ILKOM Jurnal Ilmiah Vol. 16, No. 3, December 2024, pp. 356-370
Accredited 2nd by KEMDIKTISAINTEK No. 10/C/C3/DT.05.00/2025; E-ISSN 2548-7779 | P-ISSN 2087-1716

The Development of Classification Algorithm Models on Spam SMS Using Feature Selection and SMOTE

Rachma Chrysanti [a,1,*]; Sony Hartono Wijaya [a,2]; Toto Haryanto [a,3]
[a] *IPB University, Jl. Raya Dramaga, Babakan, Dramaga, Bogor, Jawa Barat, 16680, Indonesia*
[1] *rachmachrysanti@apps.ipb.ac.id;* [2] *sony@apps.ipb.ac.id;* [3] *totoharyanto@apps.ipb.ac.id*
** Corresponding author*

Article history: Received April 24, 2024; Revised June 25, 2024; Accepted December 22, 2024; Available online December 31 2024

Abstract

Short Message Service (SMS) is a widely used communication media. Unfortunately, the increasing usage of SMS has resulted in the emergence of SMS spam, which often disturbs the comfort of cellphone users. Developing a classification model as a solution for filtering SMS spam is very important to minimize disruption and loss to cellphone users due to SMS spam. To address this issue, utilize the Naïve Bayes algorithm and Support Vector Machine (SVM) along with Chi-square and Information Gain. This study focuses on the classification and analysis of SMS spam on a cellular operator service in a telecommunications company using machine learning techniques. This study applies and combines a combination of classification methods including Naive Bayes and Support Vector Machine (SVM). The combination is carried out with Chi-square and Information Gain feature selection to reduce irrelevant features. This study also applies a combination with data balancing techniques using the Synthetic Minority Oversampling Technique (SMOTE) to balance the number of unbalanced classes. The results show that SMOTE improves classification performance. SVM performs spam SMS classification or not spam Model 7 (SVM) achieves accuracy 98,55% and it has improved the performance when it was combined with SMOTE Model 10 (SVM + SMOTE) achieves F1-score 99,23% in performing spam SMS classification or not this outperforms all other models. These results indicate that the SVM algorithm achieved better performance in detecting spam SMS compared to Naive Bayes, which demonstrated a lower level of accuracy. These results illustrate the effectiveness of combining machine learning models to enhance classification accuracy with balanced data, emphasizing the model that exhibited the most substantial improvement in performance.

Keywords: Chi-square; Feature Selection; Naïve Bayes; SMOTE; Spam Detection.

Introduction

Currently, the form of digitalization in human life is familiar. Various applications have been developed with various purposes to meet the needs of the community. In the current era of rapidly developing technology, the use of smartphone applications is widespread and has become an inevitable need for the community. Mobile applications can be used and utilized by everyone to simplify and accelerate various business processes and activities including for payment purposes, government services, hospital services, and learning tools [1], [2], [3], [4], [5]. Along with the development of various applications, there is also a need for a telephone number that must be registered in the application to be used. With the user's phone number data recorded, it is easier for the application to send promotional messages and notifications. At present, the vulnerabilities in smartphones have increased in tandem with their growing adoption. In Android devices, these vulnerabilities arise from system flaws, unsafe internet connections, and insufficient user awareness [6]. Promotional messages are often considered spam SMS where the SMS appears as an unimportant SMS or a message that is not expected by the user [7]. In modern era, the financial industries and other related agencies regard SMS as an essential tool for communicating with their customers, however, this reliance also introduces a vulnerability that spammers can exploit, thereby jeopardizing customer security [8]. There are several reasons contributing to the popularity of spam messages. First, the large number of mobile phone users worldwide increases the potential targets for spam attacks. Second, the low cost of sending spam messages serves as an advantage for spam perpetrators. Finally, the spam classification capabilities of most mobile phones are relatively weak due to

http://dx.doi.org/10.33096/ilkom.v16i3.2220.356-370

limited computational resources, restricting their ability to accurately and efficiently identify spam messages [9]. To avoid user discomfort, it is necessary to carry out a detection of spam SMS on mobile devices with the best accuracy value.

Several algorithms have been conducted and obtained various performance values from the classification results, two of which are Naïve Bayes and Support Vector Machine (SVM). SVM are recognized for their resilience to noise, as they concentrate on the support vectors nearest to the decision boundary, thereby reducing the influence of outliers, while Naïve Bayes is fast and works well when the assumption of feature independence is valid [10], [11]. The advantages of the Naive Bayes algorithm include being easy to implement and able to perform computational processes quickly [12]. There are major reasons for using the SVM algorithm, these include its ability to handle large amounts of data, its small number of hyperparameters, its theoretical guarantees, and its excellent performance [13].

In classifying the messages data of SMS, it will certainly use a lot of words or features to be analyzed, this is also called high dimensional data [14]. High dimensional data can cause the curse of dimensionality problems. This phenomenon causes the possibility of wasted storage space, poor visualization capabilities, and overfitting [15]. More time to perform the computational process is also inevitable in this phenomenon [16]. Thus, it can be said that a method of classification that can provide efficient computation time is necessary. Feature selection is done to reduce computation time and remove insignificant or redundant factors or variables where it intends to be able to improve efficiency by reducing computation time and increasing the effectiveness of machine learning algorithm model performance [16]. The use of Chi-square feature selection in Naïve Bayes classification provides better accuracy than without using Chi-square feature selection [17], [18]. This also applies to Information Gain feature selection performed in classification using the SVM algorithm, with better accuracy than other feature selection methods such as Gain Ratio, ReliefF, and Correlation [19].

According to the previous research, this research aimed to develop classification methods by comparing the performance of Naïve Bayes and SVM algorithms. The study was conducted with Information Gain and Chi-square feature selection to further reduce the space complexity and improve model performance. During the preprocessing stage, a data imbalance was found. Research by Rupapara shows that using SMOTE can improve the accuracy of classifying malicious comments on social media. Before applying SMOTE, the accuracy rate was 93%, but after its application, it rose to 95% [24]. To tackle this issue, this study also utilized the SMOTE data balancing technique. By combining the classification algorithm and feature selection with SMOTE, it aims to manage the class imbalance effectively, delivering optimal performance and reducing the risk of overfitting. Furthermore, with this research, it is hoped that it can provide new insights in creating the best model for detecting spam SMS on mobile devices.

Method

A. Research Stages

The research stages of the model used was divided into four types of method stages which can be seen in Figures 1-4. The stage without feature selection in **Figure 1**, began with data collection, followed by preprocessing in the form of data cleaning, then the processing stage, followed by the evaluation stage.

