A Comparison of SVM and Naïve Bayes Classifier in Binary Sentiment Reviews for PeduliLindungi Application

Isal Firmansyah
Department of Statistics, Faculty of Mathematics and Natural Science
Universitas Padjadjaran
Bandung, Indonesia
isal19001@mail.unpad.ac.id

Mohammad Hamid Asnawi
Department of Statistics, Faculty of Mathematics and Natural Science
Universitas Padjadjaran
Bandung, Indonesia
mohammad19011@mail.unpad.ac.id

Rafly Novian
Department of Statistics, Faculty of Mathematics and Natural Science
Universitas Padjadjaran
Bandung, Indonesia
rafly19001@mail.unpad.ac.id

Syifa Auliyah Hasanah
Department of Statistics, Faculty of Mathematics and Natural Science
Universitas Padjadjaran
Bandung, Indonesia
syifa19010@mail.unpad.ac.id

Anindya Apriliyanti Pravitasari
Department of Statistics, Faculty of Mathematics and Natural Science
Universitas Padjadjaran
Bandung, Indonesia
anindya.apriliyanti@unpad.ac.id

Abstract*— COVID-19 statistics in Indonesia show more than 4.2 million active confirmed cases with more than 140 thousand deaths. The Indonesian government has made several policies to reduce the number of COVID-19 cases, one of them is by implementing the PeduliLindung application. The government has socialized and recommended this application as an effort to fulfill the tracking, tracing, and fencing program. Various kinds of responses appear in the community to this application, therefore sentiment analysis is needed to find out public trends so that the government can evaluate the policies that have been made. This study aims to determine the best model from the comparison of the Naïve Bayes algorithm and the Support Vector Machine, besides that this study will also see whether a simpler model such as Naive Bayes is still good in handling binary sentiment for PeduliLindung data reviews. The data was obtained by web scraping from the PeduliLindung application review on the Google Play Store. The Naïve Bayes accuracy value is 81%, smaller than the Support Vector Machine which has an accuracy of 84%, although the Support Vector Machine is the best model we have, Naive Bayes itself can still be used to handle binary sentiment data because the difference in accuracy values is not too far.

***Keywords*— Sentiment Analysis, Naïve bayes, Support Vector Machine, PeduliLindungi, COVID-19**

I. INTRODUCTION

The outbreak of the COVID-19 virus has become a scourge of problems throughout the world and to be the main focus for world leaders in finding ways to deal with this problem. The global pandemic was caused by a virus first appeared in Wuhan, China. It has had a great impact and influence on various sectors of life in Indonesia, especially in the socio-economic. Indonesia's economic condition is greatly affected by the COVID-19 pandemic, in response to these conditions, in 2020 the government allocated Rp695.2 trillion for the national economic recovery (PEN) program. In February 2021, the government announced a budget of Rp699.43 trillion for the PEN program due to the unfinished condition of COVID-19 in Indonesia [1] . Not only economic, the impact of COVID-19 is also felt in various other fields, such as education, health, and even social.

Various policies have been done by the government to be able to stop the COVID-19 pandemic in Indonesia, one of which is through the Regulation of the Minister of Health of the Republic of Indonesia No. 18 of 2021 concerning the Implementation of Vaccination in the Context of Coping with COVID-19 helps reduce the number of positive cases of COVID-19 in Indonesia. It is necessary to continue to hold efforts to control COVID-19 cases in Indonesia through 3T (Testing, Tracing, Treatment) by utilizing current technological advances. The decree of the Minister of Communication and Informatics No. 171 of 2020 stipulates that the basis for carrying out tracing, tracking, and fencing activities that utilize telecommunication systems and applications to support health surveillance is the PeduliLindungi application. In helping the government to carry out tracking to suppress COVID-19 cases in Indonesia, PeduliLindungi is expected to be the right solution to deal with these cases, every information about COVID-19 is available including vaccine, etc. The users will also get notification if they are in crowd or in red zone. Health Minister Budi Gunadi Sadikin confirmed that PeduliLindungi application will be included in some community activities. The application has started to be implemented in trading sector, both in tradional market or modern market such as super market, mall, etc. This application will also be applied in various sectors such as transportation, tourism (hotels, restaurants, performances), work, religious (mosques, churches, temples, monasteries, religious activities), and education sectors [2].

