Translation of article 10.22146/jnteti.v13i4.14440

Classification of Emotions in English Texts Using the Ensemble Bagging Approach

Erfian Junianto[1,2], Mila Puspitasari[1,3], Salman Ilyas Zakaria[1,3], Toni Arifin[1,2], Ignatius Wiseto Prasetyo Agung[1,2]

[1] ARS Digital Research and Innovation (ADRI), Adhirajasa Reswara Sanjaya University, Bandung, Jawa Barat, 40282, Indonesia
[2] Informatics Engineering Study Program, Faculty of Information Technology, Adhirajasa Reswara Sanjaya University, Bandung, Jawa Barat 40282, Indonesia
[3] Information Systems Study Program, Faculty of Information Technology, Adhirajasa Reswara Sanjaya University, Bandung, Jawa Barat 40282, Indonesia

[Submitted: 14 Augustus 2024, Revised: 30 October 2024, Accepted: 19 November 2024]
Corresponding Author: Mila Puspitasari (email: mileuups14@gmail.com)

ABSTRACT — This study highlights the importance of emotion classification in English text, particularly in human interaction on social media, which often involves unstructured data. Emotions play a crucial role in communication; a better understanding of these emotions can aid in analyzing user behavior. The main objective of this research is to enhance accuracy, recall, precision, and F1-score in emotion classification by applying an ensemble bagging approach, combining the naïve Bayes, logistic regression, and k-nearest neighbor (KNN) algorithms. The methodology used included data collection from various sources, followed by data cleaning and analysis using text mining and machine learning techniques. The collected data were then analyzed to detect emotions such as anger, happiness, sadness, surprise, shame, disgust, and fear. Performance evaluation was conducted by comparing the results of the ensemble bagging method with individual algorithms to measure its effectiveness. The findings reveal that the logistic regression method achieved the highest accuracy at 98.76%, followed by naïve Bayes and KNN. This ensemble method overcame the limitations of each individual algorithm, enhancing overall classification stability and reliability. These findings provide valuable insights into text-based emotion analysis techniques and demonstrate the potential of ensemble methods to improve classification accuracy. Future research directions can explore additional ensemble techniques and optimize model complexity for improved performance in emotion analysis across broader datasets.

KEYWORDS — Text Mining, Bagging, Emotion Classification, Machine Learning.

I. INTRODUCTION

Human life is deeply connected to emotions. Emotions are responses that arise in reaction to people or events and play a significant role in daily life. They are generally categorized into two main types—positive and negative—which include various categories such as happiness, anger, fear, sadness, and others. The study of emotions is essential in fields such as psychology, cognitive science, and social media analysis, as emotions help us understand human behavior and social interactions [1].

People can easily express their emotions verbally, especially through writing on social media. Emotions are not only a form of personal expression but also a driving force behind cognitive processes and strategic thinking. They encompass various forms of language that are widely spoken and understood [2]. Furthermore, emotions have a profound impact on memory, social relationships, and even decision-making [3].

Emotion identification can be conducted in two ways: verbal and nonverbal. Verbal emotions are conveyed through speech or writing, while nonverbal emotions are expressed through body language, such as facial expressions, gestures, and hand or foot movements. Detecting emotions in text-based conversations is more challenging without these facial expressions or vocal cues [4]. Therefore, a specialized method is required to identify emotions in text-based dialogue, known as emotion classification, which is the process of grouping or mapping documents into predefined emotion categories [5].

To classify emotions in text with unstructured data, text analysis must be conducted using text mining techniques [6]–[10]. The goal of text mining is to identify and explore interesting patterns within data sources, typically collections of documents. These patterns are often not found in conventional databases but are highly relevant in the context of unstructured textual data. Consequently, text mining plays a crucial role in uncovering valuable insights from large and diverse text datasets [11].

Text mining is commonly used in sentiment analysis research [12]–[15] and emotion classification of textual datasets [16], [17]. Sentiment analysis involves processing data to identify feelings, sentiments, and emotions expressed in text [18]. Various algorithms, such as naïve Bayes NB, logistic regression, and k-nearest neighbor (KNN), are frequently employed for this purpose [19]. These algorithms typically operate under the naïve assumption that each observed condition or event is independent of others [20]–[25].

Numerous previous studies have applied naïve Bayes, logistic regression, and KNN algorithms to classification tasks, achieving relatively high accuracy [26]. However, there remains room for improvement. For instance, KNN is vulnerable to uneven data distribution and outliers; naïve Bayes relies on the often-unrealistic assumption of feature independence; and logistic regression is prone to overfitting, especially with small datasets containing noise. These factors can impact model accuracy and generalizability [27]. Consequently, this study sought to optimize classification performance by employing an ensemble bagging approach that combined these three algorithms.

Research on text mining in emotion classification has extensively involved the combination of ensemble bagging techniques with methods such as naïve Bayes, logistic regression, and KNN. Various studies have reported accuracy rates ranging from 46% to 77%. For instance, previous research conducted a study on the classification of COVID-19 vaccine types using the KNN method, achieving an accuracy of 46.20% [28].

Another study on Twitter users' perceptions regarding COVID-19 cases used the logistic regression method reached an accuracy of 77% [29]. Additionally, another research on emotion identification on Twitter used the naïve Bayes method combined with feature fusion, reporting an accuracy of 50.55% [30]. These accuracy results were influenced by the large volume of data and the variety of emotion labels used, which included seven emotion classes (anger, happiness, sadness, fear, disgust, shame, and surprise) that were employed in this research.

Subsequent research on the classification of prospective members using the bagging method based on naïve Bayes achieved a higher accuracy rate of 67.92%, compared to the use of the single naïve Bayes algorithm, which only reached an accuracy of [31]. The bagging method using naïve Bayes also recorded a recall or sensitivity value of 0.76, a precision of 0.74, and an F1-score of 0.75. On the other hand, the single naïve Bayes algorithm recorded a recall or sensitivity value of 0.65, a precision of 0.85, and an F1-score of 0.74. These findings indicated that the use of the bagging method could significantly improve classification performance in the context of the study.

Further research focused on text mining on social media to detect user emotions [11]. Testing using the support vector machine (SVM) and KNN methods yielded an average precision of 0.4564 and a recall of 0.502, with an accuracy of 0.8104 for SVM. Meanwhile, for KNN, the average precision was 0.3421, recall was 0.4595, and accuracy reached 0.797. This study has provided an important contribution to the development of text-based emotion analysis techniques on social media platforms.

Another study focused on comparing the logistic regression and random forest methods in the case study of emotion classification in tweets [32]. The results of the comparison showed that logistic regression achieved an accuracy of 78.22%, while random forest reached an accuracy of 72.41%. This study has provided insights into the effectiveness and relative performance of both approaches in the context of text processing and analysis.

The ensemble bagging technique (bootstrap aggregating) is an optimization strategy that integrates multiple machine learning models into a single entity. In this approach, each model is trained separately, and their results are combined after the individual learning process is completed. The research also evaluated the performance improvement between ensemble and non-ensemble methods by considering accuracy, precision, recall, and F1-score as the main metrics [27], [31], [33].