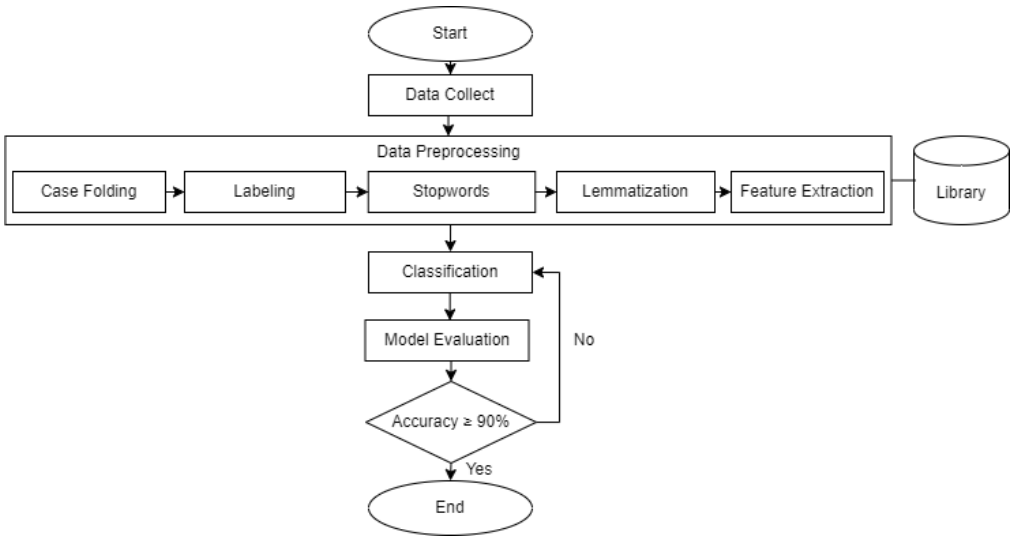

Figure 1. Flowchart of Research Stages without Feature Selection

The stage used in feature selection in **Figure 2** begins with data collection, then preprocessing in the form of data cleaning and feature selection, then the processing stage, followed by the evaluation stage.

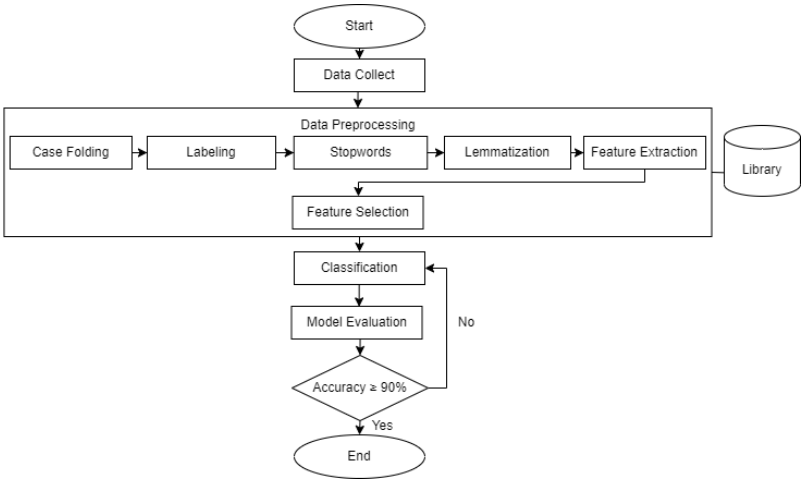

Figure 2. Flowchart of Research Stages Using Feature Selection

This research used the SMOTE oversampling technique to balance the unbalanced number of classes that affect the evaluation/performance score. **Figure 3** is the research stage of classification by applying the SMOTE oversampling technique. SMOTE was carried out after feature extraction and before the classification stage.

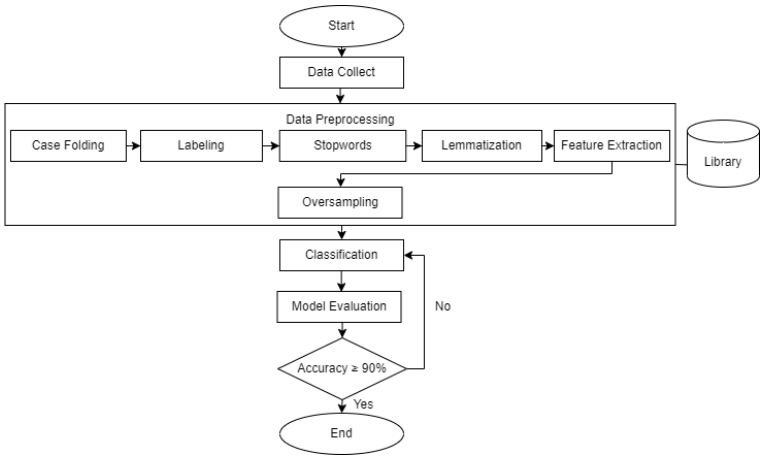

Figure 3. Flowchart of Research Stages Using SMOTE Oversampling

Figure 4 shows the stages of research when classification was carried out by applying the SMOTE oversampling technique which was carried out after feature extraction before the feature selection method, while feature selection was carried out before the classification process.

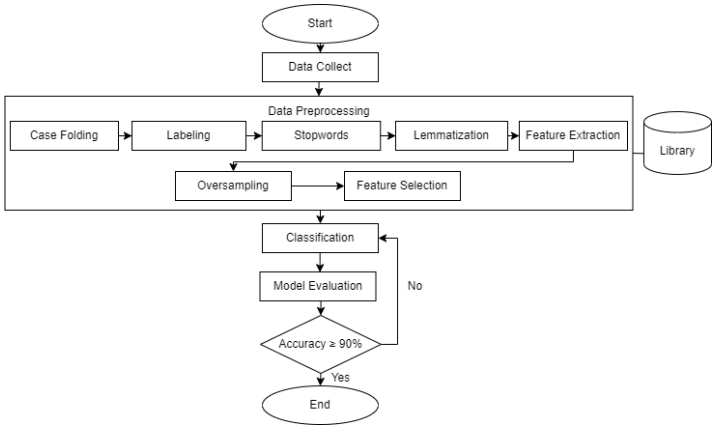

Figure 4. Flowchart of Research Stages Using SMOTE Oversampling and Feature Selection

B. Dataset Collection

The research data were from PT XYZ. Data were collected through Elasticsearch, Kibana, Beats, and Logstash (ELK Stack) analytic website applications. The data collected were in the form of SMS messages in May 2023 which consisted of date information, phone number, number of messages sent, and message content. **Figure 5** shows the stages in collecting research data. The data to be processed was text data on the contents of SMS messages which would be grouped into two classes including spam and no spam classes. The spam class contained SMS data containing promotional messages. The no spam class did not contain promotional messages, such as notifications of finance transactions, registration processes in an application, schedules for appointments with doctors, and so on.