Various responses from the public to the PeduliLindungi application emerged along with the government's recommendation to use this application, this can be analyzed using sentiment analysis based on user reviews on the Google Play Store application. Sentiment analysis is a combination of text mining with natural language processing which aims to find opinions, identify what sentiments they express, and then classify them based on the values contained in them [3]. The benefit of looking for public sentiment is to know the position of this application in the community, so that the government can take another approach in recommending and socializing this application.

This paper is inspired by former paper research entitled Analysis of User Reviews for the PeduliLindungi Application on Google Playstore Using the Support Vector Machine and Naive Bayes Algorithm Based on Particle Swarm Optimization by A. Mustopa, et al [4], the conclusion from this paper can be different due to changes in situations and conditions that Indonesia faced such as the PPKM regulation, this caused different patterns of public reviews toward PeduliLindungi application, this regulation also makes the PeduliLindungi application users number increased starting August 2021, this increase was also triggered by the opening of several public services that require visitors to have PeduliLindungi application. The previous study that conducted sentiment analysis was Kristiyanti, et al [5] who applied Support Vector Machine (SVM) and Naïve Bayes Classifier for sentiment analysis on the West Java Governor Candidate, then there was Rana and Singh [6] who compared Naïve Bayes and SVM for user reviews about movies. Comparison between other methods is given by Poornima and Priya [7] which results that Naïve Bayes being more suitable for simpler data. Inspired from previous research, it appears that SVM and Naive Bayes are often used in text analysis. The two methods have different algorithms, SVM is more complex than simple Naive Bayes. This study uses SVM and naive Bayes to predict public sentiment towards PeduliLindungi with only two categories, i.e., Positive and Negative (Binary Sentiment Analysis). This study aim to determine the best model in predicting public sentiment of PeduliLindungi application based on performance of these two algorithms. Besides it, this study also want to see is the simpler Naïve Bayes algorithm able to handle binary sentiment of Pedulilindungi application review after some new provision applied by Indonesia's government, or is it necessary to use a more complex algorithm such as SVM to predict sentiment.

II. METHODOLOGY

Sentiment analysis of the PeduliLindungi review begins with scraping data from Google Playstore (the data in Bahasa) with WebHaryy followed by pre-processing such as case folding, data cleaning, tokenization, stopword removal, normalization, and stemming. The next step after pre-processing is weighting with the TF-IDF algorithm which will then be followed by Naïve Bayes and Support Vector Machine Classifier. The flow chart of sentiment analysis can be seen in Fig 1.

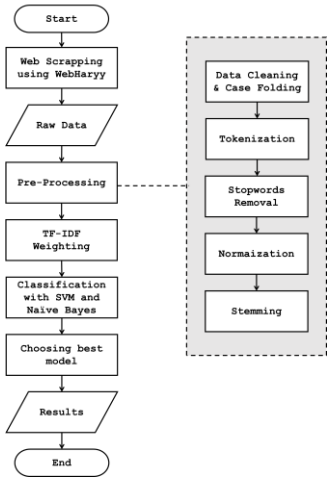

Fig. 1. Flow Chart of Sentiment Analysis for this study

A. *Pre-Processing*

Pre-processing is one of the most important steps in analysis, pre-processing ensures that the output of a data set is ready for analysis. Preprocessing identifies and fixes problems in the raw data in-to cleaner information that can be used for further processing. Here is the explanation per steps.

1. *Case Folding* and *Data Cleaning*

The case folding stage basically uniforms all text into lowercase, at this step the capital letters are changed to lowercase using the "RegEx" module in python. The purpose of case folding is to eliminate redundancy, which is the repetition of the same data in a database which results in wastage of storage. If case folding is not done, the same word may be counted or defined into two entities due to differences in writing system that have not been removed.

The data cleaning step also utilizes the same module as the case folding, namely "RegEx", at this step we clean elements in the text that have no meaning at all on the results of sentiment analysis, therefore at this stage the author performs several element deletions such as removing punctuation, eliminate numbers, and so on.

2. Tokenizing

The tokenizing stage is the step of cutting sentences into a list of words that make up the sentence separated by commas and spaces so that the results are single words that are collected in the array data which will later be used in the weighting process, examples of tokenized sentences: “halo warga indonesia” to “halo, warga, indonesia”.

3. Stopword Removal

Stopword removal is the process of removing words that are included in the stopword category, stopwords are words that often appear but have no meaning on the analysis. At this step we use the nltk package in python. The examples of words that are stopwords are conjunctions/prepositions and slang words that are inserted at the end of sentences that has a function only to make informal impression, such as the word "dong".