This study aimed to improve the accuracy, recall, precision, and F1-score of emotion classification using an ensemble bagging approach that combines the naïve Bayes, logistic regression, and KNN algorithms. This research evaluated how effectively the combination of these approaches addresses the limitations of previous studies, particularly in terms of prediction accuracy and reliability. By employing this method, it is expected that the classification results will be more accurate and reliable and provide a more effective solution for analyzing complex data.

II. CLASSIFICATION OF EMOTIONS

This chapter discusses emotion classification, ensemble bagging, text mining, machine learning, naïve Bayes, KNN, and logistic regression.

A. CLASSIFICATION

Classification is the process of assigning an object to a predetermined label or category, often using models such as classification rules (IF-THEN), decision trees, mathematical formulas, and neural networks. A classification algorithm begins by building rules based on existing data, followed by a learning phase in which the algorithm processes data to improve accuracy. Subsequently, a testing phase is conducted using test data to evaluate and validate the algorithm's performance in predicting or classifying new data [34].

The classification process consists of two main stages: learning and grouping. In the learning stage, the algorithm analyzes data to create a model, which is then used in the grouping stage to classify data. The process of developing a model that accurately describes and distinguishes data classes is central to classification. The primary objective is for the model to reliably predict the class of previously unseen objects [35].

B. EMOTION

Emotion can be defined as a condition typically triggered by an event that holds significance for an individual. Emotions involve mental aspects, including desires and thoughts. To address the complexity of human emotions in real-life situations, states, processes, and computational models must often be simplified [36].

Emotions are not inherently positive or negative, they are fundamentally human reactions to various situations. Individuals with emotional intelligence are capable of identifying, recognizing, and evaluating their own qualities as individuals. Emotional intelligence refers to the mental capacity to assess, analyze, and solve both simple and complex problems, as well as to make sound decisions [37].

C. ENSEMBLE BAGGING

The meta-algorithm known as bagging, or bootstrap aggregating, is intended to improve the performance of machine learning algorithms. In the first stage, the training data is resampled to create multiple sub-datasets from the main dataset. In the second stage, predictions from various models trained on these sub-datasets are combined into a single final prediction. This approach helps reduce variance and overfitting, resulting in a more stable and accurate model [27].

Bagging, introduced by Leo Breiman, is an ensemble technique aimed at reducing the variance in predictions by aggregating multiple models [38]. The goal is to improve the classification accuracy of single tree-based models. The bagging method uses random sampling with replacement from the original dataset to create new datasets. Each new dataset is the same size as the original, with random samples drawn with replacement, creating what are known as bootstrap samples. These samples are then used to generate classification trees across multiple iterations. The predictions from each iteration are aggregated to produce the final result [38].

D. TEXT MINING

Text mining is a process that utilizes documents and various analytical tools to gain deep insights. It enables the extraction of valuable information, discovery of patterns, and classification of textual data, as demonstrated in this research to explore interesting trends [17]. Essentially, the working mechanisms of text mining algorithms are quite similar to those of data mining algorithms. However, while text mining operates on unstructured data, data mining focuses on structured data. Text mining can assist in solving problems related to classification, clustering, and prediction of textual data [39].

The application of text mining algorithms is typically performed after the textual data has been transformed into a more optimal format. Common methods include classification, which categorizes texts into predefined groups, and information extraction, which identifies and retrieves significant details, such as names, relationships, and events. Additionally, sentiment analysis is used to detect opinions or emotions within text, categorizing them as positive, negative, or neutral. Evaluation and validation are conducted through measures like precision, recall, and F1-score to assess model performance on test data, ensuring accurate and reliable results [40].

E. MACHINE LEARNING

Research on algorithms designed to learn how to perform specific tasks—typically done automatically by humans—is known as machine learning. Learning refers to the ability to complete previously performed tasks or to draw conclusions from observed patterns, which includes processing new information. The primary focus of machine learning is to develop algorithms that enable systems to learn autonomously, without human intervention [41].

Machine learning is divided into two main learning paradigms. The first is supervised learning, which uses techniques to build functions based on existing training data. Examples of algorithms in supervised learning include naïve Bayes, KNN, and logistic regression. The second is unsupervised learning, a technique in machine learning that seeks to identify patterns in training data without preexisting classifications. Unlike supervised learning, this method does not rely on pre-classified data for guidance. An example of an unsupervised learning algorithm is clustering [42].

F. NAÏVE BAYES

Bayesian classification is a statistical method used to predict the class of an instance by considering its probability. Naïve Bayes is a classification technique based on the assumption that each attribute within a class is independent of the others [34].

The advantage of the naïve Bayes classification method is that it requires a relatively small amount of training data. This is because only the mean and variance parameters are needed for classification. During training, documents (training data) are categorized and processed to generate knowledge in the form of probability values for each word. This process identifies a set of words for each document that signifies its category. The naïve Bayes method is suitable for this context due to its ability to handle classification with limited data while producing reliable results based on the probability of each category [43].

G. K-NEAREST NEIGHBORS (KNN)

The KNN method classifies objects by considering training data with the closest similarity to the object based on a specified value of k [44]. In other words, this method uses a supervised algorithm where the classification result for a new object is determined by the majority of its nearest neighbors in the training set [45].

The advantages of KNN include its robustness against noisy training data and effectiveness in environments with large datasets. KNN is also relatively simple to implement and can be applied to both classification and regression tasks. However, KNN has weaknesses, including the need to specify the parameter k, or the number of nearest neighbors, which can affect the model performance. Additionally, distance-based methods often present challenges, such as selecting the most appropriate type of distance and identifying relevant attributes to achieve optimal results. Lastly, KNN's computational cost is relatively high, as it requires calculating the distance between each new instance and all training samples [46].

H. LOGISTIC REGRESSION

Logistic regression is one of machine learning algorithms used for classification that predicts the likelihood of a categorical dependent variable. In logistic regression, a binary variable coded as 1 or 0, serving as the dependent variable. This technique, an extension of linear regression, maps a set of numerical variables to a binary or probabilistic outcome. Logistic regression is often chosen for its ability to clearly interpret the relationship between independent and dependent variables and its effectiveness in handling classification tasks [47].

Logistic regression is widely applied in sentiment analysis problems, such as predicting positive, negative, or neutral outcomes [48]. It is suitable for cases where the output variable has only two binary values. For data with more than two possible outcomes, however, multinomial logistic regression is more appropriate [29]. Ordinal logistic regression is used to examine relationships between a response variable and predictor variables on an ordinal scale, using a model suited for polytomous response variables [49]. Equation (1) represents a simple logistic regression [50]:

$$ln\left[\frac{p(y=1)}{1-p(y=1)}\right] = \beta_\circ + \beta x. \tag{1}$$

Equation (2) represents the general form of multiple binary logistic regression:

$$ln\left[\frac{p(y=1)}{1-p(y=1)}\right] = \beta_\circ + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k \tag{2}$$

Equation (3) calculates the probability in multiple binary logistic regression:

$$p(y=1) = \frac{e^{\beta^\circ} + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k}{1 + e^{\beta^\circ} + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k} \tag{3}$$

III. METHODOLOGY

The flow or stages of the emotion classification model using the ensemble bagging approach, from text data input to the emotion classification result, is presented in Figure 1.

A. DATA COLLECTION

Data collection was the initial stage of this research. The data for this study were obtained from previous research [51] and were categorized into seven emotion classes: happiness, sadness, surprise, disgust, anger, fear, and shame. In total, the dataset consists of 2,414 texts representing these seven emotions.

