Available online at www.HighTechJournal.org

HighTech and Innovation Journal

Vol. 3, No. 4, December, 2022

ISSN: 2723-9535

Multi-class SVM Classification Comparison for Health Service Satisfaction Survey Data in Bahasa

Gede Indrawan [1*], Heri Setiawan [1], Aris Gunadi [1]

[1] *Universitas Pendidikan Ganesha, Bali 81116, Indonesia.*

Received 17 September 2022; Revised 09 November 2022; Accepted 18 November 2022; Published 01 December 2022

Abstract

This study aimed to compare the Multi-class Support Vector Machine (MSVM) classification with the One-versus-One (OvO) and One-versus-Rest (OvR) approaches using unigram and bigram features. The study used the service satisfaction survey report of Denpasar public health centers by the Center for Public Health Innovation (CPHI), Medical School, Udayana University. As Bali is known as the world's main tourism destination, it is important to know about its supporting public health service through its representative capital city, Denpasar. Moreover, this study laid the foundation for the classification process using the available methods to fit in Indonesian health service satisfaction survey data, which assists in making decisions to improve health services. Since Bali is one of the provinces in Indonesia and all of those provinces refer to the same national regulation, health service satisfaction survey data that is in the Indonesian language (Bahasa) should have the same aspects, like category, priority, word-related matters (including abbreviations, acronyms, terminology), etc. that overall make it unique and need specific processing. That work was considered a contribution since there is no such study to the best of the author's knowledge and the foundation would be useful as a part of the future vision for the integrated system of Indonesian health big data. Since in reality, satisfaction survey data tends to be unbalanced, this study also compares the developed models using unigram and bigram features without and with feature selection (FS). Those features were then processed using the OvO MSVM and OvR MSVM models. *k*-fold cross-validation was used to divide training data and testing data and, at the same time, validate the models. Through experiments without and with FS, the OvO MSVM and OvR MSVM models with unigram features had better performance in general than the same models with bigram features. Without FS and with unigram features, comparable differences were found where the OvO MSVM model was slightly better on *accuracy* and *precision*, while the OvR MSVM model was slightly better on *recall* and the *F1* score. Without FS and with bigram features, comparable differences were also found, where the OvR MSVM model had slightly better performance than the OvO MSVM model. With FS and with unigram and bigram features, the OvR MSVM model had better performance in general than the OvO MSVM model.

Keywords: Bahasa; Classification; Multi-Class; Satisfaction Survey; Support Vector Machine.

1. Introduction

One of the foci of Indonesian research is information technology for big data development [1]. Health service goes towards this trend through Satu Data Kesehatan Indonesia (Indonesian One Health Data), which comes from the vision for an integrated system of health big data. Naturally, health services satisfaction survey data should be part of this integrated system to provide some good insights for future decisions regarding health services improvement. This study contributes to putting the foundation for the analysis of this health services satisfaction survey data, which is in Indonesian Language (Bahasa), through the classification process using the available methods to fit this kind of data.

* Corresponding author: gindrawan@undiksha.ac.id

http://dx.doi.org/10.28991/HIJ-2022-03-04-05

The dataset used was constructed by service satisfaction data of Denpasar public health centers provided by the Center for Public Health Innovation (CPHI), the Medical School, Udayana University [2]. In Indonesia, Public Health Centers, including Puskesmas (Bahasa acronym for "Pusat Kesehatan Masyarakat", the smallest health service unit in a certain area that directly serves the public or gives recommendations for the next health treatment at the higher-level unit) and RSUD (Bahasa abbreviation for "Rumah Sakit Umum Daerah", the regency public hospital, which is an upper-level unit above Puskesmas and a lower-level unit below the province public hospital).

Since all public health centers in Indonesia refer to the same national regulation [3, 4], health service satisfaction survey data should have the same aspects, like category, priority, word-related matters (including abbreviations, acronyms, and terminology), etc. that overall make it unique and need specific processing. Based on that, satisfaction survey data provided by CPHI could reflect general Indonesian health service satisfaction survey data. The limitation of this study relates to the relatively small number of data points provided that affect the developed models testing performance. Since this is the foundation laid by this study, that limitation could be improved through additional incremental satisfaction survey data on future implementation. The constraint of this study related to the classification method used, which is a multi-class Support Vector Machine (MSVM) since multi-class labels were involved in the satisfaction survey data. Related to the necessity of doing this research, Mishbahuddin [5] stated that health institutions must immediately evaluate themselves and develop strategic plans to improve the performance and competitiveness of health services by empowering strengths, weaknesses, opportunities, and threats (SWOT) factors. According to Sabilla [6], the quality of health services can be achieved through users' suggestions and input related to the user satisfaction level. The SWOT factors and user satisfaction level can be obtained through reviews or reports, as in the CPHI report that exposes the sentiment data [7].

Related to the use of MSVM, this study reviewed several classification methods for comparison. The review took several references from relatively older and more recent years to get insight in general into the method during that time. Hsu and Lin [8] found that in experiments on small datasets (the Statlog collection and the UCI Repository of machine learning databases), the "one-against-one" (One-versus-One, OvO) and Directed Acyclic Graph SVM (DAG SVM) methods were more suitable for practical use than other methods, like two such "all-together" methods and the binary-classification-based method "one-against-all" (One-versus-Rest, OvR). Lei and Govindaraju [9] proposed a Half-Against-Half (HAH) MSVM whose structure is the same as a decision tree, with each node as a binary SVM classifier that tells a testing sample belonging to one group of classes or the other. Both theoretical estimation and experimental results (using the UCI machine learning repository) showed that HAH has advantages over OVR and OVO-based methods in terms of evaluation speed and the size of the classifier model while maintaining comparable accuracy. Hsu [10] did a comprehensive evaluation of the performance of multiple supervised learning models, such as Logistic Regression (LR), Decision Trees (DT), Support Vector Machine (SVM), AdaBoost (AB), Random Forest (RF), Multinomial Naive Bayes (MNB), Multilayer Perceptrons (MLP), and Gradient Boosting (GB) to assess the efficiency and robustness, as well as limitations, of these models on the classification of textual data. SVM, LR, and MLP had better performance in general, with SVM being the best, while DT and AB had much lower accuracy among all the tested models. Polpinij & Luaphol [11] conducted different multi-class classification methods that applied to assigning automatic ratings for consumer reviews based on a 5-star rating scale, where the original review ratings were inconsistent with the content. Two-term weighting schemes (i.e., TF-IDF and TF-IGM) and five supervised machine learning algorithms, namely, k-NN, MNB, RF, XGBoost, and SVM, were compared. The dataset was downloaded from the Amazon website, and language experts helped to correct the real rating for each consumer review. The multi-class classifier model developed by SVM along with TF-IGM returned the best results for automatic ratings of consumer reviews.

Since this study involved text data in Bahasa, it is logical to review other related works in more detail, as shown in Table 1. Based on all of those studies, the use of the SVM algorithm on text datasets had relatively better performance, and at the same time, it raised curiosity about the performance of this algorithm and its several processing variants on satisfaction survey data from the CPHI report. Satisfaction survey data in this study were divided into six classes in total, consisting of five classes in Bahasa (refer to health service sectors [4]), namely "Pelayanan" (Service), "Administrasi dan Manajemen" (Administration and Management), "Sarana dan Prasarana" (Facility and Infrastructure), "Peralatan" (Equipment), and "Sumber Daya Manusia" (Human Resources), and an additional "Netral" (Neutral) class was added if satisfaction survey data did not match the previous classes. The use of a dataset in the form of text is strongly influenced by data preprocessing and the selection of relevant features to be used as input to the algorithm [12, 13]. The influence of the number of words commonly referred to as *n*-grams also affects the results of the *accuracy* score of the algorithm [14]. Therefore, in this study, unigram and bigram features were used to test the effect of *n*-gram on the algorithm, and because the dataset had more than two classes, OvO and OvR approaches were used (also as another constraint) in the MSVM classification.

Table 1. Related works using text data in Bahasa

No	Authors, Year	Problem/Objective	Methods	Results/Conclusions
1	Perdana et al. (2018) [15]	To investigate the classification of schizophrenia to reduce barriers to treating the disease.	The dataset came from medical record data of schizophrenia patients which were grouped into five classes and processed using SVM with the One-against-All (OvR) approach and testing using *k*-fold cross-validation.	The results obtained in this study had an *accuracy* rate of 59.09%. The result obtained was categorized as low *accuracy*. This is because the data used was unbalanced in each class, and also the patterns in each class are different so it is difficult to determine the best pattern.
2	Widyawati & Sutanto, (2019) [16]	To identify incoming messages from mobile phones in the form of SMS and classify them as unwanted, advertisements, fraud, and so on.	The dataset came from secondary data obtained from an existing source, namely the dataset of spam SMS in Bahasa, then uses Naïve Bayes Classifier (NBC) and SVM to classify.	NBC had the largest and best *precision* and *recall* test values if the algorithm did not go through stopword removal. It was also found that the initial misclassification of actual data was at least done by NBC using or not using stopword stages.
3	Alita et al. (2020) [17]	To identify public opinion regarding cellular telecommunication networks and Indonesian social security agency of health (BPJS) services, either categorized as positive, negative, or neutral sentiments.	Collecting the dataset from Twitter and classifying data using NBC and SVM with One-against-One (OvO) optimization and One-against-All (OvR) optimization.	The optimized SVM had better *accuracy*, *precision*, *recall*, and *F1* score compared to NBC. Among SVMs, OvO SVM was better on precision, recall, and F1 score, while OvR SVM was better on accuracy.
4	Pangestu (2020) [18]	To investigate Twitter users' opinions on mental health during the COVID-19 pandemic.	Collecting the dataset from Twitter and classifying data using NBC and SVM.	*Accuracy* results using NBC, SVM with the polynomial kernel, SVM with RBF kernel, and SVM with Linear kernel were 70.71%, 80.81%, 78.79%, and 71.73%, respectively.
5	Hermanto et al. [19]	To obtain the most accurate algorithm in the classification of student complaints data.	Collecting the dataset from academic system information and classifying data using NBC and SVM.	SVM with an *accuracy* of 84.45% and an *Area Under Curve* (AUC) of 0.922 outperformed NBC with an *accuracy* of 69.75% and an AUC of 0.679.
6	Fitriana & Sibaroni (2022) [20]	To classify public sentiments of the Indonesian railway's service for tweet data (positive, negative, and neutral sentiments) and to find the best accuracy when processed with large amounts of data.	The dataset came from Twitter which was then preprocessed and then processed using the multi-class SVM (MSVM) by combining several binary SVMs, namely One Against All (OvR) and One Against One (OvO). Five different weighting features were also investigated.	TF-IDF feature extraction approach with unigram feature outperformed other methods allowing the classifier to achieve the highest accuracy when working with larger datasets. The unigram TF-IDF combined with MSVM had the highest average accuracy of 80.59% compared to the other four models namely, bigrams (52.53%), trigrams (53.54%), unigrams + bigrams (76.13%), and word cloud (70.33%).
7	Dhammajoti et al. [21]	To implement several numerical representations and implementing resampling techniques (to handle imbalanced data), which then are followed by evaluating some popular supervised machine learning classification algorithms on user feedback in an educational institution.	Collecting the dataset from the e-learning system and evaluating it on Logistic Regression (LR), Random Forest (RF), SVM, NBC, and Decision Tree (DT) algorithms.	SVM performed the best in TF-IDF and BOW, and it indicated that SVM is the least biased of the other classifier in the case of highly imbalanced data. Relate to comparing RF and DT, RF was better than DT in almost all numerical representations and with or without the resampling technique. NBC performed the worst because it assumed an independent feature, but in text classification, each feature is co-related.
8	Sujadi et al. (2021) [22]	To investigate public opinion on the Covid-19 outbreak through Twitter given by the Indonesian people.	The dataset came from Twitter secondary data obtained from https://bisa.ai/ which was then preprocessed and then processed using NBC and SVM.	The *accuracy* results for NBC and SVM algorithms were 78.3% and 81.6%, respectively. If using *10*-fold cross-validation testing, the results for NBC and SVM algorithms were 69.8% and 74.4%, respectively.
9	Cikania (2021) [23]	To classify sentiments of user reviews of the HALODOC, an Indonesian telemedicine service application, during the Covid-19 pandemic.	The dataset was obtained from users' comments on the HALODOC application which were then used as input for NBC and SVM algorithms by testing using *accuracy*, *recall*, and specificity.	NBC had an accuracy rate of 87.77% with an AUC value of 57.11%, and a G-Mean of 40.08%, while SVM with RBF Kernel had an accuracy value of 86.1% with an AUC value of 60.149%, and a G-Mean value of 49.311%. Based on that, SVM with RBF Kernel model was better than NBC.