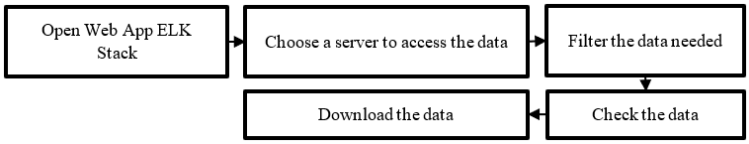

Figure 5. Data collection stages

A total of 10.000 datasets were collected. The data consists of 6122 features. The dataset consisted of two types: spam and no spam. Upon review of the collected data, it was observed that there is a significant imbalance in the composition, with 2053 entries classified as spam and 7947 as no spam.

C. Preprocessing

Data preprocessing aimed to clean and form numerical feature vectors and labels from the dataset so that it can be used as input data for machine learning models [20]. Data preprocessing can also improve performance by reducing data dimensions [17], [21]. Data preprocessing in this research consisted of data cleaning, text vectorization, data balancing, balancing technique, and feature selection.

1. Data Cleaning

At this stage, filtering using some techniques such as case folding, labeling, stopwords, and stemming is conducted to clean the SMS dataset. The case folding converted capital letters into lowercase letters and eliminated the use of numbers and symbols. Labeling of spam and nonspam messages was conducted on SMS data based on the number of words from the spam and nonspam dictionaries. Stopwords were used to remove words that have no significance. Stemming simplifies the affixed words into base words using a library [4]. The final stage of data preprocessing involved removing duplicate messages from the data set.

a. Case folding

At this stage, filtering is performed to clean the SMS dataset. Case folding is used to change capital letters to lowercase letters and remove the use of numbers and symbols. **Table 1** presents the results of the case folding stage in changing from capital letter to lowercase letter.

Table 1. Case folding changes capital letters to lowercase letters

Doc	Content
1	*yth%2c+transfer+idr+-4%2c007%2c500.00+tgl+25-05-2023+ref+ft23145x4fwr%2c+info+1500153*
2	*qasir+- +kode+otp+anda+%3a+1601.+jangan+memberitahu+kode+rahasia+ini+ke+siapapun+termasuk+pihak+qasir*
3	*rating+servis%0a3%0a1009494927550%0a2023-04-13+15%3a00%3a00.0000000*

In addition to changing letters to lowercase, case folding also performs the stages of removing symbols and numbers. **Table 2** is the result of case folding in removing symbols and numbers.

Table 2. Case folding removes symbols and numbers

Doc	Content
1	*yth c transfer idr c c tgl ref ft x fwr c info*
2	*qasir kode otp anda a jangan memberitahu kode rahasia ini ke siapapun termasuk pihak qasir*
3	*rating servis a a a a a*

Symbols such as plus signs, percent signs, and periods, as well as numbers are removed in this case folding stage.

2. Labeling

Labeling of spam and nonspam messages is done on SMS data based on the number of words from the spam and nonspam dictionary. Spam and nonspam word dictionaries are created manually to get the right labeling. The spam word dictionary consists of promotional words. The nonspam word dictionary consists of words that in their sentences contain the meaning of payment bills, electronic bank transactions, account registration, account modification, account security, health check-up appointment schedules, and customer satisfaction surveys. Words such as **Table 3** are examples of words included in the spam SMS dictionary and **Table 4** is an example of words included in the nonspam SMS dictionary.

Table 3. Spam words dictionary

No	Spam words	No	Spam words
1	*Dapatkan*	6	*Top up*
2	*Discount*	7	*Manfaatkan*
3	*Nikmati*	8	*Hanya*
4	*Pencairan*	9	*Potongan*
5	*Promo*	10	*Diskon*

Table 4. Nonspam words dictionary

No	Nonspam words (notifications)						
	Payment Bills	***e-Banking Transactions***	***Account Registration***	***Account Modification***	***Account Security***	***Appointment Scheds***	***Customers satisfaction surveys***
1	*Bayar*	*Transfer*	*Terdaftar*	*Ubah*	*Jangan*	*Tanggal*	*Partisipasi*
2	*Jatuh tempo*	*Setor*	*Terima-kasih*	*Password*	*Berikan*	*Jam*	*Survei*
3	*Segera*	*Rekening*	*Verifikasi*	*Username*	*PIN*	*Jadwal*	*Survey*
4	*Tagihan*	*Tunai*	*OTP*	-	*Membagi-kan*	*Praktek*	*Isi*
5	*Tepat waktu*	*Pembayaran*	*Membuat*	-	*Siapapun*	*Reser-vasi*	*Aspirasi*

In the no spam dictionary, all categories such as payment bills, electronic bank transactions, account registration, account modification, account security, health check-up appointment schedules, and customer satisfaction surveys are combined into one meaning as the word no spam. In the labeling stage, this is done by giving weight to the words contained in each document. For each word, the resulting weight is as follows:

a. For the word spam, the weight given is 1.

b. For the word no spam, the weight given is -1.

c. For words that are not included in both, the weight given is 0.

After being calculated, it is accumulated and the total is stored in the "count" column. If the calculation result is less than or equal to 0, it is labeled as non_spam, conversely if it is more than 0, it is labeled as SMS_spam. **Table 5** below shows the results obtained from the labeling stage using the SMS spam dictionary and the SMS nonspam dictionary.

Table 5. Labeling

Doc	Content	Count	Label
1	*yth c transfer idr c c tgl ref ft x fwr c info*	-1	non_spam
2	*qasir kode otp anda a jangan memberitahu kode rahasia ini ke siapapun termasuk pihak qasir*	-3	non_spam
3	*rating servis a a a a a*	-1	non_spam

After labeling, label encoding is also carried out which aims to change the label into a numeric code. **Table 6** is the result of label encoding.

Table 6. Label encoded

Doc	Content	Label	Label encoded
1	*yth c transfer idr c c tgl ref ft x fwr c info*	non_spam	0

Doc	Content	Label	Label encoded
2	*qasir kode otp anda a jangan memberitahu kode rahasia ini ke siapapun termasuk pihak qasir*	non_spam	0
3	*rating servis a a a a a*	non_spam	0

It can be defined that the spam SMS label code is marked with the number 1, while the nonspam SMS label code is marked with the number 0.

3. Stop words

Stop words are used to remove words that have no important meaning. The stop word stage uses a corpus taken from the Kaggle dataset collection website, the corpus was compiled by Oswin Rahadiyan Hartono at the website address: https://www.kaggle.com/datasets/oswinrh/indonesian-stoplist, in addition, words that need to be used as stop words in spam SMS but are not yet in the dictionary were also added. **Table 7** shows the documents that have undergone the stop words stage.