Table I shows the distribution of the dataset by emotion class [32]. Ach row in Table I represents a specific emotion identified in the text—happiness, sadness, surprise, disgust, anger, fear, or shame. The numbers indicate the sample count for each emotion category, which is crucial for assessing data balance or imbalance that may influence the classification outcome.

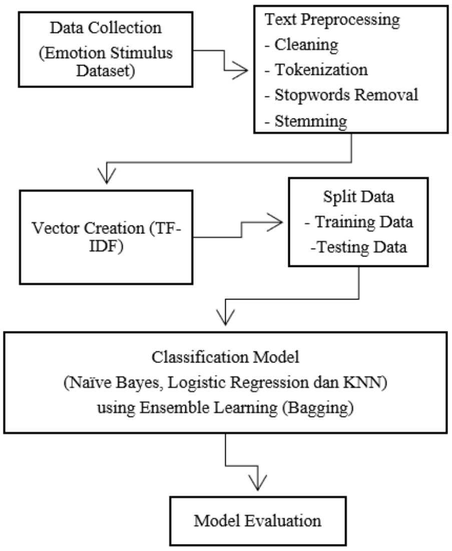

Figure 1. Research methods.

TABLE I
NUMBER OF DATASETS

Class	Amount
Happiness	479
Sadness	575
Surprise	213
Disgust	95
Anger	483
Fear	423
Shame	146

Table II presents the sample distribution of emotional data used in the study. The data were divided into seven emotion classes: happiness, sadness, anger, shame, fear, disgust, and surprise. Table II helps to visualize the proportion of each emotion in the dataset, which is essential for analyzing and interpreting the classification results.

B. TEXT PREPROCESSING

Text preprocessing was the initial step following the data analysis process, where raw data were processed and prepared for further analysis. This stage involved various techniques, such as cleaning, tokenization, stopword removal, and stemming. These steps aim to ensure that the data is in optimal condition before being used in machine learning models [52].

1) CLEANING

Cleaning is an essential part of text data preprocessing, focused on removing noise and making the text more consistent for in-depth analysis. Noise refers to irrelevant data or information that may interfere with the analysis process. In this stage, irrelevant elements in the data—such as special characters (!@#$%^&*():{}.,?~/[]), hashtags, URLs, mentions, and emoticons—are removed [45].

Figure 2 defines the ‘clean_text’ function, which cleans text within a DataFrame by converting all characters to lowercase and removing punctuation and numbers using regular expressions. This function was applied to the “text” column in the DataFrame “df,” resulting in a new column called “cleaned” that contained the cleaned text. The cleaning results were displayed using “head” to print the first five rows of the cleaned column. For example, sentence “The ANC, while expressing delight at their decision, nevertheless continued to call for the

TABLE II
TEXT OF DATA SAMPLE

Class	Text
Happy	Lennox has always truly wanted to fight for the world title and was happy taking the tough route
Sad	She was heartbroken to hear the voice of Camilla
Surprise	I was astonished at the smallness of the area which these enclosed
Disgust	The cave-man got fed-up with walking
Anger	I felt a stifled anger against her
Fear	I am horrified that somebody would do such a thing
Shame	But I’m not ashamed of my body

```
# Lowercase dan hapus tanda baca serta angka
def clean_text(text):
    text = text.lower()
    return re.sub(r'[^a-zA-Z\s]', '', text)

df['cleaned'] = df['Text'].apply(clean_text)
print("Cleaned Text:\n", df['cleaned'].head(), "\n")
```

```
Cleaned Text:
 0    i suppose i am happy being so  tiny  it means ...
1    lennox has always truly wanted to fight for th...
2    he was a professional musician now  still sens...
3    holmes is happy having the freedom of the hous...
4    i had problems with tutors trying to encourage...
Name: cleaned, dtype: object
```

Figure 2. Source code cleaning and output.

dissolution of the tricameral parliament, which it regarded as essentially racist” is changed into “The ANC while expressing delight at their decision nevertheless continued to call for the dissolution of the tricameral parliament which it regarded as essentially racist”

2) TOKENIZATION

The tokenization process involves dividing text into sentences, words, or tokens with features like capitalization type, the presence of numbers, punctuation marks, special characters, and similar attributes. At this stage, the input string is segmented into its constituent words [29].

Tokenization also refers to separating sentences in a document into individual words, using spaces as the primary delimiter. However, in Indonesian and many other languages, some words should not be separated, as this can alter or even eliminate their meaning [53], [54].

Figure 3 displays the source code for tokenizing cleaned text within a DataFrame using the “nltk” module. If not already installed, the code would download the “punkt” package. Then, the “tokenize_text” function, which utilized ‘word_tokenize,’ broke the text into tokens. This function was applied to the “cleaned” column in the DataFrame “df”, generating a new column named “tokens.” As a result, the text split into individual words was displayed in the first five rows of the “tokens” column. For example, sentence “The ANC while expressing delight at their decision nevertheless continued to call for the dissolution of the tricameral parliament which it regarded as essentially racist” is changed into “the anc” while expressing delight at their decision nevertheless continued to call for the dissolution of the tricameral parliament which it regarded as essentially racist”

3) STOPWORD REMOVAL

Stopword removal is the process of eliminating words from a document that are considered insignificant, do not convey

 p-ISSN 2301–4156 | e-ISSN 2460–5719

```
import nltk
from nltk.tokenize import word_tokenize


nltk.download('punkt')

# Tokenize teks yang sudah dibersihkan
def tokenize_text(text):
    return word_tokenize(text)


df['tokens'] = df['cleaned'].apply(tokenize_text)
print("Tokens:\n", df['tokens'].head(), "\n")
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
Tokens:
 0    [i, suppose, i, am, happy, being, so, tiny, it...
1    [lennox, has, always, truly, wanted, to, fight...
2    [he, was, a, professional, musician, now, stil...
3    [holmes, is, happy, having, the, freedom, of, ...
4    [i, had, problems, with, tutors, trying, to, e...
Name: tokens, dtype: object
```

Figure 3. Source code tokenizer and output.

emotional meaning, or do not enhance emotional expressions. This process is carried out by comparing words with an existing stopword dictionary. It also involves removing conjunctions or affixed words, such as “at,” “while,” or “to,” leaving only essential words. Additionally, there is a step to correct misspelled words and certain abbreviations to ensure consistency with the data format used [17], [52].

Stopword removal is a common preprocessing step in various applications. Its purpose is to eliminate frequently occurring words within each document in a corpus. Pronouns and prefixes are often included as stopwords. In several natural language processing (NLP) tasks, such as information retrieval and classification, these words are considered insignificant and can make it challenging to differentiate between documents [55].

Figure 4 illustrates the process of removing stopwords from tokens within a DataFrame using the “nltk” library. The list of English stopwords was accessed by importing “stopwords” from “nltk.corpus” and downloading the “stopwords” package if not already available. Next, the “remove_stopwords” function was applied to the list of tokens in the “tokens” column in DataFrame “df,” resulting in a new column named “tokens_no_stopwords” that displayed tokens cleansed of stopwords. For example, sentence “the anc while expressing delight at their decision nevertheless continued to call for the dissolution of the tricameral parliament which it regarded as essentially racist” is changed into “anc expressing delight decision nevertheless continued call dissolution tricameral parliament regarded essentially racist.”