This paper is organized into several sections, i.e., Introduction, Methods, Result and Discussion, and Conclusion. Section Introduction describes the problem, related works, and motivation in this work. Section Methods covers the source data collection and raw data processing, dataset preprocessing, modeling, and the testing mechanism. Section Result and Discussion provides the testing result and related discussion. Section Conclusions consists of some important conclusion points.

2. Methods

Figure 1 shows the research process in the comparison of the OvO MSVM and OvR MSVM models.

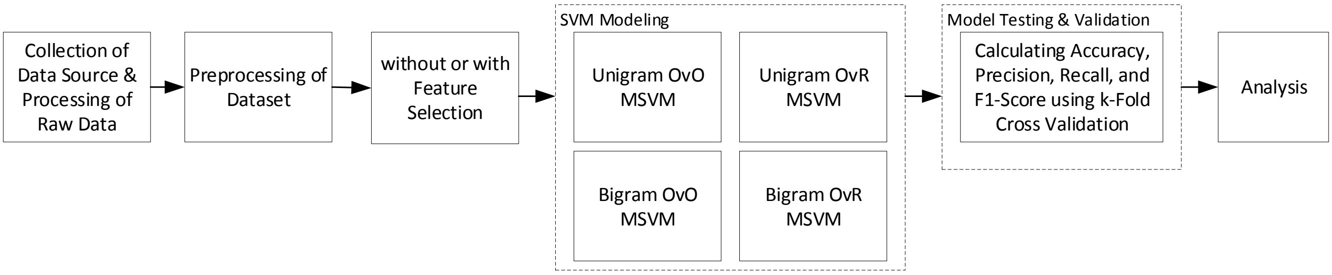

Collection of Data Source & Processing of Raw Data
Preprocessing of Dataset
without or with Feature Selection
SVM Modeling
Unigram OvO MSVM
Unigram OvR MSVM
Bigram OvO MSVM
Bigram OvR MSVM
Model Testing & Validation
Calculating Accuracy, Precision, Recall, and F1-Score using k-Fold Cross Validation
Analysis

Figure 1. Research methods

2.1. Source Data Collection and Raw Data Processing

The dataset in this study was obtained from the Denpasar Public Health Centers service satisfaction survey report 2021 by the CPHI. From the report, satisfaction survey data were obtained from users of health institutions in the Denpasar area. Suggestions/criticisms obtained from the report were still not in the format needed as input for the MSVM model, so it is necessary to change the format to suit the needs. The dataset created was labeled manually by experts and a suggestion/criticism inside the dataset was labeled by its highest priority class if it was appropriate across multiple classes (see the previous Introduction section). The order from the highest priority class (in Bahasa), namely "Pelayanan", "Administrasi dan Manajemen", "Sarana dan Prasarana", "Peralatan", and "Sumber Daya Manusia". For example, criticism 2 of Table 2 was labeled as "Peralatan" (Service) even though it was also appropriate for the class "Sarana dan Prasarana" (Facilities and Infrastructure). Note that suggestions/criticisms in Bahasa were written as it is based on the user input.

Table 2. Class labeling

No	Suggestions/Criticisms	Label
1	Pelayanan agar lebih ditingkatkan (Services to be further improved)	Pelayanan (Service)
2	Ada Petugas yg Main HP saat ada pasien, lahan parkir mobil kurang (There is staff play cellphone when there are patients, car parking space is less)	Pelayanan
3	Waktu pelayanan agar dipercepat (Service time to be faster)	Administrasi dan Manajemen (Administration and Management)
4	Perbaikan pada sistem antrian (Improvements to the queue system)	Administrasi dan Manajemen
5	Lahan parkir diperluas (The parking area should be expanded)	Sarana dan Prasarana (Facilities and Infrastructure)
6	Loket diperbanyak, ada tempat bermain untuk anak agar tidak bosan (There should be more counters and a playground area for children so they don't get bored)	Sarana dan Prasarana
7	Tidak ada alat cek darah, katanya rusak. Padahal mau disini kalau berobat atau rawat inap misalnya, tapi takut ga ada alat (There is no blood check tool, still broken as informed. The plan is to come here for treatment or hospitalization, but cancel because there's no such equipment)	Peralatan (Equipment)
8	Obat-obatannya kurang tersedia lengkap. Obat hipertensi. (The medicines are not fully available. Hypertension medication.)	Peralatan
9	Dokter spesialis ditingkatkan (Specialist doctors should be increased in number)	Sumber Daya Manusia (Human Resources)
10	Tambah tenaga medis agar lebih mudah dan cepat (Add medical personnel to make it easier and faster)	Sumber Daya Manusia

Changing the format is the process of inputting suggestions/criticisms into spreadsheet processing software and saving those data in that tool's file format. In this study, Microsoft Excel was used, and save the data in the ".xlsx" file format. Based on the results of the format change, 1031 lines of class-labeled suggestions/criticisms were obtained namely, 274 lines went into "Pelayanan", 240 lines went into ”Sarana dan Prasarana", 156 lines went into ”Sumber Daya Manusia", 104 lines went into ”Administrasi dan Manajemen", 29 lines went into ”Peralatan", and 228 lines went into "Netral".

2.2. Dataset Preprocessing

Before becoming input into the algorithm model, the dataset is changed which was originally text data into numeric data or numbers. This change process is also known as dataset pre-processing. This stage is the processing of raw data with several processing stages which can later be used as input for data visualization, machine learning, deep learning, and others [24]. In this case, the results of data processing will be used as input to the MSVM model. The steps in the process are shown in Figure 2.

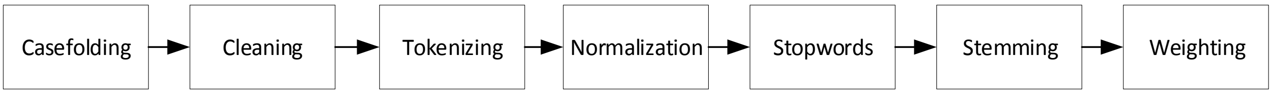


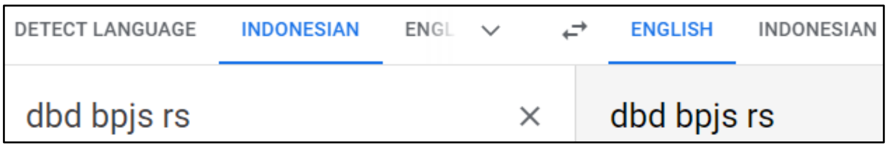
Figure 2. Dataset preprocessing

The case folding stage aims to change capital letters to lowercase letters in sentences. An example of case folding results were shown in Table 3. The cleaning process is the process of removing unnecessary text formatting, including tabs, new lines, back slices, ASCII codes, numbers, punctuation marks, excess spaces, and a character. Several Python libraries are used in this process [25]. The tokenizing stage aims to break a sentence into words [26]. The process utilizes the library from the Natural Language Toolkit (NLTK) [27] to process the dataset into tokens.

Table 3. Process result of case folding

No	Suggestions/Criticisms
1	pelayanan agar lebih ditingkatkan
2	ada petugas yg main hp saat ada pasien, lahan parkir mobil kurang
3	waktu pelayanan agar dipercepat
4	perbaikan pada sistem antrian
5	lahan parkir diperluas
6	loket diperbanyak, ada tempat bermain untuk anak agar tidak bosan
7	tidak ada alat cek darah, katanya rusak. padahal mau disini kalau berobat atau rawat inap misalnya, tapi takut ga ada alat
8	obat-obatannya kurang tersedia lengkap. obat hipertensi.
9	tambah tenaga medis agar lebih mudah dan cepat
10	dokter spesialis ditingkatkan

The normalization stage is the stage of changing abbreviations, non-standard words, and acronyms to become standard words of abbreviations, words, and acronyms. For example, the word “sy” (a non-standard abbreviation that means I), “aqu” (a non-standard word that means I), “RSUD” (an abbreviation that means regency public hospital), and “Puskesmas” (an acronym that means public health center) become “saya”, “aku”, “Rumah Sakit Umum Daerah”, and “Pusat Kesehatan Masyarakat”, respectively. This stage uses a list of words that are often used in short message sentences from sources which are then readjusted manually [28]. The word list contains 1029 words that have been given equivalent words according to Bahasa standard words. An additional list of words related to health in Bahasa (from the satisfaction survey data and the Indonesian Ministry of Health [29]) was also developed and strengthened the contribution to this classification study, specifically in this normalization stage. Figure 3 shows several examples of health terminology in Bahasa, like “dbd” (abbreviation for “demam berdarah dengue” or dengue fever), “bpjs” (abbreviation for “badan penyelenggara jaminan sosial” or Indonesian social security agency of health, exists in satisfaction survey data), and “rs” (abbreviation for “rumah sakit” or hospital, exists in satisfaction survey data). From a different perspective (still related to word processing), even Google does not understand them for translation. neither do existing classification algorithms, to the best authors’ knowledge.



dbd bpjs rs

dbd bpjs rs

Figure 3. Google translation of several examples from the health acronyms in Bahasa

The stopwords stage is a process for removing words that are not used, for example, Bahasa words “di” (at), “nggak” (no), “tadi” (just now), etc. The deletion uses the library from NLTK and the corpus source uses research results from Tala [30] and updates with additional words manually. At the stemming stage, words that have affixes are changed to basic words. This process uses the PySastrawi Python library [31]. Another swifter library is also used [32] which functions to help speed up the stemming process [33]. An example of the results of the stages from casefolding to stemming can be seen in Table 4.

