SHM PUBLISHER
SURYA HIJAU MANFAAT

Journal of Student Research Exploration

https://shmpublisher.com/index.php/josre
p-ISSN 2746-7686 | e-ISSN 2746-0991

Sentiment analysis of comments on youtube on react native vs flutter using the support vector machine and naïve bayes classifier algorithm

Adika Akbar Kurniawan[1]
[1] Information Systems Study Program, Universitas Negeri Semarang, Indonesia

Article Info

Article history:

Received July 10, 2024
Revised February 9, 2026
Accepted April 2, 2026

Keywords:

Sentiment analysis
Classification
Naïve bayes classifier
Support vector machine
Preprocessing

ABSTRACT

The study analyzed the sentiment of YouTube comments about React Native and Flutter, two popular platforms in cross-platform mobile app development, using the algorithms of Support Vector Machine and Naïve Bayes Classifier. The data was collected through YouTube comment scraping and analyzed using Google Colab with preprocessing techniques such as case folding, text cleaning, stopword removal, and stemming. Frequency-Inverse Term Document Frequency (TF-IDF) is used to extract features. The results showed that SVM was more effective in classifying positive and neutral sentiments, while NBC was superior in positive sentiment. Both algorithms have difficulty identifying negative sentiment, especially with the dominance of positive sentiment data. The study suggests the use of advanced strategies such as better feature selection or ensemble learning to improve the accuracy of sentiment classification on social media.

1. Introduction

Mobile app development is typically done per platform, which means that apps cannot be released on platforms that are not intended to be used. Because apps are made with distinct tools and languages for every platform, this kind of development is often referred to as a native development strategy. If the app wants to be used on multiple platforms, it must be written twice: utilizing Objective-C or

[1] ***Corresponding Author:***
Adika Akbar Kurniawan,
Faculty of Mathematics and Natural Sciences,
Semarang State University,
Sekaran, Mt. Pati District, Semarang City, Central Java, Indonesia
Email: adikaakbarlove@students.unnes.ac.id
DOI: https://doi.org/10.52465/josre.v3i2.411

Swift with Xcode for iOS and Android Studio with Java, Kotlin, or C++. This is a significant consequence of this development methodology [1].

Many mobile apps require compatibility with multiple operating systems. Undoubtedly, developing an individual native app for each platform requires a lot of time and financial resources. That's the reason why cross-platform solutions have emerged [2]. The term "cross-platform mobile development" is frequently used to describe it, and it covers a broad spectrum of technical frameworks and conceptual development techniques [3]. Typically, It is possible to define or develop apps that work across platforms with minimal to no platform-specific changes using a single codebase. But there are differences in the amount of code sharing between frameworks and techniques on different platforms. According to certain case studies, businesses and industry practitioners employ cross-platform development frameworks to create live platform apps even though they can assist in creating applications that can run across many platforms [4].

There are many frameworks that can be used to create cross-platform mobile apps, and React Native is one of the most widely used by large companies. However, learning this framework has some challenges, such as understanding the concepts and ecosystem of React and how declarative programming works [5].

Google introduced a new mobile SDK named Flutter at the end of 2016. React Native served as inspiration for Flutter apps, which are also cross-platform and reduce the expense and complexity of creating iOS and Android apps. Since Flutter was created entirely from scratch, only Google had used it for commercial projects at the time of writing this report in August 2017 [6]. Flutter aims to make multiplatform software development easier by using a single source code [7]. Using Flutter, mobile apps are mostly made up of widgets, which are collections of lines of code in the form of a tree [8].

Cross-platform frameworks, like React Native and Flutter, have been considered and utilized by a variety of organizations on a number of occasions. But none of them are adequate to meet the demands of industrial expansion. Despite all of their predecessors, React Native and Flutter have garnered a lot of attention and people are quite excited about their future with assistance from Facebook and Google [6].

The debate between React Native and Flutter has been a hot topic for the past few years, one of which is on a platform as big as YouTube. Because YouTube Social media and search engine are the second most popular worldwide. By 2020, YouTube will have over 2.1 billion users, over a billion hours of video watched daily, and over 500 hours of fresh content submitted every minute. Published data show that over 95% of Internet users frequently interact with YouTube in over 88 different languages [9]. In one of the videos on YouTube titled "React Native vs Flutter - I built the same chat app with both", the video contains a comparison between React Native and Flutter and mobile developers also commented on the video. Negative and positive comments from them can be identified through sentiment analysis [10].