4) STEMMING

In NLP, stemming is the process of removing affixes (prefixes, suffixes, or other inflections) from a word to reduce it to its base form, known as the “stem” or “root word.” The goal is to facilitate text analysis by reducing word variations to a uniform base form. This enables words with similar meanings to be grouped or simplifies the text data structure. For example, words like “regarding” and “regarded” are transformed into the root form “regard.” The most commonly used stemming algorithm in English is the Porter stemmer [1], [56].

Figure 5 illustrates the stemming process using the Porter stemmer algorithm from Natural Language Toolkit (NLTK) to convert text tokens to their base forms. The next step was to create a new column in the DataFrame containing the stemmed text and tokens, followed by printing the results for further analysis. For example, sentence “anc expressing delight

```
nltk.download('stopwords')

# Inisialisasi stopwords
stop_words = set(stopwords.words('english'))

# Hapus stopwords dari token
def remove_stopwords(tokens):
    return [word for word in tokens if word not in stop_words]

df['tokens_no_stopwords'] = df['tokens'].apply(remove_stopwords)
print("Tokens without Stopwords:\n", df['tokens_no_stopwords'].head(), "\n")
```

```
Tokens without Stopwords:
 0    [suppose, happy, tiny, means, able, surprise, ...
1    [lennox, always, truly, wanted, fight, world, ...
2    [professional, musician, still, sensitive, hap...
3                  [holmes, happy, freedom, house]
4    [problems, tutors, trying, encourage, diversit...
Name: tokens_no_stopwords, dtype: object

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

Figure 4. Source code stopwords and output.

```
stemmer = PorterStemmer()

def stem_tokens(tokens):
    return [stemmer.stem(word) for word in tokens]

df['stemmed_tokens'] = df['tokens_no_stopwords'].apply(stem_tokens)
df['processed_text'] = df['stemmed_tokens'].apply(lambda tokens: ' '.join(tokens))
print("Stemmed Tokens:\n", df['stemmed_tokens'].head(), "\n")
print("Processed Text:\n", df['processed_text'].head(), "\n")
```

```
Stemmed Tokens:
 0    [suppos, happi, tini, mean, abl, surpris, peop...
1    [lennox, alway, truli, want, fight, world, tit...
2    [profession, musician, still, sensit, happi, s...
3                    [holm, happi, freedom, hous]
4    [problem, tutor, tri, encourag, divers, work, ...
Name: stemmed_tokens, dtype: object

Processed Text:
 0    suppos happi tini mean abl surpris peopl gener...
1    lennox alway truli want fight world titl happi...
2    profession musician still sensit happi someth ...
3                            holm happi freedom hous
4    problem tutor tri encourag divers work experi ...
Name: processed_text, dtype: object
```

Figure 5. Source code of stopwords and output.

decision nevertheless continued call dissolution tricameral parliament regarded essentially racist” is changed into “anc express delight decis nevertheless continu call dissolut tricam parliament regard essenti racist.”

The overall steps of preprocessing applied to text before conducting emotion classification are presented in Table III. This preprocessing stage included text cleaning, tokenization, stopword removal, and stemming. For example, the original text, “The ANC, while expressing delight at their decision, nevertheless continued to call for the dissolution of the tricameral parliament, which it regarded as essentially racist,” underwent several steps to become a simpler base token format, such as “anc express delight decis nevertheless continu call dissolut tricam parliament regard essenti racist.” Each step plays an essential role in preparing the data for machine learning algorithms, ultimately enhancing accuracy in the emotion classification process.

C. VECTOR CREATION

Vector creation is the process of transforming data or entities into vectors for further analysis or processing. This is often achieved by applying term frequency-inverse document frequency (TF-IDF) weighting techniques [56]. The TF-IDF method, widely used in text retrieval and preprocessing, assigns weighting values to extracted words, helping to assess their importance within a document relative to a collection of other documents. This facilitates the analysis process and enables more effective retrieval of relevant information.

TABLE III
TEXT PREPROCESSING

Text	Preprocessing
The ANC, while expressing delight at their decision, nevertheless continued to call for the dissolution of the tricameral parliament, which it regarded as essentially racist	Text
The ANC while expressing delight at their decision nevertheless continued to call for the dissolution of the tricameral parliament which it regarded as essentially racist	Cleaning
the anc while expressing delight at their decision nevertheless continued to call for the dissolution of the tricameral parliament which it regarded as essentially racist	Tokenizer
anc expressing delight decision nevertheless continued call dissolution tricameral parliament regarded essentially racist	Stopwords
anc express delight decis nevertheless continu call dissolut tricam parliament regard essenti racist	Stemming

TABLE IV
RESULTS OF THE TF-IDF DISTRIBUTION

No.	Text	TF-IDF Results
1.	lin foh quite at being to so	0.068
2.	many of the as the wore on were to find the door not heard of the noon and so they had all day	0.115
3.	lin yuan was with proud of his and that he had tsu ma ‘s	0.229
4.	even so was when we were from last	0.351
5.	you ‘re not just for lucy	0.153
6.	that to when they heard of his ‘s grim fate	0.272

TF-IDF is a statistical method that evaluates the significance of a word within a document. Term frequency (TF) measures how often a word appears in a specific document, indicating its significance within that context. Document frequency (DF) indicates how frequently the word appears across a collection of documents, showing how common the word is within the entire corpus. TF-IDF combines these metrics to provide a more precise assessment of a word’s relevance within a specific document, while also reducing the impact of common words that appear frequently across multiple documents [57].

Table IV presents the distribution of TF-IDF values for each word in the emotion dataset. A high TF-IDF value indicates that the word is particularly significant, with strong distinguishing power for identifying specific emotions within the text.

D. SPLIT DATA

The following step involves dividing the dataset into two parts: training data and testing data.

1) TRAINING DATA

Training data are a portion of the dataset used to train machine learning algorithms to predict or perform various functions. During the training process, training data samples are used as input, allowing the algorithm to learn from these features to build a learning model. The trained machine can identify patterns by following the instructions provided by the algorithm through the training process. Training data plays a critical role in teaching the machine to recognize patterns within the data so that the resulting model can make accurate predictions or perform specific tasks efficiently. Through iterations and parameter adjustments, the model’s performance can be improved, ensuring that it can handle new data effectively in the future [58].

2) TESTING DATA

Testing data are essential for evaluating the machine learning model, as it is used to assess the model’s predictions after training is complete. In this phase, the model is used to make predictions on previously unseen testing or production data. This testing data includes samples that represent real-world scenarios encountered in actual applications. Evaluating the model with testing data helps measure its accuracy and performance, as well as identify strengths and weaknesses that need improvement before wider implementation. Therefore, testing data is crucial not only for preventing overfitting but also for ensuring that the model can effectively handle new issues it may encounter [58]. In this research, the data were split into 90% training data and 10% testing data [32].

Figure 6 shows the source code for splitting the dataset into training and testing data using the “train_test_split” function from the “sklearn.model_selection” library. First, the “emotion” column from DataFrame the “df” was stored in the “y” variable, which contained the labels or targets for classification. Subsequently, the dataset, which had been transformed into TF-IDF features “(X_tfidf),” along with the labels “y,” was divided into 90% training data and 10% testing data, using “random_state=42” to ensure consistent results each time the code was executed.

E. CLASSIFICATION MODEL

After completing the dataset partitioning phase, the next step was to apply a classification modeling approach that combines machine learning algorithms—namely, naïve Bayes, logistic regression, and KNN—using the ensemble bagging method. In this study, the integration of results from these three models was conducted during the testing phase, with each algorithm first processing the test data individually. The prediction results from each algorithm were then combined using the ensemble bagging technique to produce a more stable and accurate final prediction. Mathematically, (4) represents the formula for bagging:

$$\widehat{f_{bag}} = \widehat{f_1}(X) + \widehat{f_2}(X) + \cdots + \widehat{f_b}(X). \tag{4}$$

The left side, $\widehat{f_{bag}}$, denotes the prediction result from ensemble bagging (bagged prediction), while the right side, $\widehat{f_b}(X)$, represents the prediction from each individual learner.

F. MODEL EVALUATION

At the final stage, measurements were conducted to evaluate the accuracy level of the classification process. The results were presented in the form of a confusion matrix, which illustrates the accuracy values.

The confusion matrix is a 2 × 2 matrix that summarizes all correct and incorrect classification outcomes. From the combinations of true positive (TP), false positive (FP), false negative (FN), and true negative (TN) values, four evaluation metrics were derived to assess the classification model’s performance.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{5}$$

 p-ISSN 2301–4156 | e-ISSN 2460–5719