Table 3. The process results from casefolding to stemming

No	Suggestions/Criticisms
1	[layan, tingkat]
2	[tugas, main, hp, pasien, lahan, parkir, mobil]
3	[waktu, layan, cepat]
4	[baik, sistem, antri]
5	[lahan, parkir, luas]
6	[loket, banyak, main, anak, bosan]
7	[alat, cek, darah, rusak, obat, rawat, inap, takut, alat]
8	[obat, obat, sedia, lengkap, obat, hipertensi]
9	[tenaga, medis, mudah, cepat]
10	[dokter, spesialis, tingkat]

The weighting process is changing text token data into numeric data. TF-IDF is a numerical statistical method used to describe how important a word is in a document [34-37]. Based on the results of this process, there are 645 features for unigram and 1902 features for bigram, so the size of the input data for the MSVM model is (1031, 645) for unigram features and (1031, 1902) for bigram features.

The feature selection (FS) stage reduces the feature size of a dataset to obtain a smaller dataset subset that contains features that are relevant to the target. In addition, it eliminates data redundancies and outliers, improves learning performance, increases efficiency in computing, reduces memory usage, and can build a better general model [38, 39]. This study uses an FS technique called the Extratrees Classifier which is a classifier with an ensemble approach and is used for classification and regression problems [40-42]. Several studies have found performance improvements when using the Extratrees Classifier [43, 44] and obtaining high *accuracy* values even without parameter tuning [45]. In this study, the FS process was carried out by processing the pre-processed data from the TF-IDF weighting, then processing using the ExtraTreesClassifier module in scikit-learn [46]. Extratrees Classifier will decide tree randomly and will use the entire decision tree model to make a prediction tree. The SelectFromModel module from scikit-learn was used to retrieve the model in the Extratrees Classifier for use in the MSVM model. From the previous results, the feature size is 645 and 1902 for unigram and bigram, respectively. This dataset was then used as input in the Extratrees Classifier FS process. Several processes were carried out to obtain the best feature size from the results of *accuracy*, *precision*, *recall*, and *F1* scores. The best score was obtained for the size of 104 features for unigram and 401 features for bigram with the number of parameter settings *n_estimators* = 100. From the results of this FS, there is a reduction in the size of 541 features for unigram and 1501 features for bigram. So, for input into the MSVM model, the feature sizes to be used were (1031, 104) for unigram, and (1031.401) for bigram.

2.3. Modeling

At this stage, MSVM models were created using the unigram OvO, bigram OvO, unigram OvR, and bigram OvR approaches. The modeling used the Scikit-learn library [46] with the *SVC* module for the OvO MSVM models and the *LinearSVC* module for the OvR MSVM models. For *SVC*, given training vectors $x_i \in \mathbb{R}^p$, *i* = 1,…, *n*, in two classes, and a vector $y \in \{1, -1\}^n$, the goal is to find $w \in \mathbb{R}^p$ and $b \in \mathbb{R}$ such that the prediction given by $sign(w^T \phi(x) + b)$ is correct for most samples. *SVC* solves the following primal problem:

$$\begin{aligned} &\underset{w, b, \zeta}{min} \frac{1}{2} w^T w + C \textstyle\sum_{i=1}^{n} \zeta_i \\ &\text{Subject to } \ y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i \\ &\zeta_i \geq 0, i = 1, \dots, n \end{aligned} \tag{1}$$

Intuitively, the margin (by minimizing $||w|| = w^T w$) is trying to be maximized, while incurring a penalty when a sample is misclassified or within the margin boundary. Ideally, the value $y_i(w^T \phi(x_i) + b)$ would be ≥ 1 for all samples, which indicates a perfect prediction. But problems are usually not always perfectly separable with a hyperplane, so some samples are allowed to be at a distance ζ_i from their correct margin boundary. The penalty term *C* controls the strength of this penalty, and as a result, acts as an inverse regularization parameter. The dual problem to the primal is:

$$\begin{aligned} &\underset{\alpha}{min} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha \\ &\text{Subject to } \ y^T \alpha = 0 \\ &0 \leq \alpha_i \leq C, i = 1, \dots, n \end{aligned} \tag{2}$$

where e is the vector of all ones, Q is an n by n positive semi-definite matrix, $Q_{ij} \equiv y_i y_j K(x_i, x_j)$, where $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ is the kernel. The terms α_i are called the dual coefficients, and they are upper-bounded by C. This dual representation highlights the fact that training vectors are implicitly mapped into a higher (maybe infinite) dimensional space by the function ϕ. Once the optimization problem is solved, the output of the decision function for a given sample x becomes:

$$\sum_{i \in SV} y_i \alpha_i K(x_i, x) + b \tag{3}$$

and the predicted class corresponds to its sign. Sum over the support vectors (i.e. the samples that lie within the margin) is only needed because the dual coefficients α_i are zero for the other samples.

For *LinearSVC*, the primal problem can be equivalently formulated as:

$$\min_{w,b} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \max(0,1 - y_i(w^T \phi(x_i) + b)) \tag{4}$$

where the hinge loss is used. This is the form that is directly optimized by *LinearSVC*, but unlike the dual form, this one does not involve inner products between samples, so the famous kernel trick cannot be applied. This is why only the linear kernel is supported by LinearSVC (ϕ is the identity function).

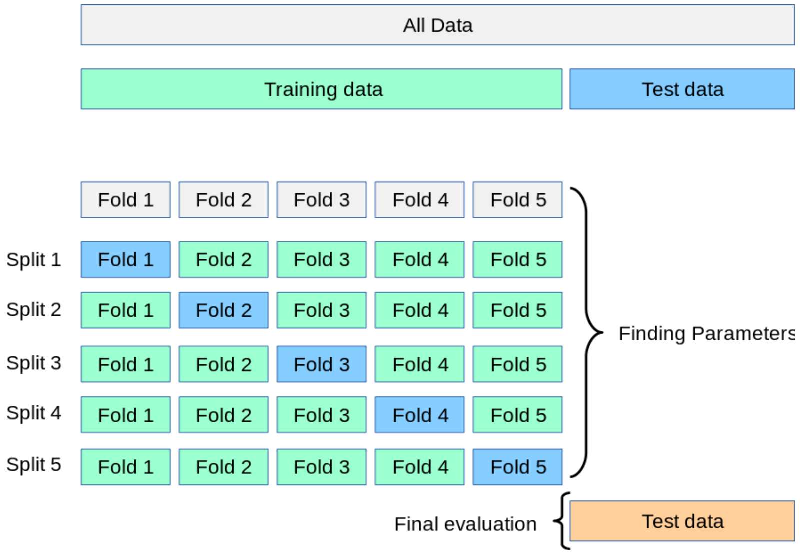
The approach to the *SVC* module used *LibSVM* calculations [47], while the approach to the *LinearSVC* module used *LibLinear* calculations [48]. The difference between *LibSVM* and *LibLinear* is that *LibSVM* is used to work on both linear and non-linear kernels, while *LibLinear* can only be used on linear kernels. In addition to that, the advantage of *LibLinear* is that it can have several variations in the regularization method and has a *speed* (*time complexity*) of $O(n)$, while *LibSVM* has a *time complexity* of $O(n^2)$ to $O(n^3)$.

Parameter settings in OvO SVM models include parameters C = 10, *kernel* = 'linear', *decision_function_shape* = 'OvO', and *max_iter* = 10000. In the OvR SVM models, the parameter settings include parameters C = 10, *multi_class* = 'OvR', and *max_iter* = 10000. Parameter C is a regular parameter that functions to control the trade-off between slack and margin variable penalties. Parameter *kernel* makes it possible to implement a model in a higher dimensional space without having to define a mapping function from input space to feature space, in which case the kernel used is a linear kernel. Parameter *decision_function_shape* determines the approach used in the SVM algorithm, i.e. the OvO approach. Parameter *max_iter* determines the maximum number of iterations performed by the algorithm. Parameter *multi_class* with value 'OvR' was set to use the OvR approach.

The difference in approach between OvO MSVM and OvR MSVM, regardless of the library used, is in determining class membership. In OvO MSVM, the determination of class membership is based on a voting strategy and if there are the same number of votes then the classification results are determined by the highest number of votes with the smallest index [8]. Meanwhile, in OvR MSVM, the determination of membership is based on the highest value of membership and if there are the same values it will be determined based on the smallest index of them [49].

2.4. Model Testing

Model testing uses the *k*-fold cross-validation method [46] where the dataset will be split into two parts, namely training data and testing/validation data. The training data will be broken down into k = 5 folds, as shown in Figure 4-a. If k is set to more than 5, there will be scores that cannot be calculated due to the imbalance in the number of data on each class label (see Figure 4-b).



(a)

(b)

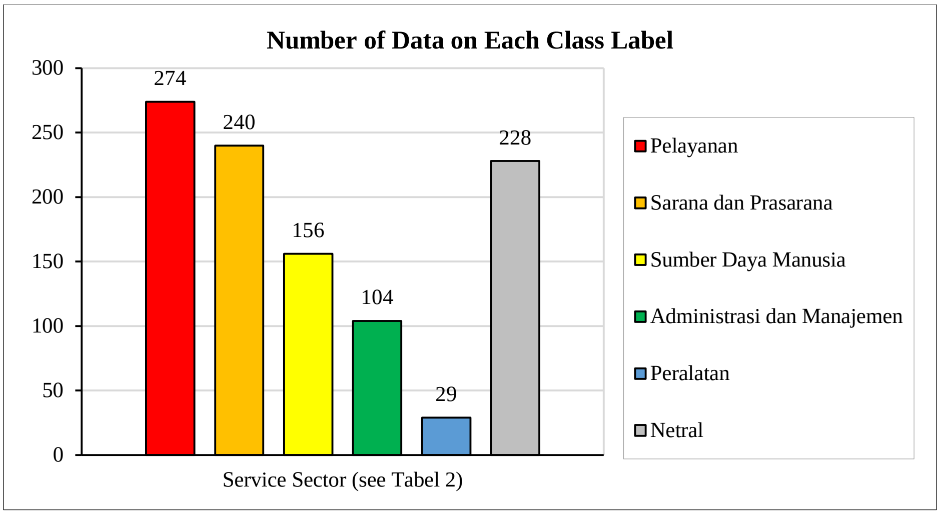


Figure 2. Model testing: (a) 5-fold cross-validation; (b) imbalance in the number of data on each class label

Testing was carried out on the OvO MSVM and OvR MSVM models, with unigram and bigram features, to see the resulting performance comparison either without or with FS.