Sentiment analysis is a component of data mining text processing, which uses artificial intelligence, statistics, and machine learning [11]. Using the supervised method, sentiment analysis will usually be carried out in four stages: preprocessing, feature extraction, model development, and performance evaluation. The main process to obtain an existing dataset classifier model is to choose the right preprocessing, feature extraction, and machine learning methods [12].

2. Method

The data collected through the scraping process will be analyzed using Google Colab. Furthermore, the calculation of the Supprot Vector Machine and Naive Bayes classification algorithm will be used to analyze sentiment.

Sentiment Analysis

Sentiment analysis, aka opinion mining, is the computational study of opinions, touches, and emotions expressed textually [13]. Sentiment analysis is used in cases where a set of text documents contains sentiments about a subject. The goal is to identify whether a comment is positive, negative, or neutral by extracting the properties and components of the object from each dataset.

Data Crawling Methods

This research utilizes data obtained from the comment column on the YouTube video platform. The procedure of gathering data is executed through the use of APIs offered by Google Cloud services. Following that, scraping methods and the Python programming language are employed to extract comments from YouTube videos.

Classification Method

Classification is an important part of data mining, a set of data that has been trained with a predefined class. Categorize features into appropriate classes. The vector of available training features, as well as its classes, are known to be used in sorter design. Identification of this pattern is called supervised [14].

According to [15], Classification is the process of categorizing a newly presented thing by looking at its features and allocating it to a predetermined class. Classification is to make decisions from invisible cases by building examples of previous decisions.

In supervised learning, special attention is paid to one variable. The variable level of the learning task is determined by the response. Dependent, outcome, explanatory, output, label, or target variables are also known as response variables. Responses and other variables are theoretically equivalent. This variable only serves as an interesting variable to predict the value when another variable (explainer) is available [16]. Among several classification algorithms, this study uses the Support Vector Machine and Naïve Bayes Classifier methods.

Support Vector Machine

Support vector machines are commonly used to instantiate data based on conventional statistical theory and classify linear and nonlinear information. All of the training vectors are classified into subclasses using a decision surface created by the support vector machine [17]. In simple terms, the SVM concept maximizes the distance between classes in an effort to find a hyperplane. SVMs can provide strong generalization capacity for subsequent data in this way [18].