Table 7. Stopwords

Doc	Content	Stopwords
1	*yth c transfer idr c c tgl ref ft x fwr c info*	*yth transfer idr tgl ref info*
2	*qasir kode otp anda a jangan memberitahu kode rahasia ini ke siapapun termasuk pihak qasir*	*qasir kode otp anda jangan memberitahu kode rahasia ini ke siapapun termasuk pihak qasir*
3	*rating servis a a a a a*	*rating servis*

4. Stemming

Stemming simplifies affixed words into basic words using a library [4]. Stemming simplifies the basic word form of the word itself using the literary module available in Python programming. **Table 8** is an example of a document that applies the stemming process.

Table 8. Stemming

Doc	Content	Stemming
1	*yth transfer idr tgl ref info*	*yth transfer idr tgl ref info*
2	*qasir kode otp anda jangan memberitahu kode rahasia ini ke siapapun termasuk pihak qasir*	*qasir kode otp anda jangan memberitahu kode rahasia ini ke siapa masuk pihak qasir*
3	*rating servis*	*rating servis*

5. Text Vectorization

At this stage, it is done using the TF-IDF method. The limitation of the scheme binary term weighting supports the use of term frequency as the weight of a term for a particular text. The frequency of a word in a text refers to how often it appears within that text [22]. TF-IDF determines the frequency value of words in an SMS document which aims to determine how far the word is related to the document by giving each word a weight. TF-IDF weighting aims to transform text SMS messages into numeric form so that modeling can be done by assigning weights to each word [23]. The number of documents in the entire document corpus where a word appears is known as its document frequency. Before the TF-IDF process, it is necessary to remove duplicate data. Data removal was conducted using python via scripting on documents that contain similar SMS content. **Table 9** is an example of duplicate data in the dataset.

Table 9. Duplicate content

Doc	Content
1	*tukar poin hsbc dg tiket vip indonesia open dozen doughnuts union jam sore booth hsbc lg central park fgrp hsbc*
2	*tukar poin hsbc dg tiket vip indonesia open dozen doughnuts union jam sore booth hsbc lg central park fgrp hsbc*
3	*tukar poin hsbc dg tiket vip indonesia open dozen doughnuts union jam sore booth hsbc lg central park fgrp hsbc*
4	*tukar poin hsbc dg tiket vip indonesia open dozen doughnuts union jam sore booth hsbc lg central park fgrp hsbc*
5	*berikan waspada penipuan kode otp utk transaksi tokopedia tdk bertransaksi hub citiphone*
6	*berikan waspada penipuan kode otp utk transaksi tokopedia tdk bertransaksi hub citiphone*
7	*berikan waspada penipuan kode otp utk transaksi tokopedia tdk bertransaksi hub citiphone*
8	*berikan waspada penipuan kode otp utk transaksi tokopedia tdk bertransaksi hub citiphone*
9	*berikan waspada penipuan kode otp utk transaksi tokopedia tdk bertransaksi hub citiphone*

After being deleted, the number of documents became reduced. The original number of documents was 10.000, reduced to 3912 documents.

Then, after the duplicate data were removed, the stages in calculating the TF-IDF method were carried out. The stages of the TF-IDF method are illustrated as follows. For example, in calculating the weight of the term "info". It is known that the term "info" appears once in the first document, the number of terms in the first document is 6, the number of occurrences of the term "info" is 2295 in the document, and the total number of documents is 3912, then:

TF_{info} = Number of occurrences of the word "info" in the first document/ number of terms in the first document
= $1/6 = 0{,}16$
IDF_{info} = log (the total number of documents/term occurrences in the first document)
= $log\ (3912/2295)$
= $0{,}231$
TF-IDF = $TF \times IDF$
= $0{,}16 \times 0{,}231$
= $0{,}394$

So from the manual calculation using the formula, the TF-IDF value for the term “info” is $0{,}394$.

6. Data Balancing

The Synthetic Minority Oversampling Technique (SMOTE), is an oversampling method used in classification to balance the unbalanced data set of malicious comment datasets [24]. The SMOTE method finds k nearest neighbors for each data in the minority class, then creates synthetic data as much as the desired percentage of minor data duplication and random nearest neighbors [25]. In general, the formula for determining synthetic data is described in Equation 1 [26].

$$x_{syn} = x_i + \delta \times (y_j - x_i) \quad (1)$$

with:
x_{syn} = replicated synthetic data
xi = n numbers randomly selected from nearest neighbors
y_j = data that has the closest distance from the data to be replicated (j = 1,2, …, k)
δ = a random number between 0 and 1

If the random number is close to 0, the synthetic data will be the same as the original minority data. If it is close to 1, the synthetic data value will be equal to the nearest neighbor value [27]. The following **Table 10** contains concepts of applying sampling techniques using SMOTE.

Table 10. SMOTE Application Concept

Doc	Term 1	Term 2	Term 3	Term 4	Label
1	a	b	c	d	x
2	$a + (e - a) \times \delta$	$b + (f - b) \times \delta$	$c + (g - c) \times \delta$	$d + (h - d) \times \delta$	x
3	e	f	g	h	x

From the application concept, it can then be calculated as can be seen in **Table 11**, an example of the application of the sampling technique using SMOTE.

Table 11. SMOTE Application Example

Doc	*info*	*jangan*	*kode*	*terimakasih*	Label
1	0,71	0,68	0,87	0,77	1
2	0,80	0,69	0,79	0,83	1
3	0,89	0,7	0,71	0,88	1

So from the illustration above, new synthetic data has been obtained in the 2nd doc for minor data using the SMOTE oversampling technique which aims to ensure that the data has the same amount of data which can improve the performance of the classification model.

7. Feature Selection

The feature selection methods used included Chi-square and Information Gain. The feature selection method was carried out with two types of processes, the first without passing the SMOTE oversampling data balancing technique process. In the second type, before entering the feature selection method, data processing was conducted through the SMOTE oversampling data balancing technique process first to balance the number of classes reduce overfitting, and improve performance.

a. Chi-square

Chi-square is a feature selection method that performs an independence test to determine whether an independent and dependent variable is related [28]. Equation 2 is a formula for calculating the expected frequency (e_{ij}), where the marginal column frequency (o_i) is multiplied by the marginal row frequency (o_j) divided by the number of samples (N).

$$e_{ij} = \frac{o_i . o_j}{N} \tag{2}$$

After obtaining the expected frequency value, determining the Chi-square score can be performed with the following Equation 3 where the square value of the observation frequency (o_{ij}) minus the expected frequency (e_{ij}) is divided by the expected frequency (e_{ij}).

$$x^2 = \sum_{i=1}^{n} \frac{(o_{ij} - e_{ij})^2}{e_{ij}} \tag{3}$$

b. Information Gain

The Information Gain feature selection method performs feature ranking by calculating the entropy of one of the classes [29]. Equation 4 below is an approach to calculate the entropy value. With the entropy value, it can determine the gain value in Equation 5.

$$Entropy(S) = \sum_i^c -P_i \, log_2 \, P_i \tag{4}$$

$$Gain(S, A) = Entropy(S) - \sum_{values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{5}$$

With:

P_i	= accumulated sample of class i
c	= value of the classification class
$Gain(S, A)$	= gain value of the feature
A	= feature
v	= possible value of feature A
$Values(A)$	= possible value of set A
Sv	= number of sample values of v
S	= total number of data samples
Entropy(Sv)	= example entropy value of v

This metric works by evaluating the importance of a term in a document and the probability that the term belongs to a particular category [30]. The feature selection process requires a value of k which is the number of feature samples used.