Stopwords are removed because they are considered unable to represent the contents of the review sentences that we have. The stopword removal process is carried out by making a stopword database. The database created will be compared with the PeduliLindungi application review data so that the result of this process is the elimination of words that exist in the database we made previously.

4. Stemming

The stemming process is the process of changing and reducing words into their basic form [8] . The stemming process removes suffixes, confixes, and prefixes in the existing text. At this stage the author uses the pipeline and StemmerFactory modules in python.

B. *TF-IDF Weighting*

In weighting process, each word will be weighted with certain rules to see the tendency of the response of the text it has. In this study, weighting was used with the TF-IDF technique. TF-IDF method combines weighting process of

TF and IDF, calculating the frequency of occurrence of a word in a particular document and reducing the weight of a word if the word appears a lot in a document, then both are multiplied [9].

C. Sentimen Analysis with Naïve Bayes Classifier

In this study, two algorithms are compared, one of them is Naive Bayes. Naïve Bayes Classifier is a classification method based on the bayes theorem, Naïve Bayes Classifier It is popular for its ease and simplicity, although this classification provides classification results equivalent to decision tree and neutral network, moreover Naïve Bayes Classifier also provides speed in processing data in large quantities [10].

Naïve Bayes Classifier assumes that the presence or absence of a feature in a class is independentthat mean a feature in a class has no connection to the existence of other features of the same data.

The Naïve Bayes Classifier process can generally be written in the following equations:

$$P(c_j|w_i) = \frac{P(c_j)xP(w_i|C_j)}{P(w_i)}, \quad (1)$$

where:

$P(c_j|w_i)$: Probability of hypothesis based on conditions $C_j w_i$(Posteriori probability)
$P(w_i|C_j)$: Probability based on conditions on w_i hypothesis C_j
$P(c_j)$: Probability hypothesis C_j (prior probability), $P(c_j) = \frac{N_{c_j}}{N}$
$P(w_i)$: Probability w_i
w_i : Unknown class
c_j : A data hypothesis that is a class C_j Specific
N_{c_j} : Documents that fall into categories c_j
N : Number of all training documents used

In the Naïve Bayes Classifier, each review is represented in attribute pairs $(k_1, k_2, \ldots, k_n)$ where k_1 is the first word, k_2 is the second word, and so on until the n-th word. In the classification process, Bayes' approach selects the category that has the highest probability (V_{MAP}) formulated in the equation as follows:

$$V_{MAP} = argmax \frac{P(c_j)xP(w_i|c_j)}{P(w_i)} \quad (2)$$

Since the value $P(W_i)$ is constant for all c_j, the value of $P(w_i)$ can be ignored so that the above equations can be written as:

$$V_{MAP} = \text{argmax } P(c)P(w_i|c_j) \quad (3)$$

This algorithm assumes that each category is independent then Naïve Bayes Classifier simplifies the equation (3) to:

$$P(k_1, k_2, \ldots, k_n|c_j) = \prod_i P(w_i|c_j) \quad (4)$$

By substituted equation (3) to equation (4) it will produce

$$V_{MAP} = argmax\ P(c_j) \prod_i P(w_i|c_j) \quad (5)$$

Value $P(c_j)$ and $P(w_i|c_j)$ calculated during the training process where the equation of the two is as follows:

$$P(c_j) = \frac{|docs_j|}{training}$$
$$P(w_i|c_j) = \frac{n_{i+1}}{vocab+n} \quad (6)$$

where:

$P(w_i|c_j)$: The probability of the word w_i in the category c_j
$|docs_j|$: Number of documents in category j
$training$: The total number of samples used in training process
n_i : The frequency of occurrence of the word w_i in the category c_j
$vocab$: Number of unique words in all training data

D. Sentimen Analysis with Support Vector Machine

Besides Naïve bayes, another algorithm to be compared is Support Vector Machine. Support Vector Machine (SVM) is one of the algorithms used to predict regression and classification. SVM is a supervised machine learning algorithm that uses kernel functions to map data point spaces. Linearly the data cannot be separated into a new space in which it is [11]. The SVM algorithm looks for the largest hyperlane value, the classification of hyperplanes is notated as follows:

$$f(x) = w^T x + b \quad (7)$$

Similarities formed according to Vapnik and Cortes [12]:

$$[(w^T x_i) + b] \geq 1 \text{ for } y_i = +1$$
$$[(w^T x_i) + b] \leq -1 \text{ for } y_i = -1$$

where:
x_i = training data set, $i = 1,2,\ldots,n$
y_i = class label of x_i

Support Vector Machine looking for the best hyperplane located in the middle of a class divider, and by maximizing the margin or distance between two sets of objects of different classes.

