrudin, and A. Nugroho, "Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TF-IDF dan K-Nearest Neighbor," *J. Intell. Syst. Comput.*, vol. 1, no. 1, pp. 43–49, Aug. 2019, doi: 10.52985/insyst.v1i1.36.

[15] A. Budianto, R. Ariyuana, and D. Maryono, "PERBANDINGAN K-NEAREST NEIGHBOR (KNN) DAN SUPPORT VECTOR MACHINE (SVM) DALAM PENGENALAN KARAKTER PLAT KENDARAAN BERMOTOR," *J. Ilm. Pendidik. Tek. Dan Kejuru.*, vol. 11, no. 1, p. 27, Nov. 2019, doi: 10.20961/jiptek.v11i1.18018.

[16] G. M. Raza, Z. S. Butt, S. Latif, and A. Wahid, "Sentiment Analysis on COVID Tweets: An Experimental Analysis on the Impact of Count Vectorizer and TF-IDF on Sentiment Predictions using Deep Learning Models," in *2021 International Conference on Digital Futures and Transformative Technologies (ICoDT2)*, Islamabad, Pakistan: IEEE, May 2021, pp. 1–6. doi: 10.1109/ICoDT252288.2021.9441508.

[17] Y. S. Mahardhika and E. Zuliarso, "Analisis Sentimen Terhadap Pemerintahan Joko Widodo Pada Media Sosial Twitter Menggunakan Algoritma Naives Bayes Classifier," in *SINTAK*, UNISBANK, 2018.

[18] A. Wendland, M. Zenere, and J. Niemann, "Introduction to Text Classification: Impact of Stemming and Comparing TF-IDF and Count Vectorization as Feature Extraction Technique," in *Systems, Software and Services Process Improvement*, M. Yilmaz, P. Clarke, R. Messnarz, and M. Reiner, Eds., in Communications in Computer and Information Science, vol. 1442. Cham: Springer International Publishing, 2021, pp. 289–300. doi: 10.1007/978-3-030-85521-5_19.

[19] T. Ahmed, S. F. Mukta, T. Al Mahmud, S. A. Hasan, and M. Gulzar Hussain, "Bangla Text Emotion Classification using LR, MNB and MLP with TF-IDF & CountVectorizer," in *2022 26th International Computer Science and Engineering Conference (ICSEC)*, Sakon Nakhon, Thailand: IEEE, Dec. 2022, pp. 275–280. doi: 10.1109/ICSEC56337.2022.10049341.

[20] Joshua Roesslein, "Tweepy Documentation," *Tweepy Documentation*, 2009. https://docs.tweepy.org/en/stable/index.html

[21] Python community, "Sastrawi," *Sastrawi*. https://pypi.org/project/Sastrawi/

[22] A. Heryana and U. Unggul, "Informan dan pemilihan informan dalam penelitian kualitatif," *Univ. Esa Unggul*, vol. 25, p. 15, 2018.

[23] M. T. Ari Bangsa, S. Priyanta, and Y. Suyanto, "Aspect-Based Sentiment Analysis of Online Marketplace Reviews Using Convolutional Neural Network," *IJCCS Indones. J. Comput. Cybern. Syst.*, vol. 14, no. 2, p. 123, Apr. 2020, doi: 10.22146/ijccs.51646.

[24] P. Nandwani and R. Verma, "A review on sentiment analysis and emotion detection from text," *Soc. Netw. Anal. Min.*, vol. 11, no. 1, p. 81, Dec. 2021, doi: 10.1007/s13278-021-00776-6.

[25] fabridamicelli, "scikit-learn," *scikit-learn*. https://github.com/scikit-learn/scikit-learn

[26] M. Birjali, M. Kasri, and A. Beni-Hssane, "A comprehensive survey on sentiment analysis: Approaches, challenges and trends," *Knowl.-Based Syst.*, vol. 226, p. 107134, Aug. 2021, doi: 10.1016/j.knosys.2021.107134.

[27] D. Vidotto, J. K. Vermunt, and K. Van Deun, "Bayesian Latent Class Models for the Multiple Imputation of Categorical Data," *methodology*, vol. 14, no. 2, pp. 56–68, Apr. 2018, doi: 10.1027/1614-2241/a000146.

[28] P. P. O. Mahawardana and G. A. Sasmita, "Analisis Sentimen Berdasarkan Opini dari Media Sosial Twitter terhadap 'Figure Pemimpin' Menggunakan Python," *J. Ilm. Teknol. Dan Komput.*, vol. 3, no. 1, 2022.

[29] B. A. Prasetyo, "Analisis Sentimen Pengguna Twitter untuk Teks Berbahasa Indonesia Terhadap Penyedia Layanan Home Fix Broadband," presented at the Seminar Nasional Teknik Industri, Yogyakarta, Indonesia: Universitas Gadjah Mada, 2021.

[30] D. A. Vonega, A. Fadila, and D. E. Kurniawan, "Analisis Sentimen Twitter Terhadap Opini Publik Atas Isu Pencalonan Puan Maharani dalam PILPRES 2024," *J. Appl. Inform. Comput.*, vol. 6, no. 2, pp. 129–135, Nov. 2022, doi: 10.30871/jaic.v6i2.4300.

[31] W. Aljedaani *et al.*, "Sentiment analysis on Twitter data integrating TextBlob and deep learning models: The case of US airline industry," *Knowl.-Based Syst.*, vol. 255, p. 109780, Nov. 2022, doi: 10.1016/j.knosys.2022.109780.

[32] D. Hazarika, G. Konwar, S. Deb, and D. J. Bora, "Sentiment Analysis on Twitter by Using TextBlob for Natural Language Processing," presented at the The International Conference on Research in Management & Technovation 2020, Jan. 2020, pp. 63–67. doi: 10.15439/2020KM20.