D. Classification

Text classification can be described as a process that algorithmically analyzes a particular electronic document, to establish a set of categories that will describe the content of the document [31]. This research used classification methods with Naïve Bayes and SVM algorithms.

1. Naïve Bayes

The Naïve Bayes algorithm searches for the probability of an event that will occur by giving another probability that has occurred [32]. The calculation of the Naïve Bayes algorithm is formulated as in Equation 6. Where the probability value of A based on the conditions in hypothesis B is the multiplication of the probability of hypothesis B based on the condition of A (posterior probability) and the probability of A divided by the probability of hypothesis B (prior probability).

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)} \quad (6)$$

2. Support Vector Machine

The purpose of the Support Vector Machine (SVM) method was to find the separator with the best margin between classes. The separator is called a hyperplane. The data point that is closest to the hyperplane is called the support vector. The SVM method can separate data linearly or nonlinearly. If a case has data with two classes that are already linearly separated, then finding the hyperplane is expressed by Equation 7.

$$w.x + b = 0 \quad (7)$$

Where w is the weight, x is the input variable, and b is the bias value. x is the normal weight of the hyperplane and b is the distance from the hyperplane to point 0. Data that falls into class A and has a label of 1 meets Equation 8. While data that falls into class B and has a label of -1 meets Equation 9.

$$w.x + b \geq 1 \quad (8)$$

$$w.x + b \leq -1 \quad (9)$$

E. Evaluation

In the model evaluation stage, a confusion matrix evaluation method was used. Test results using a confusion matrix produced parameter values of accuracy, recall, precision, and F1-score. Accuracy in Equation 10 was used to measure the correctness ratio or the number of correct predictions divided by the amount of all data that goes through the prediction process [33]. Accuracy provides an overall value of the correct results of SMS spam and correct nonspam.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \quad (10)$$

Recall in Equation 11 shows how well the model recognizes the spam SMS class. Precision in Equation 12 is the accuracy of the model in predicting positive classes by minimizing mispredicted negative data [33]. Precision in this research would illustrate how well the model is conducted in predicting spam SMS classes by avoiding misclassification of nonspam SMS classes as spam SMS classes. Precision is a measure system of how many predictions are correct [33].

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

$$Precision = \frac{TP}{TP + FP} \quad (12)$$

F1-Score as in Equation 13 is defined as a comparison of the weighted average precision and recall values to measure the overall performance of the minority class [34]. F1-Score seeks a balance between precision and recall.

$$F1 - Score = \frac{2*TP}{TP+FP+TP+FN} \quad (13)$$

Besides the confusion matrix, computation time is also taken into account as an evaluation in knowing how fast the model performs the computation process.

Results and Discussion

A. Dataset Collection

The research data were SMS messages in May 2023 containing information on the date, phone number, number of messages sent, and message content taken from the Elasticsearch, Kibana, Beats, and Logstash (ELK Stack) analytic website application. Then data processing was carried out on 10.000 SMS messages that did not have a class. **Table 12** shows the data taken to conduct the research.

Table 12. Research Data

Date	Telephone Number	Messages	Total Message
25-05-2023	6282266xx	*Yth%2C+Transfer+IDR+-4%2C007%2C500.00+ tgl+25-05-2023+Ref+FT23145X4FWR%2C+info +1500153*	1
28-05-2023	62550750xx	*Qasir+-+Kode+OTP+Anda+%3A+1601+Jangan+ memberitahu+kode+rahasia+ini+ke+siapapun+ter*	1

Date	Telephone Number	Messages	Total Message
		masuk+pihak+Qasir	
26-05-2023	62811312xx	*RATING+SERVIS%0A3%0A1009494927550%0A20*	1

B. Data Preprocessing

Data preprocessing in this research consisted of data cleaning, text vectorization, data balancing using balancing technique, and feature selection.

1. Data Cleaning

The data needed to be processed through data cleaning stages first, including case folding, labeling, and lemmatization. The case folding stage included changing column headings, changing capital letters to lowercase letters, and removing numbers and symbols. The labeling stage provided labels based on the number of words from the spam SMS dictionary and no spam SMS. In addition, the labeling stage performed coding of message content including spam message labels marked with code 1 and no spam marked with code 0. The stemming stage simplified words to basic words based on the literary Sastrawi module library and performed word deletion using a corpus according to the needs of the dataset. As part of the cleaning process, the total number of SMS data was reduced to 3708. The preprocess results of case folding, labeling, and lemmatization can be seen in **Table 13**.

Table 13. Preprocessed Data Results

Dataset	Case Folding	Labeling			Stopwords	Stemming
		Count	***Label***	***Label Encoded***		
Yth%2C+Transfer+IDR+-4%2C007%2C500.00+tgl+25-05-2023+Ref+FT23145X4FWR%2C+info+1500153	*yth c transfer idr c c tgl ref ft x fwr c info*	-1	SMS_nonSpam	0	*yth c transfer idr c c tgl ref ft x fwr c info*	*yth c transfer idr c c tgl ref ft x fwr c info*
Qasir+-+Kode+OTP+Anda+%3A+1601.+Jangan+memberitahu+kode+rahasia+ini+ke+siapapun+termasuk+pihak+Qasir	*qasir kode otp anda a jangan memberitahu kode rahasia ini ke siapapun termasuk pihak qasir*	-3	SMS_nonSpam	0	*qasir kode otp anda jangan memberitahu kode rahasia ini ke siapapun termasuk pihak qasir*	*qasir kode otp anda jangan memberitahu kode rahasia ini ke siapa masuk pihak qasir*
RATING+SERVIS%0A3%0A1009494927550%0A2023-04-13+15%3A00%3A00.0000000	*rating servis*	-1	SMS_nonSpam	0	*rating servis*	*rating servis*

2. Text Vectorization

The vectorization stage is carried out using the TF-IDF method where TF (Term Frequency) is to calculate the number of occurrences of the word divided by the total number of words in the document. IDF (Inverse Document Frequency) is the total number of documents divided by the number of documents containing the word. The results of the TF-IDF vectorization stage are shown in Table 14.