3. Result and Discussion

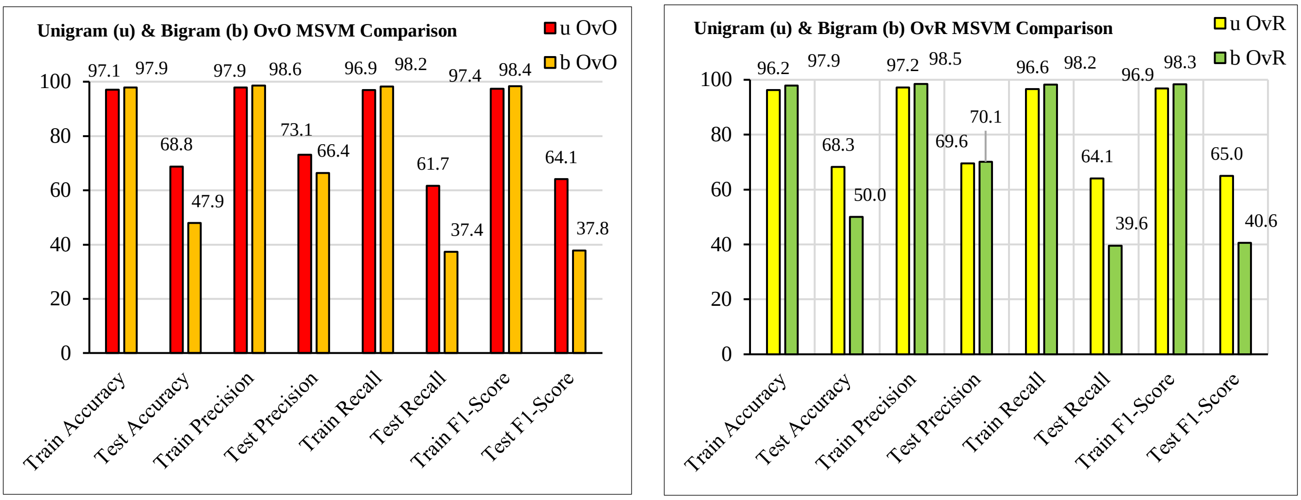
Based on the experiment, the result obtained was affected by the imbalance in the number of data on each class label, as mentioned previously and shown in Figure 4b. The “Netral” label (see the previous Introduction section) makes it worst and was unavoidable having a relatively large number of data since in reality, many suggestions/criticisms used general words, phrases, or sentences that do not match the other class labels, like “sejauh ini belum tau ingin bersaran apa” (so far don't know what to comment about), “sudah baik” (already good), “lebih ditingkatkan lagi” (improved more), or “agar lebih baik lagi” (to be even better). The process with feature selection (FS) was conducted on this kind of unbalanced dataset to know the improvement obtained compared to the proses without FS. For both processes, without and with FS, the accuracy was initially improved by the casefolding stage (see dataset preprocessing section) since this stage is important to avoid the developed model having multiple variations of the same words due to uppercase and lowercase variations, which could decrease to some extent of the *precision* score (the number of correct labels that were predicted by the model).

3.1. MSVM Models without Feature Selection (FS)

Models without FS, tested using *k*-fold cross-validation, provide performance scores of *accuracy*, *precision*, *recall*, and *F1*. Performance results of OvO MSVM models without FS using unigram and bigram features can be seen in Table 5 and the comparison chart using the average scores can be seen in Figure 5-a. Performance results of OvR MSVM models without FS using unigram and bigram features can be seen in Table 6 and the comparison chart using the average scores can be seen in Figure 5-b. The other comparison charts of OvO and OvR MSVM models without FS using unigram and bigram features can be seen in Figures 6-a and 6-b, respectively. Figure 7 shows the overall comparison of MSVM models without FS.

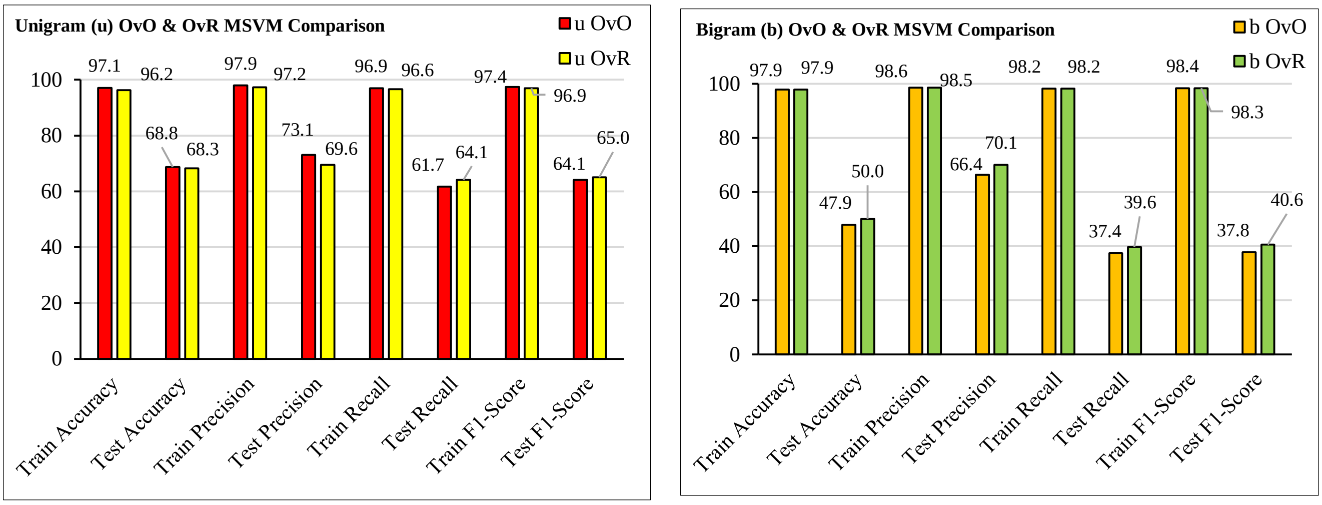
Table 4. Test results of unigram and bigram OvO MSVM models without FS

Model	*k*-fold	Train *Accuracy*	Test *Accuracy*	Train *Precision*	Test *Precision*	Train *Recall*	Test *Recall*	Train *F1*	Test *F1*
Unigram OvO MSVM without FS	1st	98.18	70.05	98.78	71.17	97.71	63.00	98.23	65.24
	2nd	97.21	78.16	98.07	82.95	96.83	70.68	97.42	73.78
	3rd	96.85	72.33	97.73	73.91	96.63	64.21	97.16	66.59
	4th	96.48	74.27	97.45	81.19	96.38	68.56	96.90	71.89
	5th	96.73	49.03	97.71	56.44	96.97	41.92	97.32	43.15
Average		**97.09**	**68.77**	**97.95**	**73.13**	**96.90**	**61.67**	**97.40**	**64.13**
Bigram OvO MSVM without FS	1st	97.94	47.34	98.57	53.12	98.32	34.47	98.41	32.43
	2nd	97.70	51.94	98.41	87.86	98.11	42.63	98.22	45.89
	3rd	97.94	51.94	98.57	69.29	98.25	41.09	98.38	43.38
	4th	98.18	53.88	98.74	87.10	98.44	43.70	98.55	46.56
	5th	97.82	34.47	98.50	34.67	98.09	25.07	98.24	20.68
Average		**97.91**	**47.92**	**98.56**	**66.41**	**98.24**	**37.39**	**98.36**	**37.79**



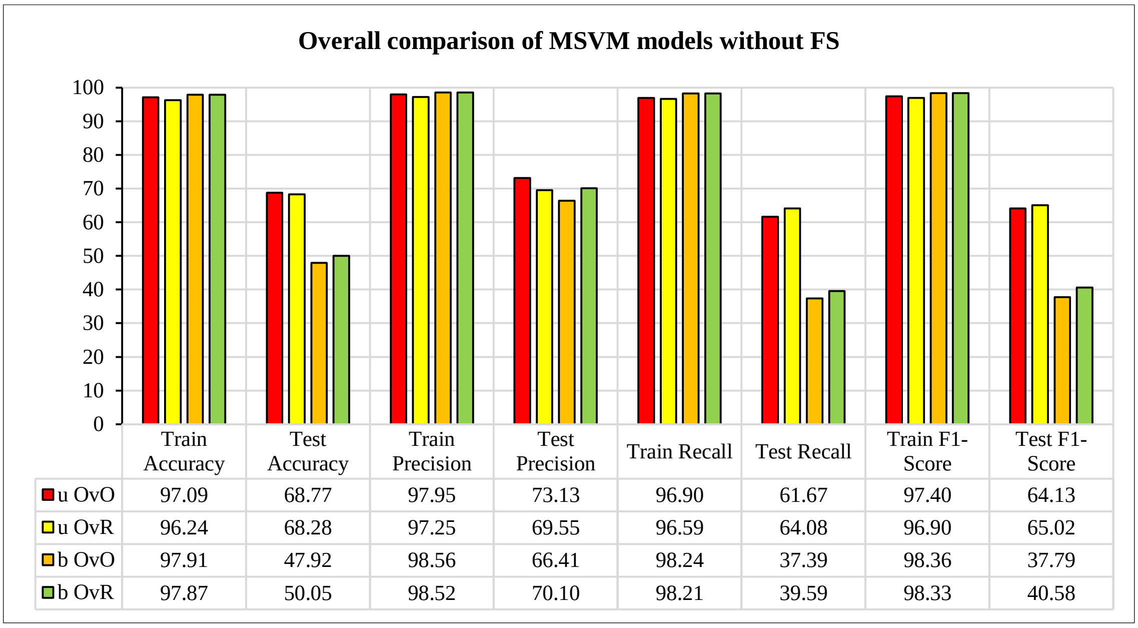
(a) (b)

Figure 3. Comparison without FS using unigram and bigram features on: (a) OvO MSVM; (b) OvR MSVM



(a) (b)

Figure 4. Comparison without FS of OvO and OvR MSVM models using features: (a) unigram; (b) bigram



	Train Accuracy	Test Accuracy	Train Precision	Test Precision	Train Recall	Test Recall	Train F1-Score	Test F1-Score
🟥 u OvO	97.09	68.77	97.95	73.13	96.90	61.67	97.40	64.13
🟨 u OvR	96.24	68.28	97.25	69.55	96.59	64.08	96.90	65.02
🟧 b OvO	97.91	47.92	98.56	66.41	98.24	37.39	98.36	37.79
🟩 b OvR	97.87	50.05	98.52	70.10	98.21	39.59	98.33	40.58

Figure 5. Overall comparison of MSVM models without FS

In all models. it was found that their training scores had already been above 90% but those performances cannot be matched by their testing scores. This was predicted due to overfitting [50, 51] caused by the use of features without prior FS so that many features become noises in the data. This FS needs to be conducted to increase the model's performance on the testing data [52, 53] (see next section).