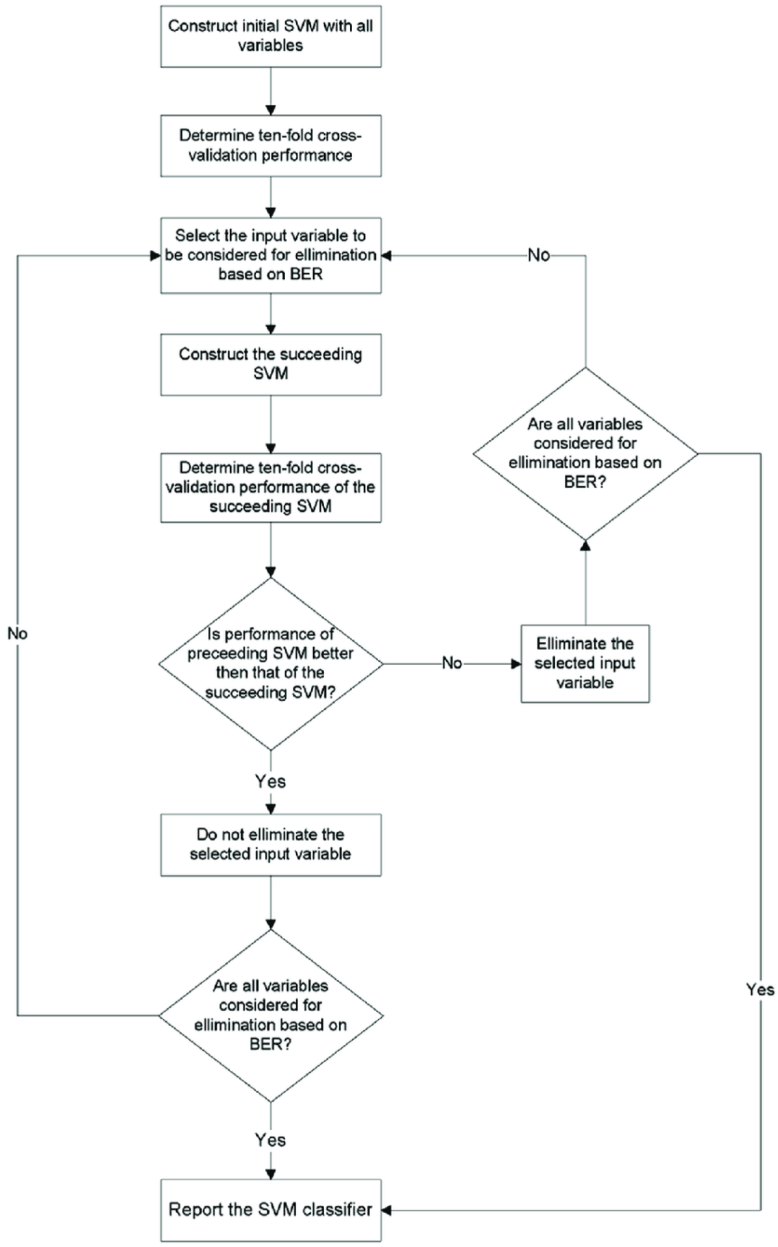

Figure 1. Support vector machine flowchart [19]

Naïve Bayes Classifier

The Naïve Bayes Classifier, created by the British scientist Thomas Bayes, utilizes statistical calculations and probabilities to predict future probabilities based on previous data. Compared to other classifier models, the Naïve Bayes Classifier theory has better performance [20]. Through the use of probabilities and statistics to classify a set of data, this strategy can select data. In this instance, the probability is computed by projecting the past probability into the future. This is the Naïve Bayes Classifier formula:

$$P\frac{H}{X} = \frac{p\frac{X}{H}.p(H)}{p(X)} \quad (1)$$

Where X is the input vector with the feature and H is the class label, Naïve Bayes can be written with P(H|X).