Table 14. Text Vectorization Result by Token

info	***jangan***	***kode***	***terimakasih***	***transaksi***
1	0	0	0	0
0	1	2	0	0
0	0	0	0	0

3. Data Balancing

Data balancing is performed using the Synthetic Minority Oversampling Technique (SMOTE) which aims to balance the amount of data distribution by increasing the amount of minority class data. There were changes after data balancing, where originally the number of spam SMS was 204 and nonspam was 3708. After applying SMOTE oversampling, the number of each class changes to 3708 spam SMS and 3708 nonspam SMS.

Feature selection carried out includes using the Chi-square and Information Gain feature selection methods. The selected features used were 1000 out of 6122 total features, this is because this number is able to provide a fairly high level of performance compared to other numbers that have been tried. The best Chi-square feature selection accuracy results were obtained after applying the SMOTE and Chi-square oversampling techniques to the SVM algorithm of 99,14%. The best Information Gain feature selection accuracy results were obtained after applying the SMOTE and Information Gain oversampling techniques to the SVM algorithm of 99,05%. **Table 15** is some of the score results from the Chi-square and Information Gain feature selection based on tokens/features before applying the SMOTE oversampling technique.

Table 15. Chi-square and Information Gain Results Before Applying SMOTE

Token	Feature order	Chi-square		Information Gain	
		Rank	***Score***	***Rank***	***Score***
info	2460	536	2,44	7	0,0586
jangan	2569	596	1,51	513	0,0048
kode	2886	575	1,84	1444	0,0025
rating	4371	985	0,15	5267	0,0000
servis	4796	941	0,17	5471	0,0000
terimakasih	5348	116	11,34	31	0,0297
transaksi	5496	112	11,87	41	0,0241

Table 16 shows some of the scores result from the Chi-square and Information Gain feature selection after applying the SMOTE oversampling technique based on tokens/features.

Table 16. Chi-square and Information Gain Result After Applying SMOTE

Token	Feature order	Chi-square		Information Gain	
		Rank	***Score***	***Rank***	***Score***
info	2460	139	24,12	2	0,2881
jangan	2569	127	27,49	2947	0,0010
kode	2886	119	31,03	102	0,0227
rating	4371	857	2,79	3179	0,0004
servis	4796	796	3,09	5514	0,0000
terimakasih	5348	15	205,66	19	0,1749
transaksi	5496	12	213,55	10	0,2088

From the first feature rank to the 1000th feature, the number of feature scores increased between before and after applying the SMOTE oversampling technique. The Chi-square method experienced an increase in the total score from 9699,66 to 22.091,75. The Information Gain method experienced an increase in the total score from 10,11 to 26,03. This proves that the SMOTE oversampling technique affects the results of the Chi-square and Information Gain feature selection.

4. Feature Selection

The feature selection stage with different number of features gives different performance results for each method used. Feature selection is done by selecting several top number of features including 500 features, 1000 features, and 4000 features. **Table 17** is the result of the comparison between each number of feature selections.

Table 17. Chi-Square and Information Gain Results

Classification Algorithm	Feature Selection	Sampling	Accuration of 500 Features	Accuration of 1000 Features	Accuration of 4000 Features
Naïve Bayes	Chi-square	-	98,98%	99,23%	98,30%
		SMOTE	98,15%	98,65%	98,70%
	Information Gain	-	97,87%	97,79%	97,53%
		SMOTE	96,13%	98,65%	98,43%

Classification Algorithm	Feature Selection	Sampling	Accuration of 500 Features	Accuration of 1000 Features	Accuration of 4000 Features
SVM	Chi-square	-	98,21%	98,13%	98,64%
		SMOTE	99,55%	99,69%	99,68%
	Information Gain	-	98,21%	98,13%	98,21%
		SMOTE	99,37%	99,37%	99,64%

The scores of the Chi-square and Information Gain feature selection after applying the SMOTE oversampling technique based on tokens/features are shown in **Table 18**.

Table 18. Chi-square and Information Gain Results with SMOTE

Token	Feature order	Chi-square		Information Gain	
		Rank	*Score*	*Rank*	*Score*
info	2460	193	36,75	146	0,0140
jangan	2569	100	135,00	518	0,0086
kode	2886	86	178,59	83	0,0185
terimakasih	5348	12	1569,00	11	0,1733
transaksi	5496	1	1832,01	4	0,1933

After applying the SMOTE oversampling technique from the first feature rank to the 1000th feature, the total feature score was increased. The Chi-square method experienced an increase in the total score from 42.446,09 to 77.541,7. The Information Gain method increased the total score from 6,3 to 14,5. This proves that the SMOTE oversampling technique affected the Chi-square and Information Gain feature selection results.

C. Modeling Process

The modeling process stage consists of feature selection using the Chi-square and Information Gain methods, as well as classification using Naïve Bayes and SVM algorithms. In the modeling process, the distribution of the training and testing data used a ratio of 7:3.

Four models of method stages have been carried out to model the classification in this study. In the first model, classification without feature selection, SVM obtained the best accuracy rate of 98,89% while Naïve Bayes obtained the accuracy rate below it, which was 97,35%. In the second model, which was classification using feature selection, SVM Chi-square got the highest accuracy rate of 98,89%. In the third model, the best was SVM without feature selection with the SMOTE oversampling technique getting an accuracy rate of 99,23%. The fourth model, which got the best accuracy rate was SVM and Information Gain which got an accuracy rate of 99,81%. From the overall model stages of the classification method carried out, it could be said that various methods with the SVM algorithm have succeeded in getting higher accuracy rates than the Naïve Bayes algorithm.

D. Modeling Evaluation

In this stage, the evaluation results have been calculated using the confusion matrix method and the best computation time has been obtained from each model applied. The best accuracy rate was obtained from SVM SMOTE of 99,23%. The best precision was obtained from SVM SMOTE of 98,74%. The best Recall was obtained when performing SVM SMOTE integrated with Information Gain of 99,81%. The best F1-score was obtained from SVM SMOTE integrated with Chi-square by 99,14%.

The best computation time was obtained from the SVM algorithm integrated with Information Gain for 0,434 seconds. However, when viewed based on the best accuracy level of 99,23% obtained from the SVM SMOTE algorithm with a computational speed of 24,390 seconds. The SVM SMOTE algorithm became faster when integrated with the feature selection method.

The same thing also happened to the Naive Bayes SMOTE algorithm with the best accuracy level of 97,25% which spent 0,399 seconds of computing time. Based on the speed of computation time, the Naive Bayes algorithm became faster when integrated with the feature selection method. **Table 19** shows all the results of the evaluation level and computation time-based on the modeling method conducted in this study.