In the model with the same approach but different *n*-gram, the unigram OvO MSVM model had a better performance in testing scores compared to its bigram OvO MSVM model, and the unigram OvR MSVM model had a better performance in general in testing scores compared to its bigram OvR MSVM model (except for *precision* which is quite comparable). Those results can be seen from the testing scores with a range of differences for the OvO MSVM model of 6.72% (*precision*) - 26.33% (*F1*) and for the OvR MSVM model of 0.55% (*precision*) - 24.49% (*recall*). The superiority of unigram features over bigram features is related to the condition that higher-order *n*-grams have a data sparsity problem that can make them less informative because so many are unseen, making the true data distribution harder to learn without more data (see the limitation of this study at the previous Introduction section). Smaller smoothing amounts give better performance than higher ones. This is because the lower ones let the model listen to the data *more*. The smoothing gives the counts that are representative of the actual data.

The *F1* score gives perspective related to the improvement of a simpler performance metric, namely *accuracy*. *Accuracy*, as the percentage of the number of correct predictions to the total number of predictions, is not a good metric to use when there are imbalanced classes (see Figure 4-b). This means that if there is a use case in which more observation is needed on data points of one class than of another, the *accuracy* is not so representative metric. One way to solve class imbalance problems is to work on samples (see next models with FS section). With specific sampling methods, resampling the dataset can be done in such a way that the data is no longer imbalanced. Accuracy as a metric then can be used again. Another way to solve class imbalance problems is to use better accuracy metrics like the *F1* score, which considers not only the number of prediction errors that the models make but also look at the type of errors that are made. *Precision* (the percentage of true positive predictions to the total number of positive predictions) and *recall* (the percentage of true positive predictions to the total number of true positive and false negative predictions) are the two most common metrics that consider class imbalance. They are also the foundation of the *F1* score, which is the harmonic mean of *precision* and *recall*. The harmonic mean is an alternative metric for the more common arithmetic mean. It is often useful when computing an average rate.

Based on those metrics perspectives and the results of Tables 5 and 6, a more confident performance difference related to the testing scores of the unigram to its bigram OvO MSVM model without FS was represented by their *F1* difference (26.33%), which is the largest difference among all the testing scores in this comparison (*accuracy* of 20.85% is the third largest difference after *recall* of 24.28%). A more confident performance difference related to the testing scores of the unigram to its bigram OvR MSVM model without FS was represented by their *F1* difference (24.44%), which is the comparable difference to *recall* of 24.49% in this comparison (*accuracy* of 18.23% is the third largest difference after them). Also note that since the *F1* score computes the average of *precision* and *recall*, models in these comparisons have medium *F1* scores because their *precision* and *recall* are low and the other is high.

Table 5. Test results of unigram and bigram OvR MSVM models without FS

Model	***k*-fold**	**Train *Accuracy***	**Test *Accuracy***	**Train *Precision***	**Test *Precision***	**Train *Recall***	**Test *Recall***	**Train *F1***	**Test *F1***
Unigram OvR MSVM without FS	1st	98.91	71.98	99.30	68.00	98.43	65.03	98.85	66.06
	2nd	94.55	76.21	95.94	72.96	95.34	72.82	95.61	72.16
	3rd	96.48	69.42	97.45	70.08	96.59	62.96	96.99	64.75
	4th	95.76	72.82	96.85	80.01	96.11	71.66	96.47	73.71
	5th	95.52	50.97	96.71	56.70	96.49	47.94	96.56	48.44
Average		**96.24**	**68.28**	**97.25**	**69.55**	**96.59**	**64.08**	**96.90**	**65.02**
Bigram OvR MSVM without FS	1st	97.82	48.79	98.49	52.80	98.24	35.46	98.33	33.35
	2nd	97.58	54.85	98.33	87.65	98.03	45.00	98.14	48.38
	3rd	97.94	55.83	98.57	86.14	98.25	46.74	98.38	51.38
	4th	98.18	55.83	98.74	87.37	98.44	45.15	98.55	48.25
	5th	97.82	34.95	98.50	36.55	98.09	25.59	98.24	21.54
Average		**97.87**	**50.05**	**98.52**	**70.10**	**98.21**	**39.59**	**98.33**	**40.58**

The other comparison of OvO and OvR MSVM models using unigram and bigram features (Figure 6) provided comparable differences in terms of performance. The results of the testing scores gave a range of differences for OvO and OvR MSVM models using unigram features of 0.49% (*accuracy*), 3.58% (*precision*), 2.41% (*recall*), 0.89% (*F1*), and for OvO and OvR MSVM models using bigram features of 2.13% (*accuracy*), 3.69% (*precision*), 2.2% (*recall*), 2.79% (*F1*). Without FS and with unigram features, the OvO MSVM model was slightly better on *accuracy* and *precision*, while the OvR MSVM model was slightly better on *recall* and *F1* score. Without FS and with bigram features, the OvR MSVM model had slightly better performance than the OvO MSVM model.

Related to those results, both OvO and OvR approaches seem to have no significant effect on the accuracy performance. They tend to affect the time complexity performance. In OvO classification, for the *n* class instances dataset, the *n*(*n*-1)/2 binary classifier models have to be generated. Using this classification approach, the primary dataset must be split into one dataset for each class opposite to every other class. Each binary classifier predicts one class label. When the test data is inputted into the classifier, then the model with the majority counts is concluded as a result. In OvR classification, for the *n* class instances dataset, the *n* binary classifier models have to be generated. The number of class labels present in the dataset and the number of generated binary classifiers must be the same. To train these *n* classifiers, *n* training datasets need to be created. After training the model, when test data is inputted into the model, then that data is considered input for all generated classifiers. If there is any possibility that the test data belong to a particular class, then the classifier created for that class gives a positive response in the form of +1, and all other classifier models provide an adverse reaction in the way of -1. Similarly, binary classifier models predict the probability of correspondence with concerning classes. By analyzing the probability scores, the result was predicted as the class index having a maximum probability score. Related to working mechanisms for both approaches, it is challenging to deal with large datasets having many numbers class instances that eventually at a certain level would decrease time complexity performance.

Based on the previous discussion and overall comparison chart in Figure 7, it was found that with the same *n*-gram, OvO MSVM and OvR MSVM models had relatively the same performance. Whereas when compared with the different *n*-gram, the unigram OvO/OvR MSVM model had better performance than its bigram OvO/OvR MSVM model. This happened because the number of features in bigram (1902) was larger than the number of features in unigram (645). That difference in number affects the test results because there was no FS used before features were inputted into OvO/OvR MSVM models.

3.2. MSVM Models with Feature Selection (FS)

In MSVM models with unigram or bigram features previously, performances on training data were very good, but scores on the testing data were quite low. It was also mentioned earlier that this was due to the overfitting of models, so it was necessary to do FS on the TF-IDF results, which were then used as input for MSVM models. After using the Extratress Classifier on the dataset, performance results of OvO MSVM models with FS using unigram and bigram features can be seen in Table 7. The average score comparison chart of OvO MSVM models without and with FS using unigram and bigram features can be seen in Table 8. Performance results of OvR MSVM models with FS using unigram or bigram features can be seen in Table 8. The average score comparison chart of OvR MSVM models without and with FS using unigram and bigram features can be seen in Figure 8.

Table 6. Test results of unigram and bigram OvO MSVM models with FS

Model	*k*-fold	Train *Accuracy*	Test *Accuracy*	Train *Precision*	Test *Precision*	Train *Recall*	Test *Recall*	Train *F1*	Test *F1*
Unigram OvO MSVM with FS	1st	84.34	72.95	87.00	67.82	81.67	65.60	83.34	66.35
	2nd	83.15	79.13	86.15	82.22	78.98	72.71	81.45	75.57
	3rd	82.91	75.24	86.07	72.70	79.51	65.97	81.70	67.69
	4th	83.52	75.73	86.53	79.45	79.78	72.67	82.19	74.03
	5th	86.18	54.37	87.63	54.28	83.47	49.41	84.81	48.10
Average		**84.26**	**71.77**	**87.52**	**71.25**	**81.11**	**66.28**	**83.26**	**67.01**
Bigram OvO MSVM with FS	1st	81.92	59.42	91.83	75.46	78.16	51.19	82.24	54.04
	2nd	81.82	63.11	91.86	83.28	77.85	59.24	82.05	63.83
	3rd	82.67	58.25	91.85	75.00	79.88	49.90	83.40	52.91
	4th	82.18	60.68	91.55	81.72	78.67	51.05	82.58	54.59
	5th	84.36	40.29	92.51	44.55	81.32	31.35	84.84	29.70
Average		**82.59**	**56.35**	**91.92**	**72.00**	**79.17**	**48.54**	**83.02**	**51.02**

Table 7. Test results of unigram and bigram OvR MSVM models with FS

Model	*k*-fold	Train *Accuracy*	Test *Accuracy*	Train *Precision*	Test *Precision*	Train *Recall*	Test *Recall*	Train *F1*	Test *F1*
Unigram OvR MSVM with FS	1st	88.96	78.26	88.62	79.05	87.36	76.75	87.63	77.76
	2nd	84.12	83.01	85.60	84.27	82.61	74.93	83.59	77.89
	3rd	85.09	70.87	86.95	66.50	83.06	64.49	84.44	64.58
	4th	85.21	75.24	86.06	76.44	83.49	71.81	84.39	73.28
	5th	87.64	57.77	88.77	56.39	87.28	54.83	87.73	53.08
Average		**86.20**	**73.03**	**87.20**	**72.53**	**84.76**	**68.56**	**85.56**	**69.32**
Bigram OvR MSVM with FS	1st	86.17	65.70	90.20	71.45	82.75	60.85	85.15	61.45
	2nd	85.21	66.50	89.73	83.25	81.69	62.15	84.28	66.40
	3rd	85.82	67.48	90.75	80.16	83.25	57.40	85.54	60.64
	4th	86.30	62.14	91.01	68.73	83.33	51.77	85.86	53.80
	5th	88.00	44.17	92.81	43.84	84.56	34.61	87.41	33.13
Average		**86.30**	**61.20**	**90.90**	**69.49**	**83.12**	**53.35**	**85.65**	**55.08**