With Quadratic Programming (QP) Problem Method, The Lagrange function used to optimize found by Vapnik is as follows:

$$L(w, b, a) = \frac{1}{2} w^T w - \sum_{i=1}^{n} \alpha_i \{y_i[(w^T x_i) + b] - 1 \quad (8)$$

where α_i = the langrange and $i = 1,2,\ldots,n$ is function multiplier

After the QP method is found, the solution in finding the class of data to be predicted is as follows:

$$f(x_t) = \sum_{s=1}^{ns} a_s y_s x_s . x_t + b \quad (9)$$

where:
x_t : data testing
x_s : data support vector, $s = 1, 2 \ldots ns$
ns : number of supports vectors

If the data set cannot be separated linearly, classification is impossible because there is no separating hyperplane and it

measure of misclassification, the presence of noise also greatly affects the margins [13]. The barriers for non-*separable* cases are as follows:

$$y_i(w^T x_i) + b \geq 1 - \xi_i, i = 1, 2 \ldots, n \quad (10)$$

It becomes a minimizing equation

$$\frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i \quad (11)$$

Kernel functions in SVM algorithm can be used to lower dimensions map into higher dimensions on non-linear data. Some kernel functions include:

1. Gaussian Radial Basic Function (RBF) Kernel

$$K(x_i, x_j) = exp\left(-\frac{||x_i, x_j||^2}{2\sigma^2}\right)$$

2. Polynomial Kernel

$$K(x_i, x_j) = \left((x_i, x_j) + c\right)^d$$

E. Metric of Evaluation

Confusion matrix or error matrix is a matrix table that displays a description of the performance of the classification model on a test data set (testing) whose actual value has been known, confusion matrix provides information comparing the results of the system classification with the actual results. Confusion matrix is used to determine accuracy, precision, recall, and error rate.

TABLE I
CONFUSION MATRIX

	Positive	Negative
Positive	*True Positive* (TP)	*False Positive* (FP)
Negative	*False Negative* (FN)	*True Negative* (TN)

Four important terms that represent the results of the classification process in the matrix confussion are: True Positive, True Negative, False Positive, and False Negative.

1. Accuracy

Accuracy is the correct prediction ratio of the number of diagonal elements to the sum of the total matrix elements, mathematically accuracy can be formulated as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \quad (12)$$

The result of the calculation above is the percentage of the predicted amount of data that is correctly valued against the overall amount of data. Accuracy is only suitable if used at the time of comparison of the number of actual data labels relatively the same.

2. Precision

Precision can be defined as the degree of reliability of a model when the model produces a positive prediction. In calculating precision, only the first or second line of the confusion matrix is required. Precision is the proportion of positively correct predictions against the overall positive prediction. Mathematically precision can be formulated as follows:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (13)$$

3. Recall

Recall or sensitivity is a method used to measure how well a test can identify a true positive, a recall describes the success of a model in rediscovering information. Recall is a comparison of positive correct predictions with overall positive correct data. Mathematically the recall value can be formulated as follows:

$$\text{Recall} = \frac{TP}{TP+FN} \quad (14)$$

4. *F*1-*Score*

Calculations that summarize precision and sensitivity/recall by taking harmonic average calculations of both. Mathematically the value of F1-Score can be made the following formula:

$$F1 - Score = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} \times \text{Recall}} \quad (15)$$

III. RESULTS AND DISCUSSIONS

The data used in this study is from the review of PeduliLindungi application in Play Store. This study mining the documents of about 2,840 comments or reviews. Table II shows the comparison of one example of a comment that has done through the text pre-processing stage.