```python
from sklearn.model_selection import train_test_split


y = df['Emotion']

# Bagi dataset
X_train, X_test, y_train, y_test = train_test_split(X_tfidf, y, test_size=0.1, random_state=42)
```

Figure 6. Source code for dataset sharing.

Equation (5) represents accuracy, which measures the ratio of correct predictions (positive and negative) to all predictions.

$$Precision = \frac{TP}{TP+FP}. \tag{6}$$

Equation (6) represents precision, which calculates how accurately the model predicts positives by comparing true positive (TP) values against all positive predictions (TP + FP).

$$Recall = \frac{TP}{TP+FN}. \tag{7}$$

Equation (7) represents recall, which measures the model's ability to detect all actual positive data, expressed as the ratio of TP to the total number of positives (TP + FN).

$$F1-score = \frac{2(Precision \cdot Recall)}{Precision+Recall}. \tag{8}$$

Equation (8) represents the F1-score, which is the harmonic mean of precision and recall, providing a balance between the two.

IV. RESULTS AND DISCUSSION

In this study, the analysis was conducted using Google Colab. To improve classification performance, this research employed the ensemble bagging method in combination with three classification algorithms: naïve Bayes, KNN, and logistic regression. This approach was expected to produce more accurate and reliable predictions than using each algorithm individually.

Table V presents the performance evaluation results of the three classification algorithms used in this study—naïve Bayes, logistic regression, and KNN—without the ensemble bagging approach. The results include accuracy, F1-score, precision, and recall values for each algorithm. From Table V, it can be observed that logistic regression demonstrated the best performance with the highest accuracy of 85.95%, followed by naïve Bayes with an accuracy of 66.94% and KNN with the lowest accuracy of 52.48%. Furthermore, logistic regression also achieved high precision and recall values, at 88.08% and 85.95%, respectively. These F1-score, precision, and recall values provide a more comprehensive view of each model's ability to classify data accurately and consistently.

Table VI displays the performance evaluation results of the three classification algorithms with the ensemble bagging approach. The ensemble bagging method aims to improve prediction accuracy by combining the outputs of multiple models during the testing phase. The results presented include accuracy, F1-score, precision, and recall values for each algorithm. From table VI, it is evident that logistic regression with ensemble bagging achieved the highest accuracy of 98.76%, followed by KNN and naïve Bayes with lower accuracies. Precision, recall, and F1-score values are also provided to offer a comprehensive overview of each algorithm's performance in classifying emotions in text. These values are essential to understand each model's reliability in producing accurate and consistent predictions.

TABLE V
CLASSIFICATION REPORT WITHOUT ENSEMBLE BAGGING

Result	Accuracy (%)	F1-Score	Precision	Recall
Naïve Bayes	66.94	64.85	71.65	66.94
Logistic regression	85.95	83.52	88.08	85.95
KNN	52.48	51.73	52.02	52.48

TABLE VI
CLASSIFICATION REPORT WITH ENSEMBLE BAGGING

Results	Accuracy (%)	F1-Score	Precision	Recall
Ensemble bagging naïve Bayes	73.55	72.03	77.11	73.55
Ensemble bagging logistic regression	98.76	98.76	98.79	98.76
Ensemble bagging KNN	82.64	81.72	83.98	82.64

Figure 7 illustrates the performance of the naïve Bayes algorithm in emotion classification for English text, evaluated using three main metrics: precision, recall, and F1-score. The graph shows that disgust and shame have the highest precision at 100%, indicating that this model is highly effective in accurately identifying these emotions without many false positives. Meanwhile, the sad had the highest recall value at 88%, demonstrating the model's high sensitivity in detecting all true instances of the sad emotion in the dataset. Then, anger had the highest F1-score at 81%, a combination of precision and recall, indicating that the model maintains a good balance in accurately and comprehensively detecting this emotion. This analysis suggested that, while naïve Bayes was effective for certain emotion categories, its performance varied depending on the type of emotion classified.

Figure 8 presents the performance analysis of the logistic regression algorithm in classifying emotions within English text, evaluated using precision, recall, and F1-score metrics. The graph shows that the emotions of disgust, shame, and surprise achieved the highest precision at 100%, indicating that the logistic regression model was highly accurate in identifying these emotions without false positives. In terms of recall, sad emotion had the highest value at 100%, demonstrating the model's effectiveness in detecting all instances of the sadness in the dataset. For F1-score, the emotions of fear and shame attained the highest score at 97%, reflecting an optimal balance between precision and recall for these emotions. Overall, these results suggested that logistic regression is highly effective in emotion classification, particularly for emotions with high precision and recall values.