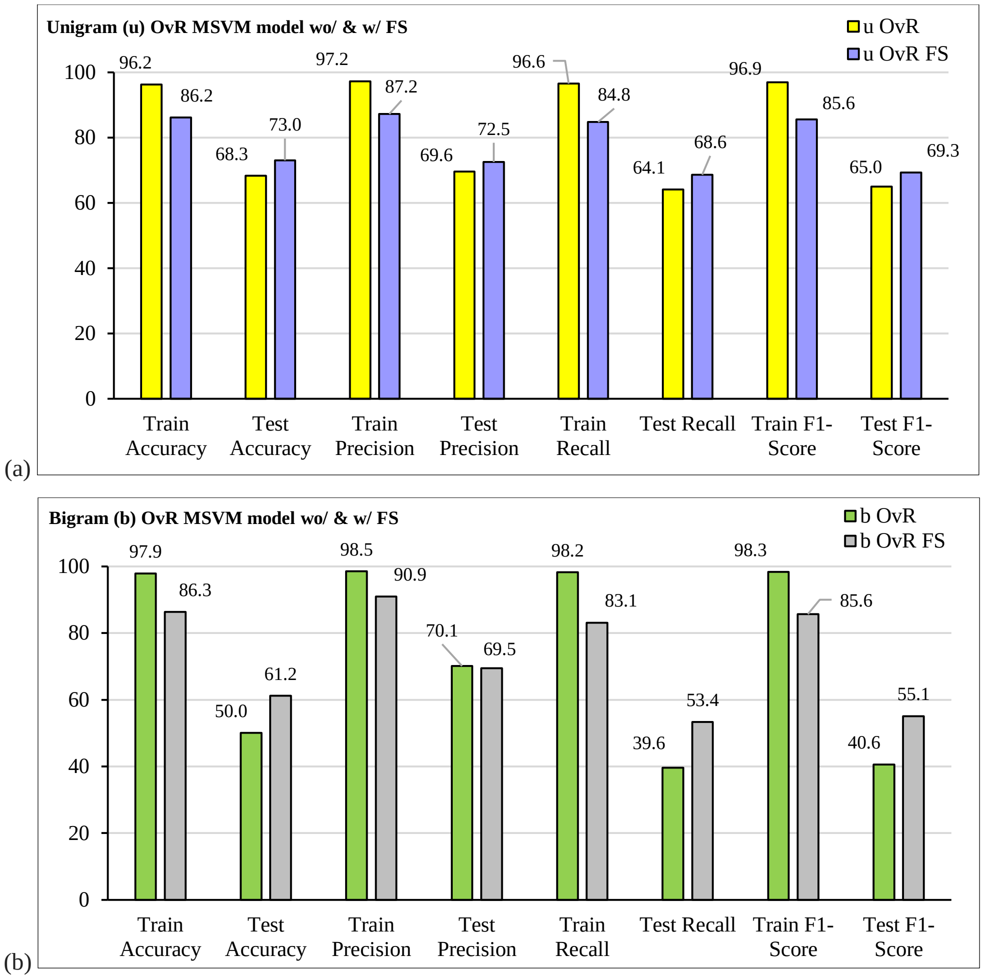
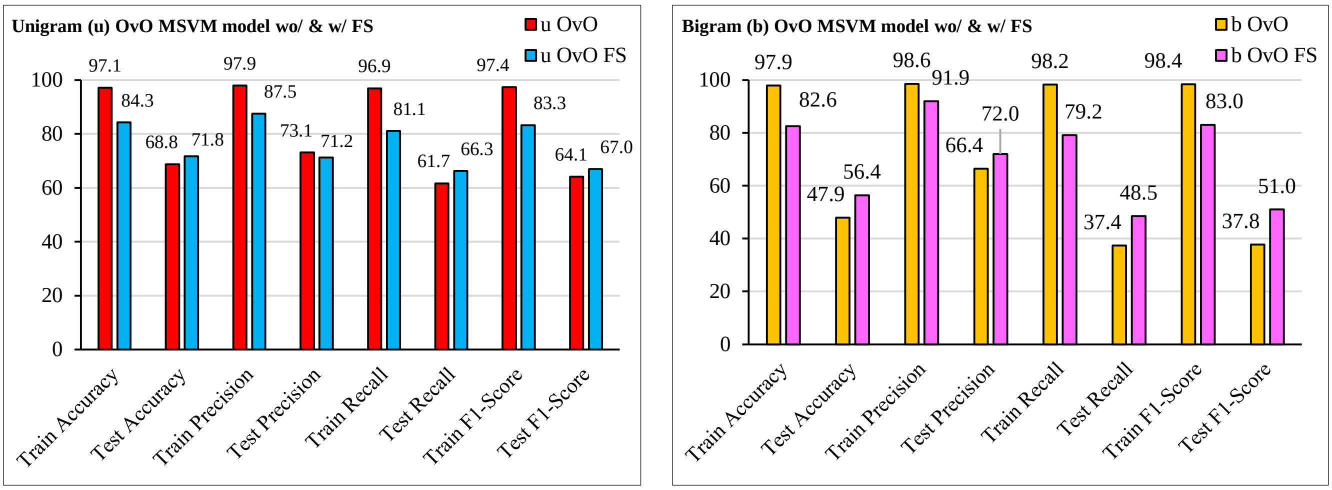


Figure 6. Comparison of OvR MSVM models without and with FS using features: (a) unigram; (b) bigram

From the comparison charts of OvO MSVM models without and with FS using unigram and bigram features in Figure 9, the reduction in feature size by FS results in decreasing training scores. For unigram OvO MSVM models without and with FS (Figure 9-a), decreasing differences were 12.83% (*accuracy*), 10.43% (*precision*), 15.79% (*recall*), and 14.13% (*F1*). For bigram OvO MSVM models without and with FS (Figure 9-b), decreasing differences were 15.32% (*accuracy*), 6.64% (*precision*), 19.07% (*recall*), and 15.34% (*F1*). On the other hand, an increase in testing scores occurred in general (except for *precision* which is quite comparable in the unigram OvO MSVM model). More significant increases occurred in the bigram OvO MSVM model with FS compared to the unigram OvO MSVM model with FS. For unigram OvO MSVM models without and with FS (Figure 9-a), increasing differences were 3% (*accuracy*), -1.88% (*precision*), 4.61% (*recall*), and 2.88% (*F1*). For bigram OvO MSVM models without and with FS (Figure 9-b), increasing differences were 8.43% (*accuracy*), 5.59% (*precision*), 11.15% (*recall*), and 13.23% (*F1*).



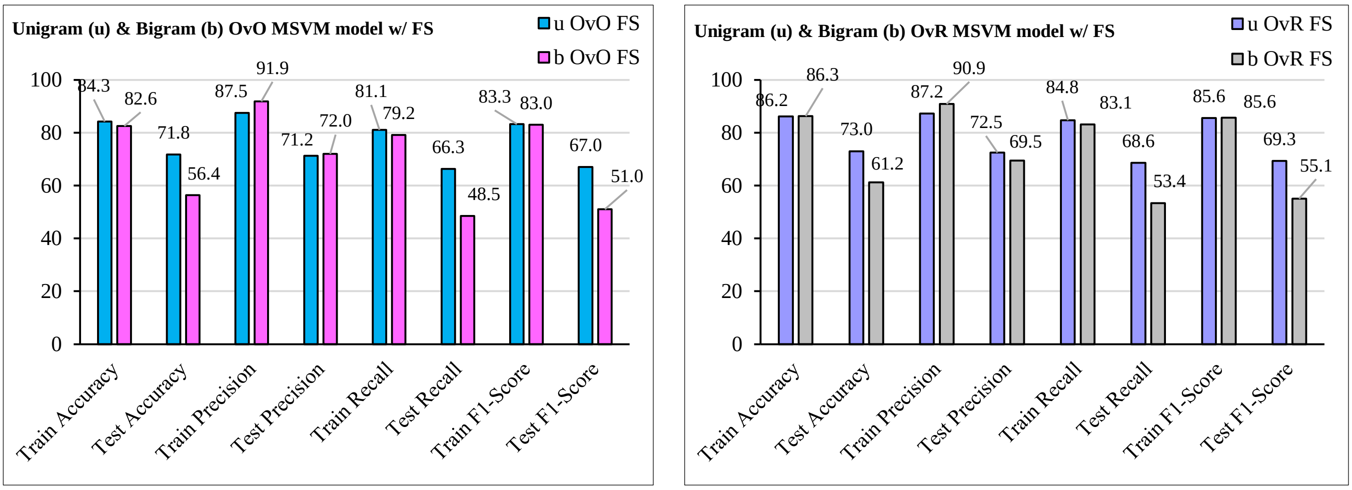
(a) (b)

Figure 7. Comparison of OvO MSVM models without and with FS using features: (a) unigram; (b) bigram

From the comparison charts of OvR MSVM models without and with FS using unigram and bigram features in Figure 8, it can be seen that scores on training data and testing data have the same pattern as previous OvO MSVM models. The difference is that *precision* decreased in testing scores (which is quite comparable) in the bigram OvR SVM model with FS (previously happened in the unigram OvO SVM model with FS). A significant increase also occurred in the bigram OvR SVM model with FS compared to the unigram OvR SVM model with FS. For unigram OvR MSVM models without and with FS (Figure 8-a), decreasing differences were 10.04% (*accuracy*), 10.05% (*precision*), 11.83% (*recall*), and 11.34% (*F1*). For bigram OvR MSVM models without and with FS (Figure 8-b), decreasing differences were 11.57% (*accuracy*), 7.62% (*precision*), 15.09% (*recall*), and 12.68% (*F1*). On the other hand, an increase in testing scores occurred in general (except for *precision* which is quite comparable in the bigram OvR MSVM model). More significant increases occurred in the bigram OvR MSVM model with FS compared to the unigram OvR MSVM model with FS. For unigram OvR MSVM models without and with FS (Figure 8-a), increasing differences were 4.75% (*accuracy*), 2.98% (*precision*), 4.48% (*recall*), and 4.3% (*F1*). For bigram OvR MSVM models without and with FS (Figure 8-b), increasing differences were 11.15% (*accuracy*), -0.61% (*precision*), 13.76% (*recall*), and 14.5% (*F1*).

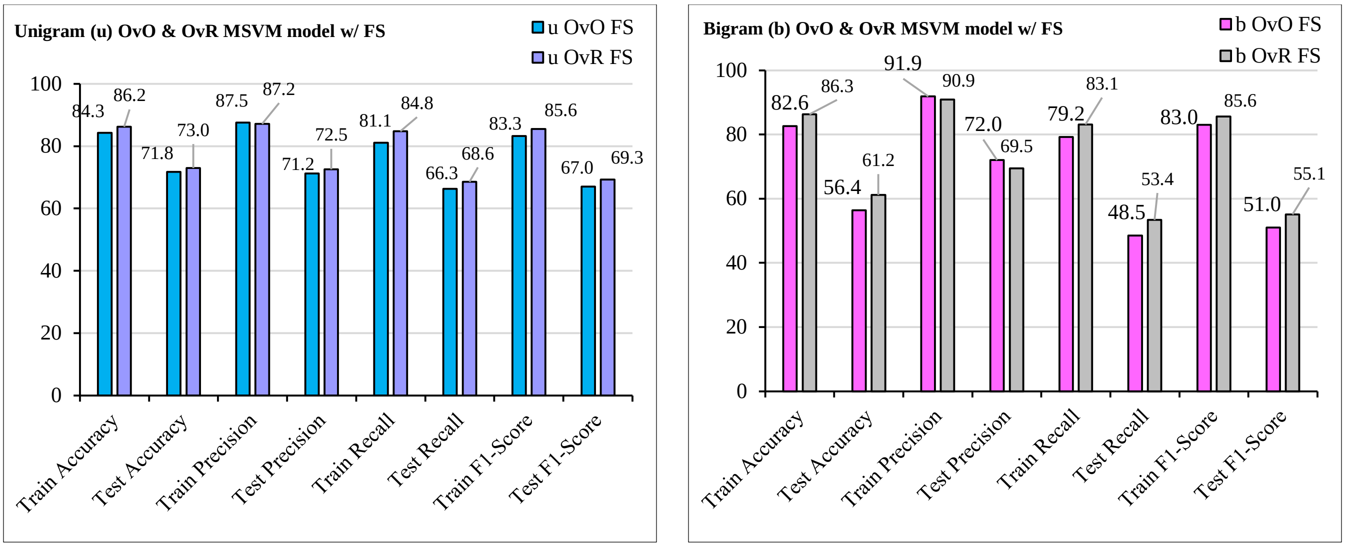
The entire MSVM model, after the FS process has been carried out, generally provides an increase in testing scores. On the other hand, there are decreasing scores of the training data. This is likely due to the imbalanced classes in the dataset used [54], thus affecting training results and also test results of MSVM models.