TABLE II
PREPROCESSING RESULT TABLE

Raw Data	Data after Preprocessing
Saya sudah vaksin 2x, tapi knapa hanya di beritahukan vaksin pertama saja, yg kedua belum vaksin ,,padahal sudah vaksin 2x, parahnya sertifkat vaksin pertama juga blum ada katanya, padahal sudah vaksin pertama bulan april, Tolong perjelas, jangan buat aplikasi abal2 dan menipu publik!!!!!!!!!!!!!!!!	vaksin kenapa beritahukan vaksin vaksin vaksin parah sertifikat vaksin belum vaksin tolong jelas aplikasi abal tipu publik
⋮	⋮
Aplikasi apa ini!!!! Masa terdeteksi make fake GPS, padahal gk make tuh. Ayolah Menkominfo, saya cuman mau *download* sertifikat vaksinasi buat ke sekolah, klo masih kaya gini, mending hapus aja dari *playstore*. Daripada cuman jadi beban.	aplikasi deteksi pakai palsu gps pakai ayo menkominfo hanya unduh sertifikat vaksinasi sekolah seperti begini mending hapus hanya beban

The comments have undergone changes in the text pre-processing. at this phase, the process of case folding and data cleaning, tokenization, stop word removal, normalization, and stemming is carried out. The results of the text pre-processing phase produce text data that are ready to enter the next stage, namely the weighting stage. The following is a data visualization of the results of text pre-processing.

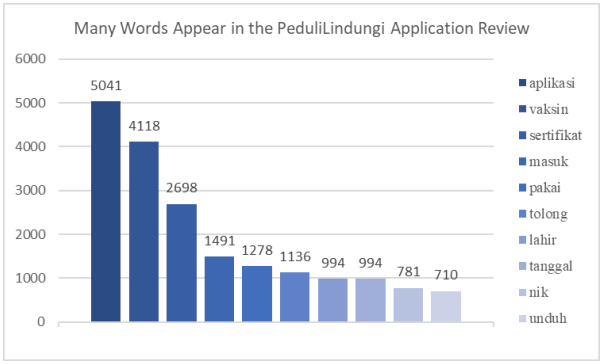

Fig. 2. Bar chart of the number of words that appear in PeduliLindungi app reviews

The two images are visualizations to represent the number of words that appear in the PeduliLindungi application review. It can be seen that there are 10 words that users often use to express their opinion about the application. Where in the wordcloud graph the word 'aplikasi' is printed larger than other words because 'aplikasi' is the word with the highest number of uses, namely 5,041 words, followed by the word 'vaksin' as many as 4,118, the word 'sertifikat' as many as 2,698, the word 'masuk' as many as 1,491, the word 'pakai' as many as 1278, the word 'tolong' as many as 1,136, the word 'lahir' as many as 994, the word 'tanggal' as many as 994, the word 'nik' as many as 781, and the word 'unduh' as many as 710 words.

At the stage of determining the TF-IDF matrix normalization is carried out on the TF and IDF to obtain the TF-IDF values and the ranking are obtained based on the weighting above as in Table III.

TABLE III
WEIGHTING RESULT TABLE

Word	Weighting
aplikasi	381.0746
vaksin	344.266
sertifikat	329.8253
tanggal	258.5432
tolong	244.0897
pakai	230.3026
masuk	228.5307
unduh	210.3529
nik	187.1637
lahir	178.803

The results obtained from the ranking of the top three words, namely the word 'aplikasi' which got the first rank with a weighting value of 381.0746. The word 'vaksin' got the second rank with a weighting value of 344.266 and for the third rank the word 'sertifikat' was obtained with a weighting value of 329.8253.

Sentiment analysis was performed using the SVM and Naïve Bayes algorithm on the PeduliLindungi application review data set, 80% split was applied to training data from all data (2,271 data), and 20% split to test data from all data (568 data). The distribution of data is given in Fig. 3.

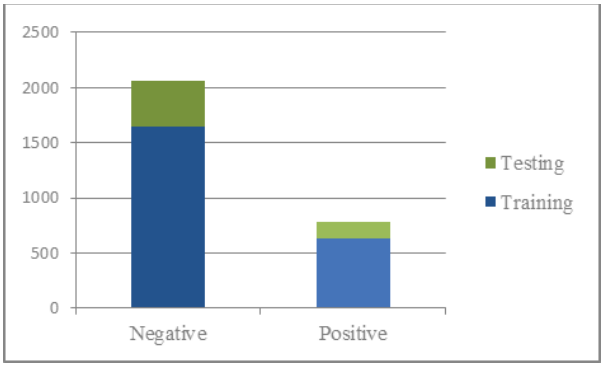

Fig. 3. Distribution of the documents labels.

Fig. 3 shown the distribution sentiment label for the data. Its appear that mostly the reviews or public sentiments towards this aplication is negative. This information can be a concern for the government to carry out a strategy to better introduce and socialize this application to the community.