Figure 9 illustrates the performance of the KNN algorithm in classifying emotions within English text, assessed through precision, recall, and F1-score metrics. The graph indicates that the emotion of disgust achieved the highest precision at 100%, suggesting that the KNN model was highly accurate in identifying this emotion. The emotions of happiness and sadness had the highest recall values at 95%, demonstrating that the model is highly sensitive and can detect nearly all instances of these emotions within the dataset. For F1-score, happiness had the highest value at 88%, indicating that the model maintained a good balance between precision and recall

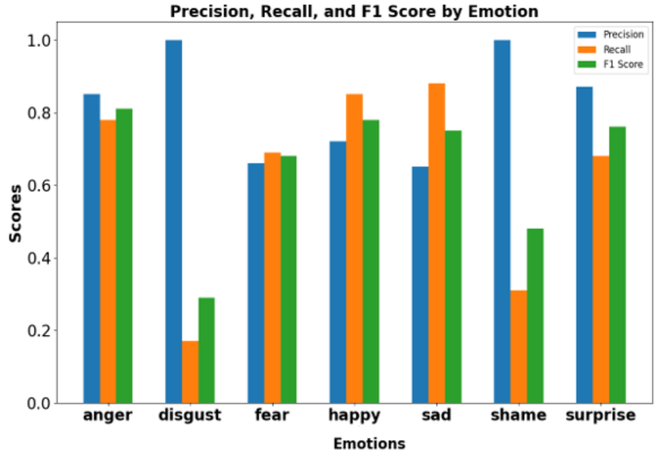

Figure 7. Graph of precision, recall and F1-score based on emotions (naïve Bayes).

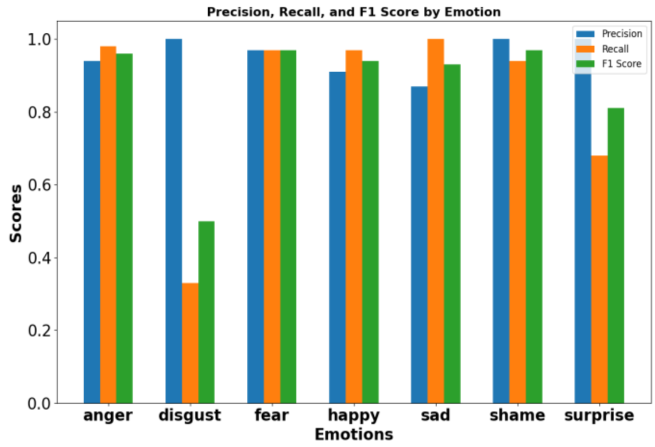

Figure 8. Graph of precision, recall and F1-score based on emotions (logistic regression).

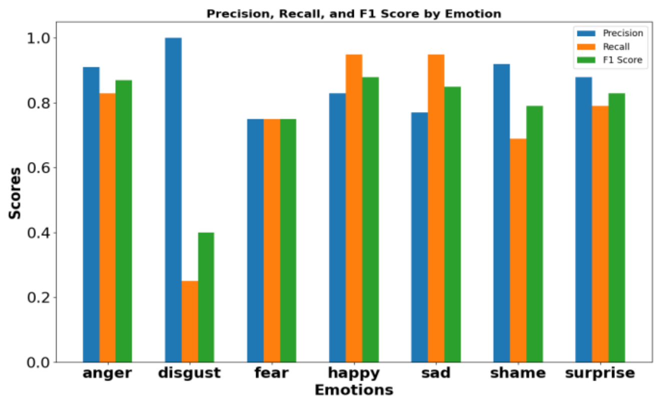

Figure 9. Graph of precision, recall and F1-score based on emotions (KNN).

in detecting the happy emotion. These results suggested that, while KNN performed well for certain emotion categories, its effectiveness varies depending on the emotion type classified.

V. CONCLUSION

This study demonstrates that the ensemble bagging technique, which combines naïve Bayes, logistic regression, and KNN algorithms, can enhance performance in emotion classification for English text. According to the evaluation results, the ensemble bagging method with logistic regression achieved the highest accuracy at 98.76%, outperforming naïve Bayes and KNN individually, which achieved accuracies of 66.94% and 52.48%, respectively. Furthermore, this ensemble method significantly improved precision, recall, and F1-score metrics compared to the results of individual algorithms. These findings support previous studies, which demonstrated that applying bagging to naïve Bayes increased accuracy from 66.94% to 73.55%. Thus, the ensemble bagging approach has proven more effective in enhancing the reliability and accuracy of emotion classification, particularly for unstructured text data.

AUTHORS' CONTRIBUTIONS

Conceptualization, Erfian Junianto; methodology, Mila Puspitasari; software, Salman Ilyas Zakaria; validation, Erfian Junianto, Mila Puspitasari, and Toni Arifin; formal analysis, Mila Puspitasari; investigation, Mila Puspitasari; resources, Ignatius Wiseto Prasetyo Agung; data curation, Salman Ilyas Zakaria; writing—original draft preparation, Mila Puspitasari; writing—review and editing, Erfian Junianto, Toni Arifin, Ignatius Wiseto Prasetyo Agung; visualization, Salman Ilyas Zakaria; supervision, Erfian Junianto; project administration, Erfian Junianto; funding acquisition, Ignatius Wiseto Prasetyo Agung.

ACKNOWLEDGMENT

The authors express sincere gratitude to Adhirajasa Reswara Sanjaya University and the Faculty of Information Technology for their invaluable support and resources, which greatly facilitated this research. We extend our thanks to the ARS Digital Research and Innovation team for their insightful discussions and technical assistance throughout the study. Additionally, we are grateful for the continuous encouragement from family and colleagues, which was instrumental in completing this research.

REFERENCES

[1] A.T. Bagus, "Klasifikasi emosi pada teks menggunakan metode deep learning," Undergraduate thesis, Universitas Islam Indonesia, Sleman, Indonesia, 2022.

[2] H. Krishnan, M.S. Elayidom, and T. Santhanakrishnan, "Emotion detection of tweets using naïve Bayes classifier," *Int. J. Eng. Technol. Sci. Res.(IJETSR)*, vol. 4, no. 11, pp. 457–462, Nov. 2017.

[3] A.N. Rohman, E. Utami, and S. Raharjo, "Deteksi kondisi emosi pada media sosial menggunakan pendekatan leksikon dan natural language processing," *J. Eksplora Inform.*, vol. 9, no. 1, pp. 70–76, Sep. 2019, doi: 10.30864/eksplora.v9i1.277.

[4] A. Chatterjee, N. Narahari, M. Joshi, and P. Agrawal, "SemEval-2019 task 3: EmoContext contextual emotion detection in text," in *Proc. 13th Int. Workshop Semant. Eval. (SemEval-2019)*, 2019, pp. 39–48, doi: 10.18653/v1/S19-2005.

[5] P.W.A. Wibawa and C. Pramartha, "Systematic literature review: Machine learning methods in emotion classification in textual data," *J. Sisfokom (Sist. Inf. Komput.)*, vol. 12, no. 3, pp. 425–433, Nov. 2023, doi: 10.32736/sisfokom.v12i3.1787.

[6] D. Ariyanti and K. Iswardani, "Teks mining untuk klasifikasi keluhan masyarakat menggunakan algoritma naive Bayes," *IKRA-ITH Inform, J. Komput Inform.*, vol. 4, no. 3, pp. 125–132, Nov. 2020.

[7] N. Andriani and A. Wibowo, "Implementasi text mining klasifikasi topik tugas akhir mahasiswa teknik informatika menggunakan pembobotan TF-IDF dan metode cosine similarity berbasis web," in *Proc. Semin. Nas. Mhs. Ilmu Komput. Apl. (SENAMIKA)*, 2021, pp. 130–137.

[8] Bimananda W. *et al.*, "Analisis text mining dari cuitan Twitter mengenai infrastruktur di Indonesia dengan metode klasifikasi naïve Bayes," *Eig. Math. J.*, vol. 2, no. 2, pp. 92–101, Dec. 2019, doi: 10.29303/emj.v1i2.36.

[9] T.W.D. Sari, "Penerapan text mining dengan menggunakan algoritma TF-IF untuk klasifikasi genre novel," *Pelita Inform., Inf. Inform.*, vol. 10, no. 1, pp. 29–37, Jul. 2021.