Based on Table 7 and Table 8, the next comparison charts among MSVM models with FS can be seen in Figures 10, 11 and 12. The average score comparison chart of OvO MSVM models with FS using unigram and bigram features can be seen in Figure 10a while the average score comparison chart of OvR MSVM models with FS using unigram and bigram features can be seen in Figure 10b. The other comparison charts of OvO and OvR MSVM models with FS using unigram and bigram features can be seen in Figure 11-a and Figure 11-b, respectively. Figure 12 shows the overall comparison of MSVM models with FS.



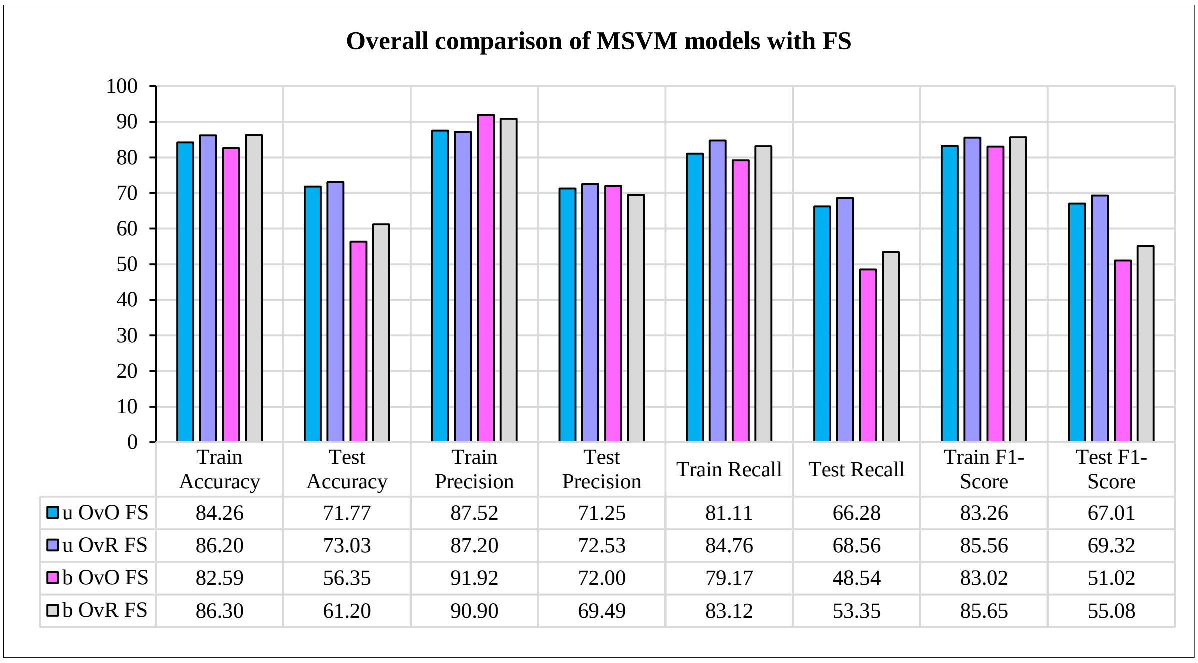
(a) (b)

Figure 8. Comparison with FS on using unigram and bigram on model: (a) OvO MSVM; (b) OvR MSVM



(a) (b)

Figure 9. Comparison with FS of OvO and OvR MSVM models using features: (a) unigram; (b) bigram



	Train Accuracy	Test Accuracy	Train Precision	Test Precision	Train Recall	Test Recall	Train F1-Score	Test F1-Score
u OvO FS	84.26	71.77	87.52	71.25	81.11	66.28	83.26	67.01
u OvR FS	86.20	73.03	87.20	72.53	84.76	68.56	85.56	69.32
b OvO FS	82.59	56.35	91.92	72.00	79.17	48.54	83.02	51.02
b OvR FS	86.30	61.20	90.90	69.49	83.12	53.35	85.65	55.08

Figure 10. Overall comparison of MSVM models with FS

Unlike all models without FS, all models with FS cannot maintain their training scores above 90% (except for bigram OvO and OvR, where their precision was at 91.92% and 90.90%, respectively), but those decreasing performances were compensated by their increasing testing scores in general. In the model with the same approach but different *n*-gram, the unigram OvO MSVM model had a better performance in general in testing scores compared to its bigram OvO MSVM model (except for *precision,* which is quite comparable), and the unigram OvR MSVM model had a better performance in testing scores compared to its bigram OvR MSVM model. The other comparison of OvO and OvR MSVM models using unigram and bigram features (Figure 11) provided comparable differences in terms of performance. The results of the testing scores gave a range of differences for OvO and OvR MSVM models using unigram features of 1.26% (*accuracy*), 1.28% (*precision*), 2.28% (*recall*), 2.31% (*F1*), and for OvO and OvR MSVM models using bigram features of 4.85% (*accuracy*), 2.51% (*precision*), 4.81% (*recall*), 4.06% (*F1*). With FS and unigram features, the OvR MSVM model had slightly better performance than the OvO MSVM model. Without FS and with bigram features, the OvR MSVM model was slightly better on *accuracy*, *recall*, and *F1* score, while the OvO MSVM model was slightly better on *precision.* Based on the overall comparison chart, it was found that with the same *n-gram, the* OvO MSVM and OvR MSVM models had relatively the same performance. Whereas, when compared with the different *n*-gram, the unigram OvO/OvR MSVM model had better performance than its bigram OvO/OvR MSVM model.

4. Conclusion

Classification comparisons of multi-class Support Vector Machine (MSVM) models without and with feature selection (FS) have already been conducted comprehensively on the text dataset constructed from service satisfaction survey data of Denpasar public health centers. Since all public health centers in Indonesia refer to the same national regulation, health service satisfaction survey data should have relatively the same aspects and characteristics that, overall, make it unique and need specific processing. This study lays the foundation for handling the classification of this kind of data that reflects the Indonesian health service satisfaction survey data. It is considered a contribution since there is no such study.

The foundation of the classification process laid by this study to fit in the Indonesian health service satisfaction survey data would be useful as part of the future vision for an integrated system of Indonesian health big data. Future implementation based on this study is related to the automatic incremental classification system that shows the comparison performance for several models in real-time to provide some good insights for a future decision regarding health services improvement. Automatic means that there is no longer a manual process (see Figure 1) to be carried out, as in this study. Any data transformation needed at a certain stage is provided automatically by the developed information system. Related to the automatic process, any supervision mechanism should be developed for unclear/ambiguous results for the continuous improvement of the system. Incremental means that the comparison performance is calculated immediately only for data that has already been validated as ground truth by the expert (see Table 2), including additional validation of positive, negative, and neutral sentiments (involving a Bahasa expert). Comparison metrics (*accuracy*, *precision*, *recall*, and *F1 score*) are calculated based on those ground truths. Related to the incremental process, any notification mechanism should be developed on the system so the experts can be notified immediately in real-time when suggestions/criticisms data is submitted through the feedback form of the information system (no longer through the paper-based form).

5. Declarations

5.1. Author Contributions

Conceptualization, G.I., H.S., and A.G.; methodology, G.I. and H.S.; software, G.I. and H.S.; validation, A.G.; formal analysis, G.I. and H.S.; investigation, G.I. and H.S.; resources, G.I., H.S., and A.G.; data curation, G.I. and H.S.; writing—original draft preparation, G.I. and H.S.; writing—review and editing, A.G.; visualization, G.I. and H.S.; supervision, A.G.; project administration, G.I.; funding acquisition, G.I. and A.G. All authors have read and agreed to the published version of the manuscript.

5.2. Data Availability Statement

The data presented in this study are available in the article.

5.3. Funding and Acknowledgements

The authors gratefully thank the Center for Public Health Innovation (CPHI) for providing the Denpasar public health centers service satisfaction survey report for this study. The authors also gratefully acknowledge the support of the Indonesian Ministry of Education, Culture, Research, and Technology for research funding of this work in the area of information technology for big data development.

5.4. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

6. References

[1] Indonesian Ministry of Education Culture Research and Technology. (2021). Research and Community Service Guide Book Edition XIII Revision. Jakarta, Indonesia. (In Indonesian).

[2] Center for Public Health Innovation (CPHI). (2021). Denpasar Health Centers Satisfaction Survey Report. Denpasar, Indonesia. (In Indonesian)

[3] Indonesian Ministry of Health. (2016). Regulation of the Minister of Health No. 39 of 2016 concerning Guidelines for Implementing the Healthy Indonesia Program with A Family Approach. Jakarta, Indonesia. (In Indonesian).

[4] Indonesian Ministry of Health. (2019). Regulation of the Minister of Health No. 30 of 2019 concerning Hospital Classification and Licensing. Jakarta, Indonesia. (In Indonesian).

[5] Mishbahuddin, B. (2020). Improving Hospital Health Service Management. Tangga Ilmu, Yogyakarta, Indonesia. (In Indonesian).

[6] Sabilla, A. G. (2021). The Relationship between Quality of Health Services and Patient Satisfaction Levels Using BPJS at First Level Health Facilities. Medika Hutama, 3(1), 1654–1659. (In Indonesian).

[7] Zhang, L., & Liu, B. (2017). Sentiment Analysis and Opinion Mining. Encyclopaedia of Machine Learning and Data Mining, Springer, Boston, United States. doi:10.1007/978-1-4899-7687-1_907.

[8] Hsu, C. W., & Lin, C. J. (2002). A comparison of methods for multiclass support vector machines. IEEE Transactions on Neural Networks, 13(2), 415–425. doi:10.1109/72.991427.

[9] Lei, H., & Govindaraju, V. (2005). Half-Against-Half Multi-class Support Vector Machines. In: Oza, N.C., Polikar, R., Kittler, J., Roli, F. (eds) Multiple Classifier Systems, MCS 2005. Lecture Notes in Computer Science, 3541. Springer, Berlin, Germany. doi:10.1007/11494683_16.