From Table IV and Table V we can see the confussion matrix of Naïve Bayes and SVM algorithm, the comparison of the algorithm through the accuracy, precision, recall, and F1-score from testing data is given in Table VI. From Table VI, it could see that from the several metrics of evaluation the simpler Naïve Bayes algorithm can still be used properly to handle binary sentiment analysis of PeduliLindungi application because the accuracy value is not too far from the SVM algorithm. but the SVM is considered better in predicting binary sentiment review for PeduliLindungi application. However, the difference in the accuracy is about 3%, which means the valid prediction difference is about 18 documents. If we look at the confusion matrix, the number of miss-prediction comments or reviews is quite large, this could happen due to the possibility of another type of sentiment that is not measured in this study, namely neutral sentiment.

TABLE IV
CONFUSION MATRIX NAÏVE BAYES CLASSIMETFIER

	Positive	Negative
Positive	367	48
Negative	59	*94*

TABLE V
CONFUSION MATRIX SUPPORT VECTOR MACHINE

	Positive	Negative
Positive	381	34
Negative	55	*98*

TABLE VII
MODEL EVALUATION

Performance	Naïve Bayes	SVM
Accuracy	81.16%	84.33%
Precision	86.15%	87.38%
Recall	88.43%	91.80%
F1 Score	87.27%	89.50%

ACKNOWLEDGMENT

The Authors are grateful to Universitas Padjadjaran which support this research.

REFERENCES

[1] SMERU , PROSPERA , UNDP , UNICEF, "Executive Summary Report: The Social and Economic Impacts of COVID-19 on Households and Strategic Policy Recommendations for Indonesia," SMERU RESEARCH INSTITUTE, Jakarta, 2021.

[2] P. Violleta, U. Liman and S. Haryati, "PeduliLindungi to be used in people's daily activities: Minister," ANTARA News, 7 October 2021. [Online]. Available: https://en.antaranews.com/news/193073/pedulilindungi-to-be-used-in-peoples-daily-activities-minister. [Accessed Oktober 2021].

[3] B. Liu, "Sentiment Analysis and Opinion Mining," *Morgan & Claypool Publishers.* , 2012.

[4] A. Mustopa, H. A. E. P. Pratama, A. Hendini and D. Risdiansyah, "Analysis of User Reviews for the PeduliLindungi Application on Google Play Using the Support Vector Machine and Naive Bayes Algorithm Based on Particle Swarm Optimization," 2021.

[5] D. A. Kristiyanti, A. H. Umam, M. Wahyudi, R. Amin and L. Marlinda, "Comparison of SVM & Naïve Bayes Algorithm for Sentiment Analysis Toward West Java Governor Candidate Period 2018-2023 Based on Public Opinion on Twitter," *The 6th International Conference on Cyber and IT Service Management (CITSM 2018)*, 2018.

[6] S. Rana and A. Singh, "Comparative Analysis of Sentiment Orientation Using SVM and Naive Bayes Techinques," *2016 2nd International Conference on Next Generation Computing Technologies (NGCT-2016)*, 2016.

[7] P. A and K. S. Priya, "A Comparative Sentiment Analysis of Sentence Embedding Using Machine learning Techniques," *2020 6th International Conference on Advanced Computing & Communication Systems (ICACCS)*, 2020.

[8] C. Gallagher, E. Furey and K. Curran, "The Application of Sentiment Analysis and Text Analytics to Costumer Experience Reviews to understand What Costumers Are Really saying," *International Journal of Data Warehousing and Mining*, vol. 15, no. 4, pp. 21-47, 2019.

[9] B. G. Gebre, M. Zampieri, P. Wittenburg and T. Heskes, "Improving Native Language Identification with TF-IDF Weighting," in *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, Atlanta, Georgia, 2013.

[10] C.Aggarwal, Data Classification: Algorithms and Applications, Minneapolis, Minnesota, USA: Chapman & Hall/CRC , 2015.

[11] T. Mullen and N. Collier, "Sentiment analysis using support vector machines with diverse information sources," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Barcelona, 2004.

[12] C. Cortes and V. Vapnik, "Support-Vector Networks," in *Machine Learning*, Boston, Kluwer Academic Publishers, Boston, 1995, pp. 273-297.

[13] V. K. Chauhan, K. Dahiya and A. Sharma, "Problem formulations and solvers in linear SVM: a review," *Artificial Intelligence Review*, vol. 52, p. 803–855, 2018.