Table 19. Evaluation Results of Modeling Method Implementation

No.	Methods	Original Feature Number	Total Selected Features	Execution time per second	Percentage Value			
					Accuracy	*Precision*	*Recall*	*F1-score*
1	NB	-	-	0,330	97,35	72,72	94,73	82,28
2	NB + Chi2	6122	1000	0,070	97,18	70,09	98,68	81,96
3	NB + IG	6122	1000	0,067	96,42	66,66	89,47	76,40
4	NB + SMOTE	-	-	**0,399**	**97,25**	98,06	96,37	97,21
5	NB + SMOTE + Chi2	6122	1000	0,082	97,07	97,96	96,10	97,02
6	NB + SMOTE + IG	6122	1000	0,079	95,68	96,84	94,38	95,59
7	SVM	-	-	7,145	98,89	92,00	90,78	91,39
8	SVM + Chi2	6122	1000	0,851	98,89	93,15	89,47	91,27
9	SVM + IG	6122	1000	0,434	98,29	91,17	81,57	86,11
10	SVM + SMOTE	-	-	**24,390**	**99,23**	98,74	99,72	99,23
11	SVM + SMOTE + Chi2	6122	1000	0,290	99,14	98,65	99,63	99,14
12	SVM + SMOTE + IG	6122	1000	0,434	99,05	98,30	99,81	99,05

Conclusion

The application of classification algorithms has been carried out with feature selection and SMOTE oversampling data balancing techniques. Unbalanced data made this research necessary to apply data balancing techniques. The application of the SMOTE oversampling technique provided a balance in the spam and nonspam SMS classes that could help the model get a high level of performance. In the application of feature selection, from a total of 6122 features, the best number has been searched, and the best 1000 selected features were selected to get the best accuracy rate. The best classification process data division for training and testing was 7:3.

When the SMOTE oversampling technique was applied to SMS data, the performance score results improved. Before applying the oversampling technique and feature selection, the highest accuracy rate on the SVM algorithm was 98,23%. After applying the SMOTE oversampling technique and feature selection method, the highest accuracy rate was the Chi-square feature selection in the SVM classification algorithm with an accuracy rate of 99,14%. In the application of the Naïve Bayes algorithm with the SMOTE oversampling technique and feature selection, the highest accuracy rate was obtained at 97,07% which is conducted with the Chi-square feature selection. This shows that SVM had a better accuracy rate than Naïve Bayes. Furthermore, the Naive Bayes algorithm performs better in terms of computational time than the Support Vector Machine algorithm. Meanwhile, the SVM algorithm has demonstrated a superior level of accuracy and F1-score.

In terms of performance success rate from confusion matrix evaluation results, classification algorithms with data balancing techniques can be considered to provide the best success rate. However, in terms of the speed of computing time in performing classification, the application of feature selection can be considered as a method that can provide effective time speed compared to without using feature selection or using classification algorithms only.

References

[1] P. Medina Aguerrebere, E. Medina, and T. Gonzalez Pacanowski, "Promoting Health Education Through Mobile Apps: A Quantitative Analysis of American Hospitals," *Healthc.*, vol. 10, no. 11, pp. 1–14, 2022, doi: 10.3390/healthcare10112231.

[2] S. Kamarudin, L. Tang, J. Bolong, and N. A. Adzharuddin, "A Systematic Literature Review of Mitigating Cyber Security Risk," *Qual. Quant.*, vol. 58, no. 4, pp. 3251–3273, 2024, doi: 10.1007/s11135-023-01791-9.

[3] D. M. D. Oliveira, L. Pedro, and C. Santos, "The Use of Mobile Applications in Higher Education Classes: A Comparative Pilot Study of The Students' perceptions and real usage," *Smart Learn. Environ.*, vol. 8, no. 1, p. 14, 2021, doi: 10.1186/s40561-021-00159-6.

[4] L. K. Osei, Y. Cherkasova, and K. M. Oware, "Unlocking The Full Potential of Digital Transformation in

Banking: A Bibliometric Review and Emerging Trend," *Futur. Bus. J.*, vol. 9, no. 1, 2023, doi: 10.1186/s43093-023-00207-2.

[5] Y. Song, T. Natori, and X. Yu, "Tracing the Evolution of E-Government: A Visual Bibliometric Analysis from 2000 to 2023," *Adm. Sci.*, vol. 14, no. 7, 2024, doi: 10.3390/admsci14070133.

[6] A. Qamar, A. Karim, and V. Chang, "Mobile Malware Attacks: Review, Taxonomy & Future Directions," *Futur. Gener. Comput. Syst.*, vol. 97, pp. 887–909, 2019, doi: 10.1016/J.FUTURE.2019.03.007.

[7] S. Kumar and S. Gupta, "Legitimate and spam SMS classification employing novel Ensemble feature selection algorithm," *Multimed. Tools Appl.*, vol. 83, no. 7, pp. 19897–19927, 2024, doi: 10.1007/s11042-023-16327-4.

[8] E. G. Jain, "A Comparative Analyzing of SMS Spam Using Topic Models," P. K. Singh, Z. Polkowski, S. Tanwar, S. K. Pandey, G. Matei, and D. Pirvu, Eds., Cham: Springer International Publishing, 2021, pp. 91–99. doi: 10.1007/978-3-030-66218-9_10.

[9] N. Terli, P. Chintakayala, V. M. Angaluri, and S. Sodagudi, "Detection of Spam in SMS Using Machine Learning Algorithms BT - Smart Trends in Computing and Communications," T. Senjyu, C. So-In, and A. Joshi, Eds., Singapore: Springer Nature Singapore, 2023, pp. 417–427, doi: 10.1007/978-981-99-0838-7_37.

[10] K. S. Chong and N. Shah, "Comparison of Naive Bayes and SVM Classification in Grid-Search Hyperparameter Tuned and Non-Hyperparameter Tuned Healthcare Stock Market Sentiment Analysis," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 12, pp. 90–94, 2022, doi: 10.14569/IJACSA.2022.0131213.

[11] J. Manuel, A. Acosta, and R. G. Cabrera, "Analysis of the Influence of Preprocessing Techniques on Text Classification Accuracy : An Investigation with the Naive Bayes Model and the Reuters-21578 Dataset," vol. 9, no. 10, pp. 5220–5229, 2023.

[12] N. Kewsuwun and S. Kajornkasirat, "A Sentiment Analysis Model of Agritech Startup on Facebook Comments Using Naive Bayes Classifier," *Int. J. Electr. Comput. Eng.*, vol. 12, no. 3, pp. 2829–2838, 2022, doi: 10.11591/ijece.v12i3.pp2829-2838.