[10] Hsu, B. M. (2020). Comparison of supervised classification models on textual data. Mathematics, 8(5), 851. doi:10.3390/MATH8050851.

[11] Polpinij, J., & Luaphol, B. (2021). Comparing of Multi-class Text Classification Methods for Automatic Ratings of Consumer Reviews. Multi-disciplinary Trends in Artificial Intelligence. MIWAI 2021. Lecture Notes in Computer Science, 12832, Springer, Cham, Switzerland. doi:10.1007/978-3-030-80253-0_15.

[12] Kharwar, A. R., & Thakor, D. V. (2021). An Ensemble Approach for Feature Selection and Classification in Intrusion Detection Using Extra-Tree Algorithm. International Journal of Information Security and Privacy, 16(1), 1–21. doi:10.4018/ijisp.2022010113.

[13] Khaire, U. M., & Dhanalakshmi, R. (2022). Stability of feature selection algorithm: A review. Journal of King Saud University - Computer and Information Sciences, 34(4), 1060–1073. doi:10.1016/j.jksuci.2019.06.012.

[14] Wang, H., He, J., Zhang, X., & Liu, S. (2020). A Short Text Classification Method Based on N - Gram and CNN. Chinese Journal of Electronics, 29(2), 248 - 254. doi:10.1049/cje.2020.01.00.

[15] Perdana, A., Furqon, M. T., & Indriati, I. (2018). Application of the Support Vector Machine Algorithm in the Classification of Schizophrenia Mental Illness: A Study Case on RSJ. Radjiman Wediodiningrat, Lawang. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 2(9), 3162–3167. (In Indonesian).

[16] Widyawati, W. & Sutanto, S. (2019). Comparison of the Naïve Bayes Algorithm and the Support Vector Machine in Indonesian SMS Spam Classification. Jurnal Ilmiah Sains Dan Teknologi, 3(2), 178–19. (In Indonesian).

[17] Alita, D., Fernando, Y., & Sulistiani, H. (2020). Implementation of Multiclass SVM Algorithm on Indonesian Language Public Opinion on Twitter. Jurnal Tekno Kompak, 14(2), 86. doi:10.33365/jtk.v14i2.792. (In Indonesian).

[18] Pangestu, D. A. (2020). Sentiment Analysis of Public Opinion on Mental Health During the Covid-19 Pandemic on Social Media Twitter Using the Naive Bayes Classifier and Support Vector Machine. Ph.D. Thesis, Universitas Islam Indonesia, Yogyakarta, Indonesia. (In Indonesian).

[19] Hermanto, H., Mustopa, A., & Kuntoro, A. Y. (2020). Naive Bayes Classification and Support Vector Machine Algorithms in Student Complaint Services. Jurnal Ilmu Pengetahuan Dan Teknologi Komputer, 5(2), 211–220. doi:10.33480/jitk.v5i2.1181. (In Indonesian).

[20] Fitriana, D. N. & Sibaroni, Y. (2022). Tweet Data Classification Using the Multi-Class Support Vector Machine Classification Method: A Case Study of PT. KAI. e-Proceeding of Engineering, 7(2), 8493–8505. doi:10.34818/eoe.v7i2.12746. (In Indonesian).

[21] Dhammajoti, Young, J. C., & Rusli, A. (2020). A Comparison of Supervised Text Classification and Resampling Techniques for User Feedback in Bahasa Indonesia. 2020 Fifth International Conference on Informatics and Computing (ICIC), Gorontalo, Indonesia. doi:10.1109/icic50835.2020.9288588.

[22] Sujadi, H. (2022). Analysis of the Sentiment of Twitter Social Media Users towards the Covid-19 Outbreak with the Naïve Bayes Classifier and Support Vector Machine . INFOTECH Journal, 8(1), 22–27. doi:10.31949/infotech.v8i1.1883. (In Indonesian).

[23] Cikania, R. N. (2021). Implementation of the Naïve Bayes Classifier Algorithm and Support Vector Machine in the HALODOC Telemedicine Service Review Sentiment Classification. Jambura Journal of Probability and Statistics, 2(2), 96–104. doi:10.34312/jjps.v2i2.11364. (In Indonesian).

[24] Sohrabi, M. K., & Hemmatian, F. (2019). An efficient preprocessing method for supervised sentiment analysis by converting sentences to numerical vectors: a twitter case study. Multimedia Tools and Applications, 78(17), 24863–24882. doi:10.1007/s11042-019-7586-4.

[25] Malloy, B. A., & Power, J. F. (2019). An empirical analysis of the transition from Python 2 to Python 3. Empirical Software Engineering, 24(2), 751–778. doi:10.1007/s10664-018-9637-2.

[26] Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). A brief survey of text mining: Classification, clustering and extraction techniques. arXiv preprint arXiv:1707.02919. doi:10.48550/arXiv.1707.02919.

[27] Wang, M., & Hu, F. (2021). The application of nltk library for python natural language processing in corpus research. Theory and Practice in Language Studies, 11(9), 1041–1049. doi:10.17507/tpls.1109.09.

[28] Prakoso, R. (2017). Indonesian standard words for sentiment analysis. Available online: https://github.com/ramaprakoso/analisis-sentimen (accessed on November 2022).

[29] Indonesian Ministry of Health (2022). Dictionary of terms and definitions related to Health. Indonesian Ministry of Health, Jakarta, Indonesia. Available online: https://www.kemkes.go.id/folder/view/full-content/structure-kamus.html (accessed on November 2022). (In Indonesian).

[30] Tala, F. (2003). A study of stemming effects on information retrieval in Bahasa Indonesia. Master Thesis, Universiteit van Amsterdam, Amsterdam, Netherlands.

[31] Robbani, H. A. (2018). PySastrawi: Indonesian stemmer. Python port of PHP Sastrawi project. Available online: https://github.com/har07/PySastrawi (accessed on November 2022).

[32] Carpenter, J. (2022). Swifter: A package which efficiently applies any function to a panda's data frame or series in the fastest available manner. Available online: https://github.com/jmcarpenter2/swifter (accessed on November 2022).

[33] Howard, B. E., Phillips, J., Miller, K., Tandon, A., Mav, D., Shah, M. R., Holmgren, S., Pelch, K. E., Walker, V., Rooney, A. A., Macleod, M., Shah, R. R., & Thayer, K. (2016). SWIFT-Review: A text-mining workbench for systematic review. Systematic Reviews, 5(1), 87. doi:10.1186/s13643-016-0263-z.

[34] Leskovec, J., Rajaraman, A., & Ullman, J. D. (2020). Mining of Massive Datasets. Cambridge University Press Cambridge, United Kingdom. doi:10.1017/9781108684163.

[35] Jones, K. S. (2021). A Statistical Interpretation of Term Specificity and Its Application in Retrieval (1972). Ideas That Created the Future, 339–348, Cambridge, Massachusetts, United States. doi:10.7551/mitpress/12274.003.0037.

[36] Robertson, S. (2004). Understanding inverse document frequency: On theoretical arguments for IDF. Journal of Documentation, 60(5), 503–520. doi:10.1108/00220410410560582.

[37] Uther, W. (2010). Encyclopedia of Machine Learning. Springer, Boston, United States. doi:10.1007/978-0-387-30164-8.

[38] Wang, S., Tang, J., Liu, H. (2017). Feature Selection. Encyclopedia of Machine Learning and Data Mining. Springer, Boston, United States. doi:10.1007/978-1-4899-7687-1_101.

[39] Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature Selection. ACM Computing Surveys, 50(6), 1–45. doi:10.1145/3136625.

[40] Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. Machine Learning, 63(1), 3–42. doi:10.1007/s10994-006-6226-1.

[41] Sharaff, A., & Gupta, H. (2019). Extra-Tree Classifier with Metaheuristics Approach for Email Classification. Advances in Computer Communication and Computational Sciences. Advances in Intelligent Systems and Computing, 924, Springer, Singapore. doi:10.1007/978-981-13-6861-5_17.

[42] Ossai, C. I., & Wickramasinghe, N. (2022). GLCM and statistical features extraction technique with Extra-Tree Classifier in Macular Oedema risk diagnosis. Biomedical Signal Processing and Control, 73, 103471. doi:10.1016/j.bspc.2021.103471.

[43] Ampomah, E. K., Qin, Z., & Nyame, G. (2020). Evaluation of tree-based ensemble machine learning models in predicting stock price direction of movement. Information (Switzerland), 11(6), 332. doi:10.3390/info11060332.

[44] Podder, P., Khamparia, A., Mondal, M. R. H., Rahman, M. A., & Bharati, S. (2021). Forecasting the Spread of COVID-19 and ICU Requirements. International Journal of Online and Biomedical Engineering, 17(05), 81. doi:10.3991/ijoe.v17i05.20009.

[45] Arathi Krishna, V., Anusree, A., Jose, B., Anilkumar, K., & Lee, O. T. (2021). Phishing Detection using Extra Trees Classifier. IEEE, 5th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India. doi:10.1109/ISCON52037.2021.9702372.

[46] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12(85), 2825−2830.

[47] Chang, C. C., & Lin, C. J. (2011). LIBSVM: A Library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2(3), 1–27. doi:10.1145/1961189.1961199.

[48] Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R., & Lin, C. J. (2008). LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research, 9, 1871–1874.

[49] Bharadwaj, Prakash, K.B., & Kanagachidambaresan, G.R. (2021). Pattern Recognition and Machine Learning. Programming with TensorFlow. EAI/Springer Innovations in Communication and Computing, Springer, Cham, Switzerland. doi:10.1007/978-3-030-57077-4_11.

[50] Han, H., & Jiang, X. (2014). Overcome Support Vector Machine Diagnosis Overfitting. Cancer Informatics, 13(S1), CIN.S13875. doi:10.4137/cin.s13875.

[51] Salam, M. A., Taher, A., Samy, M., & Mohamed, K. (2021). The Effect of Different Dimensionality Reduction Techniques on Machine Learning Overfitting Problem. International Journal of Advanced Computer Science and Applications, 12(4), 641-655. doi:10.14569/ijacsa.2021.0120480.

[52] Tao, Z., Huiling, L., Wenwen, W., & Xia, Y. (2019). GA-SVM based feature selection and parameter optimization in hospitalization expense modeling. Applied Soft Computing Journal, 75, 323–332. doi:10.1016/j.asoc.2018.11.001.

[53] Tatwani, S., & Kumar, E. (2019). A stable SVM-RFE feature selection method for gene expression data. International Journal of Engineering and Advanced Technology, 8(6), 2110–2115. doi:10.35940/ijeat.F8482.088619.

[54] Mazurowski, M. A., Habas, P. A., Zurada, J. M., Lo, J. Y., Baker, J. A., & Tourassi, G. D. (2008). Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. Neural Networks, 21(2–3), 427–436. doi:10.1016/j.neunet.2007.12.031.