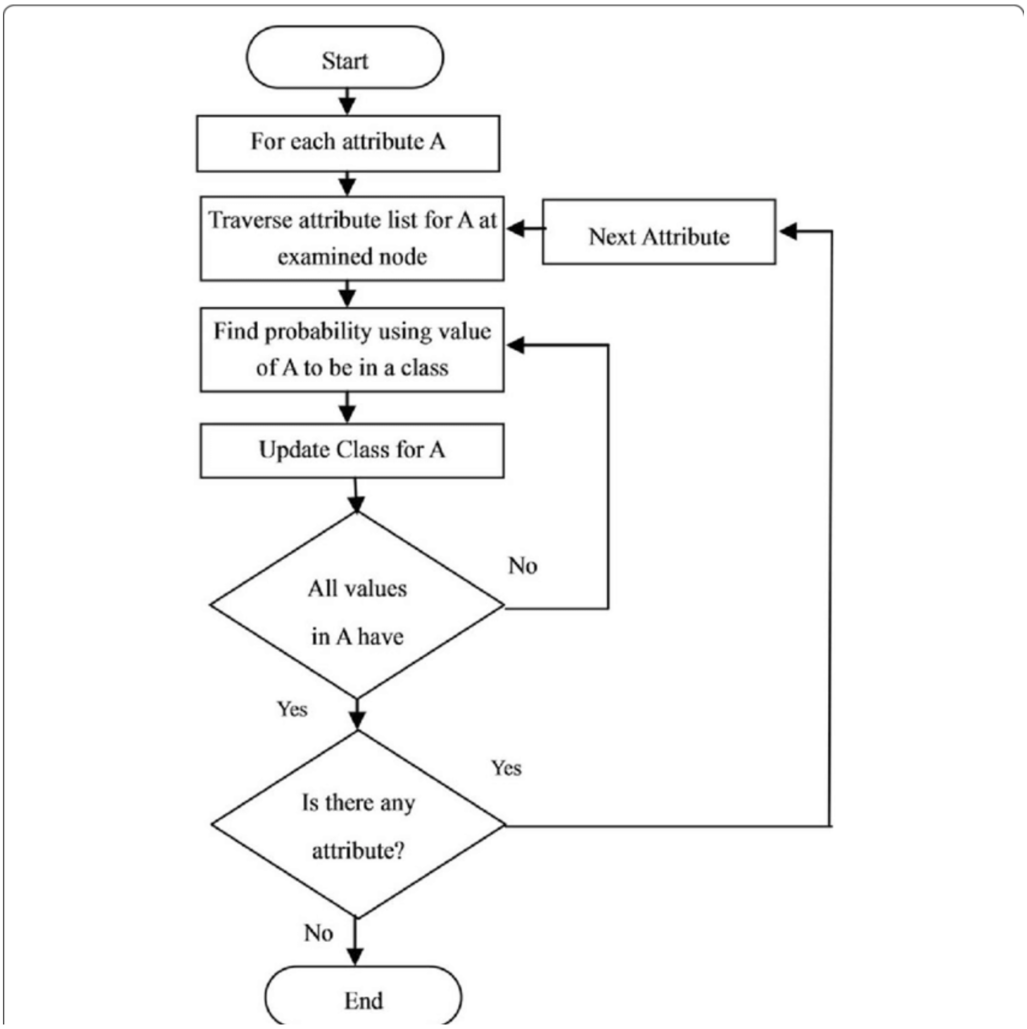

Figure 2. Naïve Bayes flowcart [21]

Kernel Support Vector Machine

Vapnik introduced SVM as an idea in the domain of pattern recognition. This study uses several SVM kernel functions, including linear, polynomial, radial base function, and sigmoid. The goal is to create separate non-linear data linearly when transferring data into a large-dimensional space [22]. Here are some types of SVM kernel functions as follows:

Kernel linear

$$K(x_i, x_j) = x \frac{T}{i} . x_j \quad (2)$$

The function of a linear kernel is the multiplication of the points of two vectors.

Kernel radial basis function

$$K(x_i, x_j) = exp(-\gamma \,||\, x_i - x_j||2), \gamma > 0 \quad (3)$$

In this case, the influence of a single training sample is measured using γ, a positive parameter.

Kernel polynomial

$$K(x_i, x_j) = (\gamma \, x \frac{T}{i} \, x_j + r)d \,, \gamma > 0 \quad (4)$$

A function with degrees d, where the parameters are r and d.

Performance Matrix

This performance matrix consists of various sizes, including True Positives, True Negatives, False Positives, and False Negatives. These measures are then used to calculate accuracy, precision, recall, and other metrics.

Accuracy
Calculate the percentage of correct predictions from the overall model's predictions.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

Recall
Calculate the percentage of correct positive predictions for each case that is actually positive.

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

Research Stage

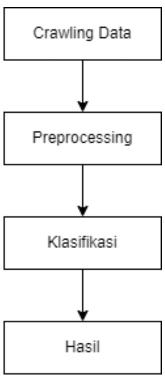

Figure 3. Research process

3. Results and Discussion

Through comments on YouTube videos, sentiment analysis was obtained through the use of SVM and NBC algorithms in the study analysis. Processing data in Google Colab is the next step after the data collection procedure. Google offers a free machine learning service called Colab. The platform facilitates user collaboration in real-time, allowing users to browse and execute Python or Jupyter code, as well as allowing users to submit and store code and data. Python code is used to import data into Google Colab and run data modeling using SVM and NBC algorithms.

Data Collection

Identifying positive or negative responses from YouTube video comments whose comments have been datasetd and processed is the main goal of this study. The video "React Native vs Flutter - I built the same chat app with both" can be accessed via the following link: https://www.youtube.com/watch?v=X8ipUgXH6jw. This is a video whose comments will be processed. We get a dataset of comments from the Fireship channel on YouTube. We managed to get 1275 comments from 1478 comments on the video. Here is the dataset that we have crawled:

Table 1. Youtube comment dataset

publishedAt	author	text	LikeCount
09/04/2024 09:07	@moccomocco	How much longer until Google decides to kill Flutter?	0
08/04/2024 05:58	@segganew	Sniffles in Kotlin/Multiplatform	0
07/04/2024 05:38	@jhoncarlogabato6352	SGXV0	0
28/03/2024 11:30	@AlexHadfield	Swift and Kotlin all the way!	1
	Total Comments: 1275		

Labeling

As part of the sentiment analysis labeling procedure, the text to be examined is labeled with a specific sentiment, such as positive, negative, or neutral. Sorting the text according to the sentiment it expresses is the purpose of this method. A total of 1275 comments were successfully categorized, there were 592 positive comments, 166 negative comments, and 517 neutral comments.