[10] M. Afdal and L.R. Elita, "Penerapan text mining pada aplikasi Tokopedia menggunakan algoritma k-nearest neighbor," *J. Ilm. Rekayasa Manaj. Sist. Inf.*, vol. 8, no. 1, pp. 78–87, Feb. 2022, doi: 10.24014/rmsi.v8i1.16595.

[11] D. Ardiada, M. Sudarma, and D. Giriantari, "Text mining pada sosial media untuk mendeteksi emosi pengguna menggunakan metode support

vector machine dan k-nearest neighbour," *Maj. Ilm. Teknol. Elekt.*, vol. 18, no. 1, pp. 55–60, Jan.-Apr. 2019, doi: 10.24843/mite.2019.v18i01.p08.

[12] T. Ridwansyah, "Implementasi text mining terhadap analisis sentimen masyarakat dunia di Twitter terhadap Kota Medan menggunakan k-fold cross validation dan naïve Bayes classifier," *Kajian Ilm. Inform. Komput.*, vol. 2, no. 5, pp. 178–185, Apr. 2022, doi: 10.30865/klik.v2i5.362.

[13] S.S. Berutu, "Text mining dan klasifikasi sentimen berbasis naïve Bayes pada opini masyarakat terhadap makanan tradisional," *J. Sist. Komput. Inform. (JSON)*, vol. 4, no. 2, pp. 254–262, Dec. 2022, doi: 10.30865/json.v4i2.5138.

[14] D.T. Alamanda *et al.*, "Sentiment analysis using text mining of Indonesia tourism reviews via social media," *Int. J. Humanit. Arts Soc. Sci.*, vol. 5, no. 2, pp. 43–53, Apr. 2019, doi: 10.20469/ijhss.5.10005-2.

[15] F.F. Mailo and L. Lazuardi, "Analisis sentimen data Twitter menggunakan metode text mining tentang masalah obesitas di Indonesia," *J. Inf. Syst. Public Health*, vol. 6, no. 1, pp. 44–51, Apr. 2021, doi: 10.22146/jisph.44455.

[16] E. Indrayuni, "Klasifikasi text mining review produk kosmetik untuk teks bahasa Indonesia menggunakan algoritma naive Bayes," *J. Khatulistiwa Inform.*, vol. 7, no. 1, pp. 29–36, Jun. 2019, doi: 10.31294/jki.v7i1.5740.

[17] W. Hermanto, B. Irawan, and C. Setianingsih, "Klasifikasi emosi pada lirik lagu menggunakan algoritma support vector machine dan optimasi particle swarm optimization," *e-Proc. Eng.*, vol. 8, no. 5, pp. 6307–6327, Oct. 2021.

[18] N. Anggraini, E.S.N. Harahap, and T.B. Kurniawan, "Text mining-text analysis related to COVID-19 vaccination issues," *J. IPTEK-KOM (J. Ilmu Pengetah. Teknol. Komun.)*, vol. 23, no. 2, pp. 141–153, Dec. 2021, doi: 10.33169/iptekkom.23.2.2021.141-153.

[19] H.P. Doloksaribu and Y.T. Samuel, "Komparasi algoritma data mining untuk analisis sentimen aplikasi Pedulilindungi," *J. Teknol. Inf., J. Keilmuan Apl. Bid. Tek. Inform.*, vol. 16, no. 1, pp. 1–11, Jan. 2022, doi: 10.47111/jti.v16i1.3747.

[20] S.D. Pramukti, A. Nugroho, and A.S. Sunge, "Analisis sentimen masyarakat dengan metode naïve Bayes dan particle swarm optimization," *Techno.Com*, vol. 21, no. 1, pp. 62–75, Feb. 2022, doi: 10.33633/tc.v21i1.5332.

[21] R. Fajar, "Implementasi algoritma naive Bayes terhadap analisis sentimen opini film pada Twitter," *J. Inovtek Polbeng Seri Inform.*, vol. 3, no. 1, pp. 50–59, Jun. 2018, doi: 10.35314/isi.v3i1.335.

[22] S. Budi, "Text mining untuk analisis sentimen review film menggunakan algoritma k-means," *Techno.Com*, vol. 16, no. 1, pp. 1–8, Feb. 2017, doi: 10.33633/tc.v16i1.1263.

[23] A.K. Fauziyyah, "Analisis sentimen pandemi COVID19 pada streaming Twitter dengan text mining python," *J. Ilm. Sinus*, vol. 18, no. 2, pp. 31–42, Jul. 2020, doi: 10.30646/sinus.v18i2.491.

[24] R. Wahyudi and G. Kusumawardhana, "Analisis sentimen pada review aplikasi Grab di Google Play Store menggunakan support vector machine," *J. Inform.*, vol. 8, no. 2, pp. 200–207, Sep. 2021, doi: 10.31294/ji.v8i2.9681.

[25] R. Siringoringo and Jamaluddin, "Text mining dan klasterisasi sentimen pada ulasan produk toko online," *J. Penelit. Tek. Inform.*, vol. 2, no. 1, pp. 314–319, Apr. 2019, doi: 10.34012/jutikomp.v2i1.456.

[26] H.P. Koapaha and N. Ananto, "Bagging based ensemble analysis in handling unbalanced data on classification modelling," *Klabat Account. Rev.*, vol. 2, no. 2, pp. 165–178, Sep. 2021, doi: 10.60090/kar.v2i2.589.165-178.

[27] R. Siringoringo and I.K. Jaya, "Ensemble learning dengan metode SMOTEBagging pada klasifikasi data tidak seimbang," *J. Inf. Syst. Dev. (ISD)*, vol. 3, no. 2, pp. 75–81, Jul. 2018.

[28] P. Arsi, I. Prayoga, and M. H. Asyari, "Klasifikasi sentimen publik terhadap jenis vaksin COVID-19 yang tersertifikasi WHO berbasis NLP dan KNN," *J. Media Inform. Budidarma*, vol. 7, no. 1, pp. 260–266, Jan. 2023, doi: 10.30865/mib.v7i1.5418.

[29] A.K. Santoso *et al.*, "Klasifikasi persepsi pengguna Twitter terhadap kasus COVID-19 menggunakan metode logistic regression," *J. Inform. Kaputama (JIK)*, vol. 5, no. 2, pp. 234–241, Jul. 2021, doi: 10.59697/jik.v5i2.247.

[30] F. Fanesya, R.C. Wihandika, and Indriati, "Deteksi emosi pada Twitter menggunakan metode naïve Bayes dan kombinasi fitur," *J. Pengembangan Teknol. Inf. Ilmu Komput.*, vol. 3, no. 7, pp. 6678–6686, Jul. 2019.

[31] S. Shofiyah, "Studi komparasi metode naive Bayes dan metode ensemble bagging," Undergraduate thesis, Universitas Islam Negeri Sunan Kalijaga, Yogyakarta, Indonesia, 2020.

[32] W.O. Simanjuntak, A.B.P. Negara, and R. Septriana, "Perbandingan algoritma logistic regression dan random foret (Studi kasus: Klasifikasi emosi tweet)," *J. Apl. Ris. Inform.*, vol. 1, no. 2, pp. 160–164, Aug. 2023, doi: 10.26418/juara.v2i1.69682.

[33] A. Helmut, Adiwijaya, and D.T. Murdiansyah, "Klasifikasi email multi kelas menggunakan ensemble bagging," *e-Proc. Eng.*, vol. 7, no. 1, pp. 2498–2505, Apr. 2020.