[13] J. Sangeetha and D. U. Kumaran, "Comparison of Sentiment Analysis on Online Product Reviews Using Optimised RNN-LSTM with Support Vector Machine," *Webology*, vol. 19, no. 1, pp. 3883–3898, 2022, doi: 10.14704/web/v19i1/web19256.

[14] A. Ghourabi and M. Alohaly, "Enhancing Spam Message Classification and Detection Using Transformer-Based Embedding and Ensemble Learning," 2023. doi: 10.3390/s23083861.

[15] A. K. Rastogi, S. Taterh, and B. S. Kumar, "Dimensionality Reduction Algorithms in Machine Learning: A Theoretical and Experimental Comparison," 2023. doi: 10.3390/engproc2023059082.

[16] Sutriawan, Muljono, Khairunnisa, Z. Alamin, T. A. Lorosae, and S. Ramadhan, "Improving Performance Sentiment Movie Review Classification Using Hybrid Feature TFIDF, N-Gram, Information Gain and Support Vector Machine," *Math. Model. Eng. Probl.*, vol. 11, no. 2, pp. 375–384, 2024, doi: 10.18280/mmep.110209.

[17] A. Sikri, N. P. Singh, and S. Dalal, "Analysis of Rank Aggregation Techniques for Rank Based on The Feature Selection Technique," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 11, no. 3s, pp. 95–108, Mar. 2023, doi: 10.17762/ijritcc.v11i3s.6160.

[18] A. Abraham, · Paramartha, D. Jyotsna, K. Mandal, A. Bhattacharya, and S. Dutta, "Emerging Technologies in Data Mining and Information Security", vol. 813. 2019. doi: 10.1007/978-981-13-1498-8.

[19] L. Gao, M. Ye, X. Lu, and D. Huang, "Hybrid Method Based on Information Gain and Support Vector Machine for Gene Selection in Cancer Classification," *Genomics. Proteomics Bioinformatics*, vol. 15, no. 6, pp. 389–395, Dec. 2017, doi: 10.1016/j.gpb.2017.08.002.

[20] M. Suhaidi, R. A. Kadir, and S. Tiun, "The Impact of Preprocessing Techniques Towards Word Embedding BT - Advances in Visual Informatics," H. Badioze Zaman, P. Robinson, A. F. Smeaton, R. L. De Oliveira, B. N. Jørgensen, T. K. Shih, R. Abdul Kadir, U. H. Mohamad, and M. N. Ahmad, Eds., Singapore: Springer Nature Singapore, 2024, pp. 421–429, doi: 10.1007/978-981-99-7339-2_35.

[21] S. Bahassine, A. Madani, M. Al-Sarem, and M. Kissi, "Feature selection Using an Improved Chi-Square for

Arabic Text Classification," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 32, no. 2, pp. 225–231, Feb. 2020, doi: 10.1016/j.jksuci.2018.05.010.

[22] M. Bordoloi and S. K. Biswas, "Sentiment Analysis: A Survey on Design Framework, Applications and Future Scopes, " vol. 56, no. 11. Springer Netherlands, 2023. doi: 10.1007/s10462-023-10442-2.

[23] S. Dhelim *et al.*, "Detecting Mental Distresses Using Social Behavior Analysis in the Context of COVID-19: A Survey," *ACM Comput. Surv.*, vol. 55, no. 14 S, 2023, doi: 10.1145/3589784.

[24] V. Rupapara, F. Rustam, H. F. Shahzad, A. Mehmood, I. Ashraf, and G. S. Choi, "Impact of SMOTE on Imbalanced Text Features for Toxic Comments Classification Using RVVC Model," *IEEE Access*, vol. 9, pp. 78621–78634, 2021, doi: 10.1109/ACCESS.2021.3083638.

[25] S. Mayabadi and H. Saadatfar, "Two Density-Based Sampling Approaches for Imbalanced and Overlapping Data," *Knowledge-Based Syst.*, vol. 241, p. 108217, 2022, doi: 10.1016/j.knosys.2022.108217.

[26] Y. Qu, Z., Li, H., Wang, Y., Zhang, J., Abu-Siada, A., & Yao, "Detection of Electricity Theft Behavior Based on Technique and Random Forest Classifier," *Energies*, vol. 13, no. 8, p. 2039, 2020, doi: 10.3390/en13082039.

[27] B. Santoso, H. Wijayanto, K. A. Notodiputro, and B. Sartono, "Synthetic Over Sampling Methods for Handling Class Imbalanced Problems : A Review," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 58, no. 2, p. 012031, Mar. 2017, doi: 10.1088/1755-1315/58/1/012031.

[28] F. Rustam, A. Mehmood, M. Ahmad, S. Ullah, D. M. Khan, and G. S. Choi, "Classification of Shopify App User Reviews Using Novel Multi Text Features," *IEEE Access*, vol. 8, pp. 30234–30244, 2020, doi: 10.1109/ACCESS.2020.2972632.

[29] S. Muñoz and C. A. Iglesias, "A Text Classification Approach to Detect Psychological Stress Combining a Lexicon-Based Feature Framework with Distributional Representations," *Inf. Process. Manag.*, vol. 59, no. 5, p. 103011, 2022, doi: 10.1016/j.ipm.2022.103011.

[30] R. Çekik and M. Kaya, "A New Feature Selection Metric Based on Rough Sets and Information Gain in Text Classification," vol. 10, no. 4, pp. 472–486, 2023, doi: 10.54287/gujsa.1379024.

[31] M. Krendzelak and F. Jakab, "Text Categorization with Machine Learning and Hierarchical Structures," *ICETA 2015 - 13th IEEE Int. Conf. Emerg. eLearning Technol. Appl. Proc.*, no. November 2015, 2016, doi: 10.1109/ICETA.2015.7558486.

[32] A. M. Rahat, A. Kahir, and A. K. M. Masum, "Comparison of Naive Bayes and SVM Algorithm based on Sentiment Analysis Using Review Dataset," in *2019 8th International Conference System Modeling and Advancement in Research Trends (SMART)*, 2019, pp. 266–270. doi: 10.1109/SMART46866.2019.9117512.

[33] X. Zhang, H. Zhao, S. Zhang, and R. Li, "A Novel Deep Neural Network Model for Multi-Label Chronic Disease Prediction," *Front. Genet.*, vol. 10, no. APR, pp. 1–11, Apr. 2019, doi: 10.3389/fgene.2019.00351.

[34] K. B. Lin, W. Weng, R. K. Lai, and Ping Lu, "Imbalance Data Classification Algorithm Based on SVM and Clustering Function," *Proc. 9th Int. Conf. Comput. Sci. Educ. ICCCSE 2014*, no. Iccse, pp. 544–548, 2014, doi: 10.1109/ICCSE.2014.6926521.