Preprocessing

An important step in the technique and application of data mining is the preprocessing method [23]. Because this is an important step, for this study there are four steps for preprocessing, namely case folding, cleaning text, stopword removal, and stemming. The goal is to clean and standardize the text before further analysis or processing.

Case folding
Sometimes, writing errors lead to compositions that include capital letters or similar characters, which can lead to a lack of coherence [24].

Cleaning text
Removal of special characters, symbols, numbers, and excessive whitespace.

Removing stopwords
The purpose of stopwords is to reduce the dimension of space that looks heavy by eliminating unimportant words using literature. Words that are commonly used in texts, such as prepositions and nouns that have no meaning in a document.

Stemming
The stemming process involves applying established rules to return the suffix to the root word. The removal of prefixes, infixes, suffixes, and confixes from affixed nouns is part of this process.

Stemming plays an important role in translation, grouping documents that reduce the number of different indexes for a particular document and information search for web searches. For example, Sastrawi, Sastrawi is a stemmer library. The library is open source and can be accessed through GitHub. The website states that the dictionary of root words plays a major factor in the stemming process when using this stemmer [24].

Normalizatiom

Normalization, which applies stemming and lemmatization to the text, is the final stage of preprocessing. The word-for-word separation of the text will result in a text that can stand alone in a sentence. Writing comments that usually contain typos or other errors. With the use of WordNet Lemmatizer and a network of cognitive synonyms for tokens (words) based on a sizable lexicon data bank of NLTK (Natural Language Toolkit), it is expected that the text will be understandable after this normalization procedure. After completing the text preparation step, the data is separated into individual texts and displayed as a wordcloud.

Figure 4. Wordcloud

Feature Extraction

Many feature extraction techniques are available, including Term Frequency-Inverse Document Frequency (TF-IDF), which is often used in sentiment analysis and has shown strong feature extraction performance from datasets [25]. Based on the calculation, a weight will be assigned to each term that appears in the review. Weighting terms, or assigning a value to each term in the pre-processed review data, is what we'll do during this phase of feature extraction [26]. The results of this word weighting will then be processed as data for the classification process.

Table Results

Table 2. SVM and NBC Classification Results

Method	Classification		Accuracy (%)	Precission (%)	Recall (%)
	Ratio = 20:80				
SVM	Positive	150		77	89
	Negative	6	80	83	20
	Neutral	99		84	83
	Ratio = 60 : 40				
	Positive	404		69	74
	Negative	12	70.7	100	12
	Neutral	349		72	79
	Ratio = 90 : 10				
	Positive	803		56	86
	Negative	0	59.2	0	0
	Neutral	345		66	49
	Ratio = 20 : 80				
NBC	Positive	215		60	98
	Negative	1	64.7	100	4
	Neutral	39		92	36

Ratio = 60 : 40				
Positive	651		52	96
Negative	2	56.6	100	2
Neutral	112		82	29
Ratio = 90 : 10				
Positive	1041		49	96
Negative	0	50.8	0	0
Neutral	107		71	16

Based on the table of classification results using the SVM and NBC algorithms, several important points can be drawn:

Classification performance

a) Judging from the table, SVM has better classification accuracy than Naive Bayes Classifier (NBC) for most of the training data ratios.
b) At a 90:10 training data ratio, SVM achieved the highest accuracy of 59.2%, while NBC had the lowest accuracy of 50.8%.

Precision and recall

a) SVM shows superior accuracy in the classification of positive and neutral sentiments, but for negative sentiments has lower precision
b) NBC has better recall for classification only on positive sentiment, but for negative and neutral sentiment NBC has lower recall results.

Influence of training data ratio

a) For SVM, a 20:80 training data ratio (more negative and neutral training data) resulted in the highest accuracy, while a 90:10 ratio (more positive training data) resulted in the lowest accuracy.
b) For NBC, the ratio of training data did not have much significant effect on classification accuracy.