[34] Mursyidah, H.T. Hidayat, and D.M. Sari, "Klasifikasi teks emosi bahasa Aceh menggunakan metode term frekuensi / invers dokument frekuensi," *J. Infomedia, Tek. Inform. Multimed. Jar.*, vol. 2, no. 1, pp. 14–19, Mar. 2017, doi: 10.30811/.v2i1.462.

[35] H. Susana, N. Suarna, Fathurrohman, and Kaslani, "Penerapan model klasifikasi metode naive Bayes terhadap penggunaan akses internet," *J. Ris. Sist. Inf. Teknol. Inf. (JURSISTEKNI)*, vol. 4, no. 1, pp. 1–8, Jan. 2022, doi: 10.52005/jursistekni.v4i1.96.

[36] I. Destuardi and S. Sumpeno, "Klasifikasi emosi untuk teks bahasa Indonesia menggunakan metode naive Bayes," presented at Semin. Nas. Pascasarj. IX – ITS, Surabaya, Jawa Timur, Indonesia, 12 Aug. 2015.

[37] A. Ansori, "Kepribadian dan emosi," *J. Literasi Pendidik. Nusant.*, vol. 1, no. 1, pp. 41–54, Jan.-Jun. 2020.

[38] Y.A. Jatmiko, S. Padmadisastra, and A. Chadidjah, "Analisis perbandingan kinerja cart konvesional, bagging dan random forest pada klasifikasi objek: Hasil dari dua simulasi," *Media Stat.*, vol. 12, no. 1, pp. 1–12, Jul. 2019, doi: 10.14710/medstat.12.1.1-12.

[39] A. Priyanto and M.R. Ma'arif, "Implementasi web scraping dan text mining untuk akuisisi dan kategorisasi informasi laman web tentang hidroponik," *Indones. J. Inf. Syst.*, vol. 1, no. 1, pp. 25–33, Aug. 2018, doi: 10.24002/ijis.v1i1.1664.

[40] M.P.R. Putra and K.R.N. Waedani, "Penerapan text mining dalam menganalisis kepribadian pengguna media sosial," *JUTIM (J. Tek. Inform. Musirawas)*, vol. 5, no. 1, pp. 63–71, Jun. 2020, doi: 10.32767/jutim.v5i1.

[41] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning : From Theory To Algorithms*. New York, NY, USA: Camb. Univ. Press, 2014.

[42] A. Fathurohman, "Machine learning untuk pendidikan: Mengapa dan bagaimana," *J. Inform. Teknol. Komput.*, vol. 1, no. 3, pp. 57–62, Nov. 2021, doi: 10.55606/jitek.v1i3.306.

[43] R.N. Devita, H.W. Herwanto, and A.P. Wibawa, "Perbandingan kinerja metode naive Bayes dan k-nearest neighbor untuk klasifikasi artikel berbahasa Indonesia," *J. Teknol. Inf. Ilmu Komput.*, vol. 5, no. 4, pp. 427–434, Sep. 2018, doi: 10.25126/jtiik.201854773.

[44] P.D. Nugraha, S.A. Faraby, and Adiwijaya, "Klasifikasi dokumen menggunakan metode k-nearest neighbor (KNN) dengan information gain document," *e-Proc. Eng.*, vol. 5, no. 1, pp. 1541–1550, Mar. 2018.

[45] A. S. Rezeki, "Klasifikasi emosi pada Twitter dengan metode k-nearest neighbor (KNN)," Undergraduate thesis, Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekanbaru, Indonesia, 2021.

[46] Y. Kustiyahningsih and N. Syafa'ah, "Sistem pendukung keputusan untuk menentukan jurusan pada siswa SMA menggunakan metode KNN dan smart," *J. Sist. Inf. Indones.*, vol. 1, no. 1, pp. 19–28, Apr. 2015.

[47] I.F. Ramadhy and Y. Sibaroni, "Analisis trending topik Twitter dengan fitur ekspansi fasttext menggunakan metode logistic regression," *JURIKOM (J. Ris. Komput.)*, vol. 9, no. 1, pp. 1–7, Feb. 2022, doi: 10.30865/jurikom.v9i1.3791.

[48] A.K. Santoso, "Analisis sentimen Twitter bahasa Indonesia menggunakan pendekatan machine learning," *J. Inform. Kaputama (JIK)*, vol. 6, no. 2, pp. 129–136, Jul. 2022.

[49] R. Susetyoko, W. Yuwono, E. Purwantini, and N. Ramadijanti, "Perbandingan metode random forest, regresi logistik, naïve Bayes, dan multilayer perceptron pada klasifikasi uang kuliah tunggal (UKT)," *J. Infomedia, Tek. Inform. Multimed. Jar.*, vol. 7, no. 1, pp. 8–16, Jun. 2022, doi: 10.30811/jim.v7i1.2916.

[50] N.K. Hasibuan, S. Dur, and I. Husein, "Faktor penyebab penyakit diabetes melitus dengan metode regresi logistik," *G-Tech, J. Teknol. Terap.*, vol. 6, no. 2, pp. 257–264, Oct. 2022, doi: 10.33379/gtech.v6i2.1696.

[51] D. Ghazi, D. Inkpen, and S. Szpakowicz, "Detecting emotion stimuli in emotion-bearing sentences," in *Comput. Linguist. Intell. Text Process.*, 2015, pp. 152–165. doi: 10.1007/978-3-319-18117-2_12.

[52] S. Khairunnisa, K. Adiwijaya, and S.A. Faraby, "Pengaruh text preprocessing terhadap analisis sentimen komentar masyarakat pada

media sosial Twitter (Studi kasus pandemi COVID-19)," *J. Media Inform. Budidarma*, vol. 5, no. 2, pp. 406–414, Apr. 2021, doi: 10.30865/mib.v5i2.2835.

[53] R.I. Pristiyanti, M.A. Fauzi, and L. Muflikhah, "Sentiment analysis peringkasan review film menggunakan metode information gain dan k-nearest neighbor," *J. Media Inform. Budidarma*, vol. 2, no. 3, pp. 1179–1186, Mar. 2018.

[54] J. Petrus, "Kerangka kerja tokenisasi berdasarkan struktur kalimat bahasa Indonesia," Undergraduate thesis, Universitas Sriwijaya, Palembang, Indonesia, 2023.

[55] I. Gotama, S. Hariyanto, and H. Wijaya, "Klasifikasi berita hoaks topik COVID-19 dengan klasifikasi Rocchio dan cosine similarity," *Algor,* vol. 2, no. 1, pp. 84–92, Nov. 2020.

[56] E. Junianto and R. Rachman, "Implementation of text mining model to emotions detection on social media comments using particle swarm optimization and naive Bayes classifier," in *2019 7th Int. Conf. Cyber IT Serv. Manag. (CITSM)*, 2019, pp. 1–6, doi: 10.1109/CITSM47753.2019.8965382.

[57] 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.

[58] A.C. Darmawan, "Pengembangan aplikasi berbasis web dengan python flask untuk klasifikasi data menggunakan metode decision tree C4.5," Undergraduate thesis, Universitas Islam Indonesia, Sleman, Indonesia, 2023.