Classification of negative sentiments

a) Both SVM and NBC have difficulty classifying negative sentiment well, especially at a 90:10 training data ratio where no negative data is properly classified.
b) As part of the sentiment analysis labeling procedure, the text to be examined is labeled with a specific sentiment, such as positive, negative, or neutral. Sorting the text according to the sentiment it expresses is the purpose of this method. A total of 1275 comments were successfully categorized, there were 592 positive comments, 166 negative comments, and 517 neutral comments.

4. Conclusion

Based on the study's findings, some key conclusions about the sentiment analysis of mobile developers' YouTube comments comparing Flutter vs. React Native using the Support Vector Machine (SVM) and Naïve Bayes Classifier (NBC) algorithms may be drawn. In general, the Support Vector Machine appears to be more effective at categorizing positive and neutral sentiments, while the Naïve Bayes Classifier excels at classifying positive sentiment. However, both had difficulty accurately classifying negative attitudes when the majority of training data contained positive sentiments. To improve its performance, you can explore other strategies such as better feature selection or ensemble learning.

REFERENCES

[1] T.-M. Grønli, J. Hansen, G. Ghinea, and M. Younas, "Mobile Application Platform Heterogeneity: Android vs Windows Phone vs iOS vs Firefox OS," in *2014 IEEE 28th International Conference on Advanced Information Networking and Applications*, 2014, pp. 635–641. doi: 10.1109/AINA.2014.78.

[2] H. Heitkötter, T. A. Majchrzak, and H. Kuchen, *Cross-platform model-driven development of mobile applications with MD 2.* 2013. doi: 10.1145/2480362.2480464.

[3] C. Rieger and T. A. Majchrzak, "Towards the definitive evaluation framework for cross-platform app development approaches," *J. Syst. Softw.*, vol. 153, pp. 175–199, 2019, doi: 10.1016/j.jss.2019.04.001.

[4] A. Biørn-Hansen, C. Rieger, T. M. Grønli, T. A. Majchrzak, and G. Ghinea, "An empirical investigation of performance overhead in cross-platform mobile development frameworks," *Empir. Softw. Eng.*, vol. 25, no. 4, pp. 2997–3040, 2020, doi: 10.1007/s10664-020-09827-6.

[5] A. J. Irawan, F. A. T. Tobing, and E. E. Surbakti, "Implementation of Gamification Octalysis Method at Design and Build a React Native Framework Learning Application," in *2021 6th International Conference on New Media Studies (CONMEDIA)*, 2021, pp. 118–123. doi: 10.1109/CONMEDIA53104.2021.9617171.

[6] W. Wu, "React Native vs Flutter , cross-platform mobile application frameworks," no. March, 2018.

[7] S. Santoso, D. J. Surjawan, and E. D. Handoyo, "Pengembangan Sistem Informasi Tukar Barang Untuk Pemanfaatan Barang Tidak Terpakai dengan Flutter Framework," *J. Tek. Inform. dan Sist. Inf.*, vol. 6, no. 3, pp. 589–598, 2020, doi: 10.28932/jutisi.v6i3.3071.

[8] M. B. Sinatria, Oman Komarudin, and Kamal Prihamdani, "Penerapan Clean Architecture Dalam Membangun Aplikasi Berbasis Mobile Dengan Framework Google Flutter," *INFOTECH J.*, vol. 9, no. 1, pp. 132–146, 2023, doi: 10.31949/infotech.v9i1.5237.

[9] W. Osman, F. Mohamed, M. Elhassan, and A. Shoufan, "Is YouTube a reliable source of health-related information? A systematic review," *BMC Med. Educ.*, vol. 22, no. 1, pp. 1–12, 2022, doi: 10.1186/s12909-022-03446-z.

[10] M. R. Elfansyah and M. R. Perdana, "Analisis Sentiment Cyberbullying pada media Youtube menggunakan Algoritma Naïve Bayes," 2024.

[11] S. S. Khan, "Soft computing- a journey from statistics," *Int. J. Theor. Appl. Sci.*, vol. 9, no. 2, pp. 260–268, 2017.

[12] A. S. Aribowo, H. T. Jaya, U. Pembangunan, and N. Veteran, "An Evaluation of Preprocessing Steps and Tree-based Ensemble Machine Learning for Analysing Sentiment on Indonesian YouTube Comments," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 5, pp. 7078–7086, 2020, doi: 10.30534/ijatcse/2020/29952020.

[13] D. Alita and A. R. Isnain, "Pendeteksian Sarkasme pada Proses Analisis Sentimen

Menggunakan Random Forest Classifier," *J. Komputasi*, vol. 8, no. 2, pp. 50–58, 2020, doi: 10.23960/komputasi.v8i2.2615.
[14] A. P. Wibawa, M. G. A. Purnama, M. F. Akbar, and F. A. Dwiyanto, "Metode-metode Klasifikasi," *Pros. Semin. Ilmu Komput. dan Teknol. Inf.*, vol. 3, no. 1, p. 134, 2018.
[15] D. Kaur, R. Bedi, and S. K. Gupta, "Review of Decision Tree Data mining Algorithms: ID3 and C4.5," *Proceeding Int. Conf. Inf. Technol. Comput. Sci.*, vol. 8, no. 4, pp. 4–8, 2015.
[16] D. Valkenborg, M. Geubbelmans, A. J. Rousseau, and T. Burzykowski, "Supervised learning," *Am. J. Orthod. Dentofac. Orthop.*, vol. 164, no. 1, pp. 146–149, 2023, doi: 10.1016/j.ajodo.2023.04.010.
[17] L. Mohan, J. Pant, P. Suyal, and A. Kumar, "Support Vector Machine Accuracy Improvement with Classification," in *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*, 2020, pp. 477–481. doi: 10.1109/CICN49253.2020.9242572.
[18] K. A. Rokhman, B. Berlilana, and P. Arsi, "Perbandingan Metode Support Vector Machine Dan Decision Tree Untuk Analisis Sentimen Review Komentar Pada Aplikasi Transportasi Online," *J. Inf. Syst. Manag.*, vol. 3, no. 1, pp. 1–7, 2021, doi: 10.24076/joism.2021v3i1.341.
[19] R. Sonmez and B. SÖzgen, "A support vector machine method for bid/no bid decision making," *J. Civ. Eng. Manag.*, vol. 23, pp. 641–649, May 2017, doi: 10.3846/13923730.2017.1281836.
[20] R. A. Roni Anagora, R. Rudini, R. T. Rohmat Taufiq, A. D. J. Ahmad Dedi Jubaedi, R. W. Rio Wirawan, and Arman Syah Putra, "The Classification of Phishing Websites using Naive Bayes Classifier Algorithm," *Int. J. Sci. Technol. Manag.*, vol. 3, no. 2, pp. 553–562, 2022, doi: 10.46729/ijstm.v3i2.498.
[21] S. N Sneha and T. Gangil, "Analysis of diabetes mellitus for early prediction using optimal features selection," *J. Big Data*, vol. 6, Feb. 2019, doi: 10.1186/s40537-019-0175-6.
[22] K. Saputra, "Perbandingan Kinerja Fungsi Kernel Algoritma Support Vector Machine Pada Klasifikasi Penyakit Padi," *Ijccs*, vol. x, No.x, no. x, pp. 1–5, 2023.
[23] A. Torayev and C. Monroe, "Role of Discharge-Product Morphology and Formation Rates in Li–O 2 Battery Performance," *ECS Meet. Abstr.*, vol. MA2020-02, p. 490, Nov. 2020, doi: 10.1149/MA2020-022490mtgabs.
[24] D. Mustikasari, I. Widaningrum, R. Arifin, and W. H. E. Putri, "Comparison of Effectiveness of Stemming Algorithms in Indonesian Documents," *Proc. 2nd Borobudur Int. Symp. Sci. Technol. (BIS-STE 2020)*, vol. 203, pp. 154–158, 2021, doi: 10.2991/aer.k.210810.025.
[25] P. Prihatini, K. Indah, G. Sukerti, I. Indrayana, and I. Sudiartha, "Feature Extraction Performance on Classified Methods for Text Sentiment Analysis," pp. 1235–1243, 2023, doi: 10.5220/0010962900003260.
[26] M. Jamil, H. Hadiyanto, and R. Sanjaya, "Sentiment Analysis: Classifying Public Comments on YouTube in Disaster Management Simulation in Indonesia Using Naïve Bayes and Support Vector Machine," *Ing. des Syst. d'Information*, vol. 29, no. 2, pp. 437–446, 2024, doi: 10.18280/isi.290205.